

**ARTHUR MAÇANEIRO MAGYAR, FÁBIO GLAUCO DOS SANTOS,
LUCIANO PADILHA**

**SISTEMA DE MONITORAÇÃO E CONTROLE ESTATÍSTICO DE
INCUBADORA EXPERIMENTAL**

**Trabalho de graduação apresentado à
disciplina Trabalho de Conclusão de
Curso do Curso de Tecnologia em
Informática da Escola Técnica da
Universidade Federal do Paraná.**

**Orientador: Prof. Mário de Paula
Soares Filho**

**Co-orientador: Prof. Irapuru Flório
Haruo**

**CURITIBA
2005**

TERMO DE APROVAÇÃO

**ARTHUR MAÇANEIRO MAGYAR, FÁBIO GLAUCO DOS SANTOS,
LUCIANO PADILHA**

**SISTEMA DE MONITORAÇÃO E CONTROLE ESTATÍSTICO DE
INCUBADORA EXPERIMENTAL**

Trabalho aprovado como requisito para obtenção do grau de Graduado no Curso de Tecnologia em Informática da Escola Técnica da Universidade Federal do Paraná, pela seguinte banca examinadora:

Co-orientador: Prof. M. Irapuru Florido Haruo
Departamento de Tecnologia em Informática, UFPR

Prof. M. Jaime Wojciechowski
Departamento de Tecnologia em Informática, UFPR

Prof.^a Dr.^a Jeroniza Nunes Marchaukoski
Departamento de Tecnologia em Informática, UFPR

Orientador: Prof. Esp. Mário de Paula Soares Filho
Departamento de Tecnologia em Informática, UFPR

Curitiba, 25 de janeiro de 2006

DEDICATÓRIA

Dedicamos este trabalho a todas as iniciativas de projetos inovadores e com ênfase na contribuição ao desenvolvimento de outros cursos e nas mais diversas áreas do conhecimento.

AGRADECIMENTOS

A todos que, direta ou indiretamente, contribuíram para a realização e divulgação deste trabalho.

Em especial, a:

Mário e Irapuru,
pela diferente orientação, confiança e incentivo.

Edson e Marina,
pela disponibilidade e grande envolvimento com o projeto.

Marcelo,
pela contribuição a ambos os nossos conhecimentos.

Fernando, Marlon e Roberta,
pelas explicações de uma área que não conhecíamos.

SUMÁRIO

RESUMO.....	VII
1 INTRODUÇÃO	1
2 JUSTIFICATIVA	2
3 FERRAMENTAS UTILIZADAS	3
4 OBJETIVOS	4
4.1 OBJETIVO GERAL.....	4
4.2 OBJETIVOS ESPECÍFICOS	4
5 ANÁLISE DE REQUISITOS.....	6
6 CRONOGRAMA.....	7
6.1 CALENDÁRIO DAS ATIVIDADES	7
6.2 DIAGRAMA DE GANTT	8
7 DIAGRAMA DE TELAS.....	9
8 MODELAGEM.....	34
8.1 CASOS DE USO	34
8.1.1 Descrição Resumida.....	35
8.2 DIAGRAMAS DE CLASSE.....	42
8.2.1 Modelo Físico	42
8.2.2 Modelo Lógico.....	43
8.3 DIAGRAMAS DE INTERAÇÃO	44
8.3.1 Diagramas de Seqüência	44
8.3.2 Diagramas de Colaboração	93
8.4 DIAGRAMAS DE ESTADO	116
8.4.1 Ovo.....	116
8.4.2 Projeto	117
8.4.3 Usuário.....	118
8.4.4 Incubação	119
8.4.5 Nascimento	120
8.5 DIAGRAMA ENTIDADE-RELACIONAMENTO	121
8.6 DICIONÁRIO DE DADOS	122
9 CONCLUSÃO.....	134
10 REFERÊNCIAS	135
11 GLOSSÁRIO	136
11.1 TERMOS UTILIZADOS NO BANCO DE DADOS	136
11.2 TERMOS UTILIZADOS NA MODELAGEM	136

12 CÓDIGO FONTE.....	137
12.1 BASE DE DADOS.....	137
12.1.1 Script de Criação.....	137
12.1.2 Sequências.....	161
12.1.3 Funções e Triggers.....	165
12.1.4 Índices.....	167
12.1.5 Views.....	167
12.1.6 Tela de Configuração.....	193
12.2 PACOTES.....	194
12.2.1 broker.....	194
12.2.2 classepersistente.....	321
12.2.3 controladora.....	334
12.2.4 javaUtil.....	351
12.2.5 servelet.....	352
12.3 RELATÓRIOS.....	398
12.3.1 EntidadeBloco.....	398
12.3.2 EntidadeCelula.....	399
12.3.3 EntidadeGaiola.....	401
12.3.4 EntidadeLinha.....	402
12.3.5 ProjetoIncubacao.....	404
12.3.6 ProjetoIncubacaoIncubadoraBandeja.....	406
12.3.7 ProjetoIncubacaoNascimento.....	407
12.3.8 SalaUF.....	408
12.4 PÁGINAS JSP.....	420
12.4.1 Principais.....	420
12.4.2 Menus.....	472
12.4.3 Outras.....	483
12.5 PÁGINAS CSS.....	495
12.5.1 cabecalho.....	495
12.5.2 menu.....	496
12.5.3 principal.....	497
12.5.4 submenu.....	497
12.6 SCRIPTS.....	498
12.6.1 Páginas JavaScript.....	498
12.6.2 Páginas JSP.....	508

RESUMO

O presente trabalho tem o objetivo de demonstrar o projeto de desenvolvimento do SISINC – Sistema de Monitoração e Controle Estatístico de Dados de Incubadora Experimental, a ser utilizado pelos professores e alunos do Curso de Zootecnia da UFPR – Universidade Federal do Paraná.

O sistema visa possibilitar a inserção de dados, acompanhamento estatístico e monitoramento das incubações e nascimentos experimentais realizados na Fazenda Canguiri da UFPR, permitindo entrada de informações (como incubações realizadas e entidades relacionadas) e consultas através da Intranet.

No desenvolvimento do projeto foram utilizadas ferramentas de modelagem baseadas em UML (*Unified Modeling Language*), além de conceitos de gerência de projeto.

O desenvolvimento foi realizado através das ferramentas de desenvolvimento do Java sendo o banco de dados escolhido o PostgreSQL 8.0.3.

Palavras-chave: Trabalhos de Conclusão, Desenvolvimento de Sistemas, Informática, Zootecnia, Incubação, Nascimento, Incubadora, Nascidouro, Ovos, Aves.

1 INTRODUÇÃO

A Universidade Federal do Paraná possui um laboratório chamado Fazenda Experimental Canguiri em Pinhais, onde desenvolve pesquisas em diversas áreas, sendo que algumas de suas pesquisas carecem de auxílio por um sistema computadorizado. No caso das pesquisas realizadas em seu aviário o controle de incubação e desenvolvimento das aves tem se tornado demasiado complexo, o que inviabiliza pesquisas com o grau de precisão e rastreabilidade desejados. Torna-se cada vez mais necessário um sistema que permita ao pesquisador monitorar, rastrear e controlar de forma estatística seu projeto relacionado ao aviário.

Desta necessidade surgiu o SISINC, que visa permitir a inserção de dados e o acompanhamento estatístico das incubações e nascimentos realizados. Desta forma o pesquisador pode monitorar as variáveis envolvidas e visualizar, de forma gráfica, a distribuição dos ovos nos equipamentos (incubadoras e nascedouros), além de permitir rastrear o estado dos projetos em andamento.

2 JUSTIFICATIVA

A importância deste projeto está na possibilidade de inserção das informações associadas ao acompanhamento das aves, da geração de ovos e da produtividade final de crias.

O sistema SISINC permite, também, que sejam armazenadas informações de entidades pertinentes ao aviário, como: prédio, sala, bloco ou célula, além dos principais equipamentos (incubadoras, nascedouros e bandejas) e das principais características relativas a cada projeto, como o tipo de ave associado e a entidade mínima de rastreamento. Esta característica atende às necessidades do usuário de rastreabilidade das informações e permite análise futura dos dados obtidos durante os experimentos.

O sistema SISINC pode ser utilizado por diferentes tipos de usuários, desde administradores do sistema, professores e alunos, que possuem diferentes papéis no ambiente do aviário.

A versão atual do sistema, a ser testada pelo usuário quando do retorno das aulas do ano corrente, apresenta as características básicas definidas durante fase de levantamento inicial de dados e descritas no documento de definição escopo do projeto (documento *Scope Statment*).

3 FERRAMENTAS UTILIZADAS

Na modelagem e desenvolvimento do projeto foram utilizadas as seguintes ferramentas:

- Banco de dados: PostgreSQL 8.0.3;
- Modelagem de banco de dados: DBDesigner 4.56;
- Gerenciamento de banco de dados: EMS Postgre SQL Manager 3.0.1;
- Linguagem de programação: Java J2SE 5.0 e JSP 2.0;
- Desenvolvimento: Eclipse 3.1 com plug-in do Tomcat;
- Modelagem de dados: Rational Rose 2003;
- Criação de cronogramas: MS Project 2003;
- Relatórios, planilhas e apresentações: MS Office 2003 e XP;
- Mensagens instantâneas (conferências durante treinamento): Google Talk Beta e MSN Messenger 7.0 a 8.0;
- Navegabilidade: IE 6.0 SP1 e Mozilla Firefox 1.0 a 1.5.

4 OBJETIVOS

4.1 OBJETIVO GERAL

Criar um sistema computacional que permita a inserção e o acompanhamento de informações associadas ao aviário da Fazenda Experimental Canguiri da UFPR. O sistema deve permitir:

- Armazenamento das principais características do ambiente do aviário;
- Armazenar informações associadas aos projetos experimentais desenvolvidos no laboratório (aviário);
- Armazenar principais características dos equipamentos do laboratório;
- Armazenar informações dos ovos, nascidos e não-nascidos (descartados);
- Armazenar informações das incubações e nascimentos realizados.

4.2 OBJETIVOS ESPECÍFICOS

Desenvolver sistema computacional que atenda às principais necessidades dos usuários do Curso de Zootecnia da UFPR, definidas durante a fase de levantamento de requisitos e integrantes do documento de definição de escopo do projeto (*Scope Statment*), que podem ser resumidas nos seguintes itens:

- Cadastramento de informações básicas do ambiente do aviário, como usuários e seus papéis, além das entidades como: localidades, prédios e andares;
- Cadastro de informações básicas associadas ao projeto, como objetivo e sua duração;
- Cadastrar e alterar informações de equipamentos (incubadoras e nascedouros) e bandejas;
- Cadastrar ovos descartados, motivo e momento de descarte,

diferenciando não-nascidos e não-incubados. Permitir, também, a consulta de posições nas bandejas por nível de rastreabilidade definido em cada projeto;

- Consultar registros de incubação que permitam acessar dados estatísticos de ovos incubados, nascidos e não-nascidos.

5 ANÁLISE DE REQUISITOS

O objetivo do sistema é auxiliar no cadastro, rastreabilidade e controle estatístico de informações do aviário da Fazenda Canguiri da UFPR, permitindo acesso a alunos e professores devidamente cadastrados.

Com base nas informações disponibilizadas pelo professor Edson Gonçalves e Oliveira e estagiários do Curso de Zootecnia por ele orientados foi criado o documento de escopo do projeto, (*Scope Statment*), que possui as premissas básicas do projeto e o escopo da versão corrente do sistema.










Após análise inicial dos requisitos o grupo, em conjunto com o orientador e o usuário final, optou por definir um conjunto de funcionalidades contempladas na versão atual do sistema, devido à alta complexidade exigida para que fossem atendidas todas as necessidades dos usuários do Curso de Zootecnia da UFPR. Os principais requisitos do sistema são:

- Criação de projetos;
- Criação de entidades;
- Criação de incubações e associação das entidades e ovos envolvidos;
- Criação de nascimento e associação das entidades e ovos envolvidos;
- Possibilitar rastrear número de ovos incubados, transferidos, nascidos e descartados, para um dado projeto.

6 CRONOGRAMA

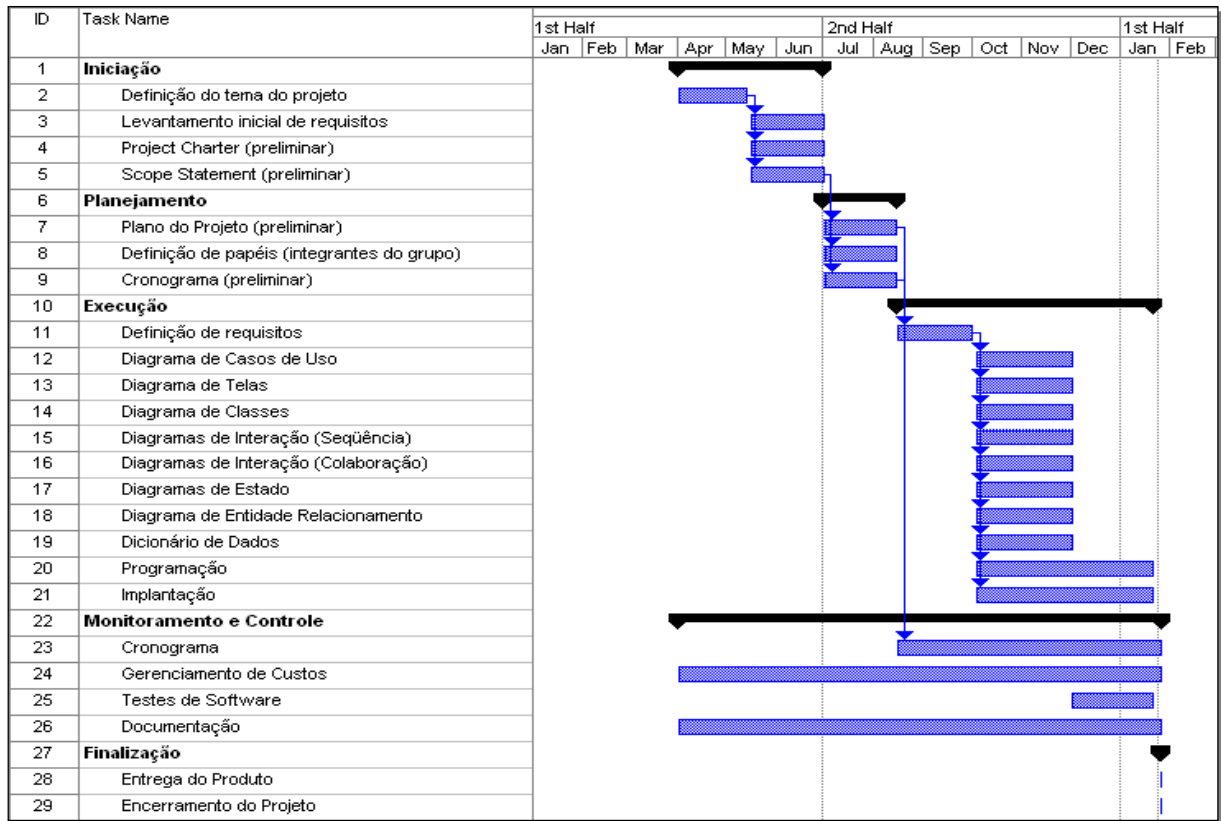
6.1 CALENDÁRIO DAS ATIVIDADES

FIGURA 1 – CALENDÁRIO DAS ATIVIDADES

ID		Task Name	Duration	Start	Finish
1		Iniciação	65 days	Fri 4/1/05	Thu 6/30/05
2		Definição do tema do projeto	31 days	Fri 4/1/05	Fri 5/13/05
3		Levantamento inicial de requisitos	34 days	Mon 5/16/05	Thu 6/30/05
4		Project Charter (preliminar)	34 days	Mon 5/16/05	Thu 6/30/05
5		Scope Statement (preliminar)	34 days	Mon 5/16/05	Thu 6/30/05
6		Planejamento	32 days	Fri 7/1/05	Mon 8/15/05
7		Plano do Projeto (preliminar)	32 days	Fri 7/1/05	Mon 8/15/05
8		Definição de papéis (integrantes do grupo)	32 days	Fri 7/1/05	Mon 8/15/05
9		Cronograma (preliminar)	32 days	Fri 7/1/05	Mon 8/15/05
10		Execução	114 days	Tue 8/16/05	Fri 1/20/06
11		Definição de requisitos	34 days	Tue 8/16/05	Fri 9/30/05
12		Diagrama de Casos de Uso	45 days	Mon 10/3/05	Fri 12/2/05
13		Diagrama de Telas	45 days	Mon 10/3/05	Fri 12/2/05
14		Diagrama de Classes	45 days	Mon 10/3/05	Fri 12/2/05
15		Diagramas de Interação (Seqüência)	45 days	Mon 10/3/05	Fri 12/2/05
16		Diagramas de Interação (Colaboração)	45 days	Mon 10/3/05	Fri 12/2/05
17		Diagramas de Estado	45 days	Mon 10/3/05	Fri 12/2/05
18		Diagrama de Entidade Relacionamento	45 days	Mon 10/3/05	Fri 12/2/05
19		Dicionário de Dados	45 days	Mon 10/3/05	Fri 12/2/05
20		Programação	80 days	Mon 10/3/05	Fri 1/20/06
21		Implantação	80 days	Mon 10/3/05	Fri 1/20/06
22		Monitoramento e Controle	214 days	Fri 4/1/05	Wed 1/25/06
23		Cronograma	117 days	Tue 8/16/05	Wed 1/25/06
24		Gerenciamento de Custos	214 days	Fri 4/1/05	Wed 1/25/06
25		Testes de Software	37 days	Thu 12/1/05	Fri 1/20/06
26		Documentação	214 days	Fri 4/1/05	Wed 1/25/06
27		Finalização	1 day	Wed 1/25/06	Wed 1/25/06
28		Entrega do Produto	1 day	Wed 1/25/06	Wed 1/25/06
29		Encerramento do Projeto	1 day	Wed 1/25/06	Wed 1/25/06

6.2 DIAGRAMA DE GANTT

FIGURA 2 – DIAGRAMA DE GANTT



7 DIAGRAMA DE TELAS

FIGURA 3 – DIAGRAMA DE TELAS

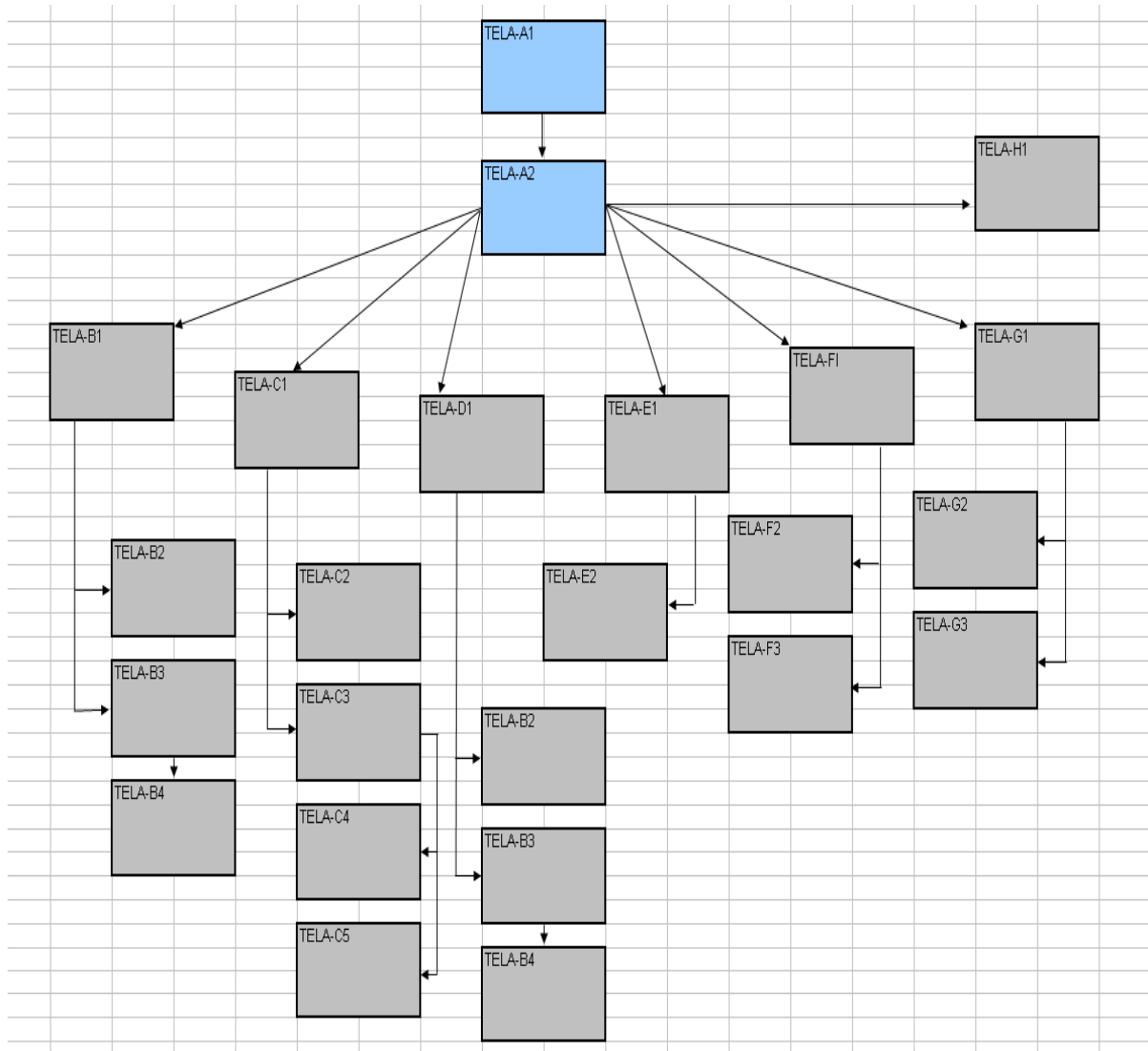


FIGURA 4 – TELA-A1

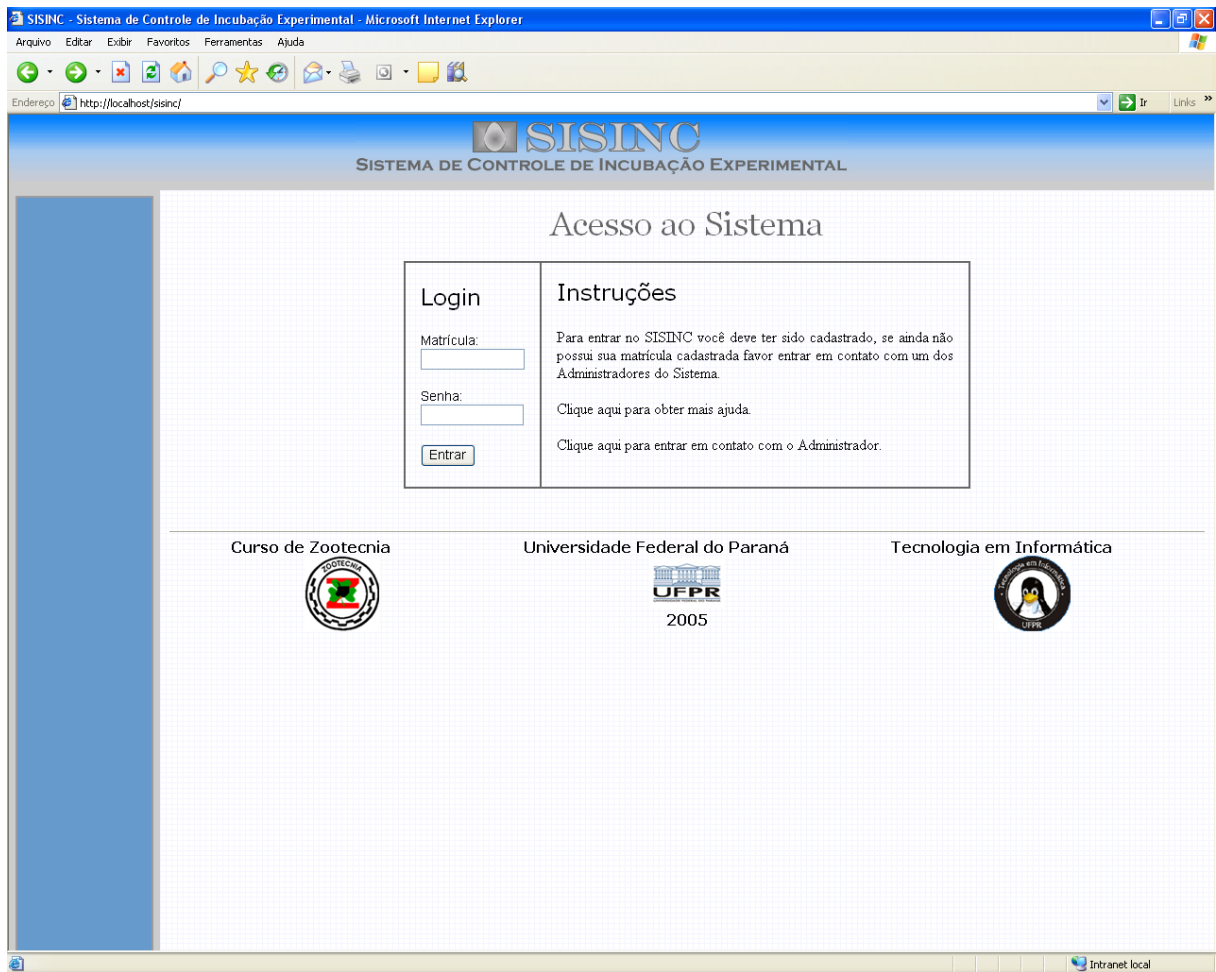


FIGURA 5 – TELA-A2

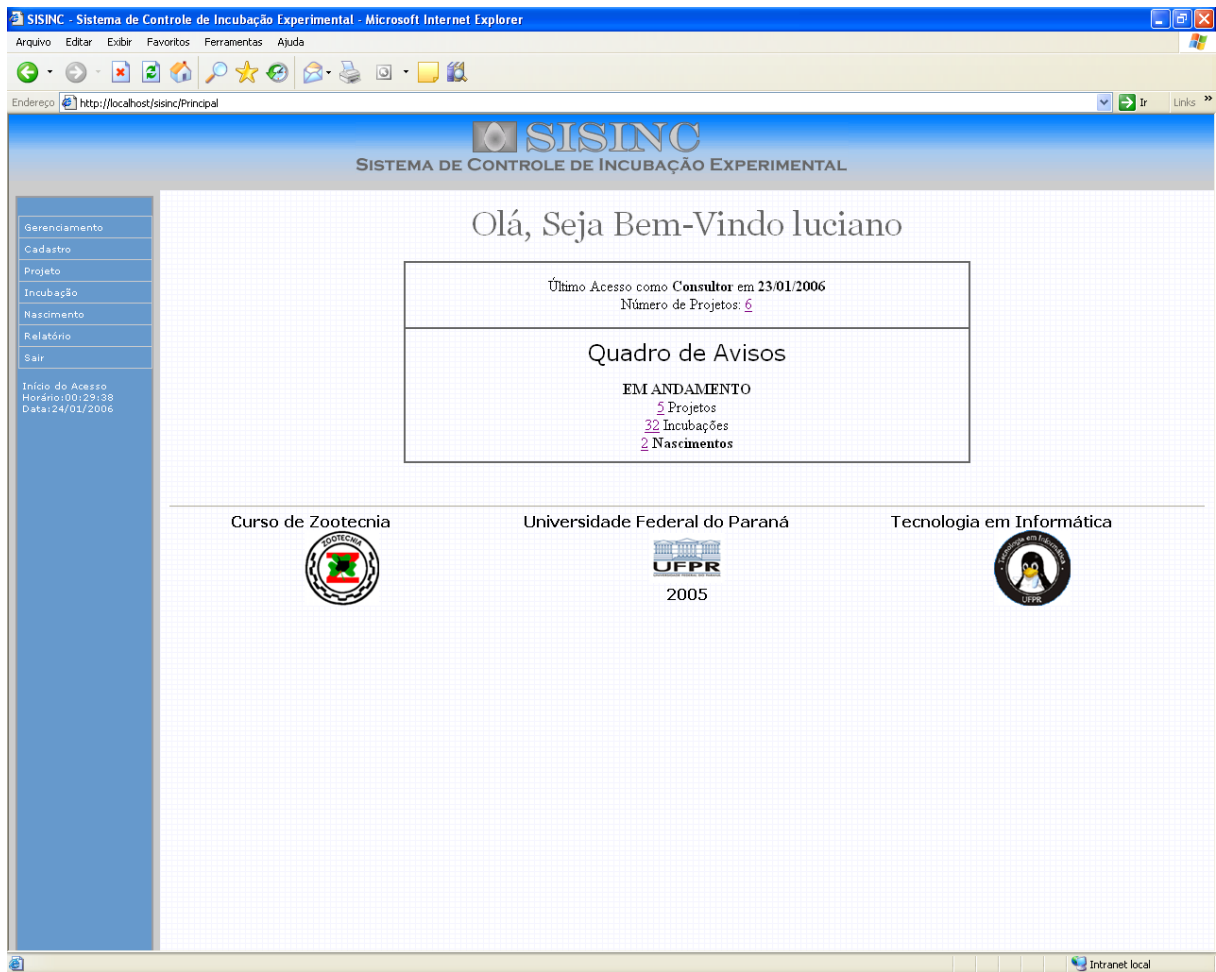


FIGURA 6 – TELA-B1

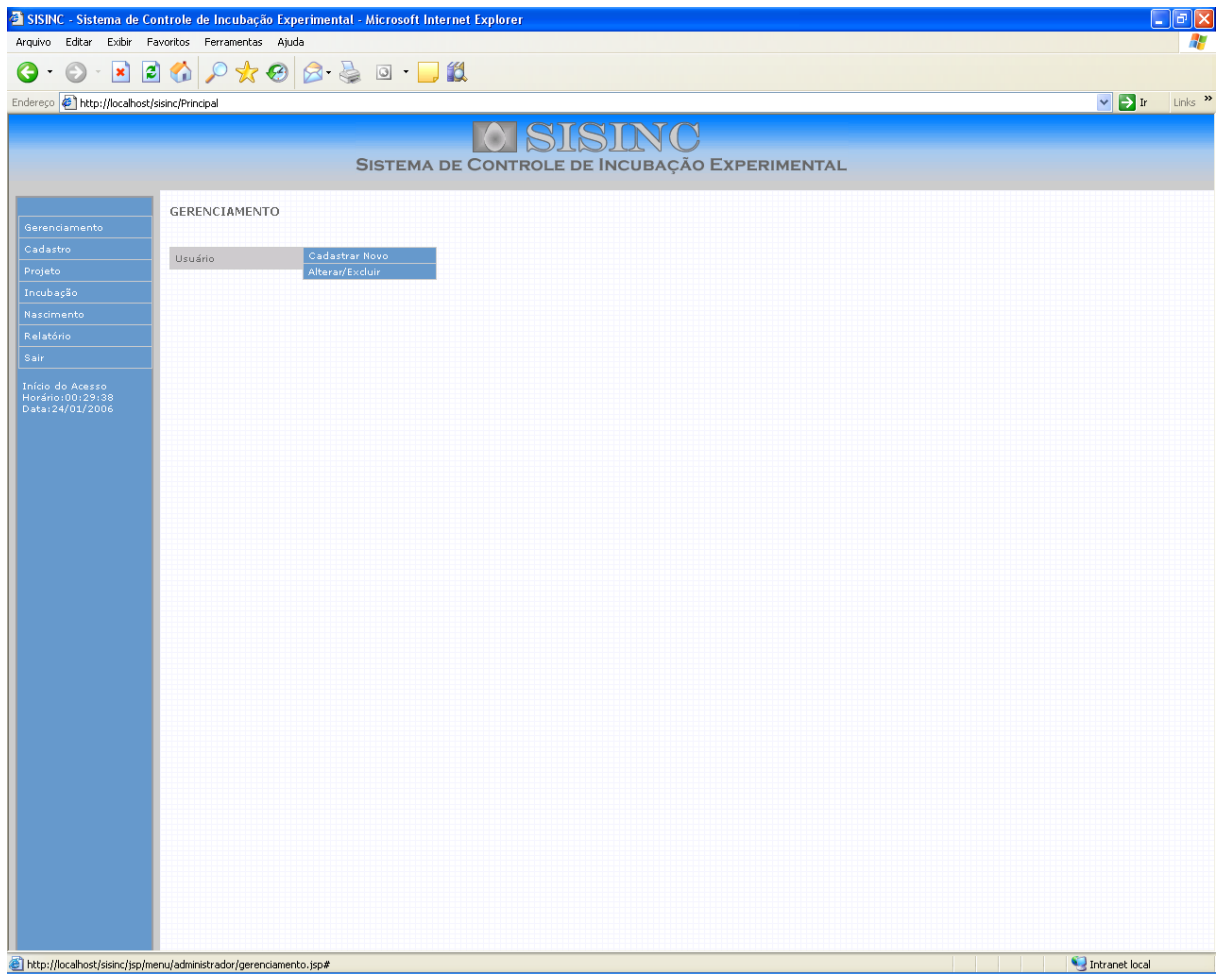


FIGURA 7 – TELA-B2

Principal \ [Gerenciamento](#)'. The browser status bar at the bottom shows 'Concluído' and 'Intranet local'."/>

SISINC - Sistema de Controle de Incubação Experimental - Microsoft Internet Explorer

Arquivo Editar Exibir Favoritos Ferramentas Ajuda

Endereço http://localhost/sisinc/Principal

SISINC
SISTEMA DE CONTROLE DE INCUBAÇÃO EXPERIMENTAL

Gerenciamento
Cadastro
Projeto
Incubação
Nascimento
Relatório
Sair

Início do Acesso
Horário: 00:29:38
Data: 24/01/2006

Cadastrar Novo Usuário

Matricula:

Nome:

Grupo:

Senha:
ATENÇÃO: Senha deve ter entre 6 e 8 caracteres.

Confirmar Senha:
ATENÇÃO: Por favor digitar novamente a Senha.

Telefone Residencial:

Telefone Comercial:

Telefone Celular:

E-mail:

Ir para: [Principal](#) \ [Gerenciamento](#)

Concluído Intranet local

FIGURA 8 – TELA-B3

SISINC - Sistema de Controle de Incubação Experimental - Microsoft Internet Explorer

Arquivo Editar Exibir Favoritos Ferramentas Ajuda

Endereço <http://localhost/sisinc/Principal> Ir Links

SISINC

SISTEMA DE CONTROLE DE INCUBAÇÃO EXPERIMENTAL

- Gerenciamento
- Cadastro
- Projeto
- Incubação
- Nascimento
- Relatório
- Sair

Início do Acesso
Horário: 00:29:38
Data: 24/01/2006

Alterar/Excluir Usuário

Matrícula:

Nome:

STATUS: Ativo

Nome	Matrícula
Administrador	1235641
luciano	2002
art	200506
luciano	123

Ir para: [Principal](#) \ [Gerenciamento](#)

Concluído Intranet local

FIGURA 9 – TELA-B4

The screenshot shows a web browser window titled "SISINC - Sistema de Controle de Incubação Experimental - Microsoft Internet Explorer". The address bar shows "http://localhost/sisinc/Principal". The page header features the SISINC logo and the text "SISTEMA DE CONTROLE DE INCUBAÇÃO EXPERIMENTAL".

On the left side, there is a vertical navigation menu with the following items: Gerenciamento, Cadastro, Projeto, Incubação, Nascimento, Relatório, and Sair. Below the menu, it displays "Início do Acesso", "Horário: 00:53:14", and "Data: 24/01/2006".

The main content area is titled "Alterar/Excluir Usuário". It contains a form with the following fields and values:

- Matricula: 2002
- Nome: luciano
- Grupo: Consultor (dropdown menu)
- Senha: (empty field) with a red warning: "ATENÇÃO: Senha deve ter entre 6 e 8 caracteres."
- Confirmar Senha: (empty field) with a red warning: "ATENÇÃO: Por favor digite novamente a Senha."
- Telefone Residencial: 231456
- Telefone Comercial: 456
- Telefone Celular: (empty field)
- E-mail: lucianoh@luciano
- STATUS: Inativo (dropdown menu)

At the bottom of the form, there are four buttons: "Alterar", "Excluir", "Voltar", and "Cadastrar Novo".

Below the buttons, there is a link: "Ir para: [Principal](#) \ [Gerenciamento](#)".

The status bar at the bottom left shows "Concluído" and the bottom right shows "Intranet local".

FIGURA 10 – TELA-C1

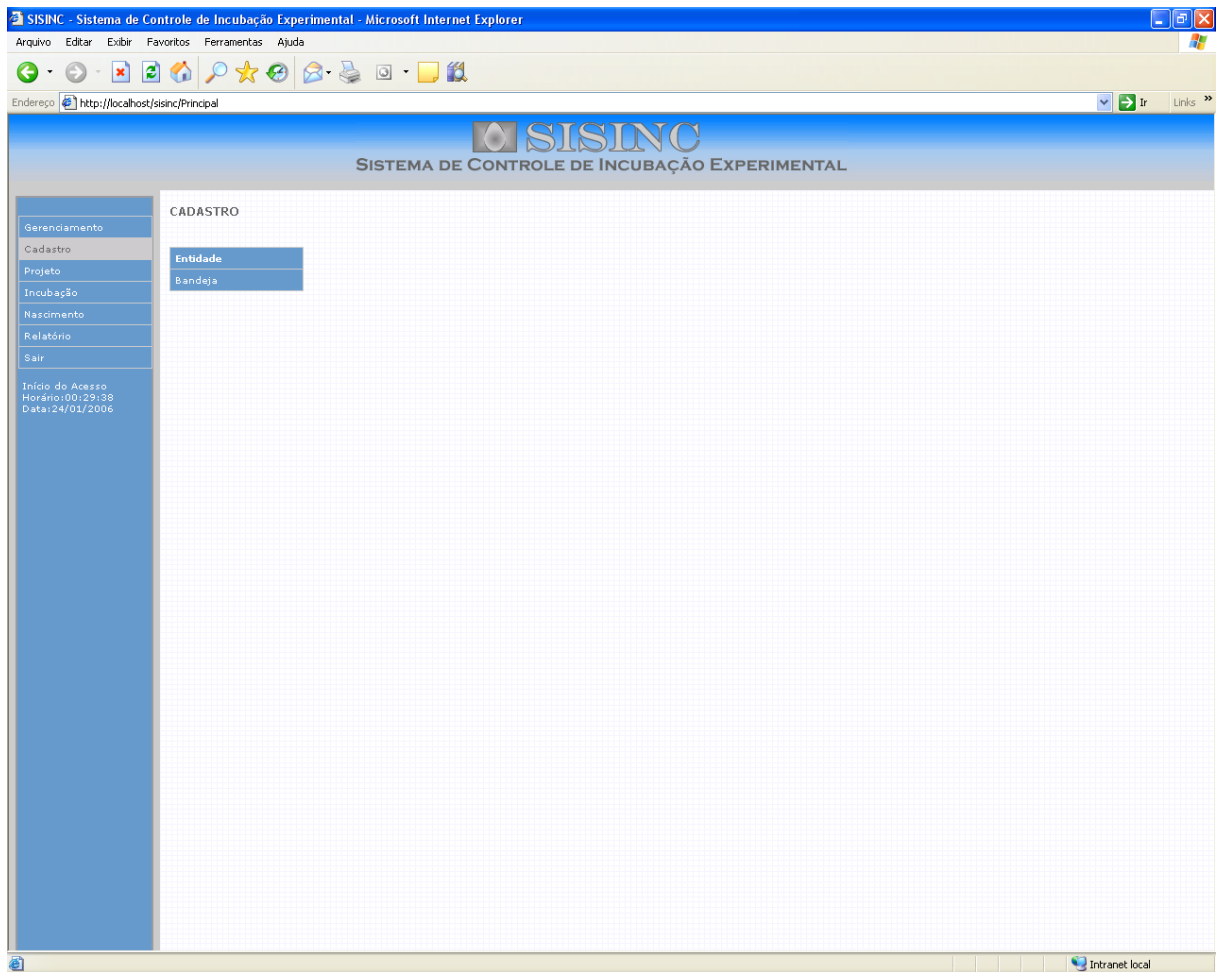


FIGURA 11 – TELA-C2

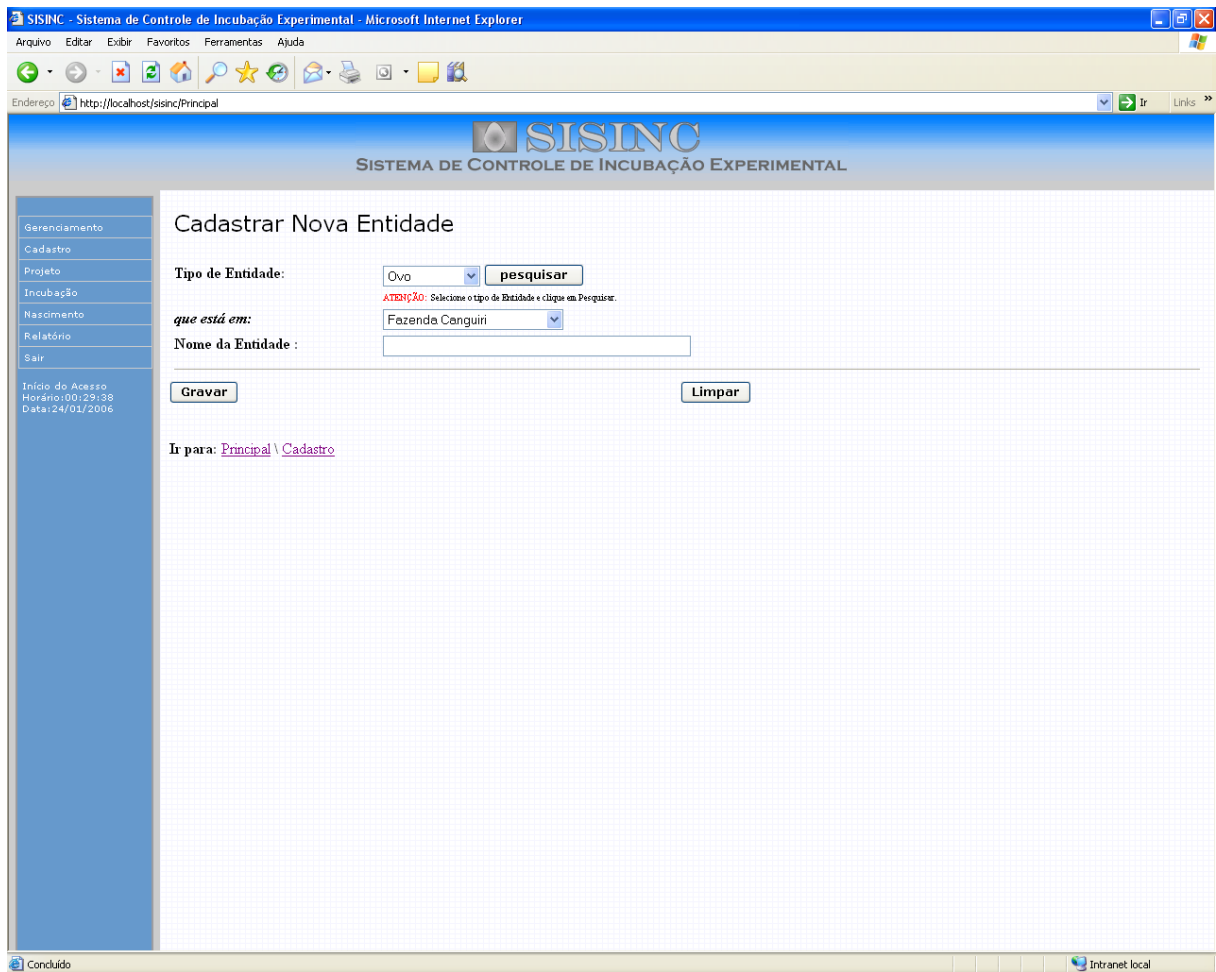


FIGURA 12 – TELA-C3

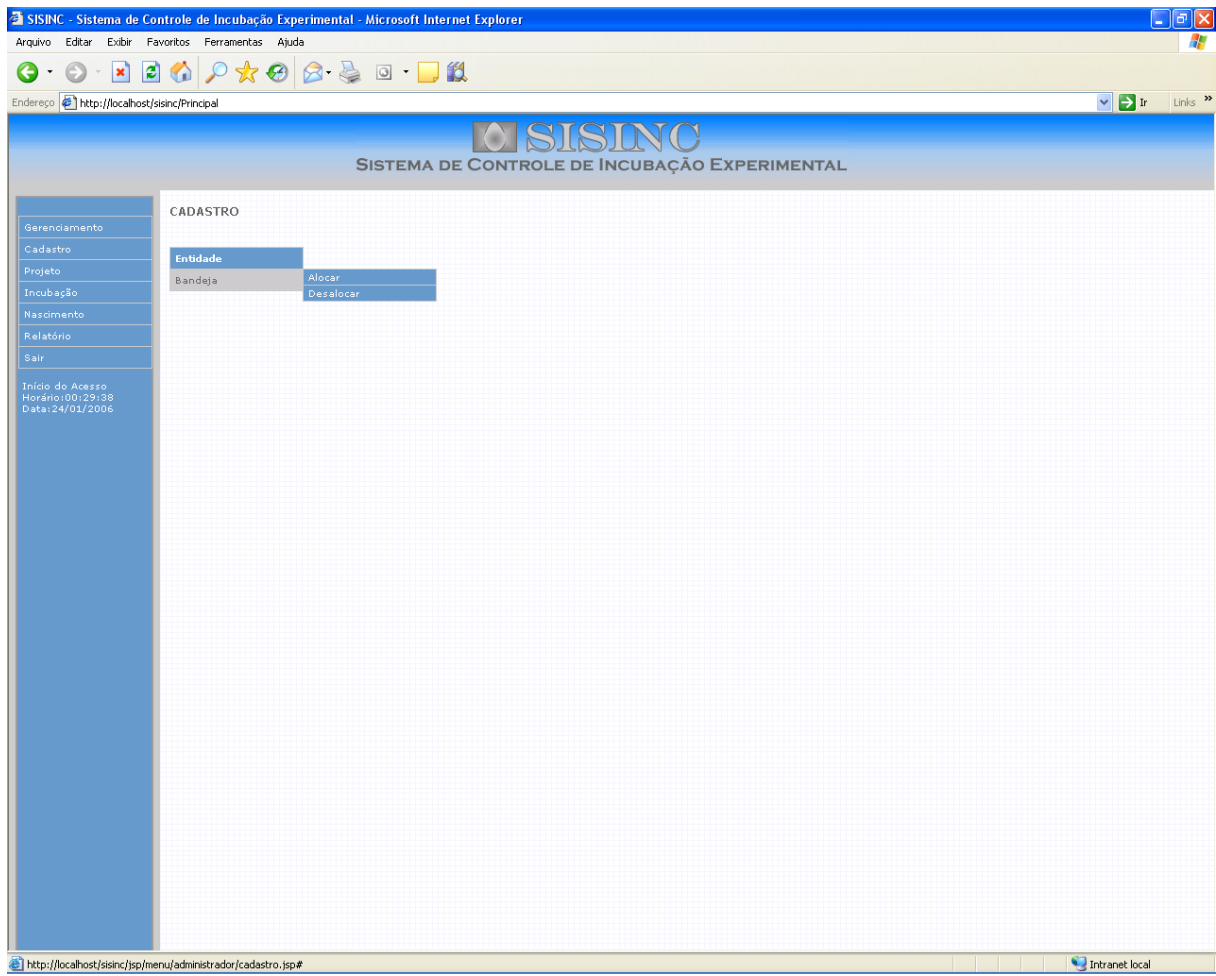


FIGURA 13 – TELA-C4

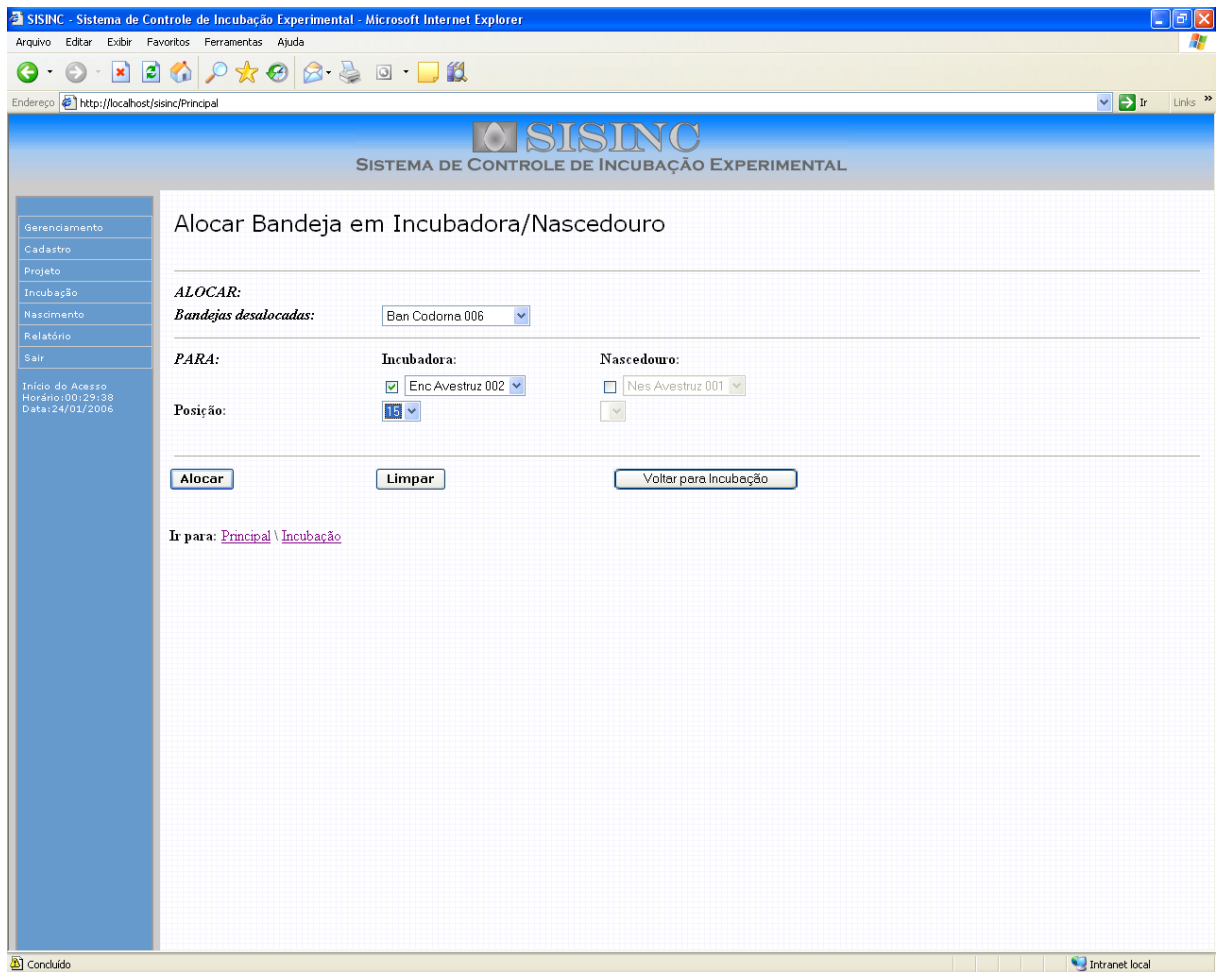


FIGURA 14 – TELA-C5

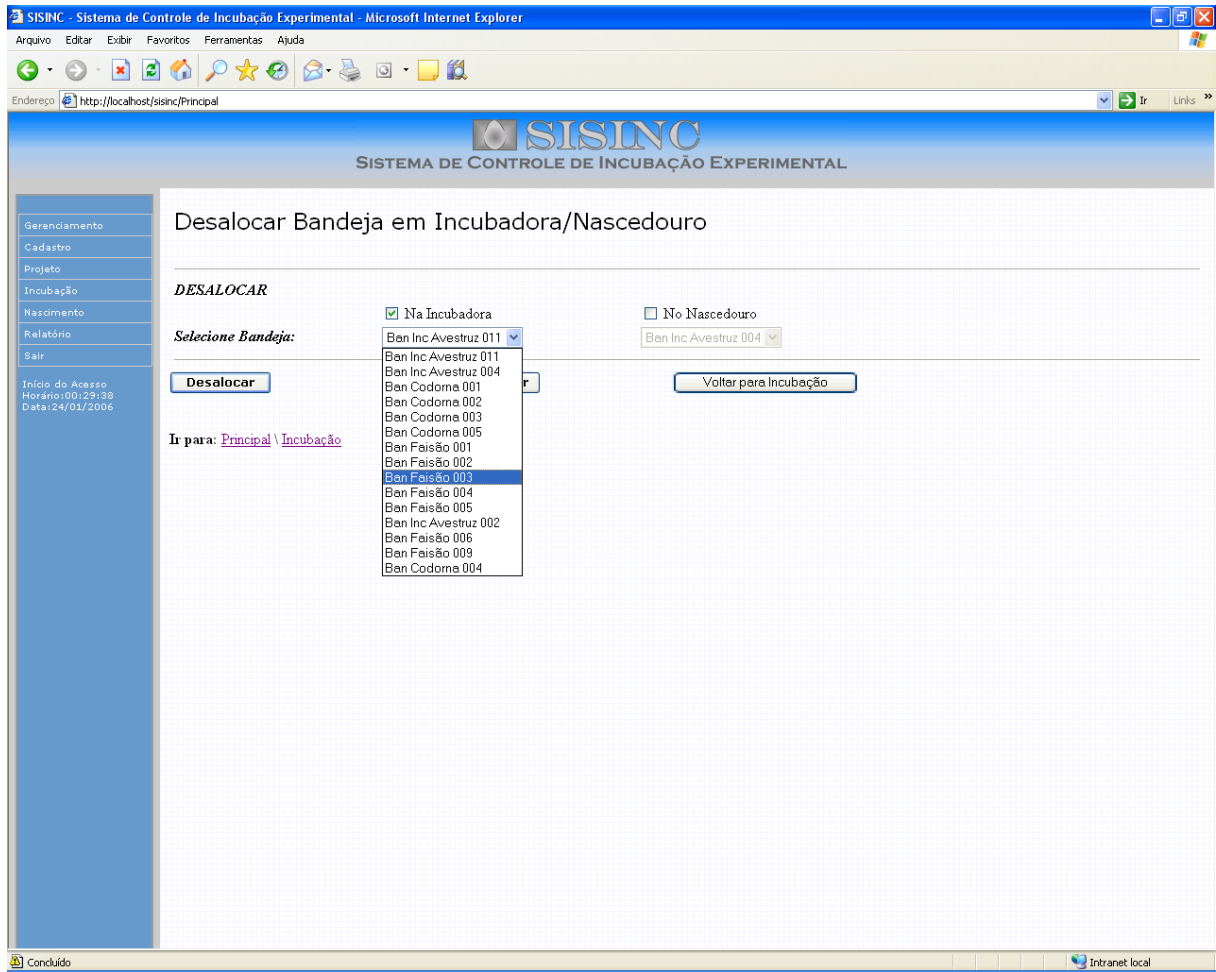


FIGURA 15 – TELA-D1

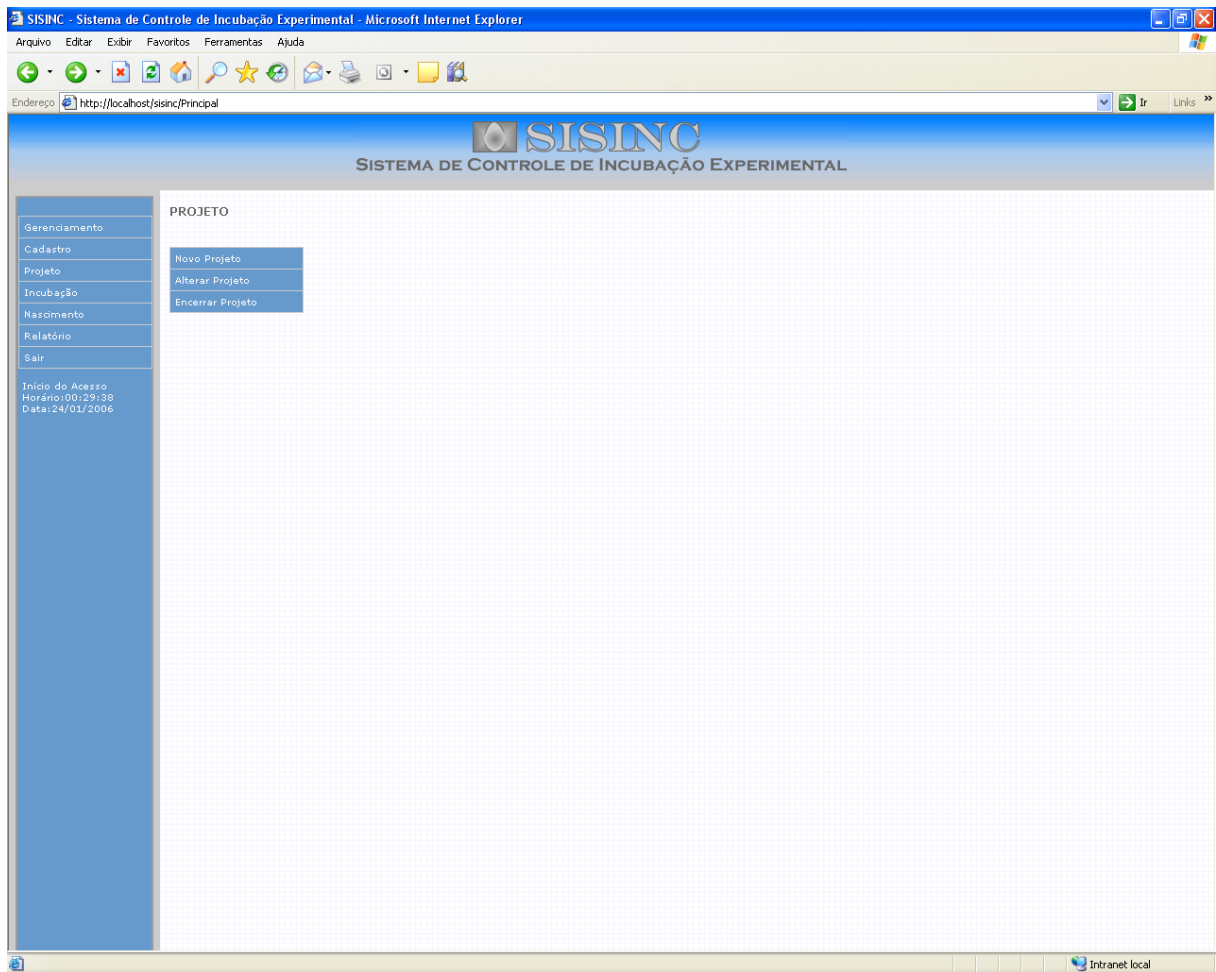


FIGURA 16 – TELA-D2

The screenshot shows a web browser window titled "SISINC - Sistema de Controle de Incubação Experimental - Microsoft Internet Explorer". The address bar shows "http://localhost/sisinc/Principal". The page header features the SISINC logo and the text "SISTEMA DE CONTROLE DE INCUBAÇÃO EXPERIMENTAL".

On the left side, there is a vertical navigation menu with the following items: Gerenciamento, Cadastro, Projeto, Incubação, Nascimento, Relatório, and Sair. Below the menu, it displays "Início do Acesso", "Horário: 00:29:38", and "Data: 24/01/2006".

The main content area is titled "Cadastrar Novo Projeto" and contains the following form fields:

- Sigla:**
- Nome:**
- Entidade:** CUIDADO: Define a entidade para tratamentos, repetições, incubações e nascimentos de **TODO** o projeto.
- Espécie de Ave:** CUIDADO: Define a espécie para tratamentos, repetições, incubações e nascimentos de **TODO** o projeto.
- Data de Início:**
- Previsão de Término:**
- Descrição:**

At the bottom of the form, there are two buttons: "Criar" and "Limpar". Below the buttons, there is a link: "Ir para: [Principal](#) \ [Projeto](#)".

The status bar at the bottom of the browser shows "Concluído" on the left and "Intranet local" on the right.

FIGURA 17 – TELA-D3

Nome:

data:

Nome	data de Início	data prevista de TÃ©rmino
projeto2	25/11/2005	12/03/2007
projeto4	25/11/2005	12/01/2007
projeto5	29/11/2005	08/07/2006
projeto7	29/11/2005	02/10/2009
projeto1	06/11/2005	12/10/2006
projeto01	20/11/2005	10/02/2006

Concluído Intranet local

FIGURA 18 – TELA-D4

The screenshot displays the SISINC web application interface within a Microsoft Internet Explorer browser window. The browser's address bar shows the URL `http://localhost/sisinc/Principal`. The application header features the SISINC logo and the title "SISTEMA DE CONTROLE DE INCUBAÇÃO EXPERIMENTAL".

On the left side, there is a vertical navigation menu with the following items: Gerenciamento, Cadastro, Projeto, Incubação, Nascimento, Relatório, and Sair. Below the menu, the access information is displayed: "Info do Acesso Horário: 00:53:14 Data: 24/01/2006".

The main content area contains a form for project registration with the following fields and values:

- Sigla:** 200508
- Nome:** projeto5
- Especie de Ave:** faisao (dropdown menu)
- Entidade Escolhida:** Geiola (dropdown menu)
- Data de Início:** 2005-11-29
- Previsão de Término:** 2006-07-08
- Descrição:** FAISÕES TRATAMENTO A BASE DE PROTEINA, EM DIFERENTES QUANTIDADES, E TAMBEM DIFERENTES REPETIÇÕES COM RELAÇÃO AO CICLO DE POSTAGEM. (text area)
- Número de Incubações:** (empty text field)
- STATUS do Projeto:** Ativo (dropdown menu)

At the bottom of the form, there are three buttons: "alterar", "excluir", and "voltar". The browser's status bar at the bottom right indicates "Intranet local".

FIGURA 19 – TELA-E1

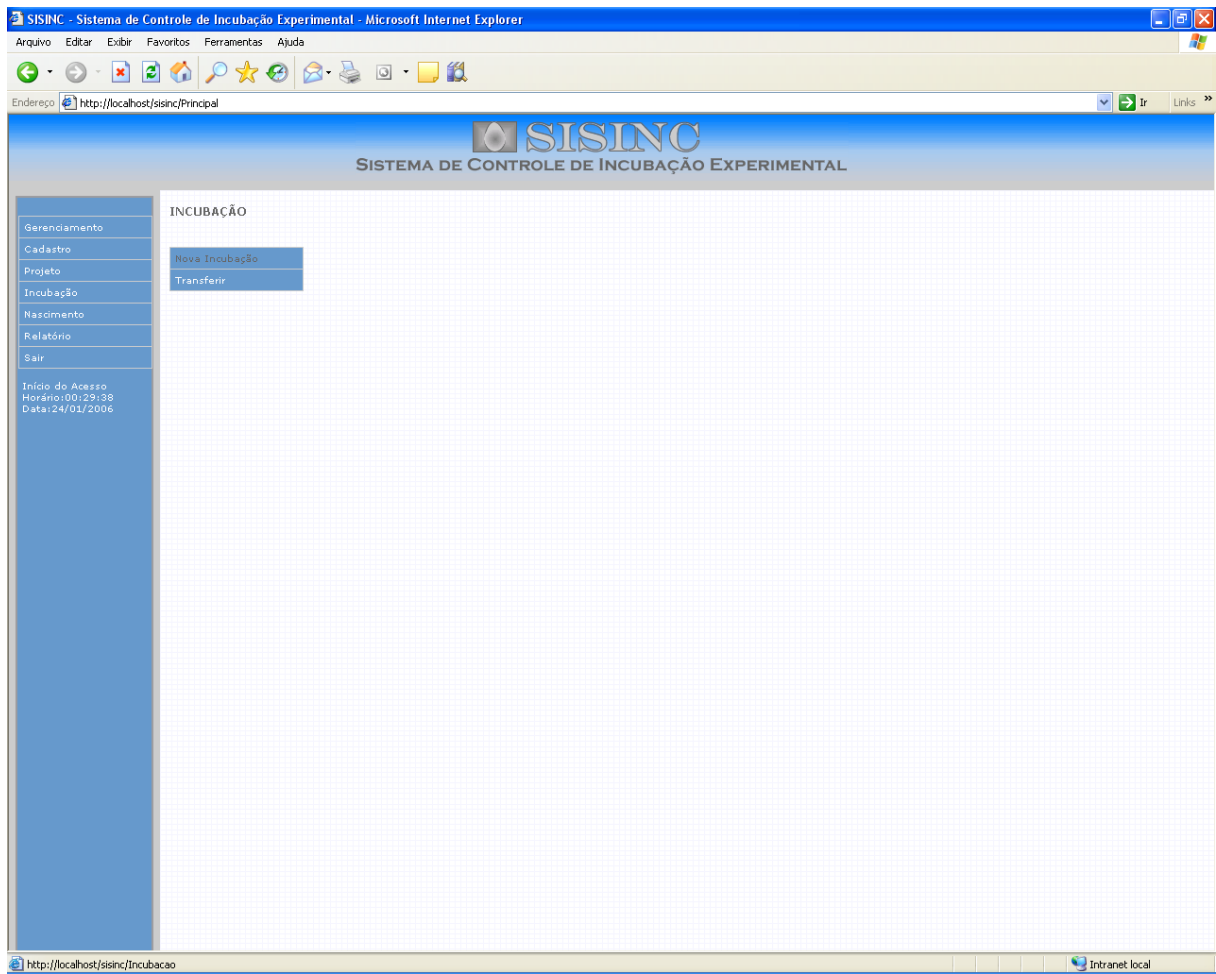


FIGURA 20 – TELA-E2

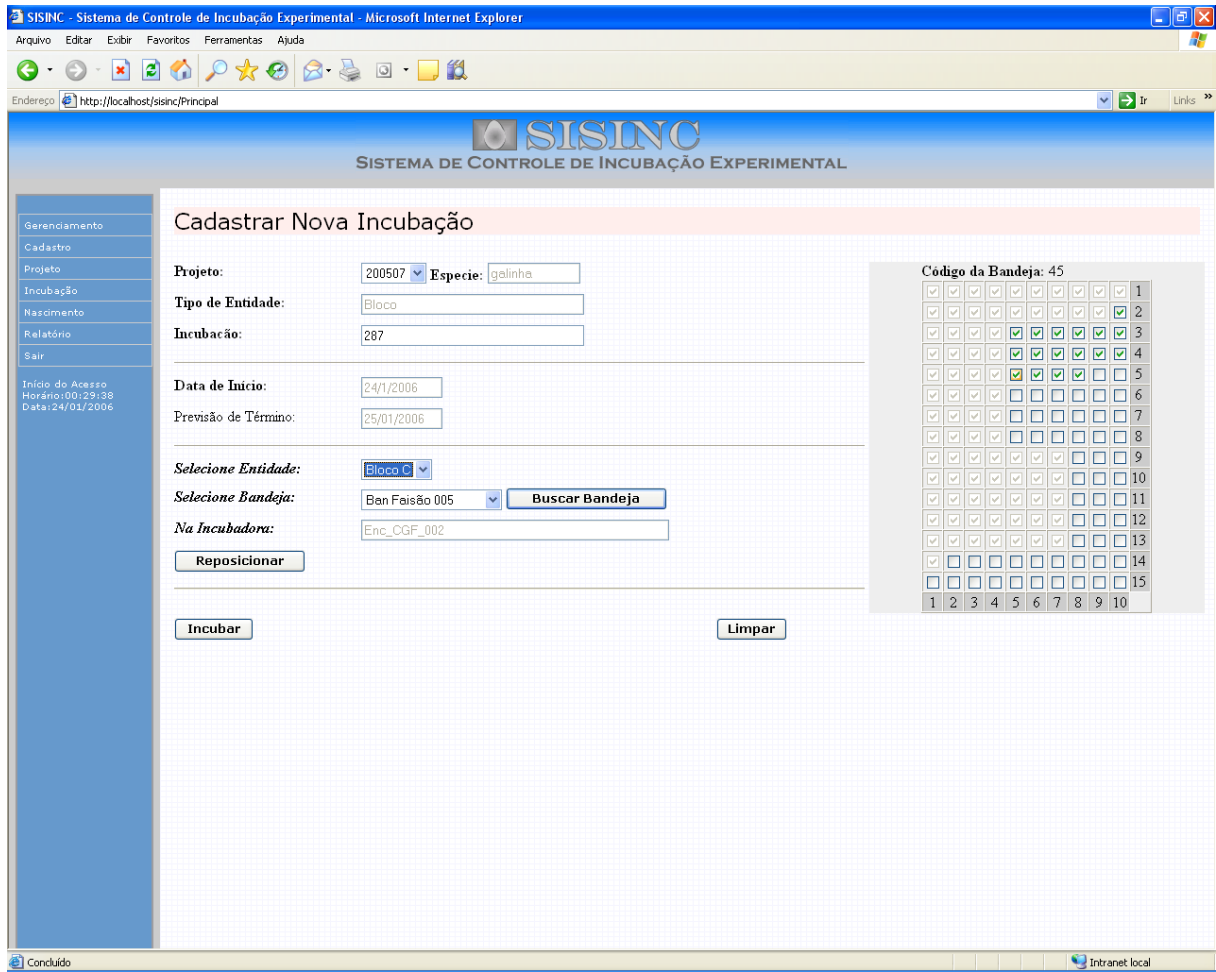


FIGURA 21 – TELA-F1

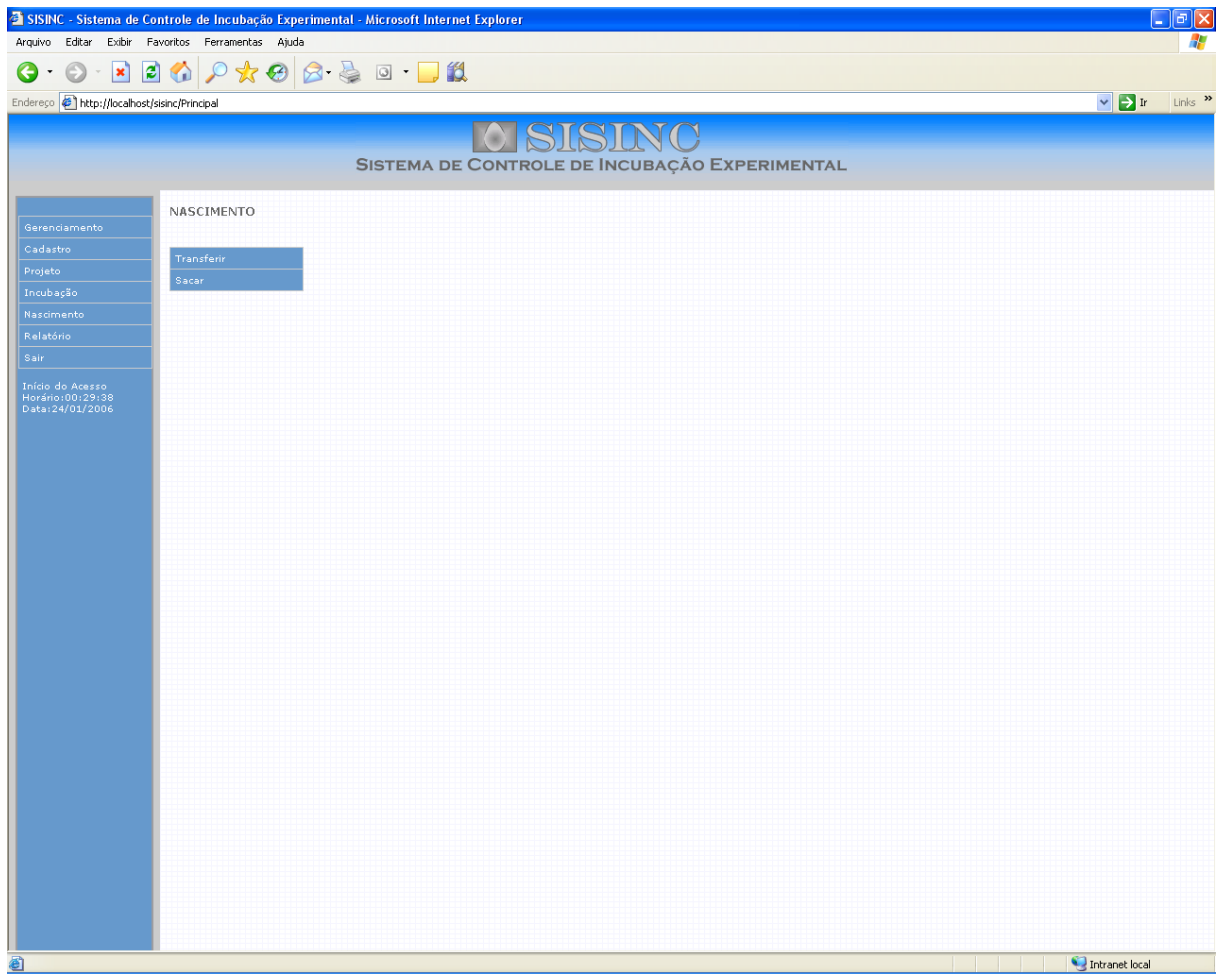


FIGURA 22 – TELA-F2

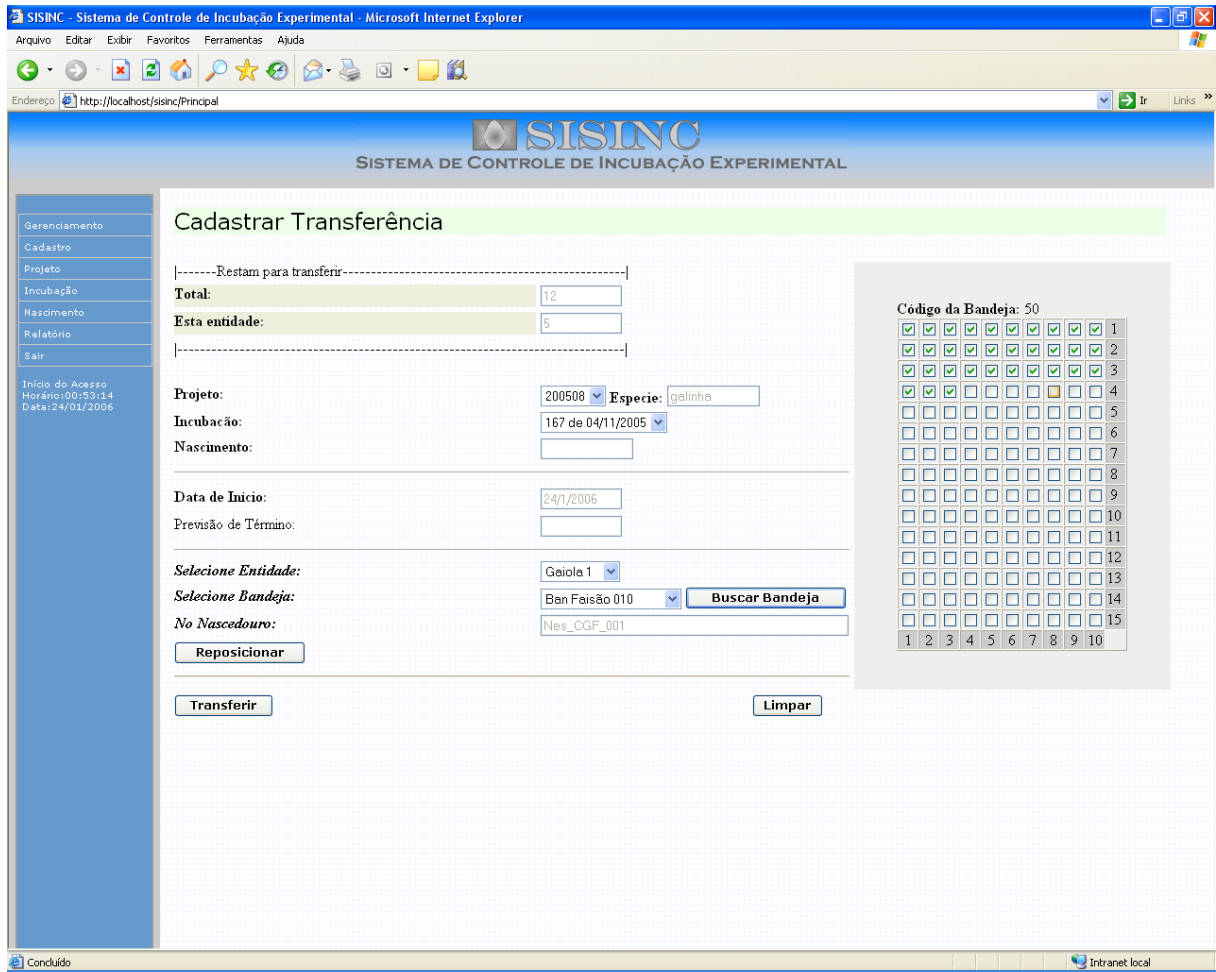


FIGURA 23 – TELA-F3

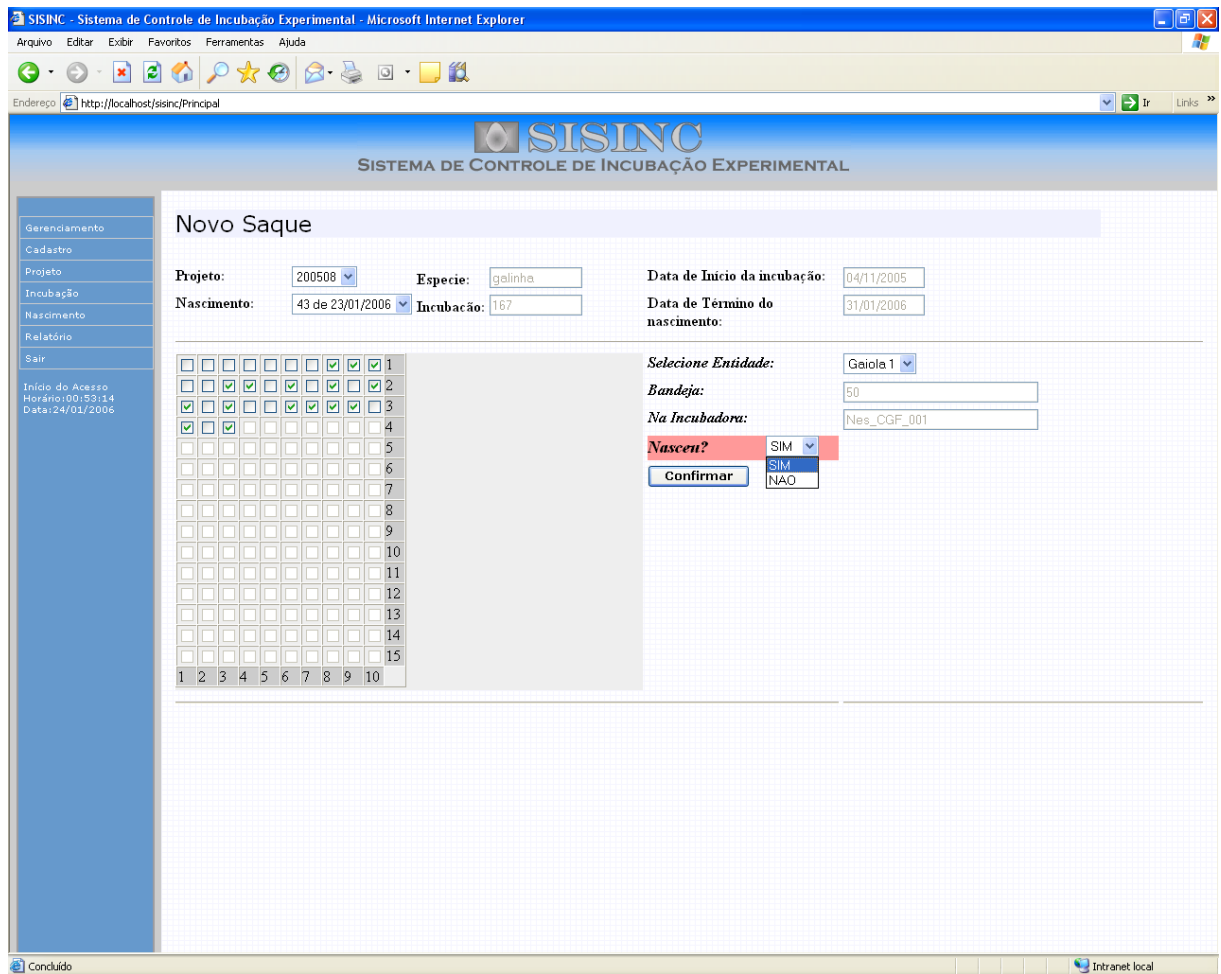


FIGURA 24 – TELA-G1

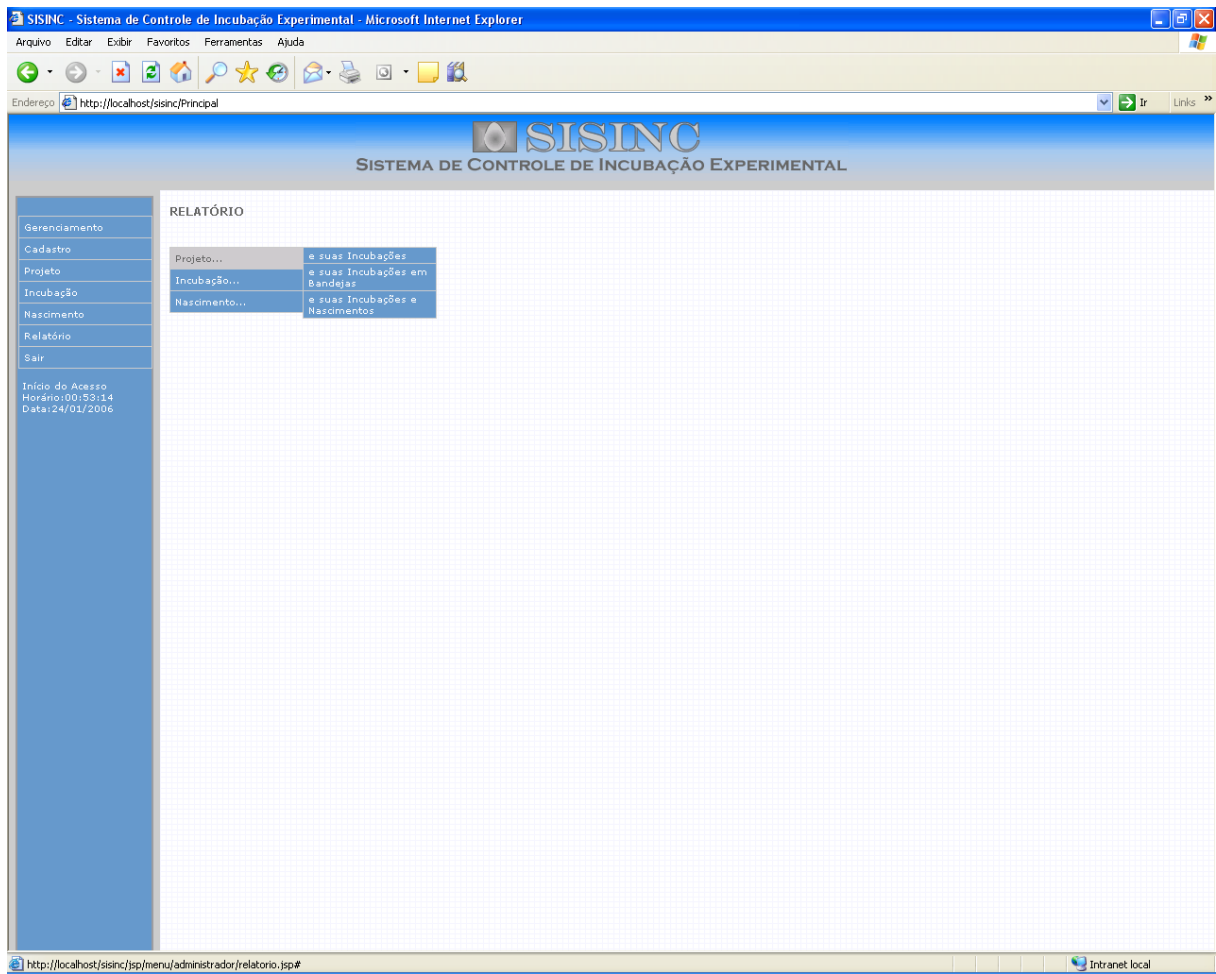


FIGURA 25 – TELA-G2

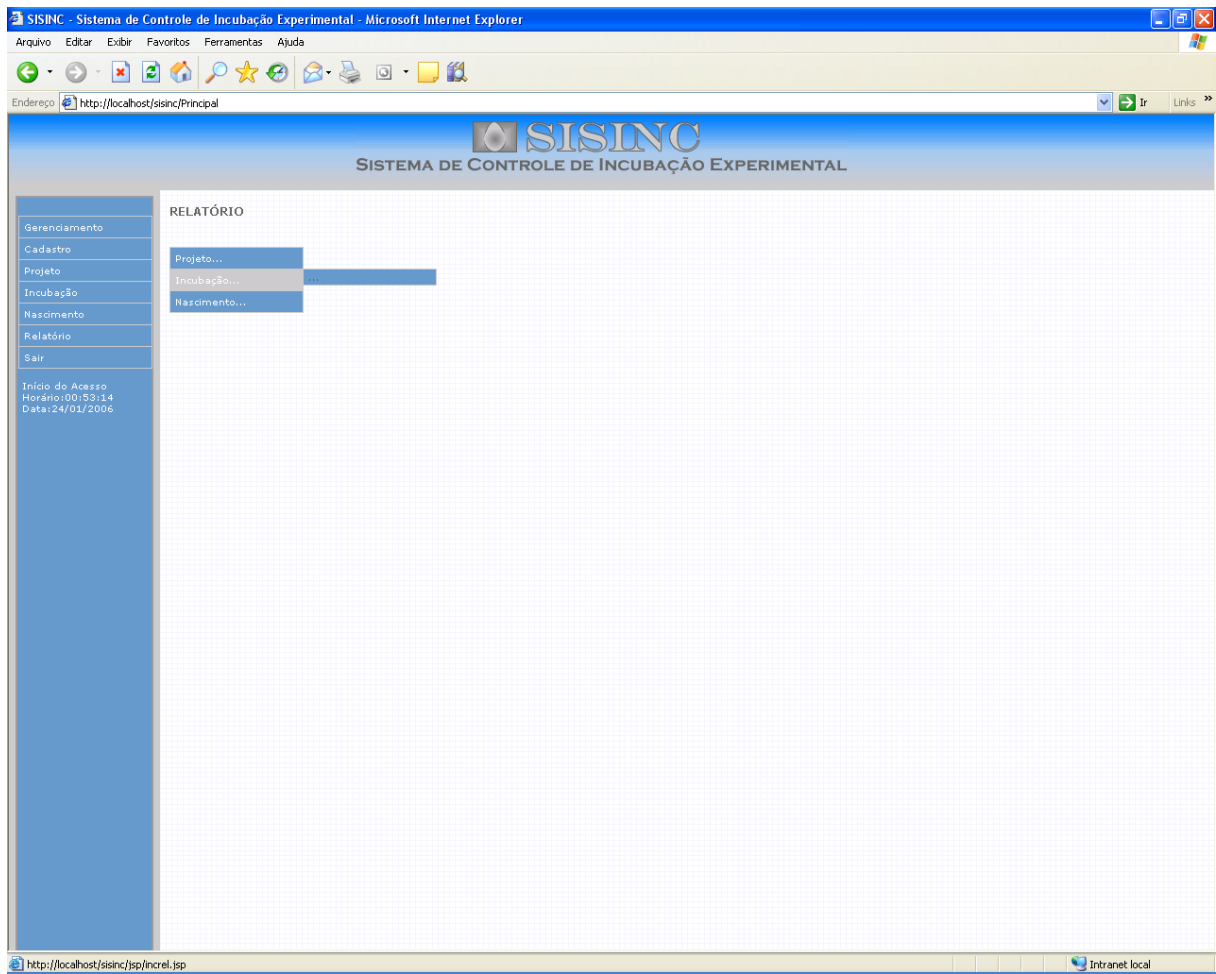


FIGURA 26 – TELA-G3

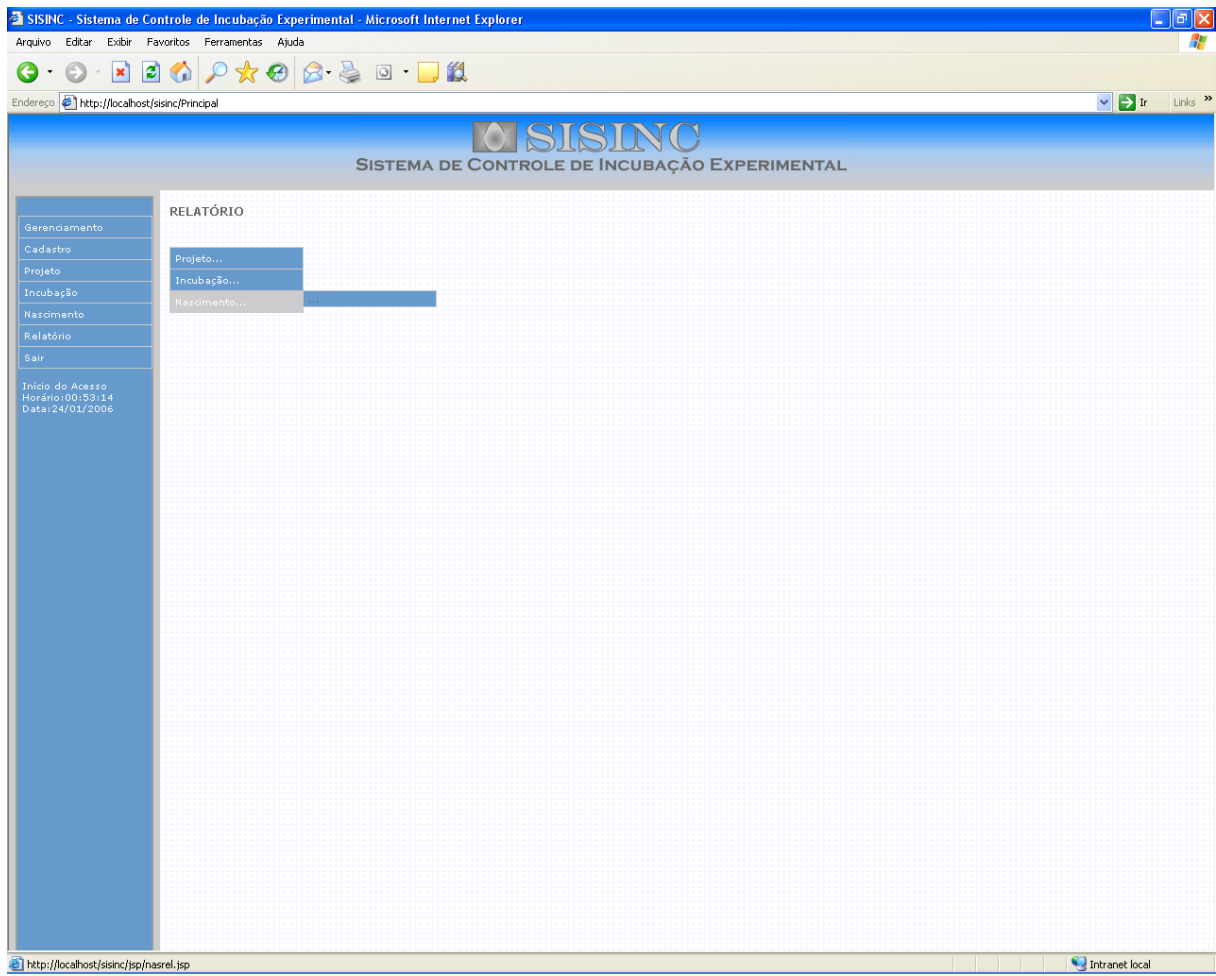
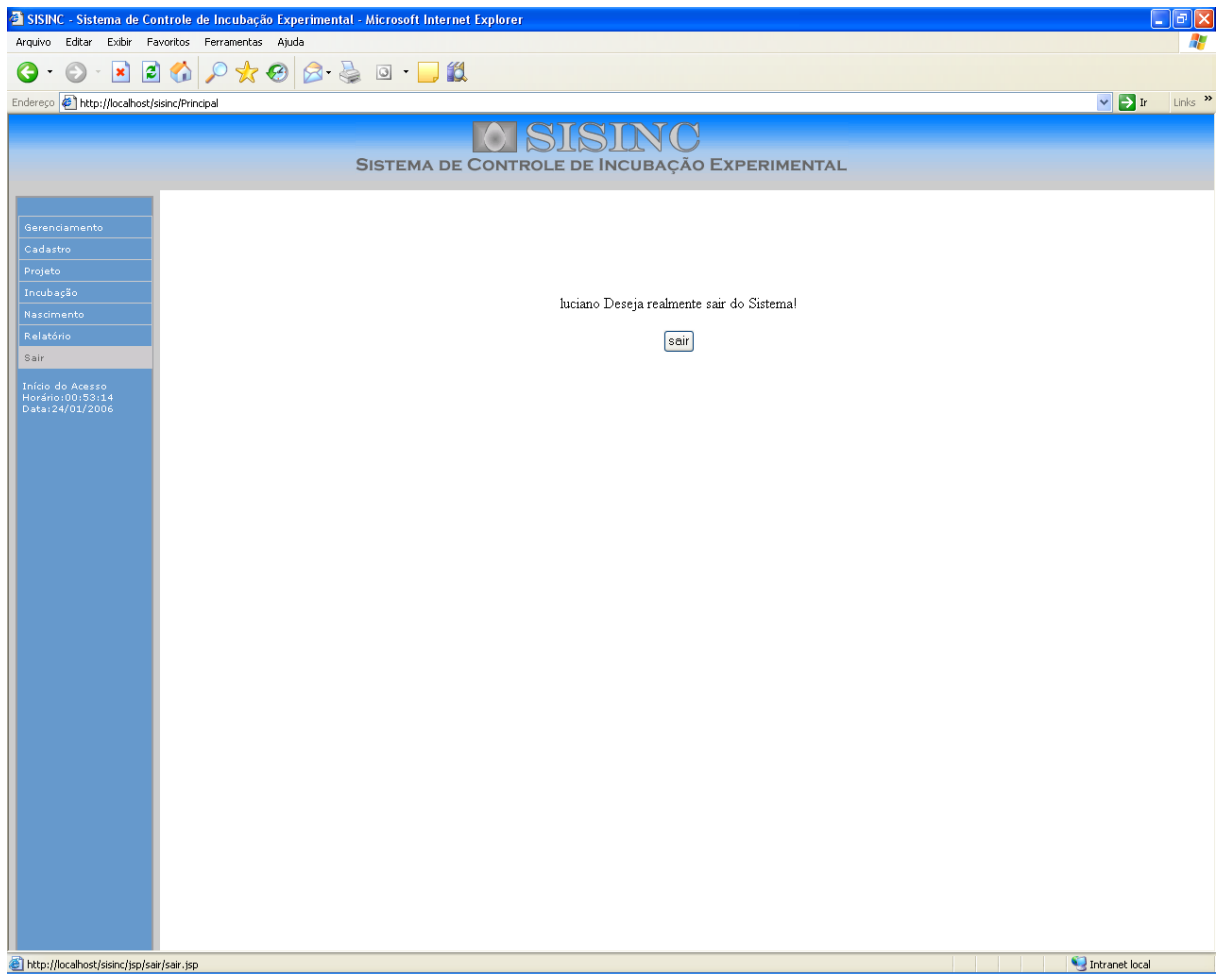


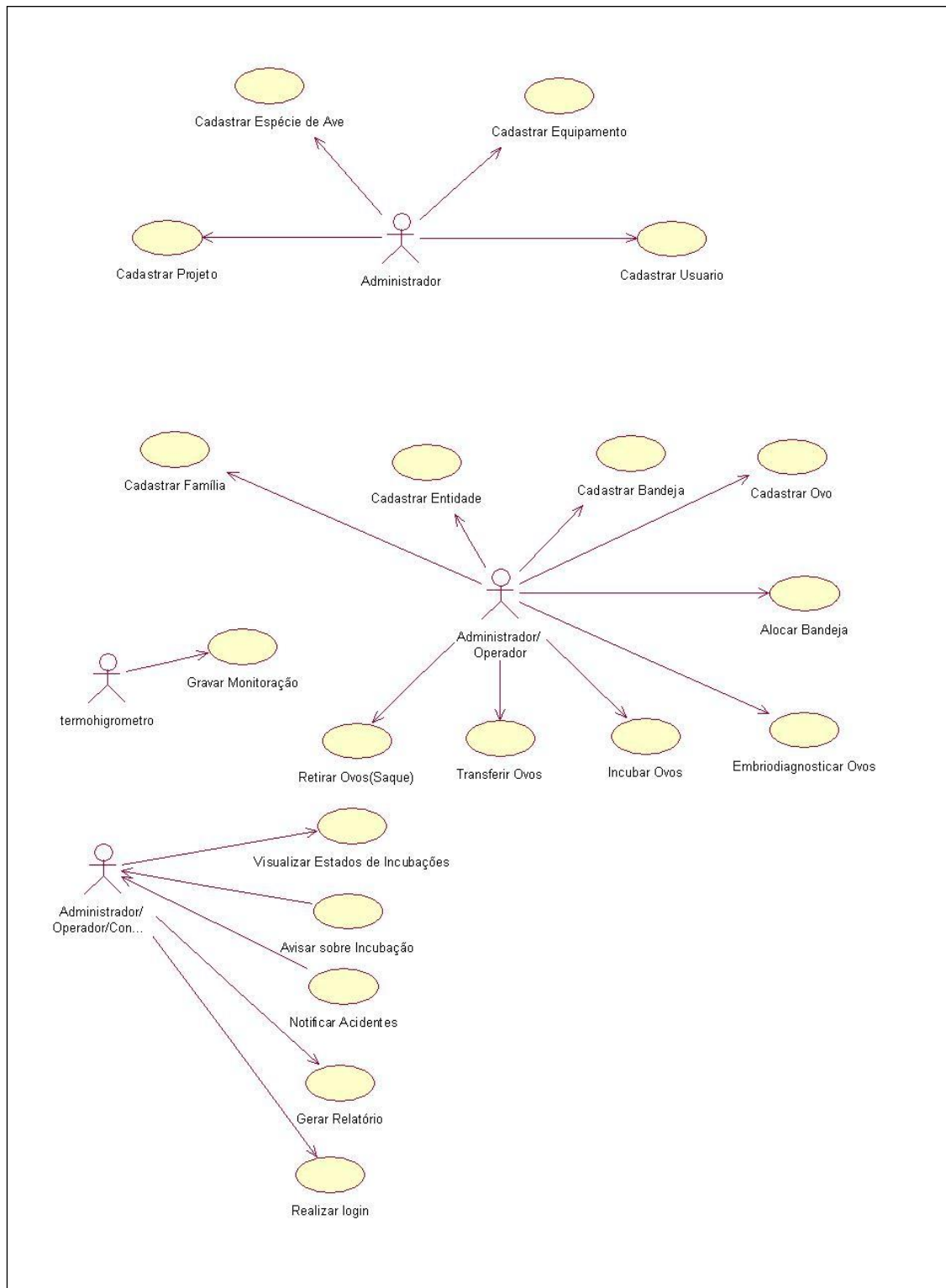
FIGURA 27 – TELA-H1



8 MODELAGEM

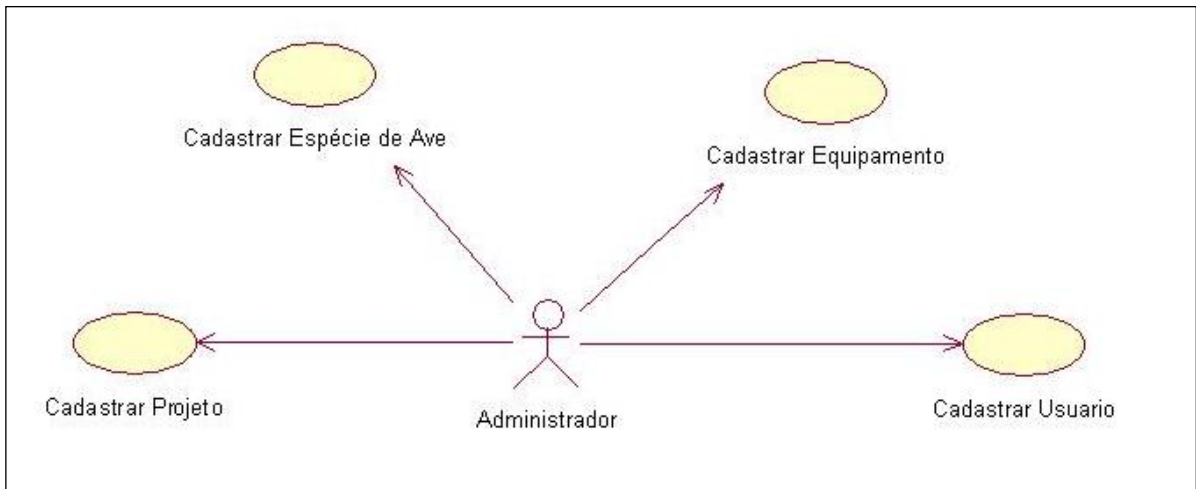
8.1 CASOS DE USO

FIGURA 28 – CASOS DE USO



8.1.1 Descrição Resumida

FIGURA 29 – ATOR ADMINISTRADOR



8.1.1.1 Cadastrar Projeto

Este caso de uso tem como objetivo a criação do Projeto, que corresponde à entidade responsável por armazenar informações associadas a um determinado estudo realizado durante um período de tempo determinado. Este caso de uso fica a cargo do ator denominado Administrador.

8.1.1.2 Cadastrar Espécie de Ave

Este caso de uso tem como objetivo a criação de espécies de aves, além do armazenamento das informações pertinentes, como tempos e umidades de incubação e nascimento. Este caso de uso fica a cargo do ator denominado Administrador.

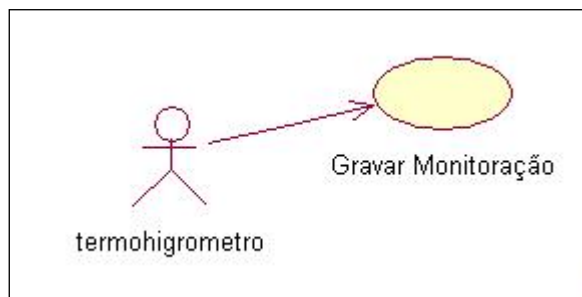
8.1.1.3 Cadastrar Equipamento

Este caso de uso tem como objetivo o cadastro de equipamentos, que correspondem às incubadoras e nascedouros da fazenda. Este caso de uso fica a cargo do ator denominado Administrador.

8.1.1.4 Cadastrar Usuário

Este caso de uso tem como objetivo o cadastramento de usuários que terão acesso disponível ao sistema, além da definição do papel a ser desempenhado por cada usuário. Este caso de uso fica a cargo do ator denominado Administrador.

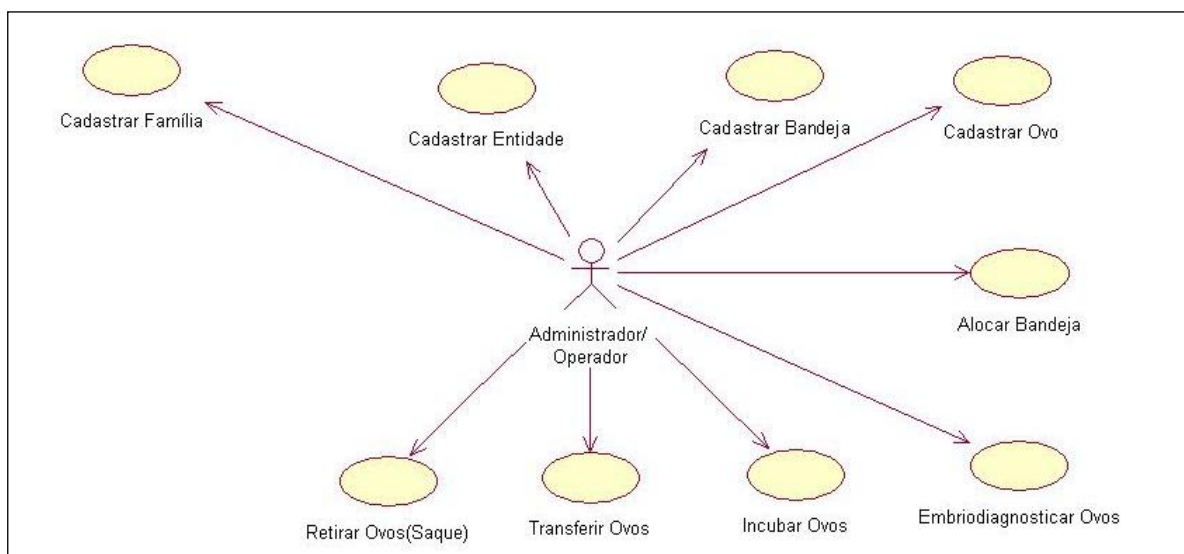
FIGURA 30 – ATOR TERMOHIGRÔMETRO



8.1.1.5 Gravar Monitoração

Este caso de uso tem como objetivo a gravação da monitoração realizada pelo equipamento denominado termohigrômetro, monitoração esta que corresponde à temperatura e umidade do equipamento (incubadora ou nascedouro) em determinado momento (data e hora). Este caso de uso fica a cargo do ator denominado termohigrômetro.

FIGURA 31 – ATOR ADMINISTRADOR/OPERADOR



8.1.1.6Cadastrar Família

Este caso de uso tem como objetivo o cadastramento de famílias, que correspondem ao agrupamento de machos e fêmeas (usualmente um macho para uma ou duas fêmeas) em determinada célula para que sejam rastreadas informações sobre produção de ovos crias, quando aplicável no dado projeto. Este caso de uso fica a cargo do ator denominado Administrador/Operador.

8.1.1.7Cadastrar Entidade

Este caso de uso tem como objetivo o cadastramento de entidades, que corresponde às propriedades, prédios, andares, salas, blocos, linhas, gaiolas, células e ovos a serem rastreados no sistema. Este caso de uso fica a cargo do ator denominado Administrador/Operador.

8.1.1.8Cadastrar Bandeja

Este caso de uso tem como objetivo o cadastramento das bandejas, onde são armazenados os ovos durante as incubações e nascimentos. Este caso de uso fica a cargo do ator denominado Administrador/Operador.

8.1.1.9Cadastrar Ovo

Este caso de uso tem como objetivo o cadastramento dos ovos, quando aplicável a determinado projeto. Nos projetos atualmente desenvolvidos somente ovos de avestruz são cadastrados individualmente. Este caso de uso fica a cargo do ator denominado Administrador/Operador.

8.1.1.10 Alocar Bandeja

Este caso de uso tem como objetivo o cadastramento das alocações de

bandeja, ou seja, associações de bandejas a dado equipamento (incubadoras ou nascedouros). Esta associação visa o início do processo de incubação ou nascimento do conjunto de ovos. Este caso de uso fica a cargo do ator denominado Administrador/Operador.

8.1.1.11 Embriodiagnosticar Ovo

Este caso de uso tem como objetivo o embriodiagnóstico dos ovos, ou seja, a análise do motivo pelo qual os ovos não geraram cria (vide figura 32). Este caso de uso fica a cargo do ator denominado Administrador/Operador.

FIGURA 32 – EMBRIODIAGNOSTICAR OVO



8.1.1.12 Incubar Ovos

Este caso de uso tem como objetivo a incubação dos conjuntos de ovos, para o dado projeto. A incubação de ovos marca o início do rastreamento das informações da dada entidade (vide figura 33). Este caso de uso fica a cargo do ator denominado Administrador/Operador.

FIGURA 33 – BANDEJA COM OVOS



8.1.1.13 Transferir Ovos

Este caso de uso tem como objetivo registrar a transferência do conjunto de ovos da incubadora para o nascedouro, onde os ovos devem permanecer até o momento da eclosão (caso gerem cria) ou do embriodiagnóstico (caso não gerem cria). Este caso de uso fica a cargo do ator denominado Administrador/Operador.

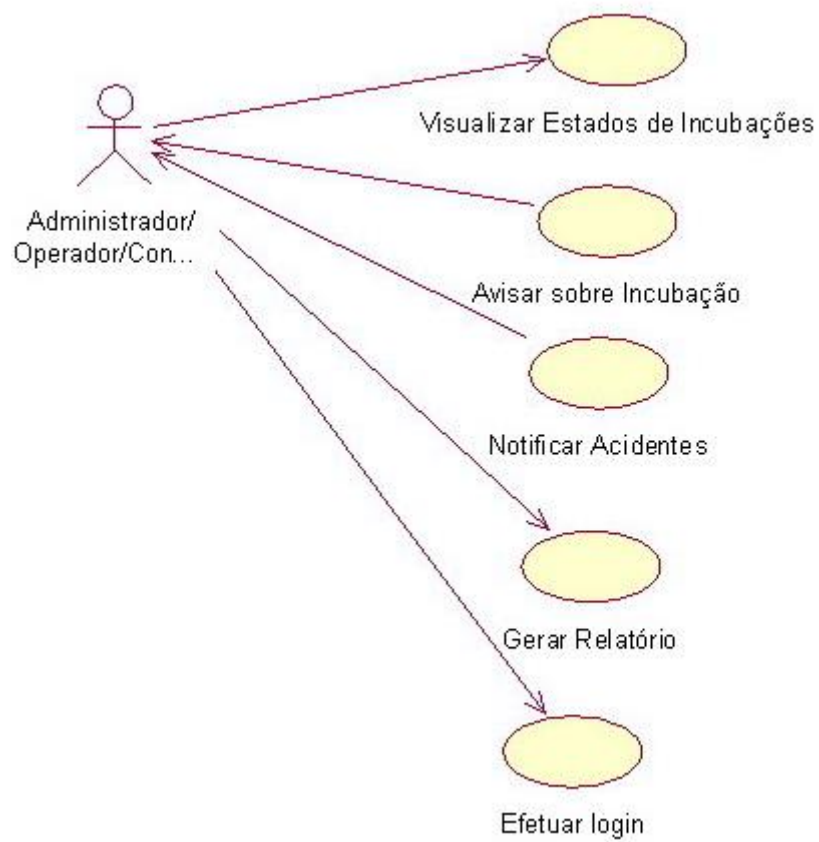
8.1.1.14 Retirar Ovos (Saque)

Este caso de uso tem como objetivo registrar o saque das crias ao final do processo de nascimento, realizado no equipamento denominado nascedouro. Este caso de uso fica a cargo do ator denominado Administrador/Operador.

FIGURA 34 – OVO RETIRADO (SAQUE)



FIGURA 35 – ATOR ADMINISTRADOR/OPERADOR/CONSULTOR



8.1.1.15 Visualizar Estado de Incubações

Este caso de uso tem como objetivo a visualização do estado das incubações realizadas e informações associadas, para um dado projeto. Este caso de uso fica a cargo do ator denominado Administrador/Operador/Consultor.

8.1.1.16 Visualizar sobre Incubação

Este caso de uso tem como objetivo o aviso do final do período de incubação de um conjunto de ovos. Este caso de uso fica a cargo do ator denominado Administrador/Operador/Consultor.

8.1.1.17 Notificar Acidentes

Este caso de uso tem como objetivo o registro de acidentes com conjunto de ovos, como queda que venham a impactar na produtividade rastreada de determinado projeto. Este caso de uso fica a cargo do ator denominado Administrador/Operador/Consultor.

8.1.1.18 Gerar Relatórios

Este caso de uso tem como objetivo a geração de relatórios de acompanhamento do projeto, através das entidades disponíveis. Este caso de uso fica a cargo do ator denominado Administrador/Operador/Consultor.

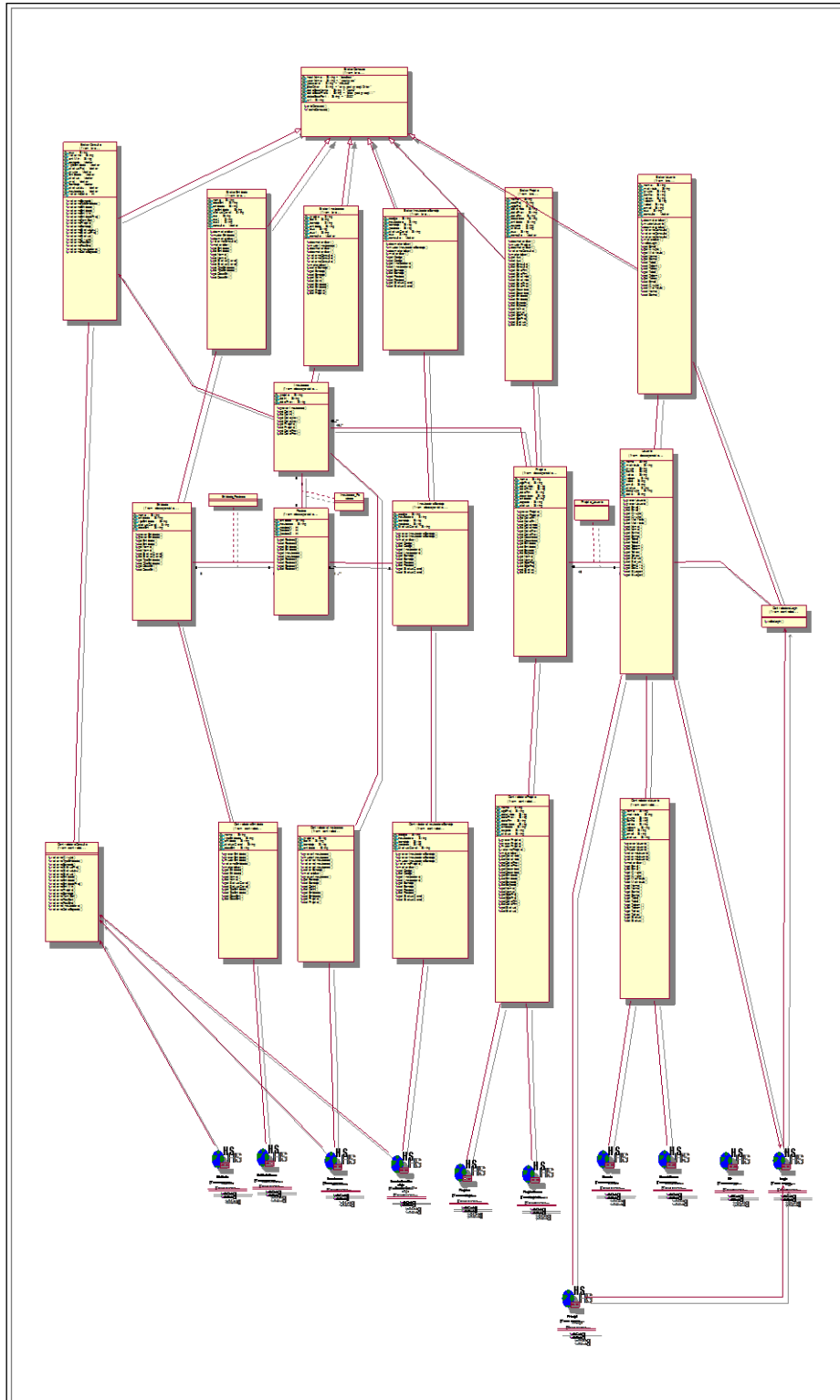
8.1.1.19 Efetuar Login

Este caso de uso tem como objetivo o login de usuários previamente cadastrados no sistema, para que sejam verificadas suas permissões e funcionalidades disponíveis de acordo com o grupo associado. Este caso de uso fica a cargo do ator denominado Administrador/Operador/Consultor.

8.2 DIAGRAMAS DE CLASSE

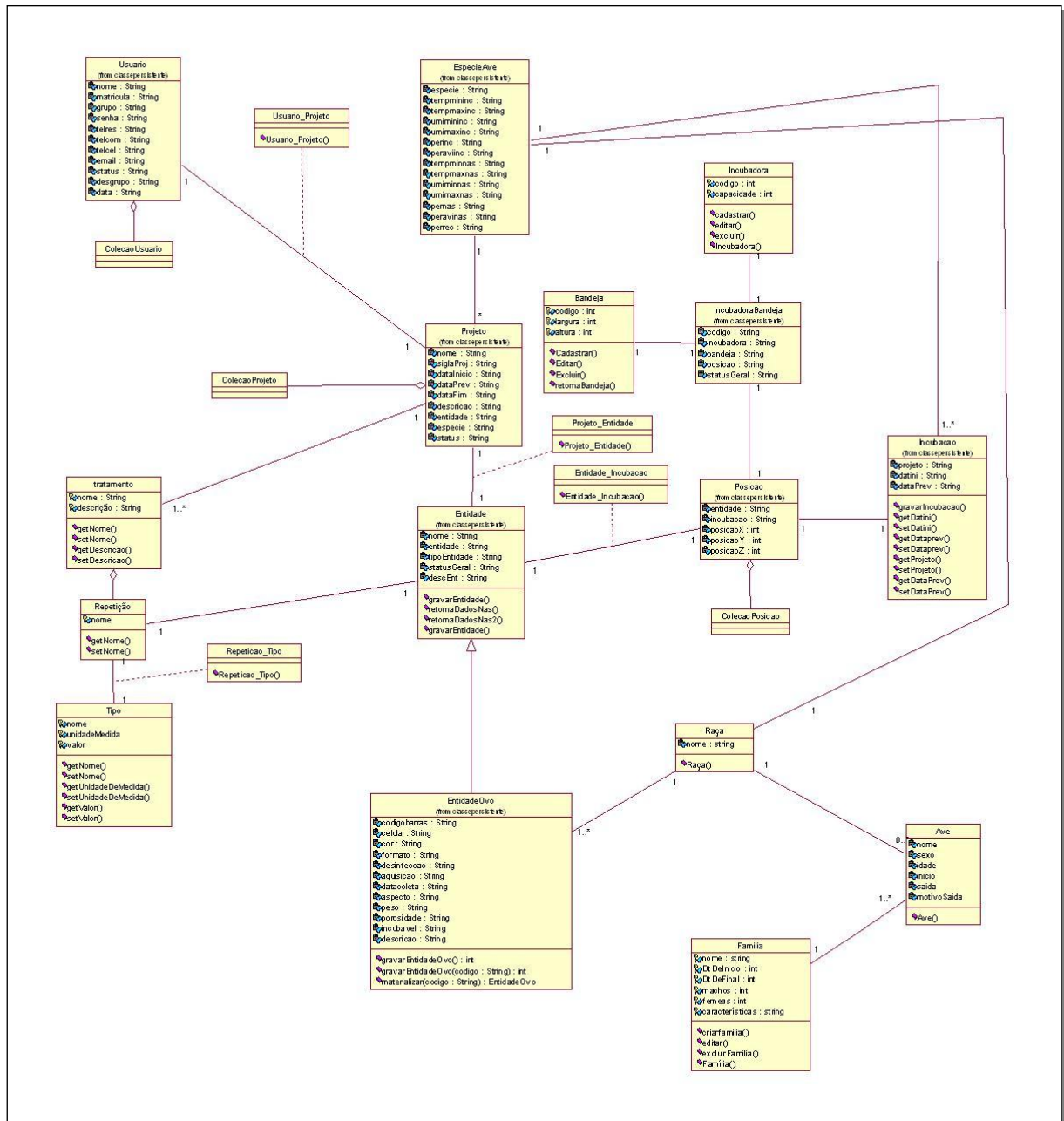
8.2.1 Modelo Físico

FIGURA 36 – DIAGRAMA DE CLASSE DO MODELO FÍSICO



8.2.2 Modelo Lógico

FIGURA 37 – DIAGRAMA DE CLASSE DO MODELO LÓGICO



8.3 DIAGRAMAS DE INTERAÇÃO

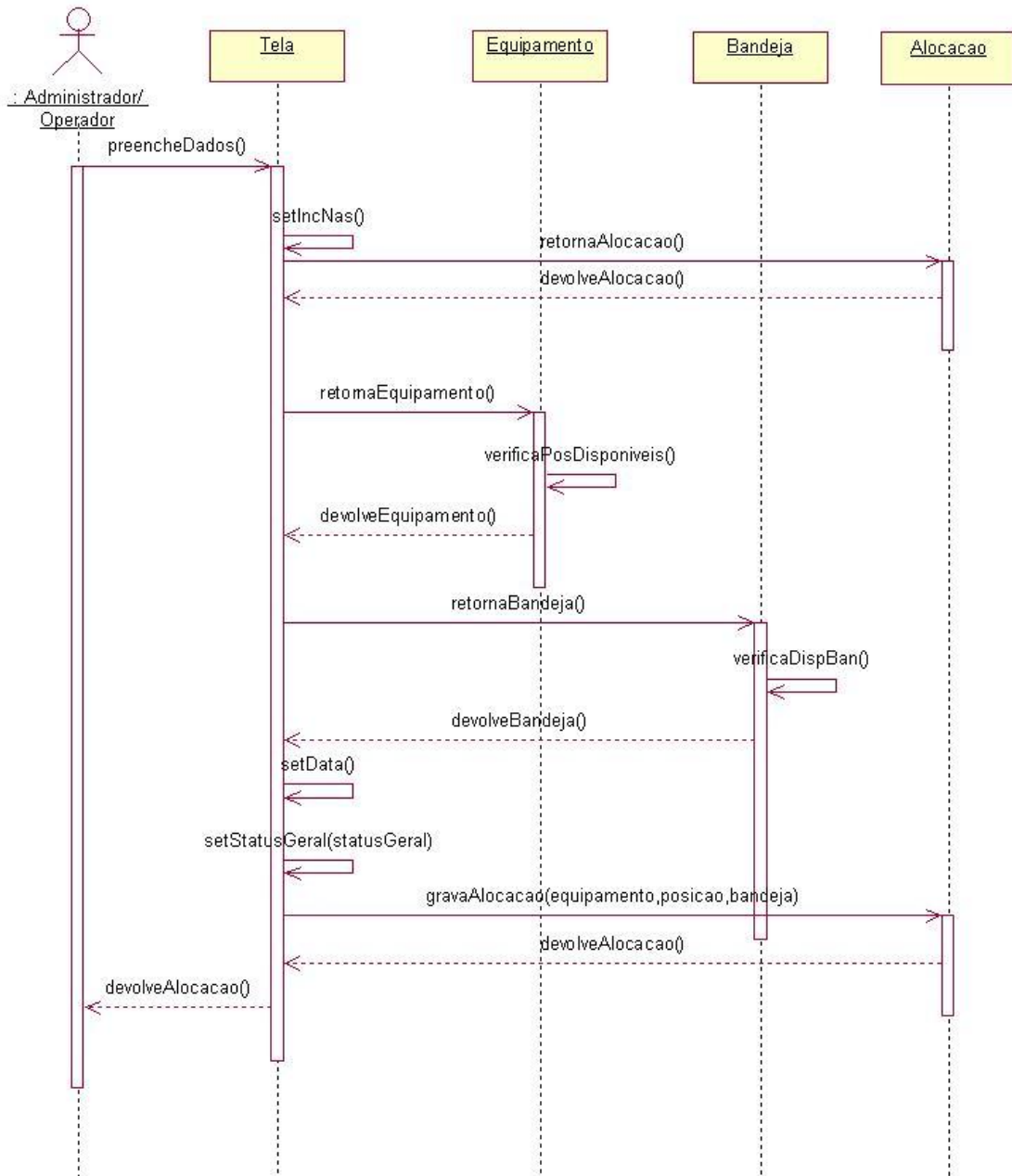
8.3.1 Diagramas de Sequência

8.3.1.1 Alocar Bandeja – Alterar

QUADRO 1 – ALOCAR BANDEJA ALTERAR

Nome do Caso de Uso	Alocar Bandeja Alterar	
Descrição	Alteração da alocação de bandejas no devido equipamento (incubadora ou nascedouro).	
Ator(es) Envolvido(s)	Administrador, Operador	
Interação entre Ator e Sistema	Ator(es)	Sistema
	O caso de uso é iniciado quando ator seleciona a opção Alocar Bandeja - Alterar/Excluir	
		Sistema apresenta tela com campos: Tipo de Equipamento e Nome do Equipamento.
	Ator preenche um dos campos, nenhum dos campos ou ambos os campos e clica no botão Consultar.	
		Sistema traz lista de equipamentos que correspondem à seleção definida pelo ator.
	Ator seleciona o Equipamento e Bandeja alocada desejada.	
		Sistema apresenta tela com campos Bandeja e Nome do Equipamento e Posição (coluna).
	Ator realiza a seleção opção desejada: Desalocar ou Alocar.	
		Sistema realiza verificação das informações inseridas pelo usuário.
		Sistema realiza cadastramento da alocação de bandeja. Neste momento este caso de uso é encerrado.
Exceções	EX1 - Os campos devem estar todos preenchidos e de acordo com o domínio do atributo.	

FIGURA 38 – DIAGRAMA DE SEQÜÊNCIA ALOCAR BANDEJA ALTERAR

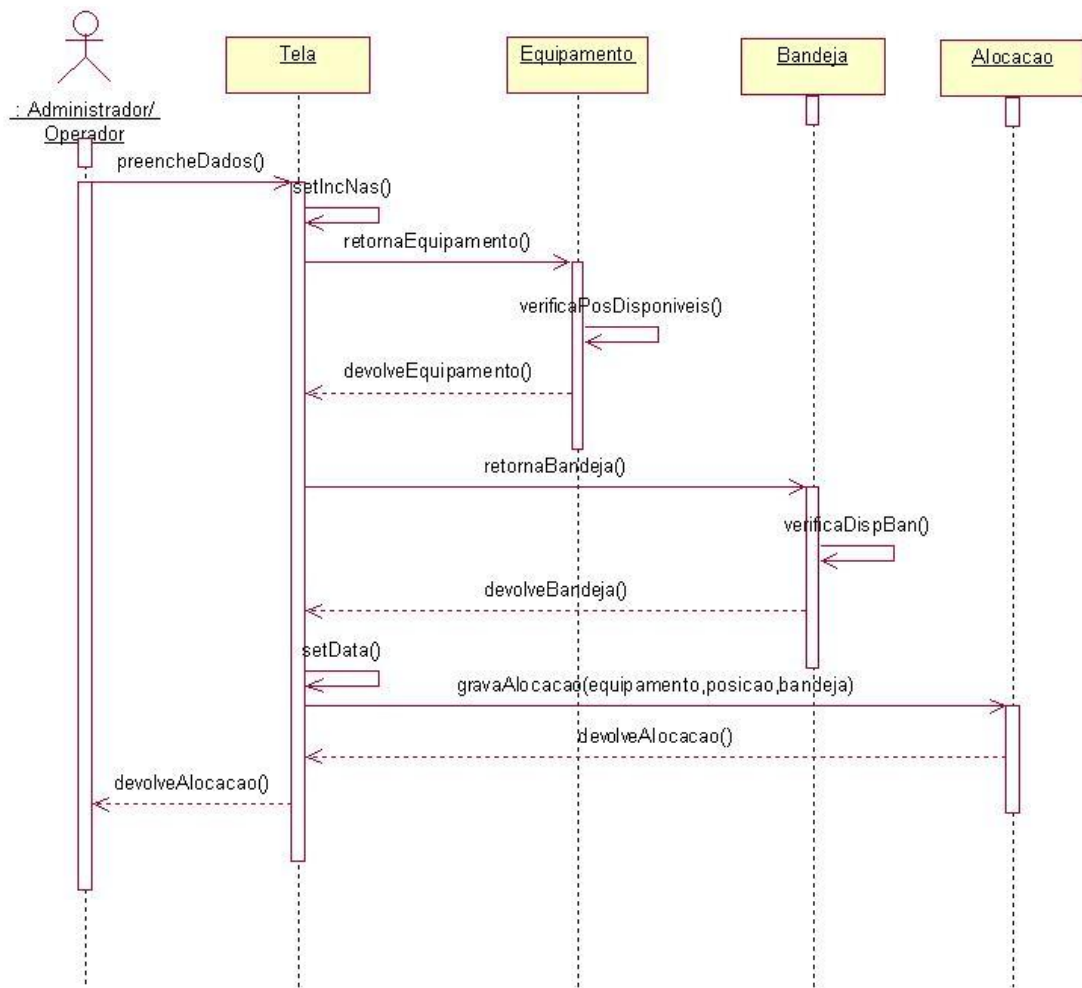


8.3.1.2 Alocar Bandeja – Novo

QUADRO 2 – ALOCAR BANDEJA NOVO

Nome do Caso de Uso	Alocar Bandeja Novo	
Descrição	Nova alocação de bandejas no devido equipamento (incubadora ou nascedouro).	
Ator(es) Envolvido(s)	Administrador, Operador	
Interação entre Ator e Sistema	Ator(es)	Sistema
	O caso de uso é iniciado quando ator seleciona a opção Alocar Bandeja - Nova Alocação	
		Sistema apresenta tela com campos Bandeja, Tipo de Equipamento e Nome do Equipamento.
	Ator realiza a seleção da Bandeja.	
	Ator realiza a seleção do Tipo de Equipamento (incubadora ou nascedouro).	
		Sistema, após a seleção realizada pelo ator do Tipo de Equipamento, traz no campo Equipamento somente os equipamentos disponíveis do tipo escolhido: Incubadora ou Nascedouro.
	Ator preenche campo Equipamento e Posição do Equipamento (coluna)	
		Sistema realiza verificação das informações inseridas pelo usuário.
		Sistema realiza cadastramento da alocação de bandeja. Neste momento este caso de uso é encerrado.
Exceções	EX1 - Os campos devem estar todos preenchidos e de acordo com o domínio do atributo.	

FIGURA 39 – DIAGRAMA DE SEQÜÊNCIA ALOCAR BANDEJA NOVO

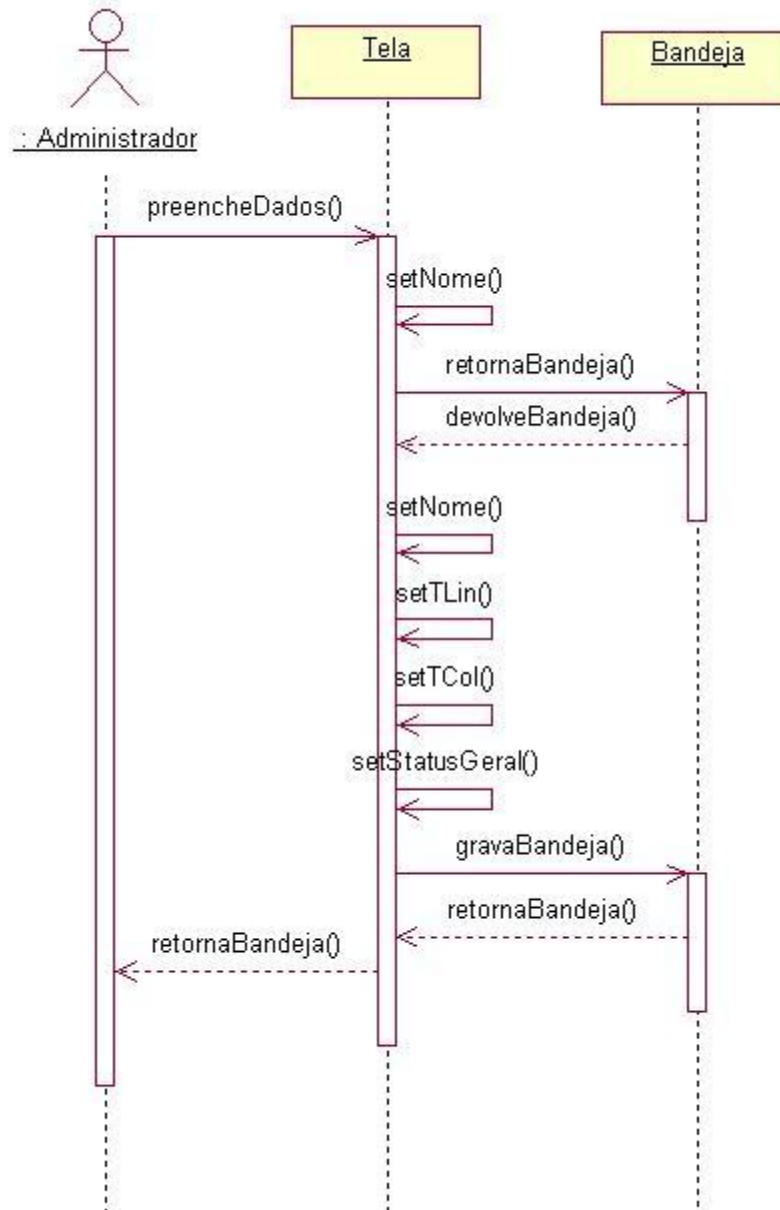


8.3.1.3 Cadastrar Bandeja – Alterar

QUADRO 3 – CADASTRAR BANDEJA ALTERAR

Nome do Caso de Uso	Cadastrar Bandeja Alterar	
Descrição	Alteração de cadastro de bandejas, definindo número de linhas e colunas associadas.	
Ator(es) Envolvido(s)	Administrador, Operador	
Interação entre Ator e Sistema	Ator(es)	Sistema
	O caso de uso é iniciado quando ator seleciona a opção Cadastrar Bandeja - Alterar/Excluir.	
		Sistema apresenta tela com campo: Nome da Bandeja.
	Ator preenche o campo com o nome da Bandeja que deseja alterar e clica no botão Consultar.	
		Sistema apresenta tela com campos: Nome da Bandeja, Número de Linhas e Número de Colunas.
	Ator preenche campos e clica no botão Alterar ou Excluir.	
		Sistema realiza verificação das informações inseridas pelo usuário.
		Sistema realiza alteração do cadastro da bandeja. Neste momento este caso de uso é encerrado.
Exceções	EX1 - Os campos devem estar todos preenchidos e de acordo com o domínio do atributo.	

FIGURA 40 – DIAGRAMA DE SEQÜÊNCIA CADASTRAR BANDEJA ALTERAR

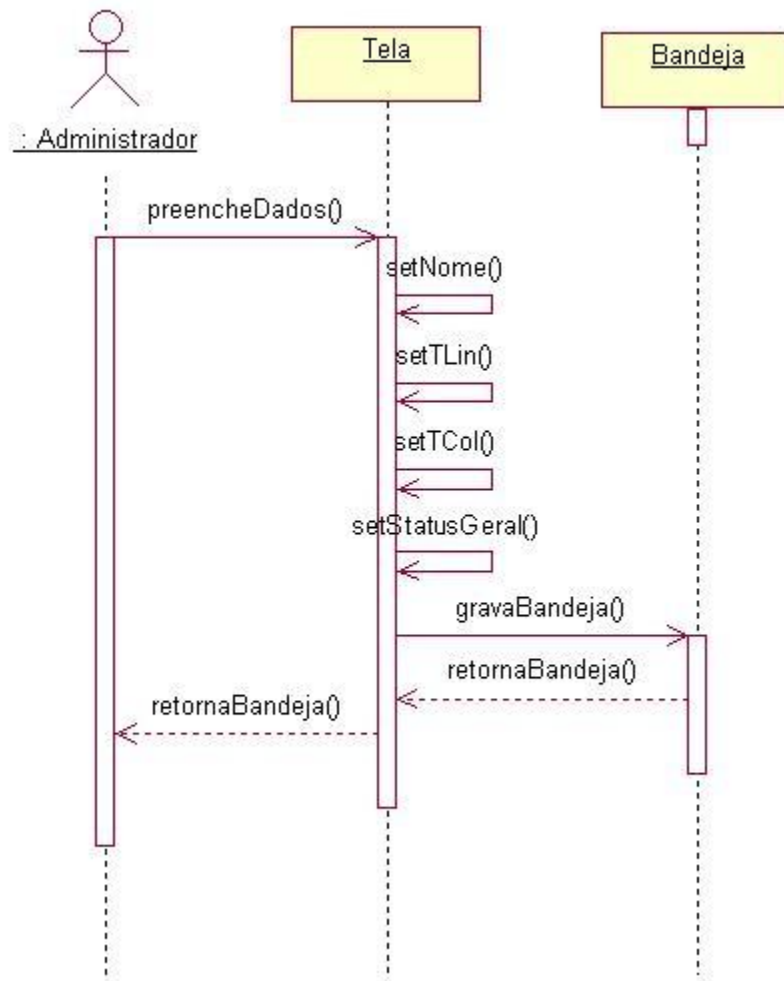


8.3.1.4 Cadastrar Bandeja – Novo

QUADRO 4 – CADASTRAR BANDEJA NOVO

Nome do Caso de Uso	Cadastrar Bandeja Novo	
Descrição	Cadastramento de novas bandejas, definindo número de linhas e colunas associadas.	
Ator(es) Envolvido(s)	Administrador, Operador	
Interação entre Ator e Sistema	Ator(es)	Sistema
	O caso de uso é iniciado quando ator seleciona a opção Cadastrar Bandeja - Cadastrar Nova.	
		Sistema apresenta tela com campos: Nome da Bandeja, Número de Linhas e Número de Colunas.
	Ator preenche campos e clica no botão Cadastrar Bandeja.	
		Sistema realiza verificação das informações inseridas pelo usuário.
		Sistema realiza cadastramento da bandeja. Neste momento este caso de uso é encerrado.
Exceções	EX1 - Os campos devem estar todos preenchidos e de acordo com o domínio do atributo.	

FIGURA 41 – DIAGRAMA DE SEQÜÊNCIA CADASTRAR BANDEJA NOVO

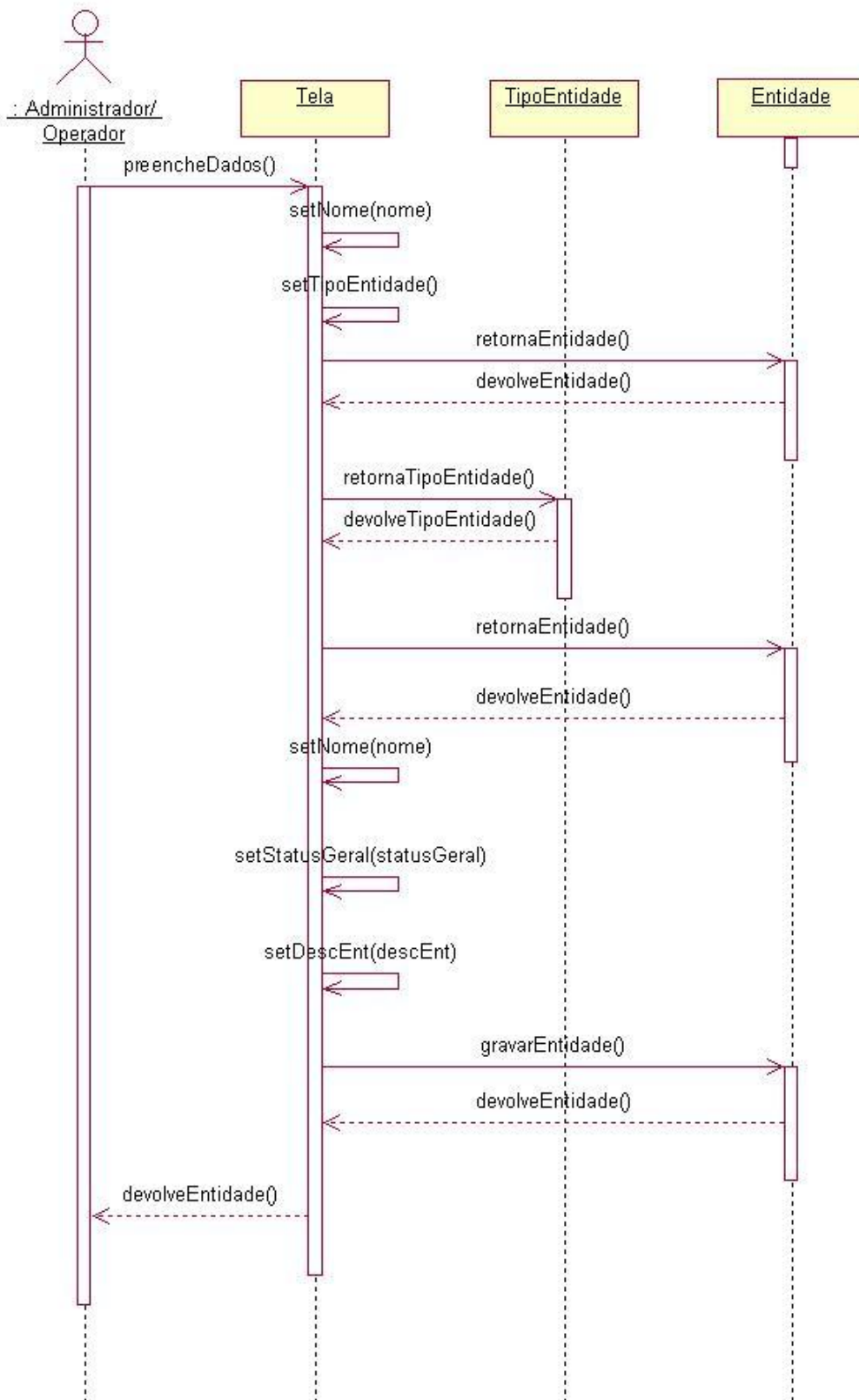


8.3.1.5 Cadastrar Entidade – Alterar

QUADRO 5 – CADASTRAR ENTIDADE ALTERAR

Nome do Caso de Uso	Cadastrar Entidade Alterar	
Descrição	<p>Alteração de entidades, que correspondem às propriedades, aos prédios, aos andares, às salas, aos blocos, às linhas, às gaiolas e às células do sistema.</p> <p>Para os ovos, que também são considerados entidades, o cadastramento e alteração são realizados em caso de uso separado devido ao número de características específicas apresentadas.</p>	
Ator(es) Envolvido(s)	Administrador, Operador	
Interação entre Ator e Sistema	Ator(es)	Sistema
	O caso de uso é iniciado quando ator seleciona a opção Cadastro - Entidade - Alterar/Excluir.	
		Sistema apresenta tela com campos: Nome e Tipo de Entidade.
	Ator preenche um dos campos, nenhum dos campos ou ambos os campos e clica no botão Consultar.	
		Sistema traz lista de entidades que correspondem aos nomes ou tipos de entidades definidos pelo ator.
	Ator seleciona a entidade que deseja alterar.	
		Sistema apresenta tela com campos: Tipo de Entidade, "que está em" e Nome da Entidade.
	Ator preenche campos que pretende alterar e clica no botão Alterar ou Excluir.	
		Sistema realiza verificação das informações inseridas pelo usuário.
		Sistema realiza cadastramento da entidade. Neste momento este caso de uso é encerrado.
Exceções	EX1 - Os campos devem estar todos preenchidos e de acordo com o domínio do atributo.	

FIGURA 42 – DIAGRAMA DE SEQÜÊNCIA CADASTRAR ENTIDADE ALTERAR

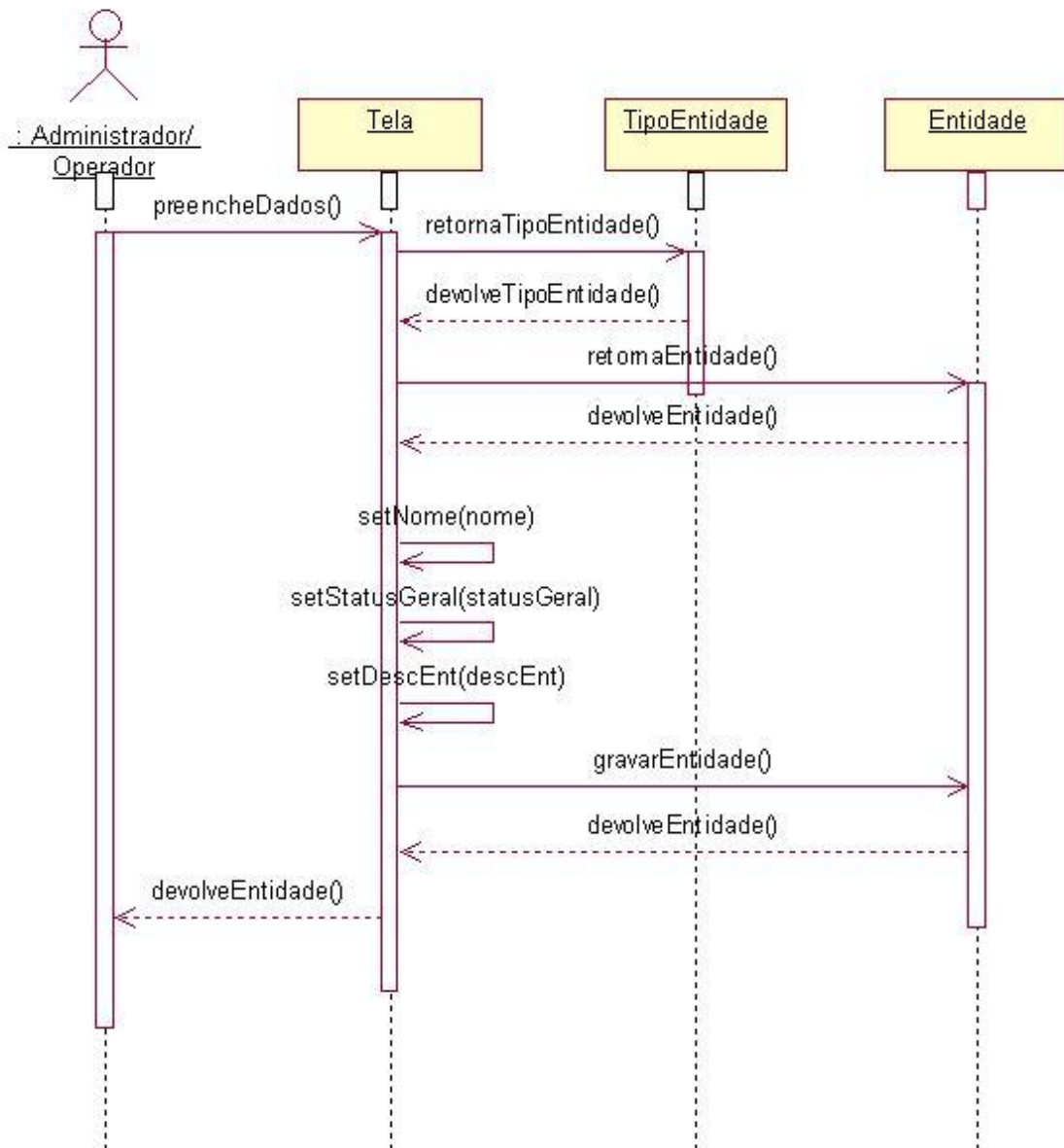


8.3.1.6 Cadastrar Entidade – Novo

QUADRO 6 – CADASTRAR ENTIDADE NOVO

Nome do Caso de Uso	Cadastrar Entidade Novo	
Descrição	<p>Cadastramento de novas entidades, que correspondem às propriedades, aos prédios, aos andares, às salas, aos blocos, às linhas, às gaiolas e às células do sistema.</p> <p>Para os ovos, que também são considerados entidades, o cadastramento é realizado em caso de uso separado devido ao número de características específicas apresentadas.</p>	
Ator(es) Envolvido(s)	Administrador, Operador	
Interação entre Ator e Sistema	Ator(es)	Sistema
	O caso de uso é iniciado quando ator seleciona a opção Cadastro - Entidade - Cadastrar Nova.	
		Sistema apresenta tela com campos: Tipo de Entidade, "que está em" e Nome da Entidade.
	Ator preenche campos e clica no botão Gravar.	
		Sistema realiza verificação das informações inseridas pelo usuário.
		Sistema realiza cadastramento da entidade. Neste momento este caso de uso é encerrado.
Exceções	EX1 - Os campos devem estar todos preenchidos e de acordo com o domínio do atributo.	

FIGURA 43 – DIAGRAMA DE SEQÜÊNCIA CADASTRAR ENTIDADE NOVO

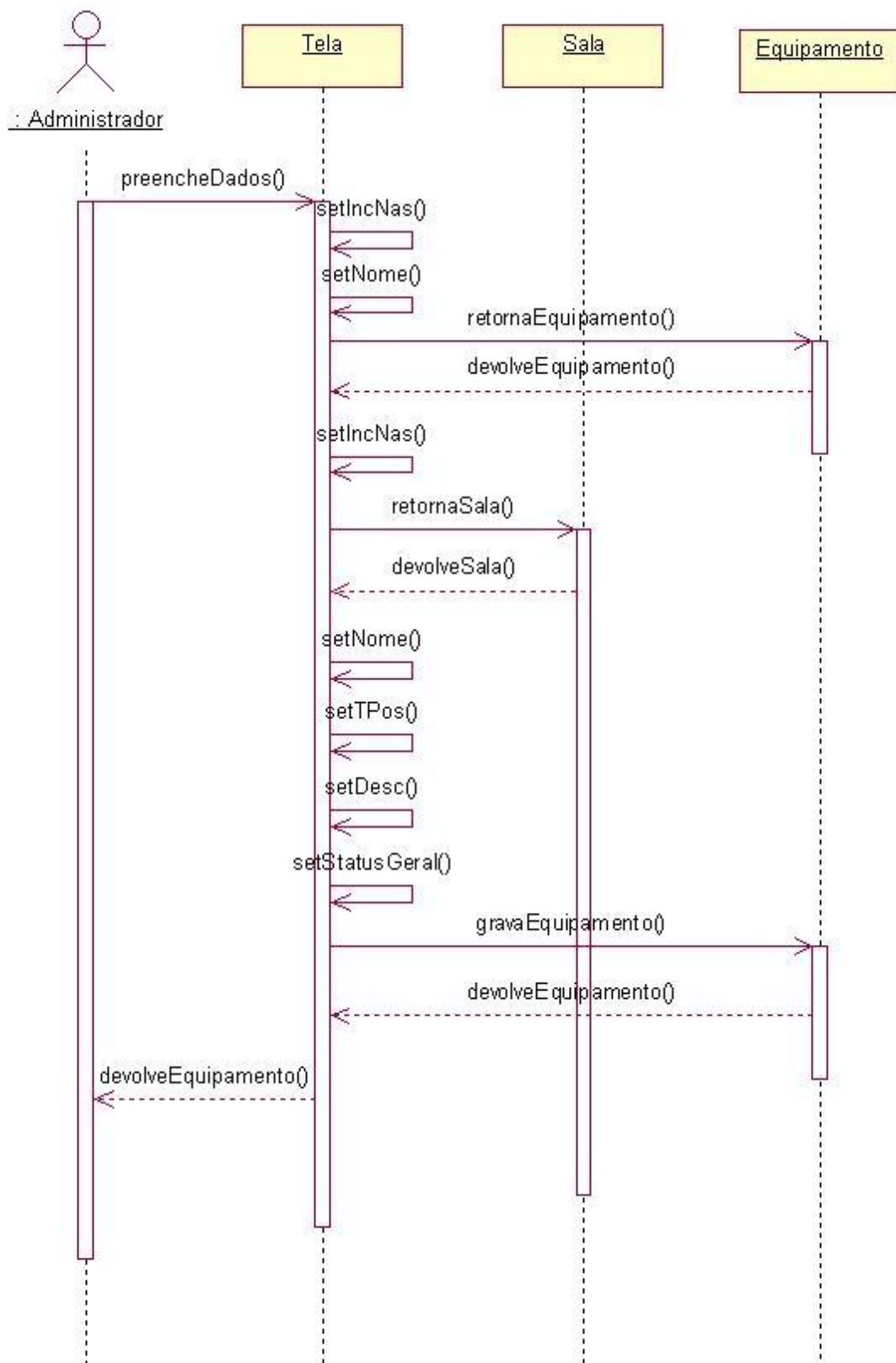


8.3.1.7 Cadastrar Equipamento – Alterar

QUADRO 7 – CADASTRAR EQUIPAMENTO ALTERAR

Nome do Caso de Uso	Cadastrar Equipamento Alterar	
Descrição	Alteração de incubadoras e novos nascedouros, definindo-se o tipo de equipamento e a sala onde o mesmo se encontra.	
Ator(es) Envolvido(s)	Administrador	
Interação entre Ator e Sistema	Ator(es)	Sistema
	O caso de uso é iniciado quando ator seleciona a opção Cadastro - Equipamento - Alterar/Excluir.	
		Sistema apresenta tela com campos: Tipo do Equipamento e Nome do Equipamento.
	Ator preenche um dos campos, nenhum dos campos ou ambos os campos e clica no botão Consultar.	
		Sistema traz lista de equipamentos que correspondem à seleção definida pelo ator.
	Ator seleciona equipamento que deseja alterar.	
		Sistema apresenta tela com campos: Tipo do Equipamento, Nome do Equipamento, Sala, Número de Posições de Bandeja e Descrição.
	Ator preenche campos que pretende alterar e clica no botão Alterar ou Excluir.	
		Sistema realiza verificação das informações inseridas pelo usuário.
	Sistema atualiza cadastramento do equipamento. Neste momento este caso de uso é encerrado.	
Exceções	EX1 - Os campos devem estar todos preenchidos e de acordo com o domínio do atributo.	

FIGURA 44 – DIAGRAMA DE SEQÜÊNCIA CADASTRAR EQUIPAMENTO ALTERAR

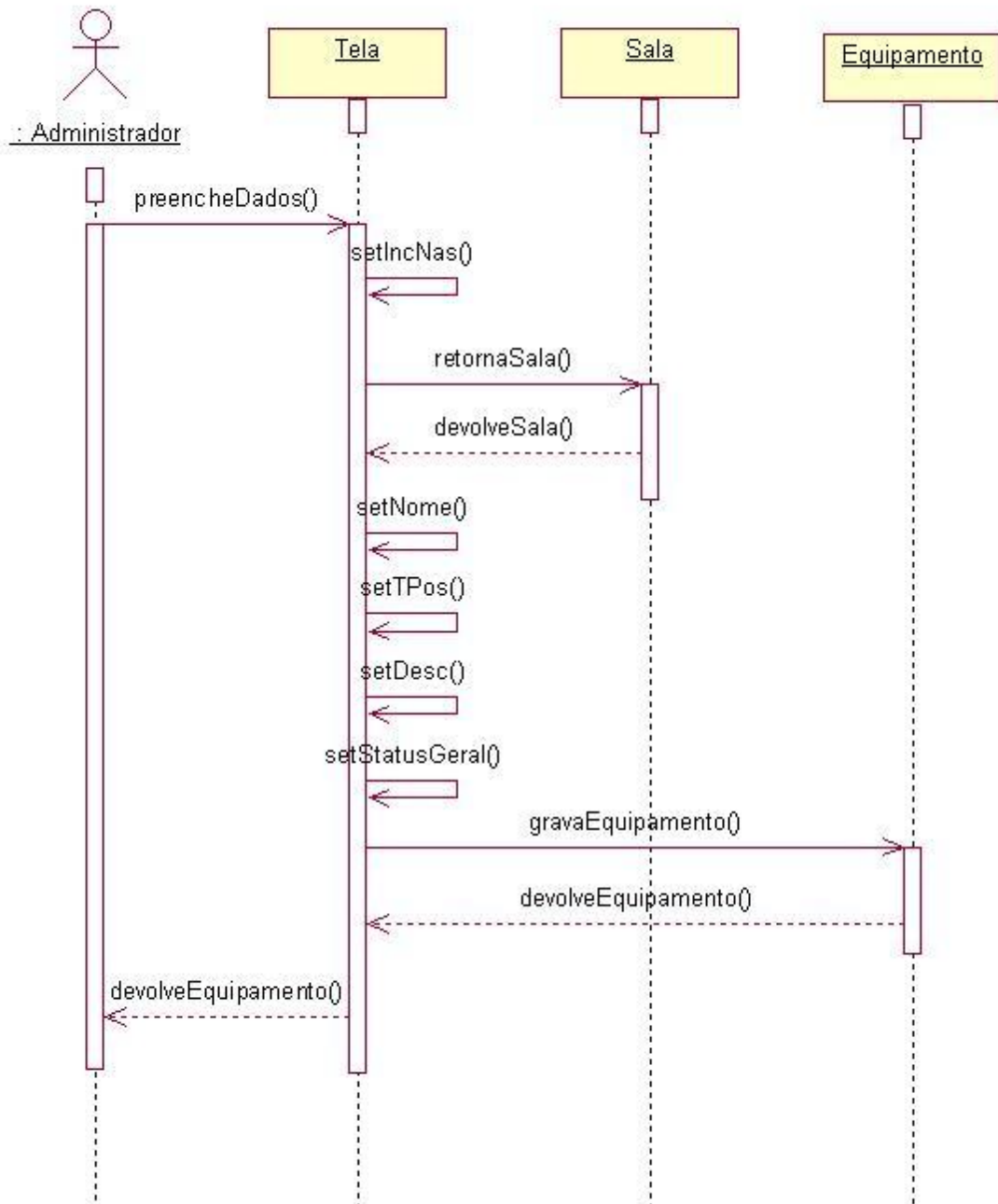


8.3.1.8 Cadastrar Equipamento – Novo

QUADRO 8 – CADASTRAR EQUIPAMENTO NOVO

Nome do Caso de Uso	Cadastrar Equipamento Novo	
Descrição	Cadastramento de novas incubadoras e novos nascedouros, definindo-se o tipo de equipamento e a sala onde o mesmo se encontra.	
Ator(es) Envolvido(s)	Administrador	
Interação entre Ator e Sistema	Ator(es)	Sistema
	O caso de uso é iniciado quando ator seleciona a opção Cadastro - Equipamento - Cadastrar Novo.	
		Sistema apresenta tela com campos: Tipo do Equipamento, Nome do Equipamento, Sala, Número de Posições de Bandeja e Descrição.
	Ator preenche campos e clica no botão Cadastrar Equipamento.	
		Sistema realiza verificação das informações inseridas pelo usuário.
		Sistema realiza cadastramento do equipamento. Neste momento este caso de uso é encerrado.
Exceções	EX1 - Os campos devem estar todos preenchidos e de acordo com o domínio do atributo.	

FIGURA 45 – DIAGRAMA DE SEQÜÊNCIA CADASTRAR EQUIPAMENTO NOVO

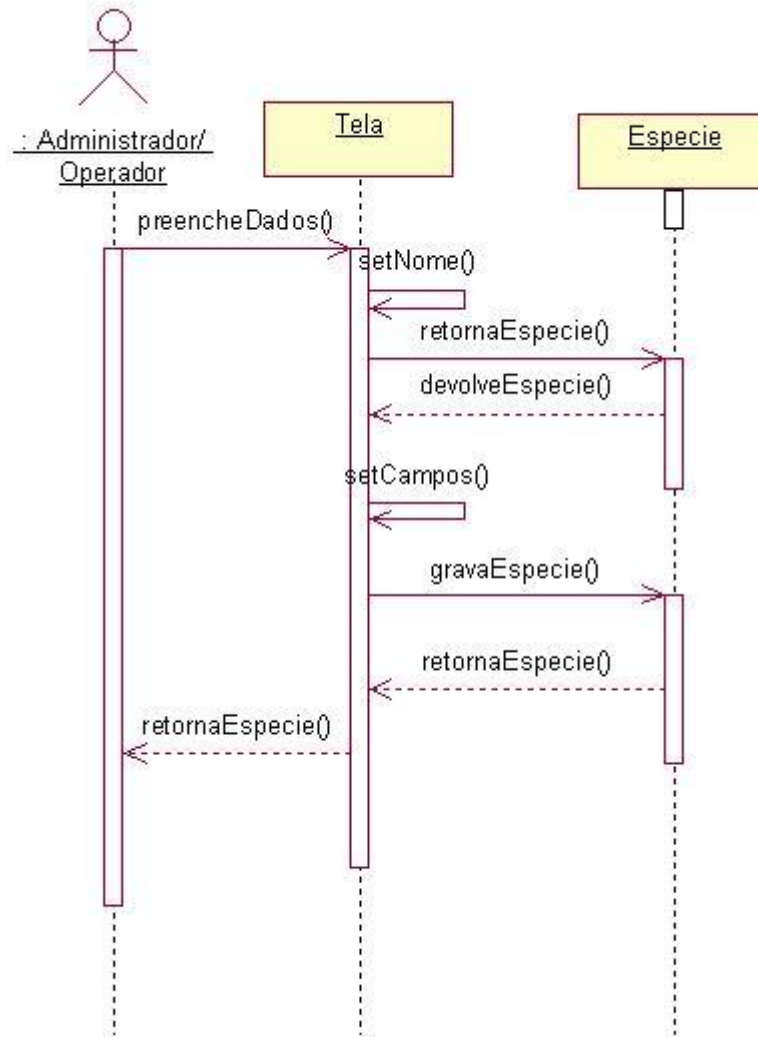


8.3.1.9 Cadastrar Espécie de Ave – Alterar

QUADRO 9 – CADASTRAR ESPÉCIE DE AVE ALTERAR

Nome do Caso de Uso	Cadastrar Espécie de Ave Alterar	
Descrição	Alteração do cadastro de Espécies de Ave, a serem utilizadas nos projetos do sistema.	
Ator(es) Envolvido(s)	Administrador	
Interação entre Ator e Sistema	Ator(es)	Sistema
	O caso de uso é iniciado quando ator seleciona a opção Cadastro - Espécie - Alterar/Excluir.	
		Sistema apresenta tela com campo: Nome da Espécie.
	Ator preenche campo Nome da Espécie e clica no botão Consultar.	
		Sistema traz lista de espécies que correspondem à seleção definida pelo ator.
	Ator seleciona Espécie desejada.	
		Sistema apresenta tela com campos: Nome da Espécie, Temperatura Mínima de Incubação, Temperatura Máxima de Incubação, Umidade Mínima de Incubação, Umidade Máxima de Incubação, Período de Incubação, Período de Aviso de Incubação, Temperatura Mínima de Nascimento, Temperatura Máxima de Nascimento, Umidade Mínima da Nascimento, Umidade Máxima de Nascimento, Período de Nascimento, Período de Aviso de Nascimento e Período de Recria.
	Ator preenche campos e clica no botão Alterar ou Excluir.	
		Sistema realiza verificação das informações inseridas pelo usuário.
	Sistema realiza alteração do cadastro da espécie. Neste momento este caso de uso é encerrado.	
Exceções	EX1 - Os campos devem estar todos preenchidos e de acordo com o domínio do atributo.	

FIGURA 46 – CADASTRAR ESPÉCIE DE AVE ALTERAR

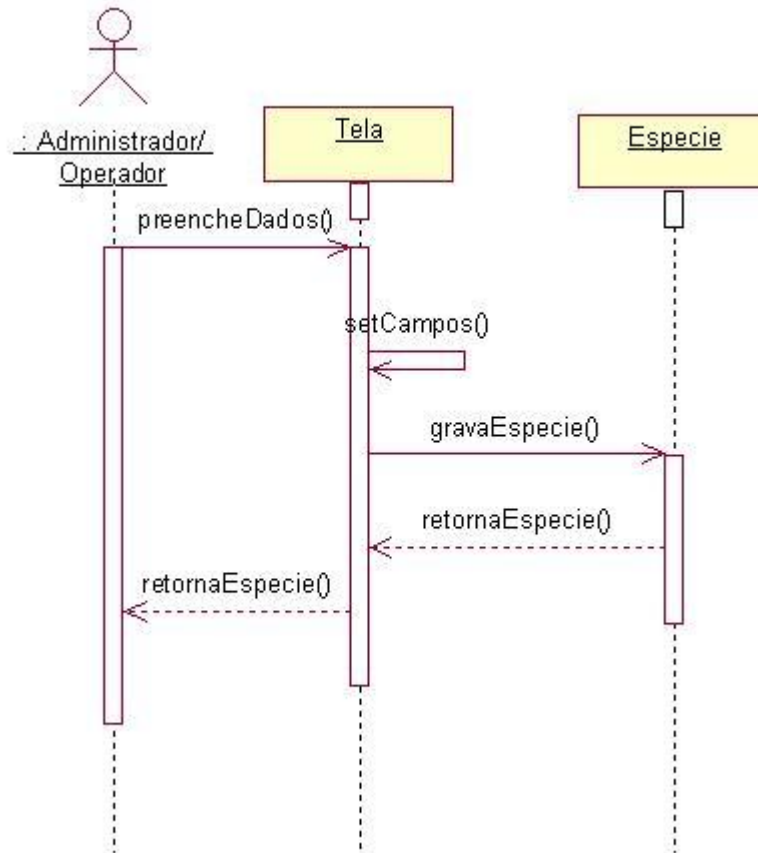


8.3.1.10 Cadastrar Espécie de Ave – Novo

QUADRO 10 – CADASTRAR ESPÉCIE DE AVE NOVO

Nome do Caso de Uso	Cadastrar Espécie de Ave Novo	
Descrição	Cadastramento de novas Espécies de Ave, a serem utilizadas nos projetos do sistema.	
Ator(es) Envolvido(s)	Administrador	
Interação entre Ator e Sistema	Ator(es)	Sistema
	O caso de uso é iniciado quando ator seleciona a opção Cadastro - Espécie - Cadastrar Nova.	
		Sistema apresenta tela com campos: Nome da Espécie, Temperatura Mínima de Incubação, Temperatura Máxima de Incubação, Umidade Mínima de Incubação, Umidade Máxima de Incubação, Período de Incubação, Período de Aviso de Incubação, Temperatura Mínima de Nascimento, Temperatura Máxima de Nascimento, Umidade Mínima da Nascimento, Umidade Máxima de Nascimento, Período de Nascimento, Período de Aviso de Nascimento e Período de Recria.
	Ator preenche campos e clica no botão Cadastrar Espécie.	
		Sistema realiza verificação das informações inseridas pelo usuário.
	Sistema realiza cadastramento da espécie. Neste momento este caso de uso é encerrado.	
Exceções	EX1 - Os campos devem estar todos preenchidos e de acordo com o domínio do atributo.	

FIGURA 47 – CADASTRAR ESPÉCIE DE AVE NOVO

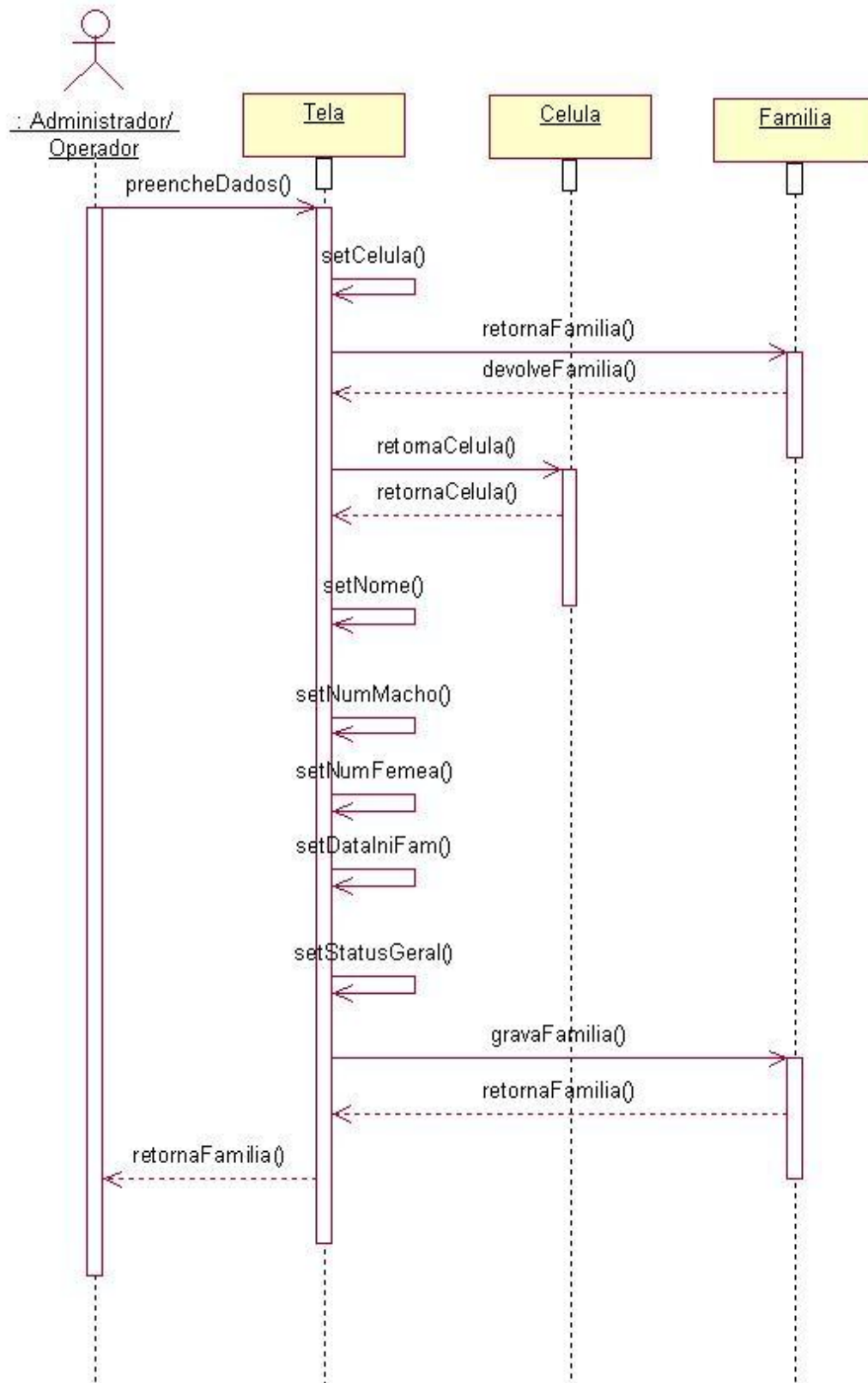


8.3.1.11 Cadastrar Família – Alterar

QUADRO 11 – CADASTRAR FAMÍLIA ALTERAR

Nome do Caso de Uso	Cadastrar Família Alterar	
Descrição	Alterar cadastro de famílias, que são agrupamentos de aves (normalmente um macho e duas fêmeas) alojadas em uma determinada célula por um período de tempo.	
Ator(es) Envolvido(s)	Administrador, Operador	
Interação entre Ator e Sistema	Ator(es)	Sistema
	O caso de uso é iniciado quando ator seleciona a opção Família - Alterar/Excluir.	
		Sistema apresenta tela com campo: Célula.
	Ator preenche campo Célula e clica no botão Consultar.	
		Sistema apresenta tela com campos: Célula, Nome da Família, Número de Machos, Número de Fêmeas e Data de Início da Família.
	Ator preenche campos e clica no botão Alterar ou Excluir.	
		Sistema realiza verificação das informações inseridas pelo usuário.
		Sistema atualiza cadastro da família. Neste momento este caso de uso é encerrado.
Exceções	EX1 - Os campos devem estar todos preenchidos e de acordo com o domínio do atributo.	

FIGURA 48 – CADASTRAR FAMÍLIA ALTERAR

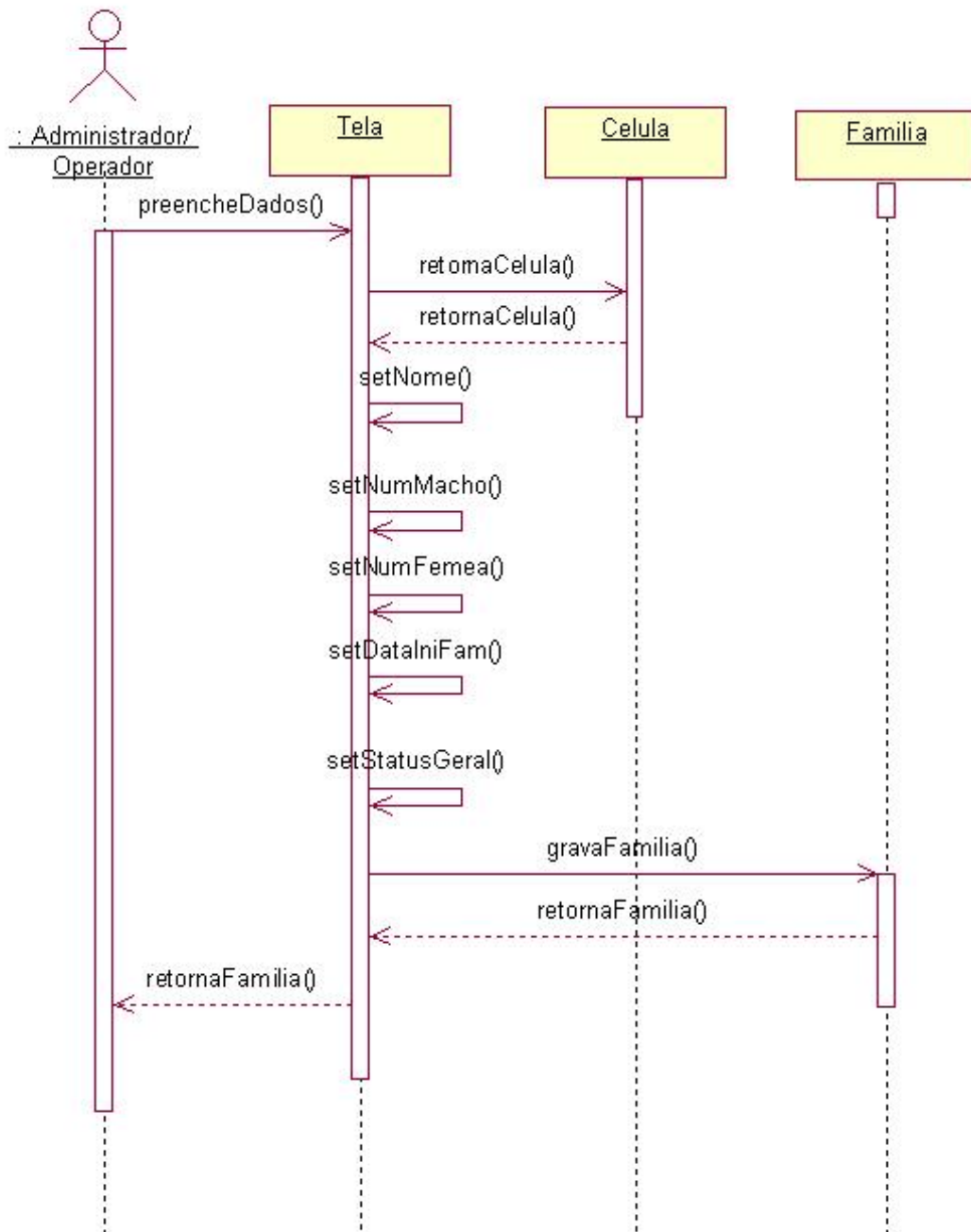


8.3.1.12 Cadastrar Família – Novo

QUADRO 12 – CADASTRAR FAMÍLIA NOVO

Nome do Caso de Uso	Cadastrar Família Novo	
Descrição	Cadastramento de novas famílias, que são agrupamentos de aves (normalmente um macho e duas fêmeas) alojadas em uma determinada célula por um período de tempo.	
Ator(es) Envolvido(s)	Administrador, Operador	
Interação entre Ator e Sistema	Ator(es)	Sistema
	O caso de uso é iniciado quando ator seleciona a opção Família - Cadastrar Nova.	
		Sistema apresenta tela com campos: Célula, Nome da Família, Número de Machos, Número de Fêmeas e Data de Início da Família.
	Ator preenche campos e clica no botão Cadastrar.	
		Sistema realiza verificação das informações inseridas pelo usuário.
		Sistema realiza cadastramento da família. Neste momento este caso de uso é encerrado.
Exceções	EX1 - Os campos devem estar todos preenchidos e de acordo com o domínio do atributo.	

FIGURA 49 – CADASTRAR FAMÍLIA NOVO

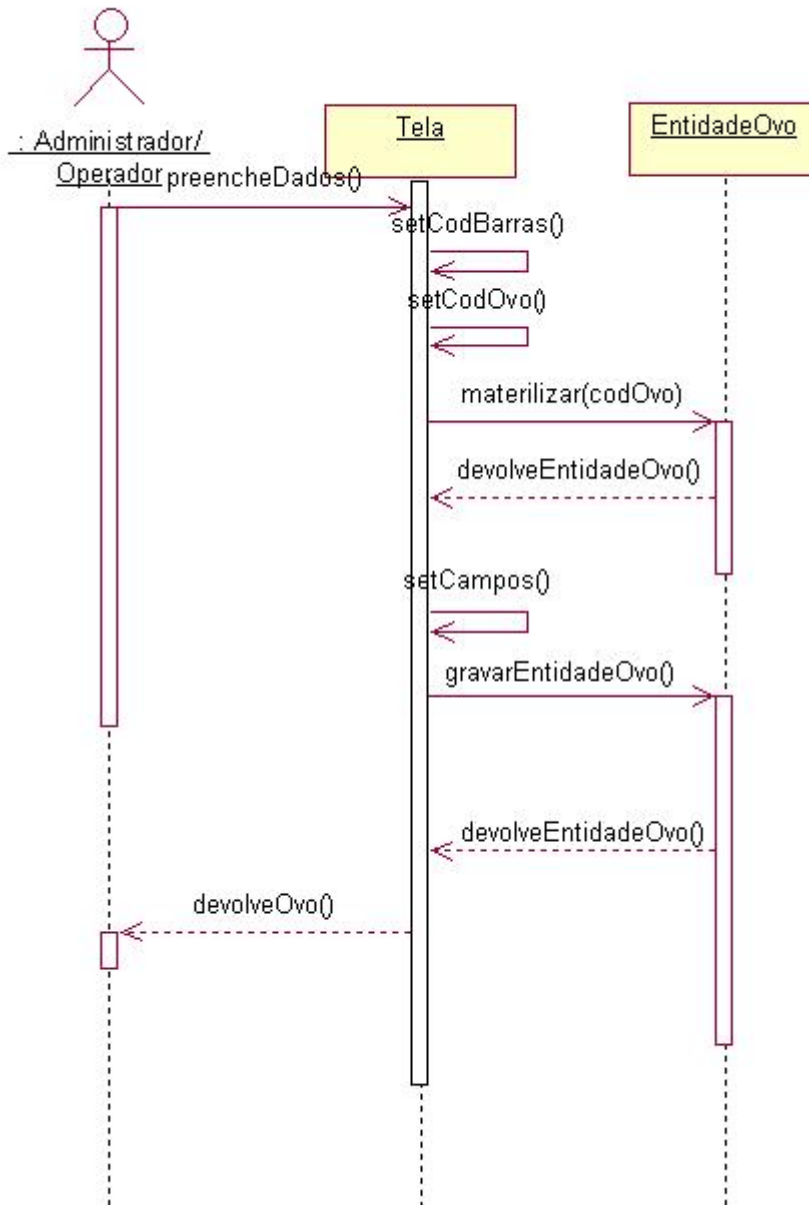


8.3.1.13 Cadastrar Ovo – Alterar

QUADRO 13 – CADASTRAR OVO ALTERAR

Nome do Caso de Uso	Cadastrar Ovo Alterar	
Descrição	Alterar cadastro individual de ovos, realizados principalmente para espécie Avestruz (ovos analisados e rastreados individualmente).	
Ator(es) Envolvido(s)	Administrador, Operador	
Interação entre Ator e Sistema	Ator(es)	Sistema
	O caso de uso é iniciado quando ator seleciona a opção Cadastrar Ovo - Altera/Excluir.	
		Sistema apresenta tela com campos: Código de Barras e Código do Ovo.
	Ator preenche um dos campos, nenhum dos campos ou ambos os campos e clica no botão Consultar.	
		Sistema traz lista de Ovos que correspondem à seleção definida pelo ator.
	Ator seleciona o Ovo desejado.	
		Sistema apresenta tela com campos: Desinfecção, Formato, Cor, Célula, Aquisição, Data de Coleta, Aspecto, Peso, Porosidade e Incubável.
	Ator preenche campos e clica no botão Alterar ou Excluir.	
		Sistema realiza verificação das informações inseridas pelo usuário.
		Sistema realiza alteração no cadastro do Ovo. Neste momento este caso de uso é encerrado.
Exceções	EX1 - Os campos devem estar todos preenchidos e de acordo com o domínio do atributo.	

FIGURA 50 – CADASTRAR OVO ALTERAR

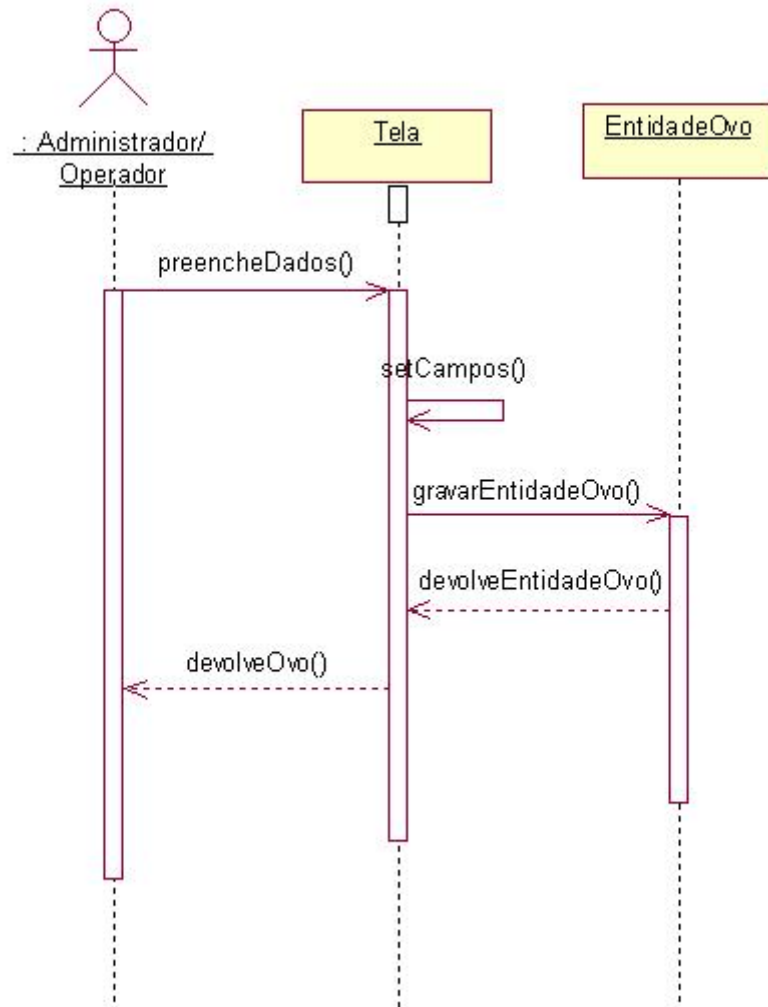


8.3.1.14 Cadastrar Ovo – Novo

QUADRO 14 – CADASTRAR OVO NOVO

Nome do Caso de Uso	Cadastrar Ovo Novo	
Descrição	Cadastramento individual de novos ovos, realizados principalmente para espécie Avestruz (ovos analisados e rastreados individualmente).	
Ator(es) Envolvido(s)	Administrador, Operador	
Interação entre Ator e Sistema	Ator(es)	Sistema
	O caso de uso é iniciado quando ator seleciona a opção Cadastrar Ovo - Novo Ovo.	
		Sistema apresenta tela com campos: Desinfecção, Formato, Cor, Célula, Aquisição, Data de Coleta, Aspecto, Peso, Porosidade e Incubável.
	Ator preenche campos e clica no botão Cadastrar.	
		Sistema realiza verificação das informações inseridas pelo usuário.
		Sistema realiza cadastramento do ovo. Neste momento este caso de uso é encerrado.
Exceções	EX1 - Os campos devem estar todos preenchidos e de acordo com o domínio do atributo.	

FIGURA 51 – CADASTRAR OVO NOVO

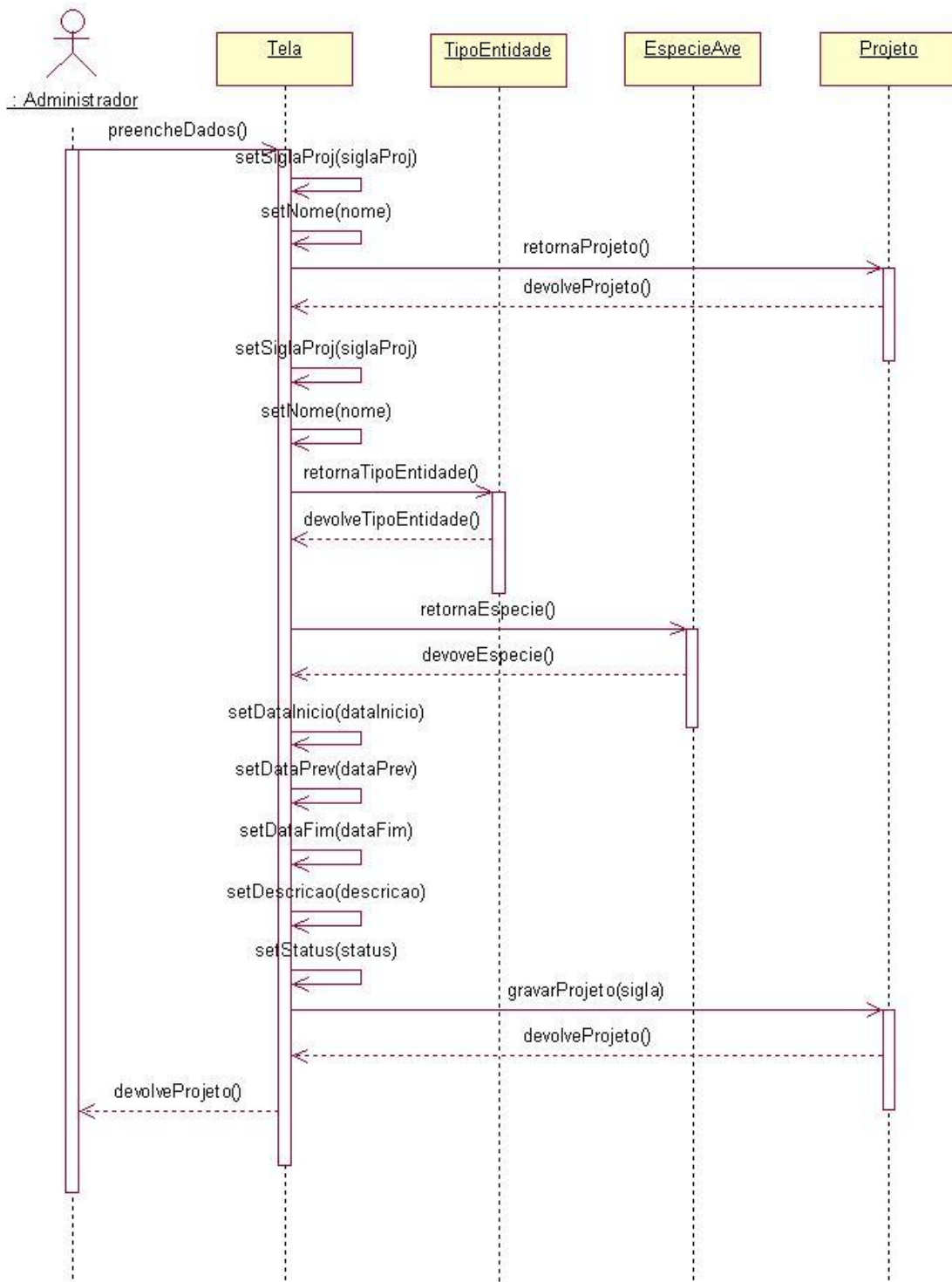


8.3.1.15 Cadastrar Projeto – Alterar

QUADRO 15 – CADASTRAR PROJETO ALTERAR

Nome do Caso de Uso	Cadastrar Projeto Alterar	
Descrição	Alterar cadastro de projetos, para os quais são realizados incubações, nascimentos, transferências e saques, e para os quais são disponibilizadas informações armazenadas no sistema.	
Ator(es) Envolvido(s)	Administrador	
Interação entre Ator e Sistema	Ator(es)	Sistema
	O caso de uso é iniciado quando ator seleciona a opção Projeto - Alterar/Excluir.	
		Sistema apresenta tela com campos: Sigla e Nome do Projeto.
	Ator preenche um dos campos, nenhum dos campos ou ambos os campos e clica no botão Consultar.	
		Sistema traz lista de projetos que correspondem à seleção definida pelo ator.
	Ator seleciona o Sigla ou Nome do Projeto desejado.	
		Sistema apresenta tela com campos: Sigla, Nome, Entidade, Espécie de Ave, Data de Início, Previsão de Término e Descrição.
	Ator preenche campos e clica no botão Alterar ou Excluir.	
		Sistema realiza verificação das informações inseridas pelo usuário.
	Sistema realiza alteração do cadastro da projeto. Neste momento este caso de uso é encerrado.	
Exceções	EX1 - Os campos devem estar todos preenchidos e de acordo com o domínio do atributo.	

FIGURA 52 – CADASTRAR PROJETO ALTERAR

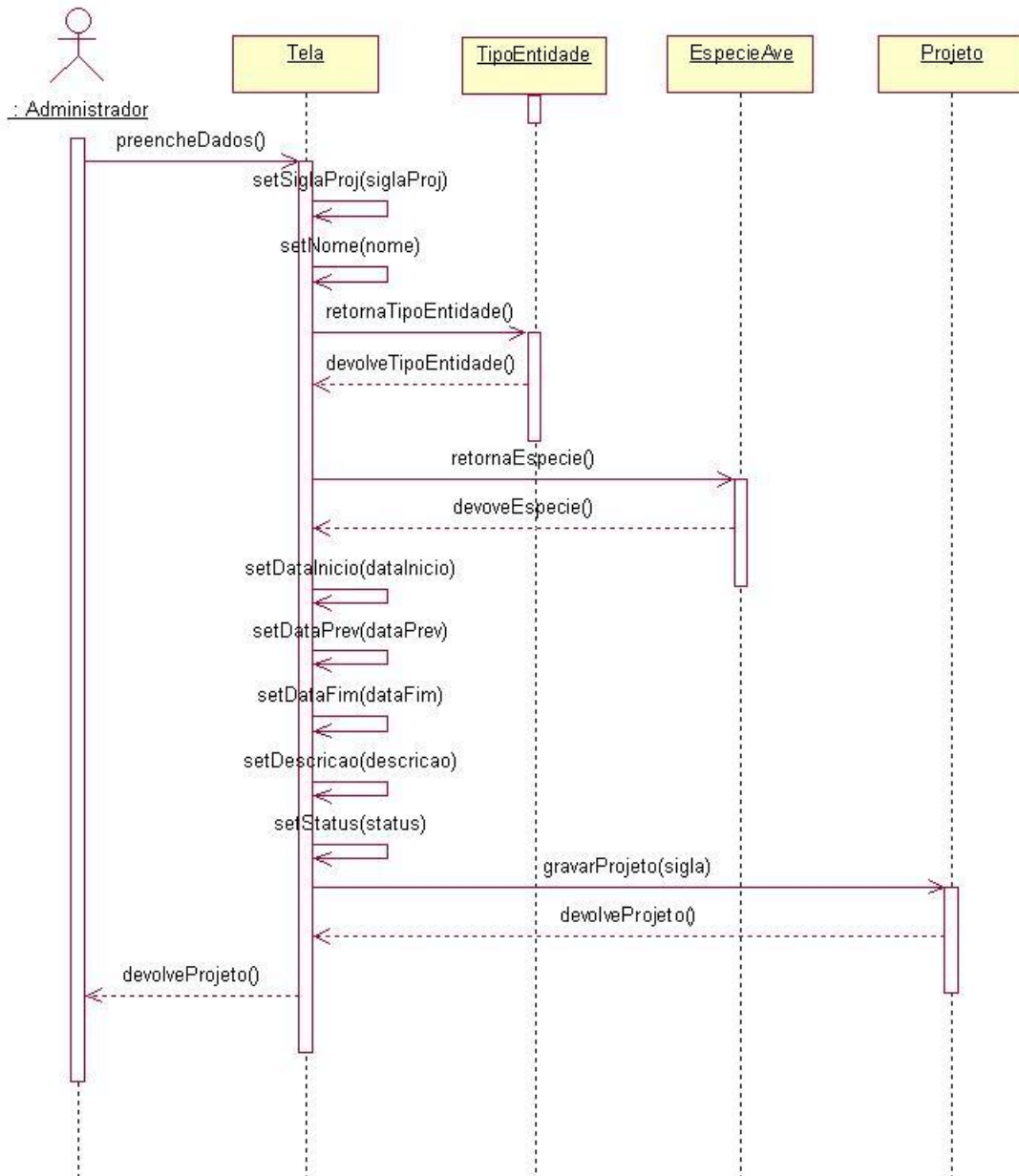


8.3.1.16 Cadastrar Projeto – Novo

QUADRO 16 – CADASTRAR PROJETO NOVO

Nome do Caso de Uso	Cadastrar Projeto Novo	
Descrição	Cadastramento de novos projetos, para os quais serão realizados, posteriormente, incubações, nascimentos, transferências e saques, e para os quais serão disponibilizadas informações armazenadas no sistema.	
Ator(es) Envolvido(s)	Administrador	
Interação entre Ator e Sistema	Ator(es)	Sistema
	O caso de uso é iniciado quando ator seleciona a opção Projeto - Novo Projeto.	
		Sistema apresenta tela com campos: Sigla, Nome, Entidade, Espécie de Ave, Data de Início, Previsão de Término e Descrição.
	Ator preenche campos e clica no botão Criar.	
		Sistema realiza verificação das informações inseridas pelo usuário.
		Sistema realiza cadastramento da projeto. Neste momento este caso de uso é encerrado.
Exceções	EX1 - Os campos devem estar todos preenchidos e de acordo com o domínio do atributo.	

FIGURA 53 – CADASTRAR PROJETO NOVO

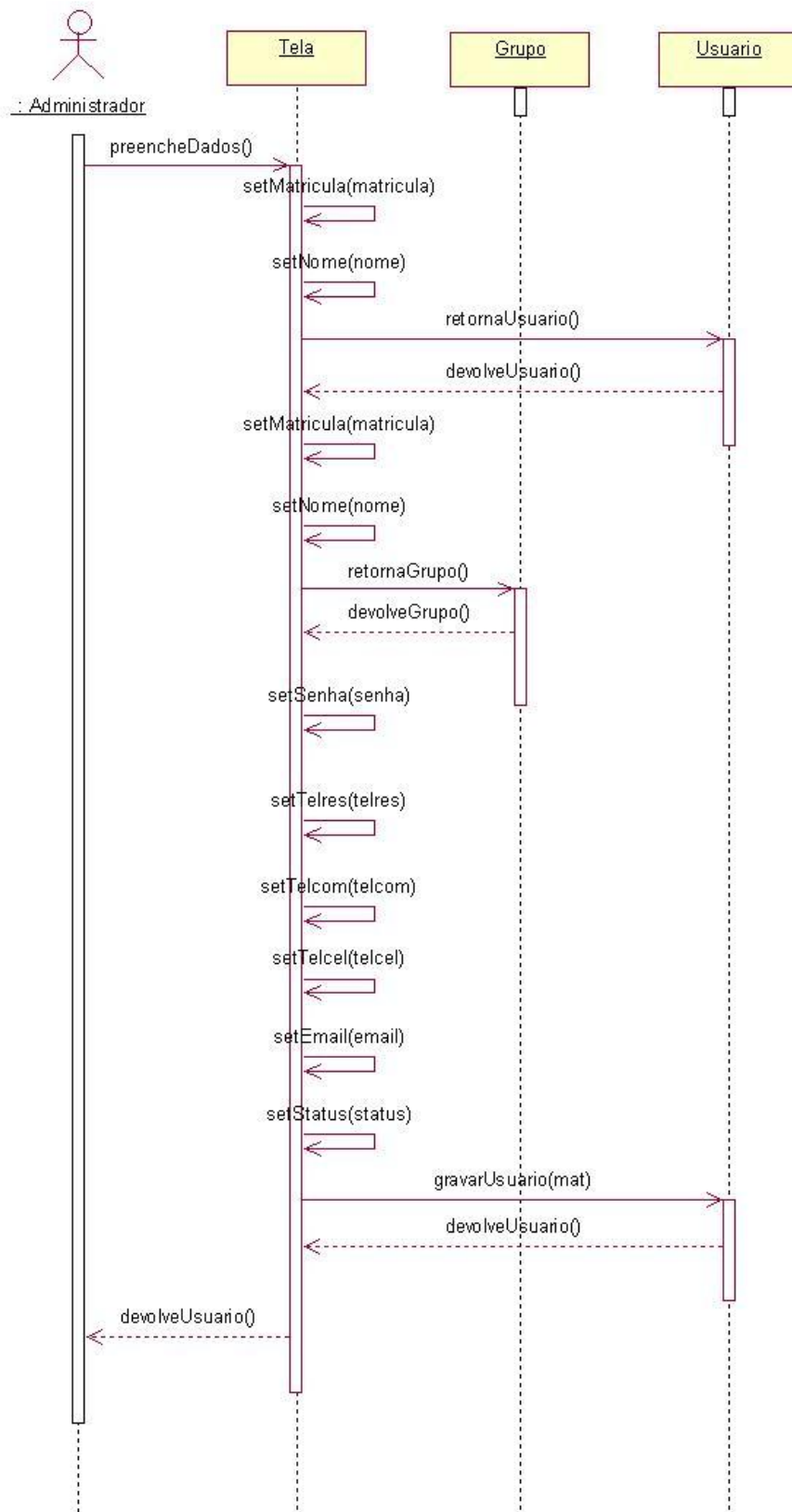


8.3.1.17 Cadastrar Usuário – Alterar

QUADRO 17 – CADASTRAR USUÁRIO ALTERAR

Nome do Caso de Uso	Cadastrar Usuario Alterar	
Descrição	Alteração de usuários com acesso ao sistema, além da definição do grupo ao qual cada usuário faz parte, associando-se, desta forma, as suas permissões no sistema.	
Ator(es) Envolvido(s)	Administrador	
Interação entre Ator e Sistema	Ator(es)	Sistema
	O caso de uso é iniciado quando ator seleciona a opção Gerenciamento - Usuário - Alterar/Excluir.	
		Sistema apresenta tela com campos: Matrícula e Nome.
	Ator preenche um dos campos, nenhum dos campos ou ambos os campos e clica no botão Consultar.	
		Sistema traz lista de usuários que correspondem às matrículas ou nomes definidos pelo ator.
	Ator seleciona o Nome ou Matrícula do usuário desejado.	
		Sistema apresenta tela com campos preenchidos: Matrícula, Nome, Grupo, Senha, Confirmar Senha, Telefone Residencial, Telefone Comercial, E-mail e Descrição.
	Ator preenche campos que pretende alterar e clica no botão Alterar ou Excluir.	
		Sistema realiza verificação das informações inseridas pelo usuário.
	Sistema atualiza cadastramento do usuário. Neste momento este caso de uso é encerrado.	
Exceções	EX1 - Os campos devem estar todos preenchidos e de acordo com o domínio do atributo.	

FIGURA 54 – CADASTRAR USUÁRIO ALTERAR

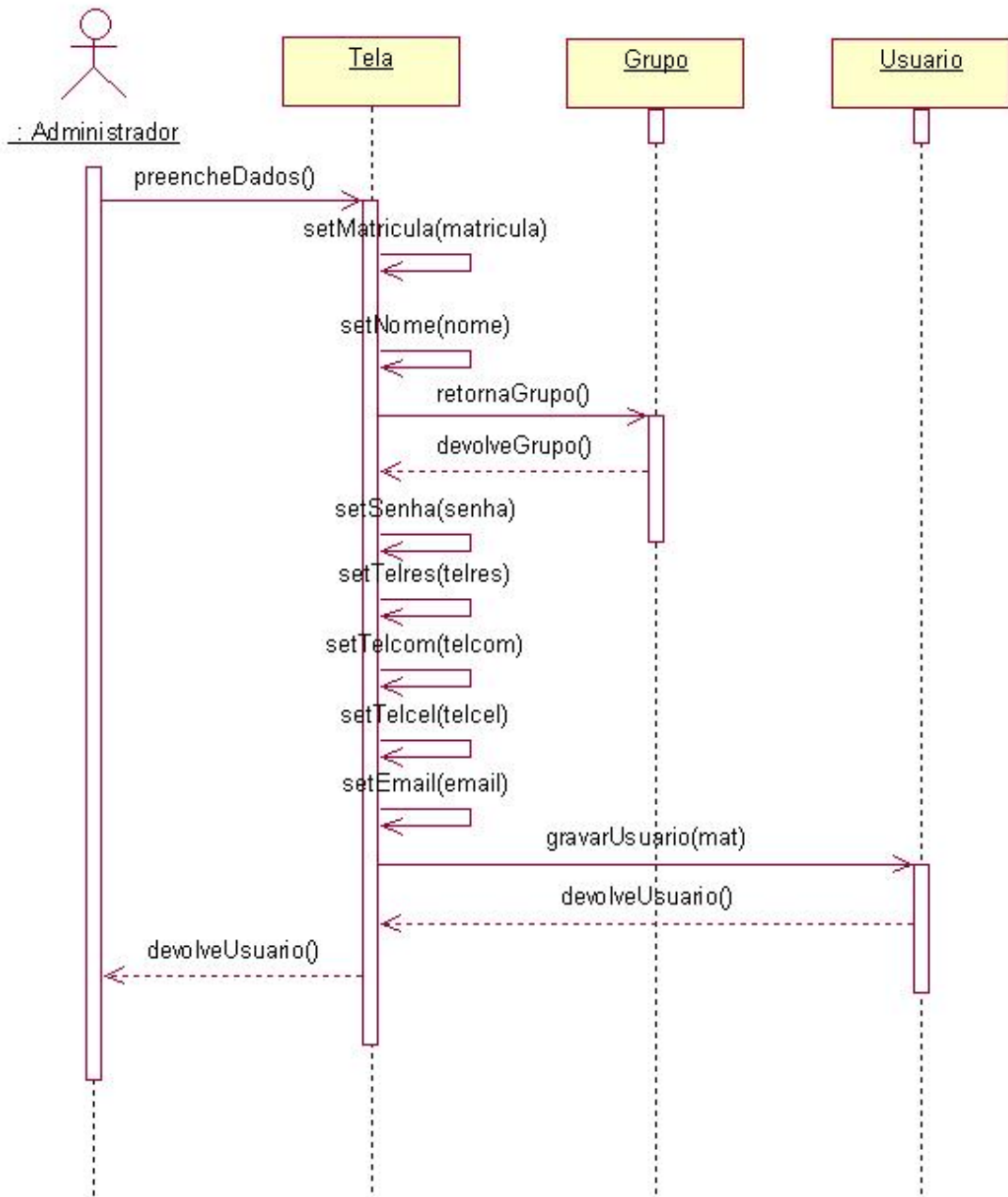


8.3.1.18 Cadastrar Usuário – Novo

QUADRO 18 – CADASTRAR USUÁRIO NOVO

Nome do Caso de Uso	Cadastrar Usuario Novo	
Descrição	Cadastramento de novos usuários que terão acesso ao sistema, além da definição do grupo ao qual cada usuário faz parte, associando-se, desta forma, as suas permissões no sistema.	
Ator(es) Envolvido(s)	Administrador	
Interação entre Ator e Sistema	Ator(es)	Sistema
	O caso de uso é iniciado quando ator seleciona a opção Gerenciamento - Usuário - Cadastrar Novo.	
		Sistema apresenta tela com campos: Matrícula, Nome, Grupo, Senha, Confirmar Senha, Telefone Residencial, Telefone Comercial, E-mail e Descrição.
	Ator preenche campos e clica no botão Gravar.	
		Sistema realiza verificação das informações inseridas pelo usuário.
		Sistema realiza cadastramento do usuário. Neste momento este caso de uso é encerrado.
Exceções	EX1 - Os campos devem estar todos preenchidos e de acordo com o domínio do atributo.	

FIGURA 55 – CADASTRAR USUÁRIO NOVO



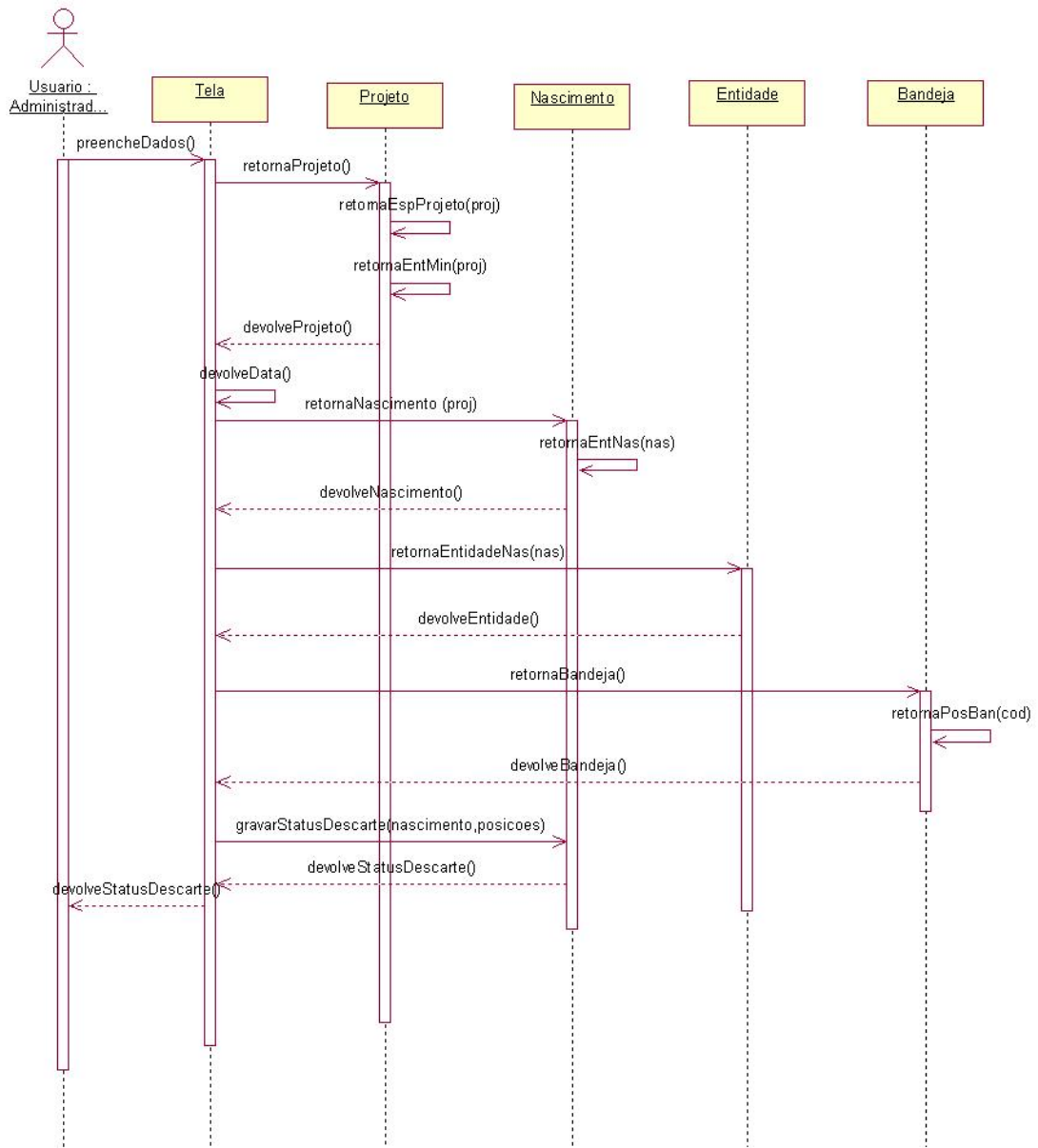
8.3.1.19 Embriodiagnosticar Ovos – Novo

QUADRO 19 – EMBRIODIAGNOSTICAR OVOS NOVO

Nome do Caso de Uso	Embriodiagnosticar Ovos Novo	
Descrição	Novo embriodiagnóstico de ovos, ou seja, possível análise da causa da morte do embrião, realizada para ovos que não eclodiram na etapa de nascimento (ovos que restaram no nascedouro após saque).	
Ator(es) Envolvido(s)	Administrador, Operador	
Interação entre Ator e Sistema	Ator(es)	Sistema
	O caso de uso é iniciado quando ator seleciona a opção Embriodiagnóstico - Novo Embriodiagnóstico	
		Sistema apresenta série de campos a serem preenchidos pelo usuário.
	Ator seleciona um dos projetos em andamento apresentados no campo Projeto.	
		Sistema identifica, para o dado projeto, a espécie de ave e o tipo de entidade associados, a data de início do nascimento e a previsão de término. O sistema busca, também, todos os nascimentos realizados para o dado projeto.
	Ator seleciona um dos nascimentos disponíveis	
		Sistema identifica todas as entidades que foram inseridas no nascimento selecionado pelo ator
	Ator seleciona uma das entidades disponíveis.	
		Sistema traz as posições da bandeja que correspondem à dada entidade, para ovos não identificados como efetivamente nascidos.
	Ator identifica quais as posições que correspondem a ovos que não eclodiram, e para os quais foi realizada análise do embrião (embriodiagnóstico), e seleciona um dos motivos no campo Motivo. Posteriormente o ator clica na opção Gravar.	

		Sistema realiza cadastramento das informações dos ovos não eclodidos, o motivo e suas respectivas posições no nascedouro.
	Caso seja necessário o ator pode selecionar outra entidade, e identificar um novo conjunto de ovos para os quais foi realizado o embriodiagnóstico.	
		Neste momento este caso de uso é encerrado.
Exceções	EX1 - O usuário pode somente selecionar ovos de entidades previamente inseridas no nascedouro.	
	EX2 - O usuário pode somente selecionar ovos que não foram previamente identificados como nascidos.	

FIGURA 56 – EMBRIODIAGNOSTICAR OVOS NOVO

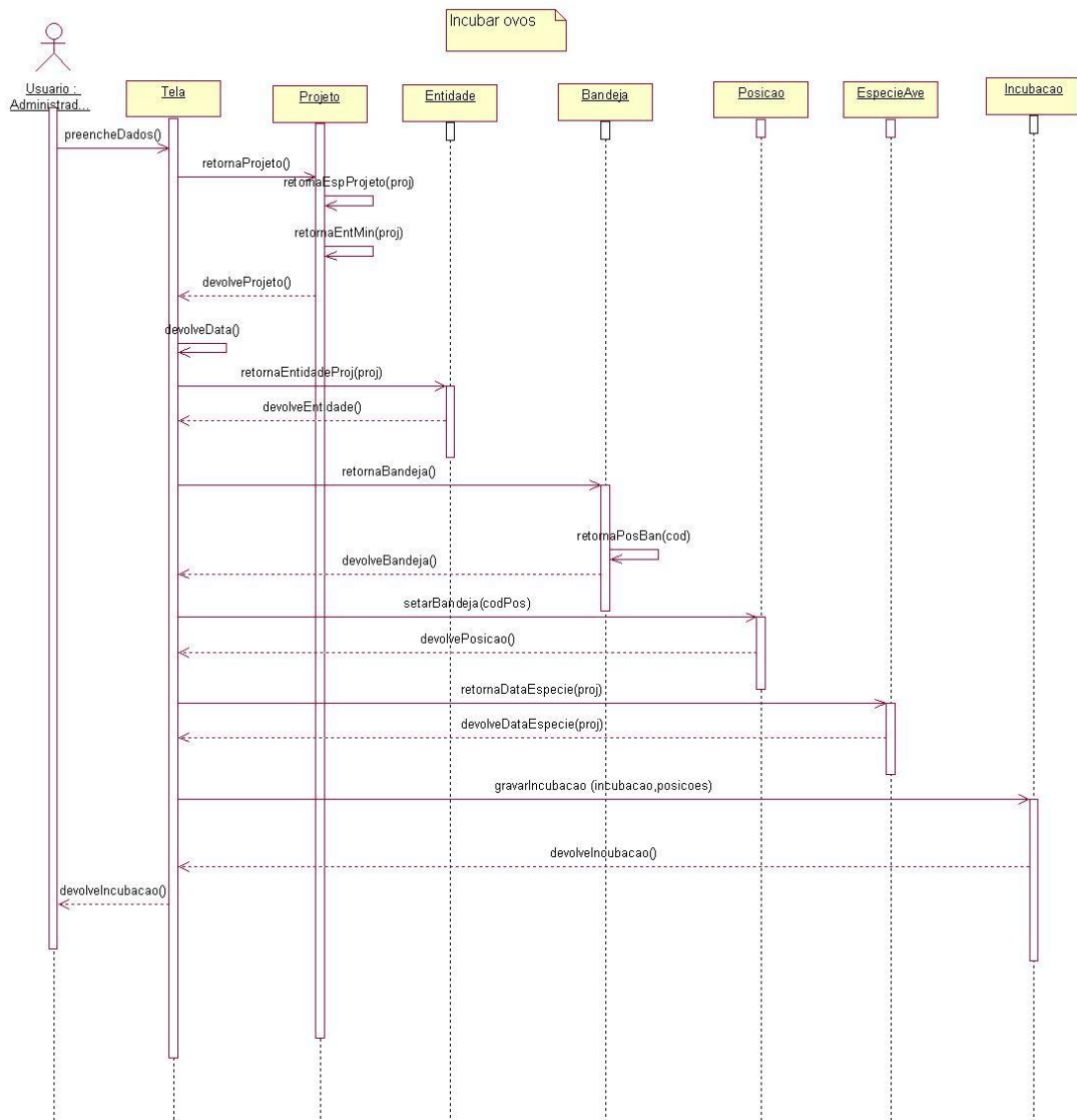


8.3.1.20 Incubar Ovos – Novo

QUADRO 20 – INCUBAR OVOS NOVO

Nome do Caso de Uso	Incubar Ovos Novo	
Descrição	Incubação de novo conjunto de ovos, associada a determinado projeto selecionado pelo usuário.	
Ator(es) Envolvido(s)	Administrador, Operador	
Interação entre Ator e Sistema	Ator(es)	Sistema
	O caso de uso é iniciado quando ator seleciona a opção Incubação - Nova Incubação.	
		Sistema apresenta série de campos a serem preenchidos pelo usuário.
	Ator seleciona um dos projetos em andamento apresentados no campo Projeto.	
		Sistema identifica, para o dado projeto, a espécie de ave e o tipo de entidade associados. A data de início da incubação corresponde à data corrente (definida pelo sistema). A previsão de término é dependente da espécie de ave associada.
	Ator seleciona uma das entidades disponíveis.	
	Ator seleciona uma das bandejas alocadas (bandejas associadas a incubadoras) e clica em Buscar Bandeja.	
		Sistema traz representação gráfica das posições disponíveis na bandeja selecionada.
	Ator clica nas posições a serem ocupadas por ovos da dada entidade na bandeja, e clica em Incubar.	
		Sistema realiza cadastramento dos ovos e suas respectivas posições na incubação.
	Caso seja necessário o ator pode selecionar outra entidade, outra bandeja, selecionar novas posições disponíveis e realizar uma nova incubação.	
	Neste momento este caso de uso é encerrado.	
Exceções	EX1 - O usuário pode somente selecionar posições disponíveis na dada alocação de bandeja.	

FIGURA 57 – INCUBAR OVOS NOVO

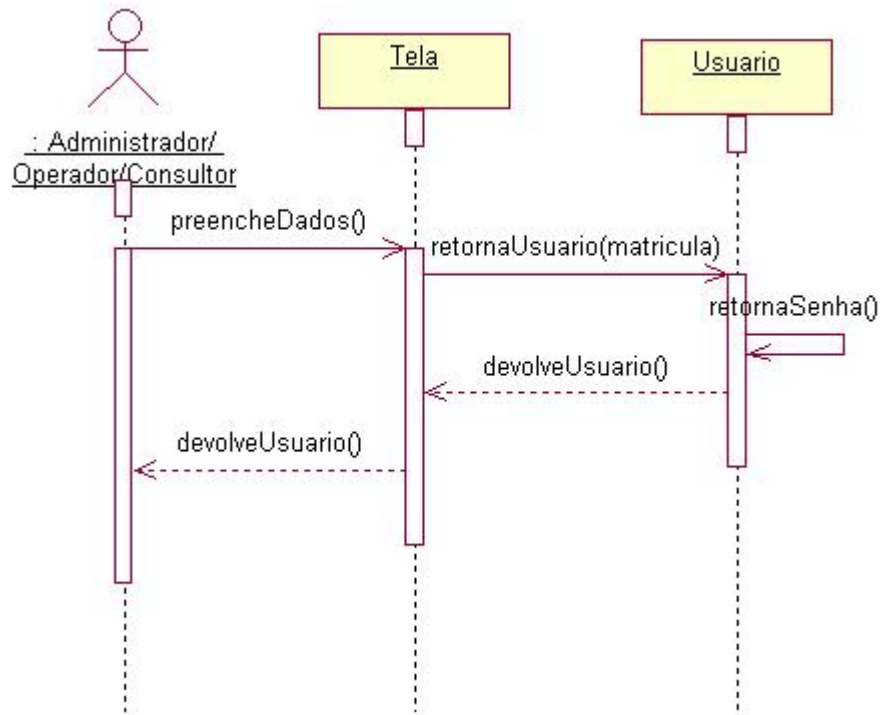


8.3.1.21 Efetuar Login – Novo

QUADRO 21 – EFETUAR LOGIN NOVO

Nome do Caso de Uso	Realizar Login Novo	
Descrição	Verificação de cadastramento de todos os usuários que realizem nova tentativa de acesso ao sistema	
Ator(es) Envolvido(s)	Administrador, Operador, Consultor	
Interação entre Ator e Sistema	Ator(es)	Sistema
	O caso de uso é iniciado quando ator realiza acesso à página inicial do sistema.	
		Sistema apresenta tela com campos: Matrícula e Senha.
	Ator preenche campos e clica no botão Entrar.	
		Sistema realiza verificação da existência do cadastramento do usuário representado pela matrícula inserida e das permissões associadas ao dado usuário.
		Sistema permite acesso ao sistema caso se trate de usuário cadastrado e devidamente identificado (senha). Caso contrário é pedido ao usuário que efetue novo login.
		Neste momento este caso de uso é encerrado.
Exceções	EX1 - Os campos devem estar todos preenchidos e de acordo com o domínio do atributo.	
	EX2 - O usuário deve ter sido devidamente cadastrado e suas permissões associadas para que seja garantido o acesso ao sistema.	

FIGURA 58 – EFETUAR LOGIN NOVO



8.3.1.22 Retirar Ovos (Saque) – Novo

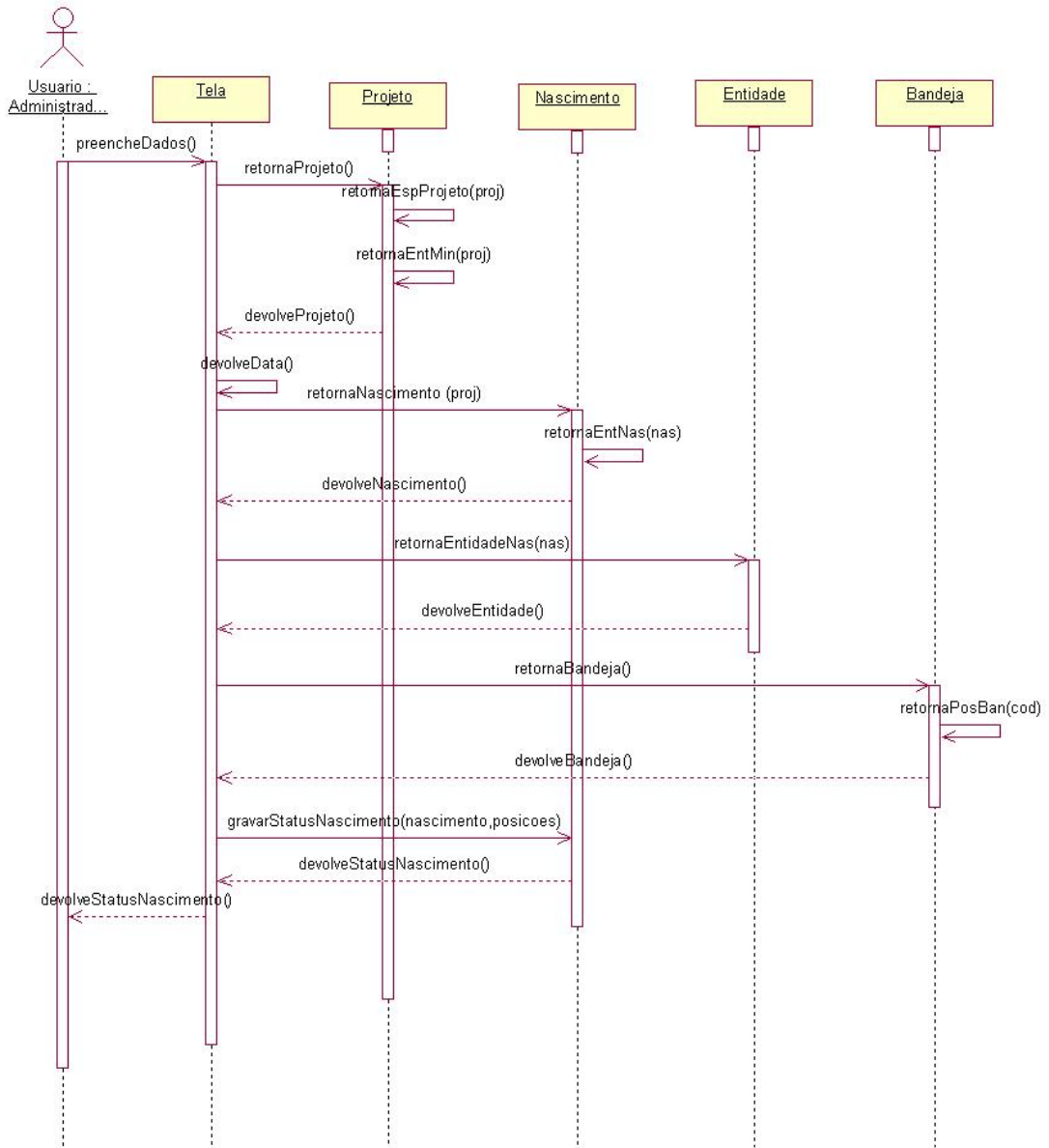
QUADRO 22 – RETIRAR OVOS (SAQUE) NOVO

Nome do Caso de Uso	Retirar Ovos (Saque) Novo	
Descrição	Novo saque de ovos, que saem do nascedouro (equipamento de nascimento).	
Ator(es) Envolvido(s)	Administrador, Operador	
Interação entre Ator e Sistema	Ator(es)	Sistema
	O caso de uso é iniciado quando ator seleciona a opção Sacar Ovos - Novo Saque.	
		Sistema apresenta série de campos a serem preenchidos pelo usuário.
	Ator seleciona um dos projetos em andamento apresentados no campo Projeto.	
		Sistema identifica, para o dado projeto, a espécie de ave e o tipo de entidade associados, a data de início do nascimento e a previsão de término. O sistema busca, também, todos os nascimentos realizados para o dado projeto.
	Ator seleciona um dos nascimentos disponíveis	
		Sistema identifica todas as entidades que foram inseridas no nascimento selecionado pelo ator
	Ator seleciona uma das entidades disponíveis.	
		Sistema traz as posições da bandeja que correspondem à dada entidade.
	Ator identifica quais as posições que correspondem a ovos que efetivamente nasceram, e clica na opção Gravar.	
		Sistema realiza cadastramento das informações dos ovos efetivamente nascidos e suas respectivas posições no nascedouro.
Caso seja necessário o ator pode selecionar outra entidade, e identificar um novo conjunto de ovos efetivamente nascidos.		
	Neste momento este caso de uso é encerrado.	

Exceções

EX1 - O usuário pode somente selecionar ovos de entidades previamente inseridas no nascedouro.

FIGURA 59 – RETIRAR OVOS (SAQUE) NOVO



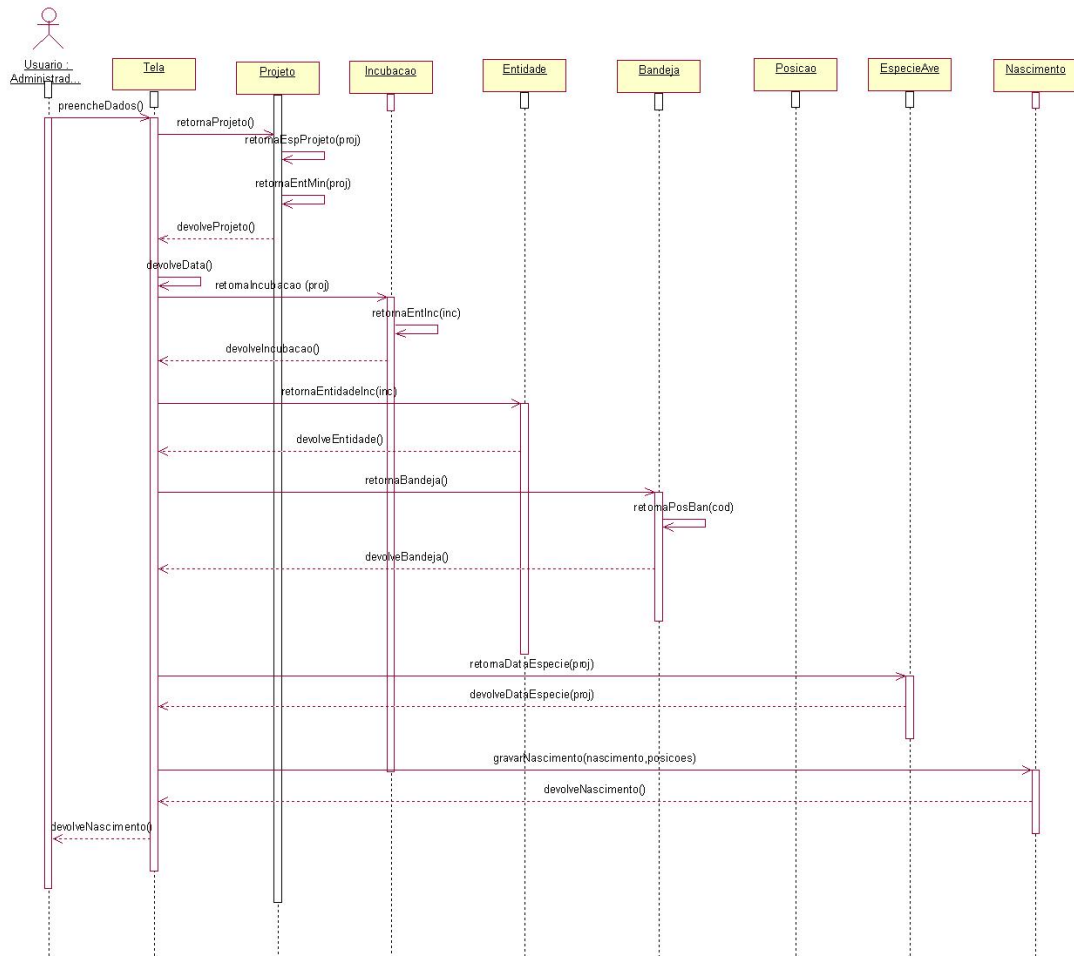
8.3.1.23 Transferir Ovos – Novo

QUADRO 23 – TRANSFERIR OVOS NOVO

Nome do Caso de Uso	Transferir Ovos Novo	
Descrição	Nova transferência de ovos, que saem da incubadora (equipamento de incubação) e são colocados no nascedouro.	
Ator(es) Envolvido(s)	Administrador, Operador	
Interação entre Ator e Sistema	Ator(es)	Sistema
	O caso de uso é iniciado quando ator seleciona a opção Transferir - Nova Transferência.	
		Sistema apresenta série de campos a serem preenchidos pelo usuário.
	Ator seleciona um dos projetos em andamento apresentados no campo Projeto.	
		Sistema identifica, para o dado projeto, a espécie de ave e o tipo de entidade associados. A data de início do nascimento corresponde à data corrente (definida pelo sistema). A previsão de término é dependente da espécie de ave associada. O sistema busca, também, todas as incubações realizadas para o dado projeto.
	Ator seleciona uma das incubações disponíveis	
		Sistema identifica todas as entidades que foram incubadas na incubação selecionada pelo ator
	Ator seleciona uma das entidades disponíveis.	
	Ator seleciona uma das bandejas alocadas (bandejas associadas a nascedouros) e clica em Buscar Bandeja.	
		Sistema traz representação gráfica das posições disponíveis na bandeja selecionada.
Ator clica nas posições a serem ocupadas por ovos da dada entidade na bandeja, e clica em Transferir.		
	Sistema realiza cadastramento dos ovos e suas respectivas posições no nascedouro.	

	Caso seja necessário o ator pode selecionar outra entidade, outra bandeja, selecionar novas posições disponíveis e realizar uma nova transferência.	
		Neste momento este caso de uso é encerrado.
Exceções	EX1 - O usuário pode somente selecionar posições disponíveis na dada alocação de bandeja.	
	EX2 - As entidades disponíveis correspondem àquelas que foram inicialmente incubadas.	

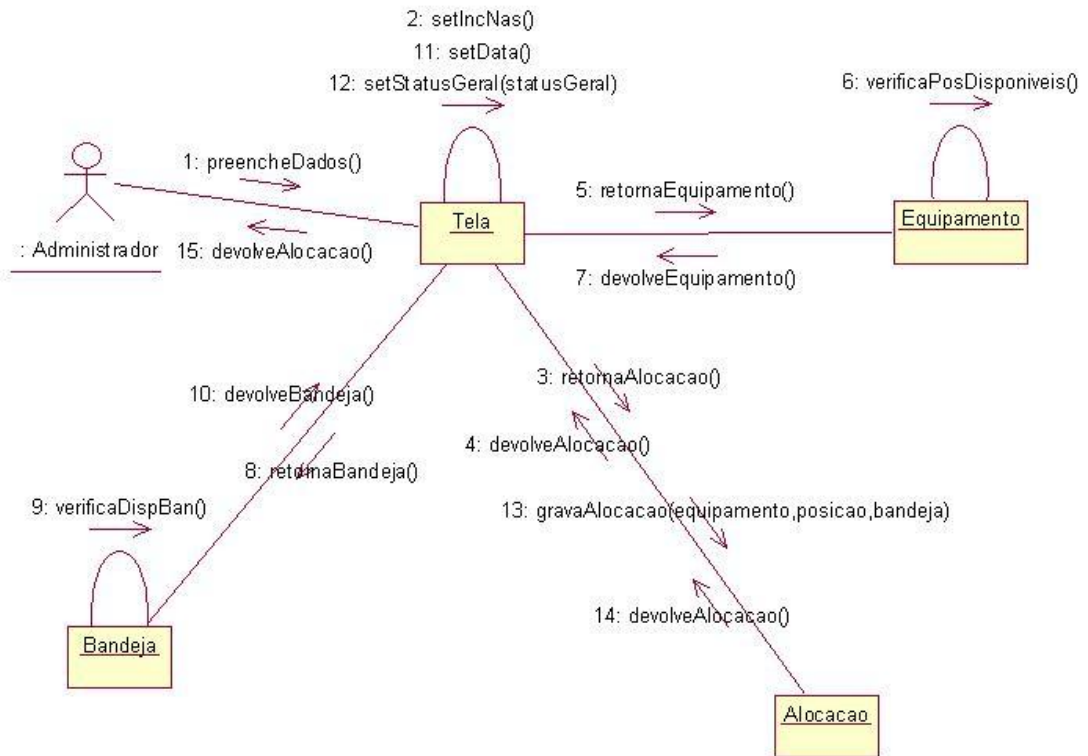
FIGURA 60 – TRANSFERIR OVOS NOVO



8.3.2 Diagramas de Colaboração

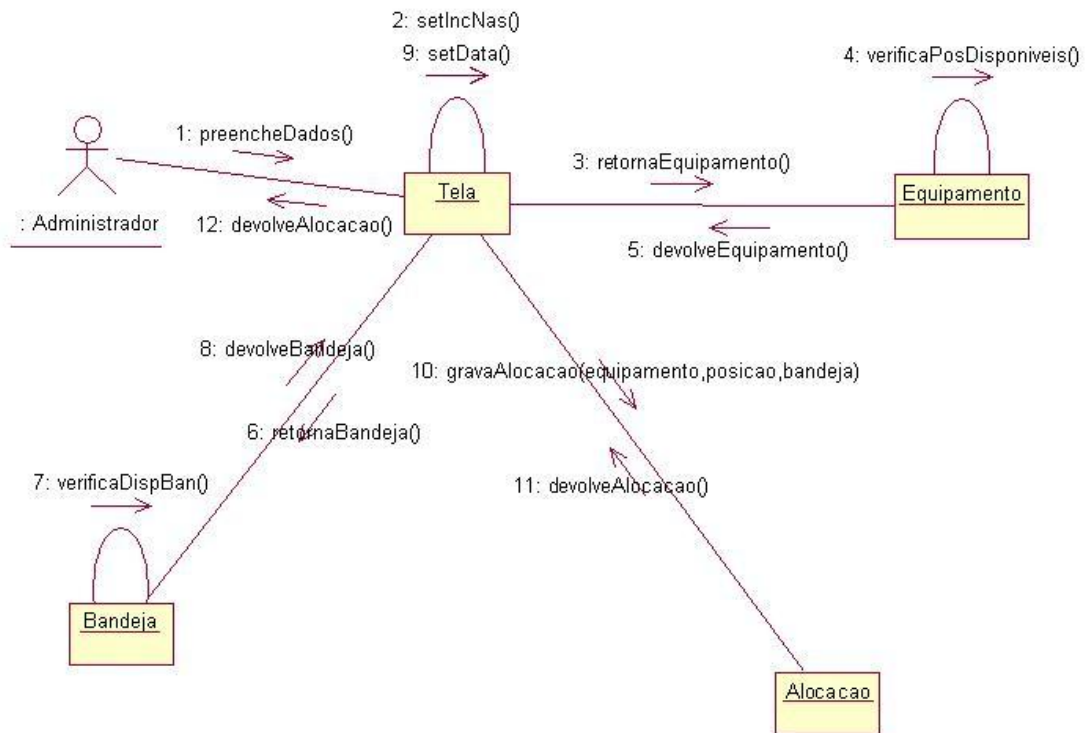
8.3.2.1 Alocar Bandeja – Alterar

FIGURA 61 – ALOCAR BANDEJA ALTERAR



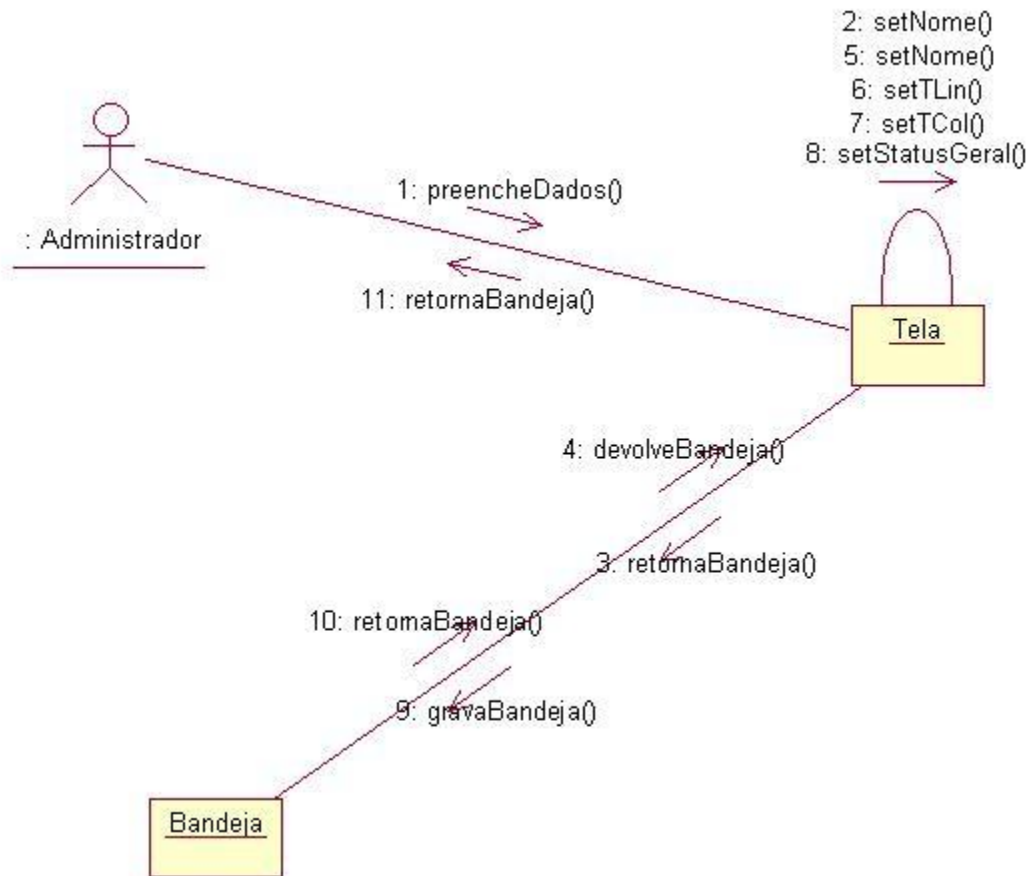
8.3.2.2 Alocar Bandeja – Novo

FIGURA 62 – ALOCAR BANDEJA NOVO



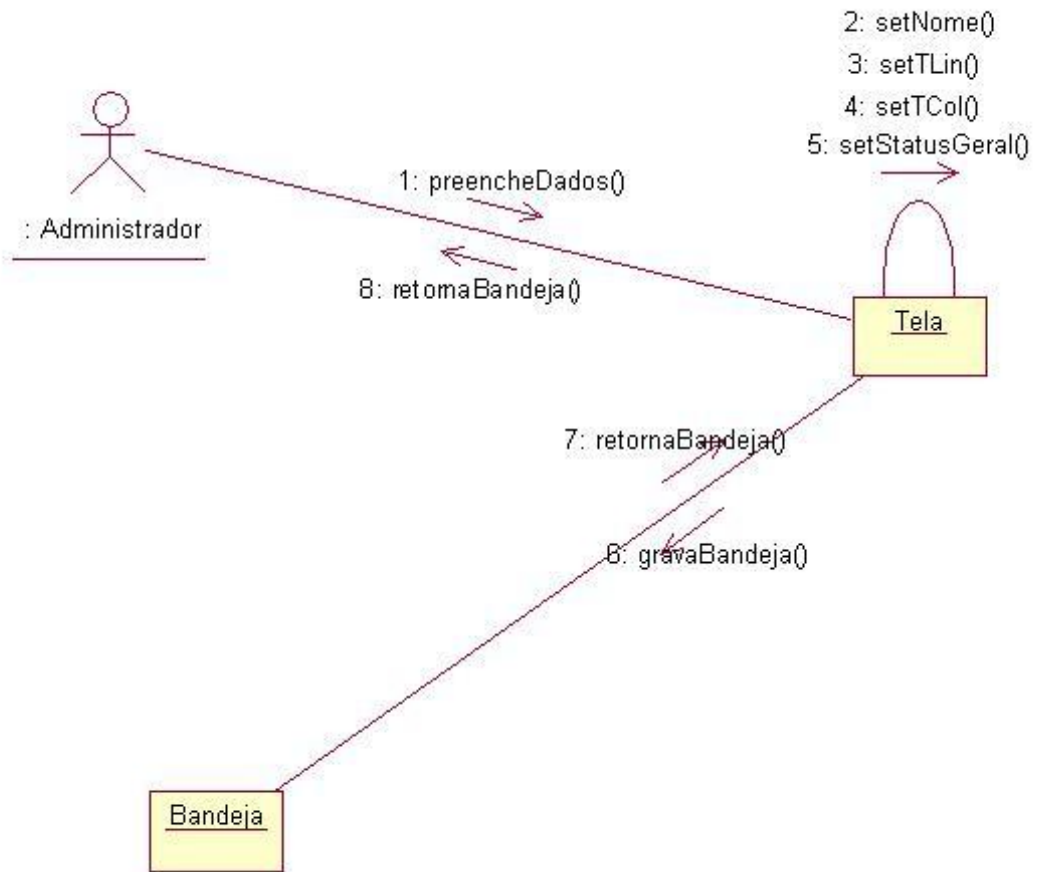
8.3.2.3 Cadastrar Bandeja – Alterar

FIGURA 63 – CADASTRAR BANDEJA ALTERAR



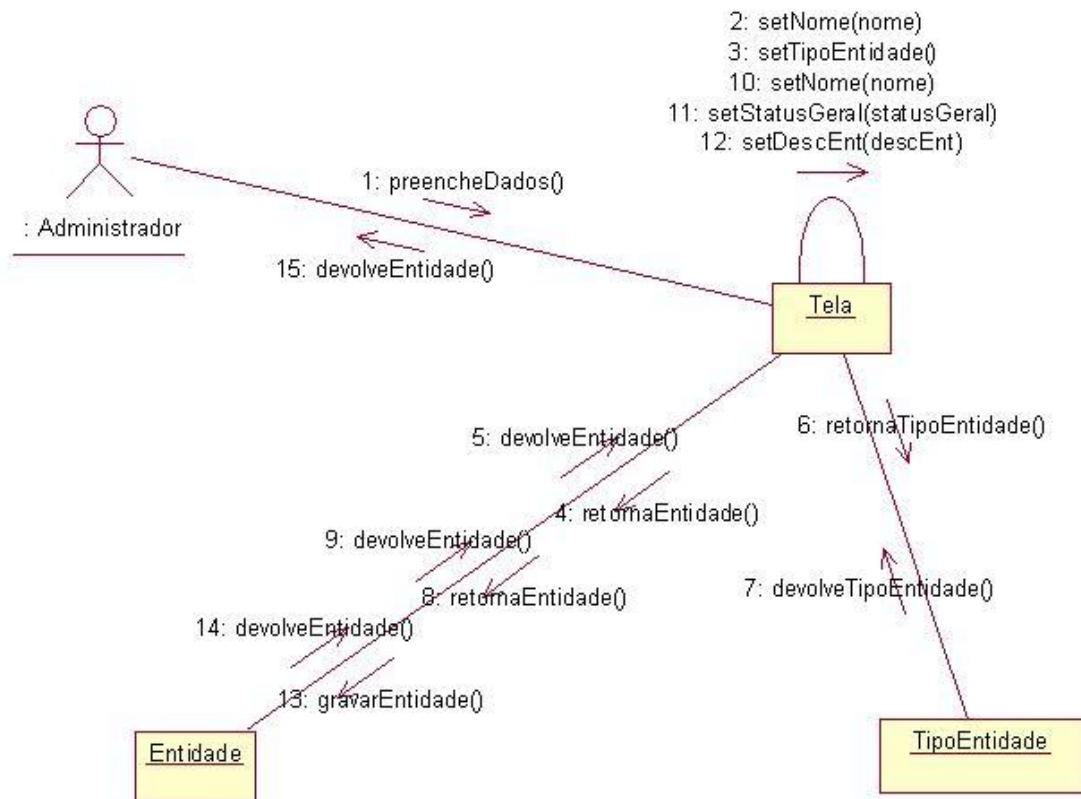
8.3.2.4 Cadastrar Bandeja – Novo

FIGURA 64 – CADASTRAR BANDEJA NOVO



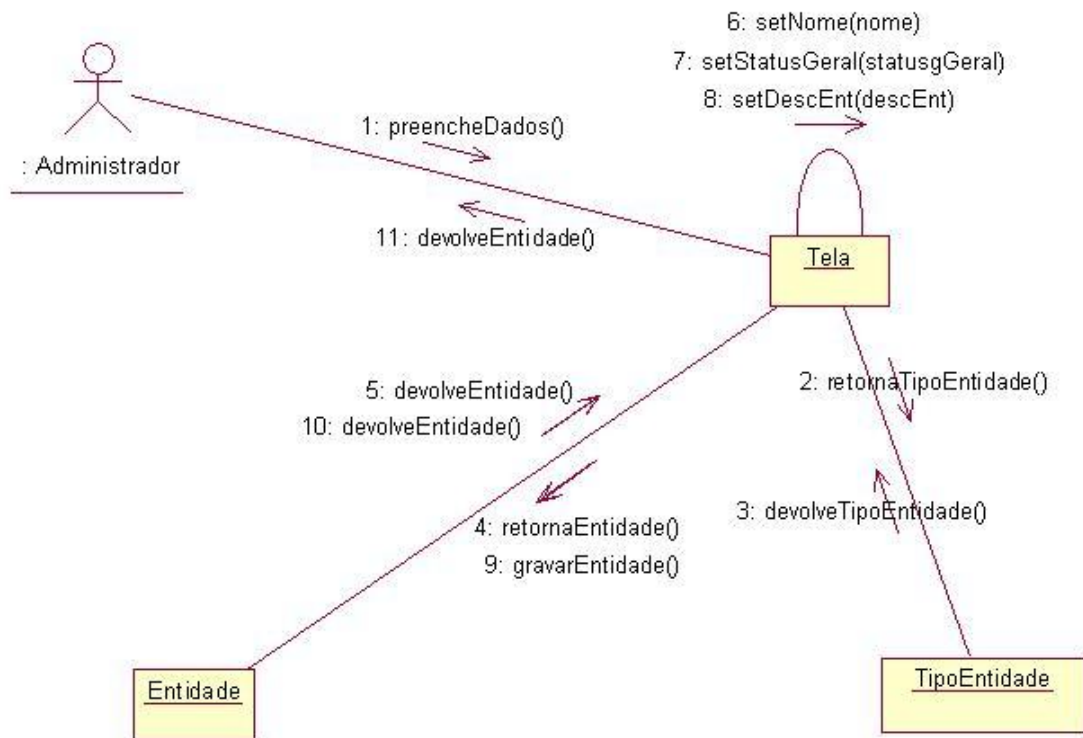
8.3.2.5 Cadastrar Entidade – Alterar

FIGURA 65 – CADASTRAR ENTIDADE ALTERAR



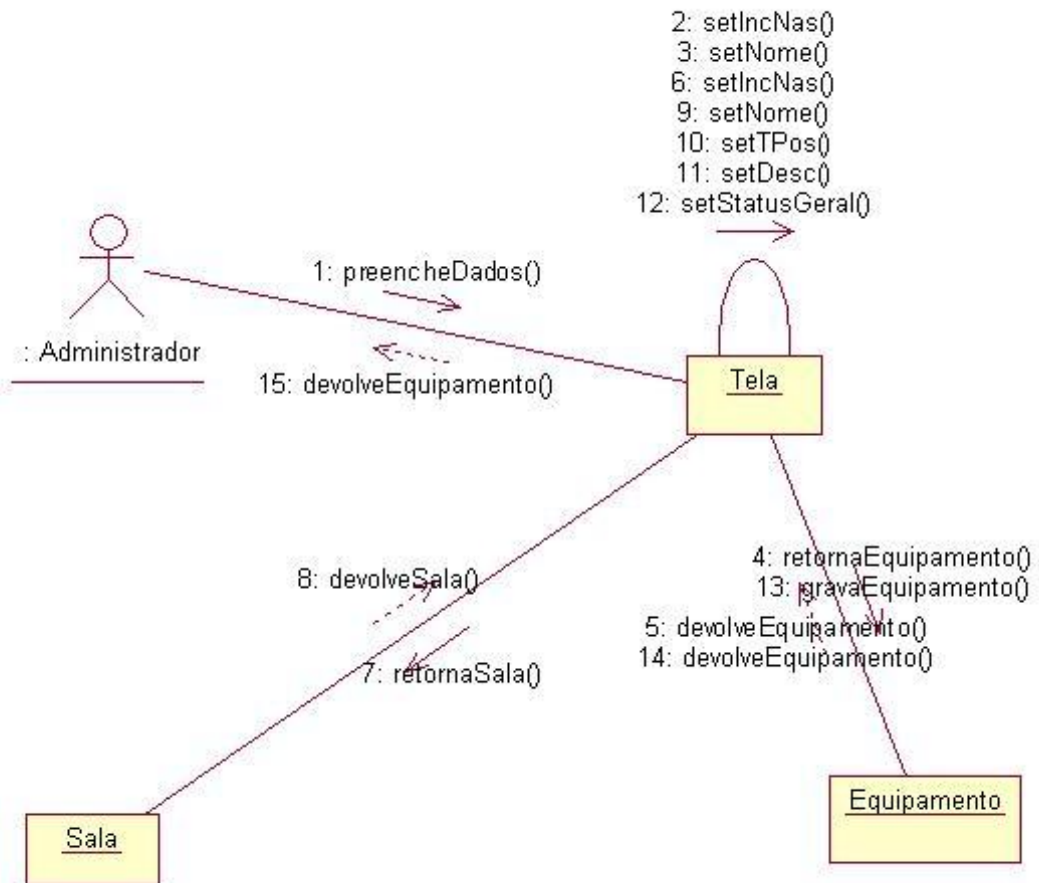
8.3.2.6 Cadastrar Entidade – Novo

FIGURA 66 – CADASTRAR ENTIDADE NOVO



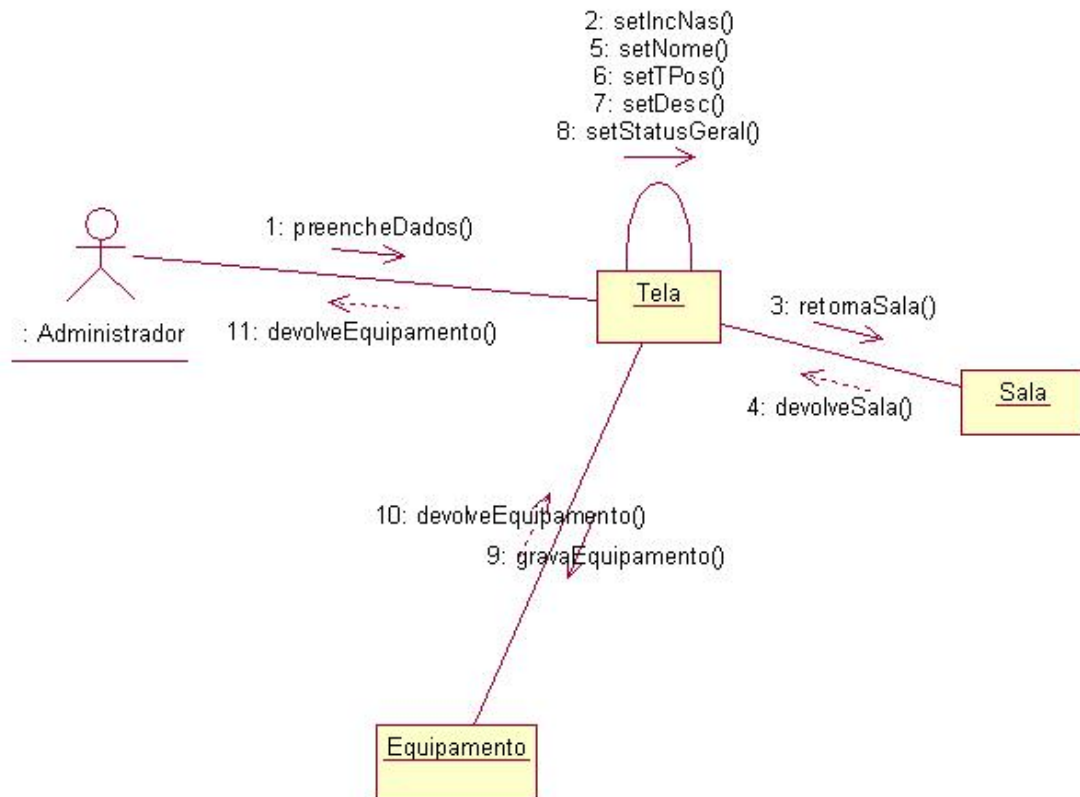
8.3.2.7 Cadastrar Equipamento – Alterar

FIGURA 67 – CADASTRAR EQUIPAMENTO ALTERAR



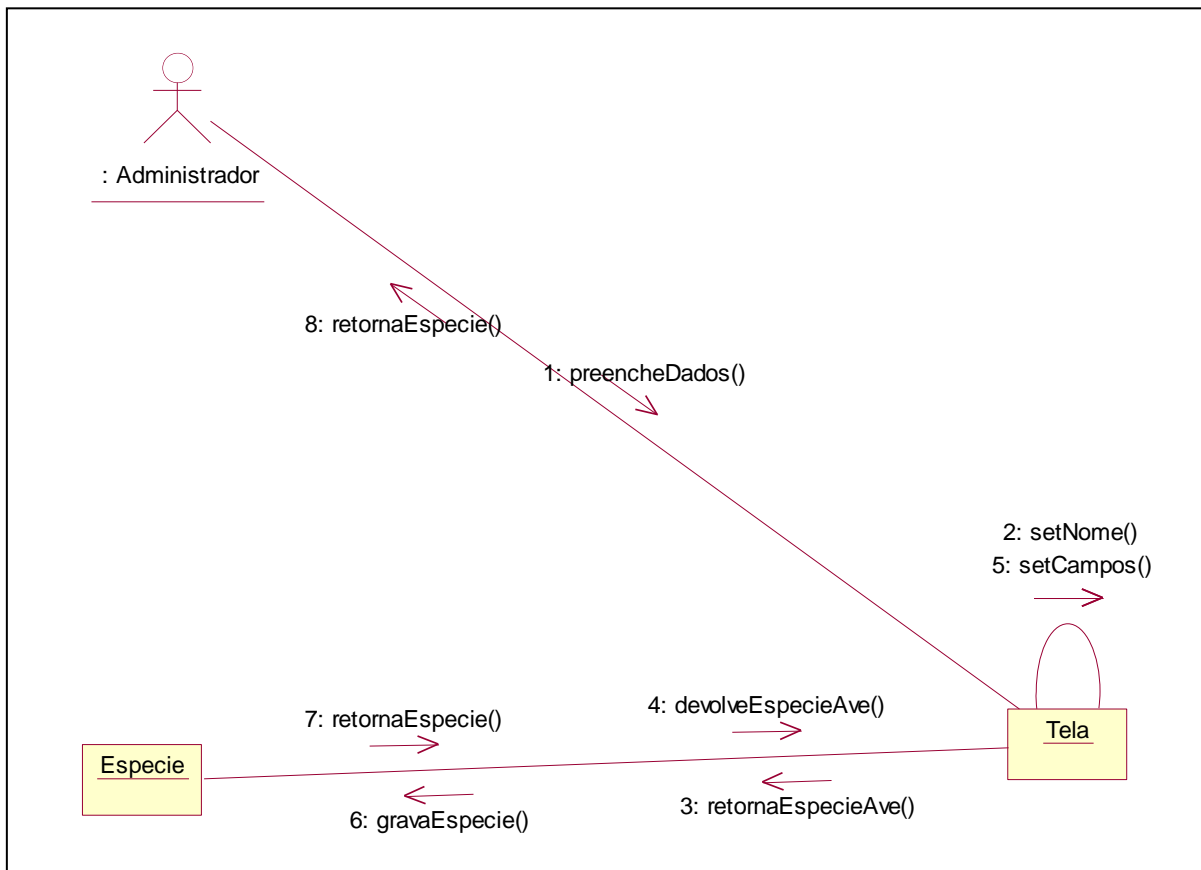
8.3.2.8 Cadastrar Equipamento – Novo

FIGURA 68 – CADASTRAR EQUIPAMENTO NOVO



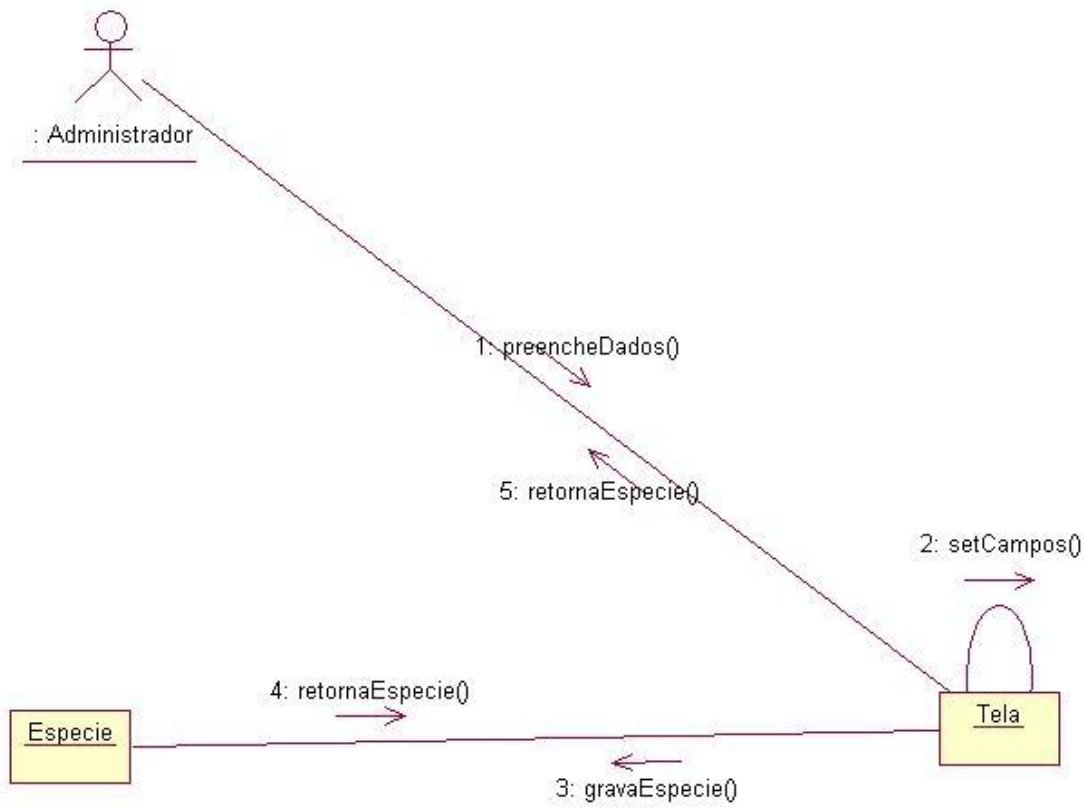
8.3.2.9 Cadastrar Espécie de Ave – Alterar

FIGURA 69 – CADASTRAR ESPÉCIE DE AVE ALTERAR



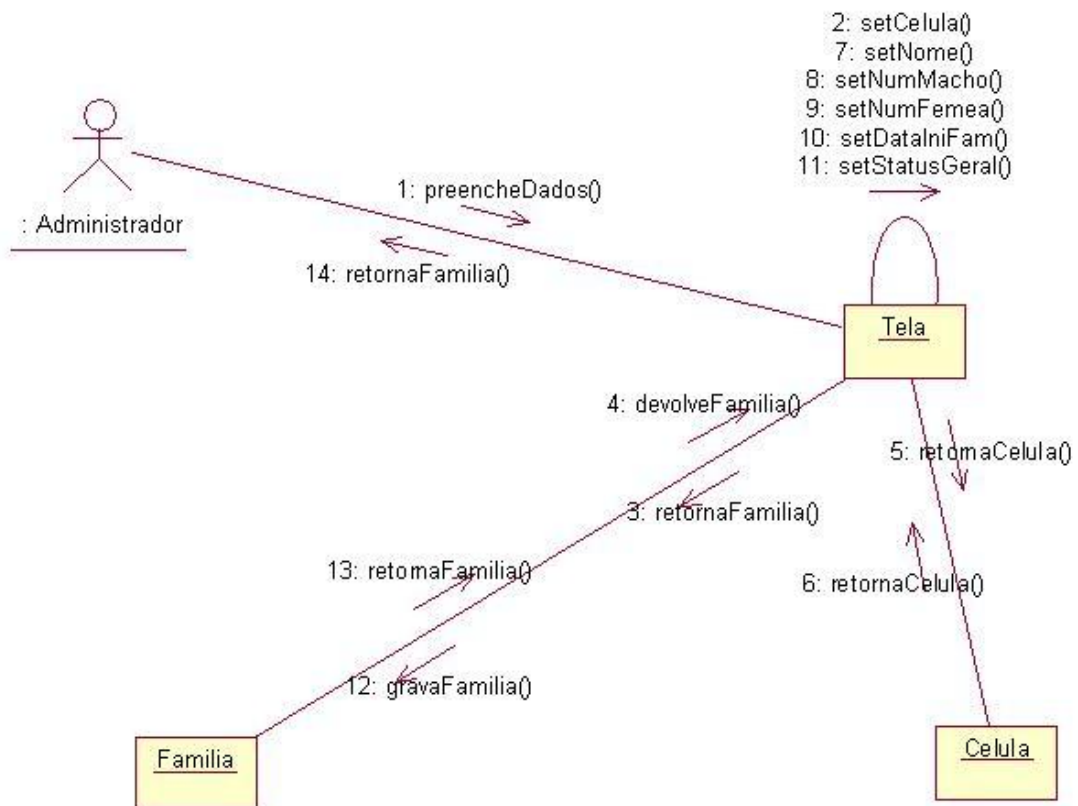
8.3.2.10 Cadastrar Espécie de Ave – Novo

FIGURA 70 – CADASTRAR ESPÉCIE DE AVE NOVO



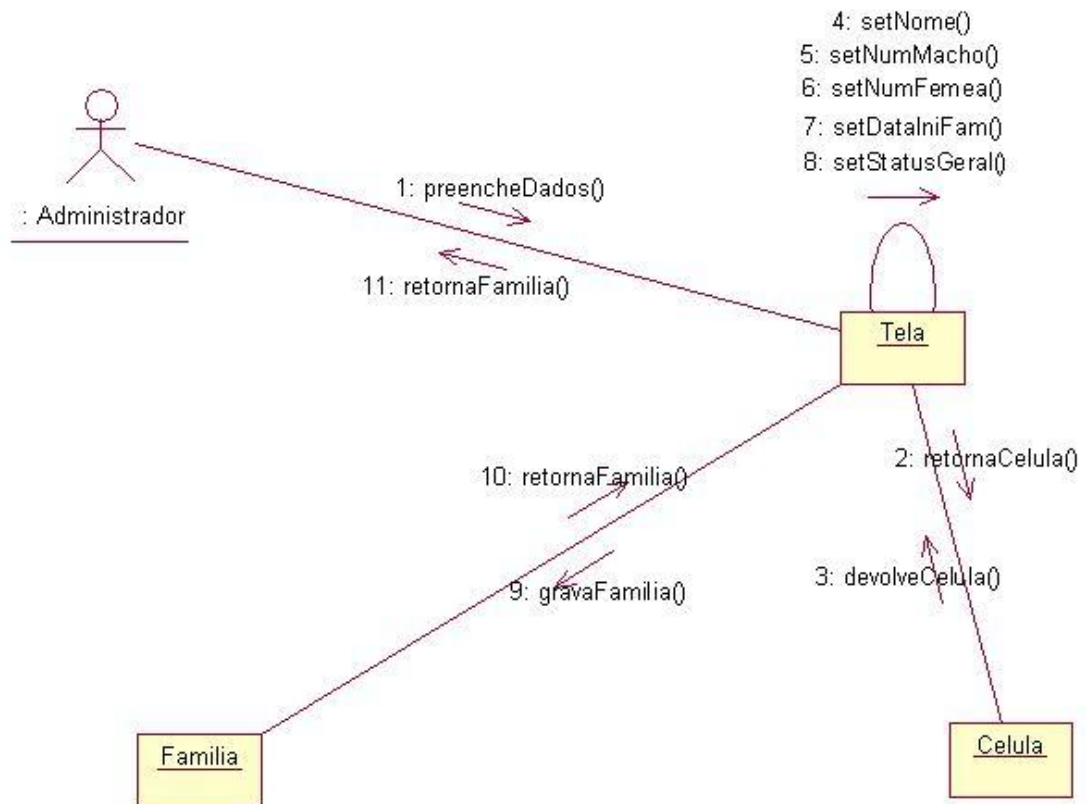
8.3.2.11 Cadastrar Família – Alterar

FIGURA 71 – CADASTRAR FAMÍLIA ALTERAR



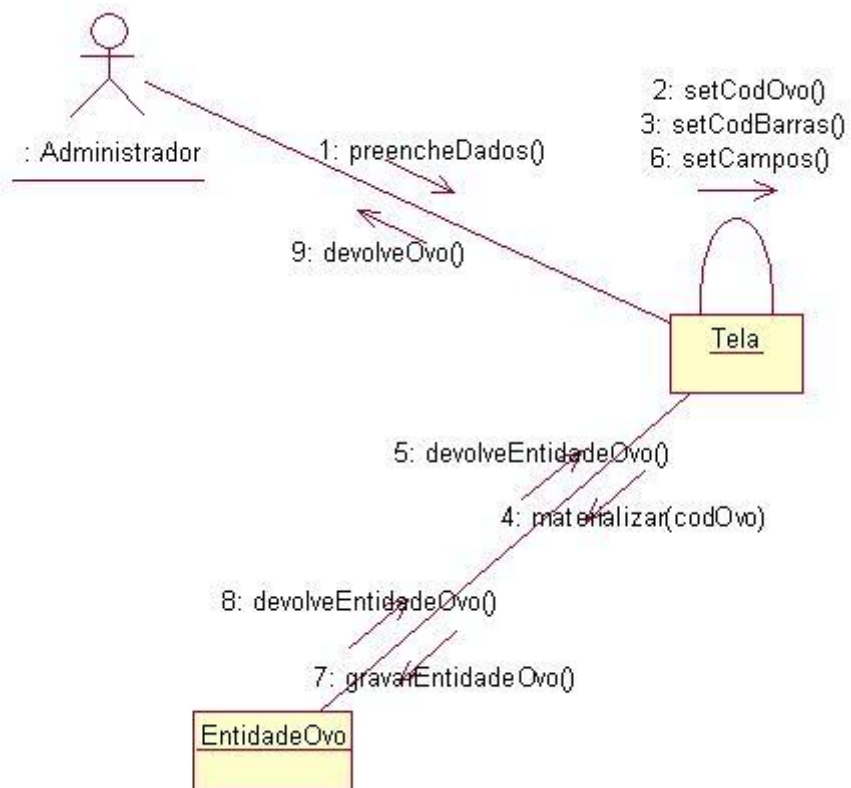
8.3.2.12 Cadastrar Família – Novo

FIGURA 72 – CADASTRAR FAMÍLIA NOVO



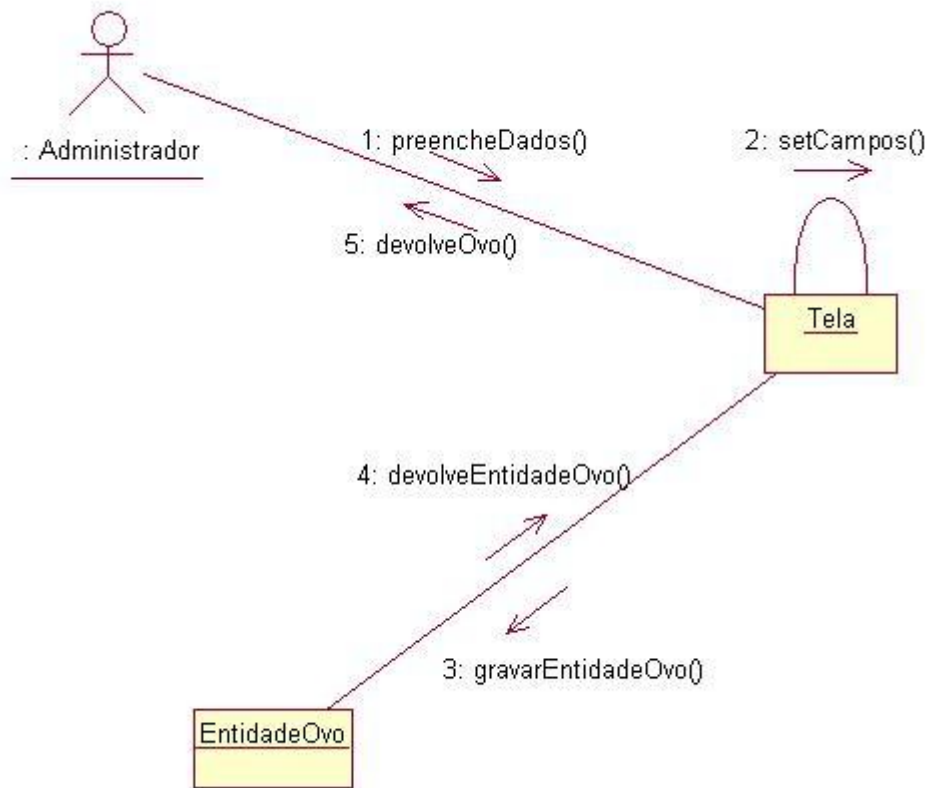
8.3.2.13 Cadastrar Ovo – Alterar

FIGURA 73 – CADASTRAR OVO ALTERAR



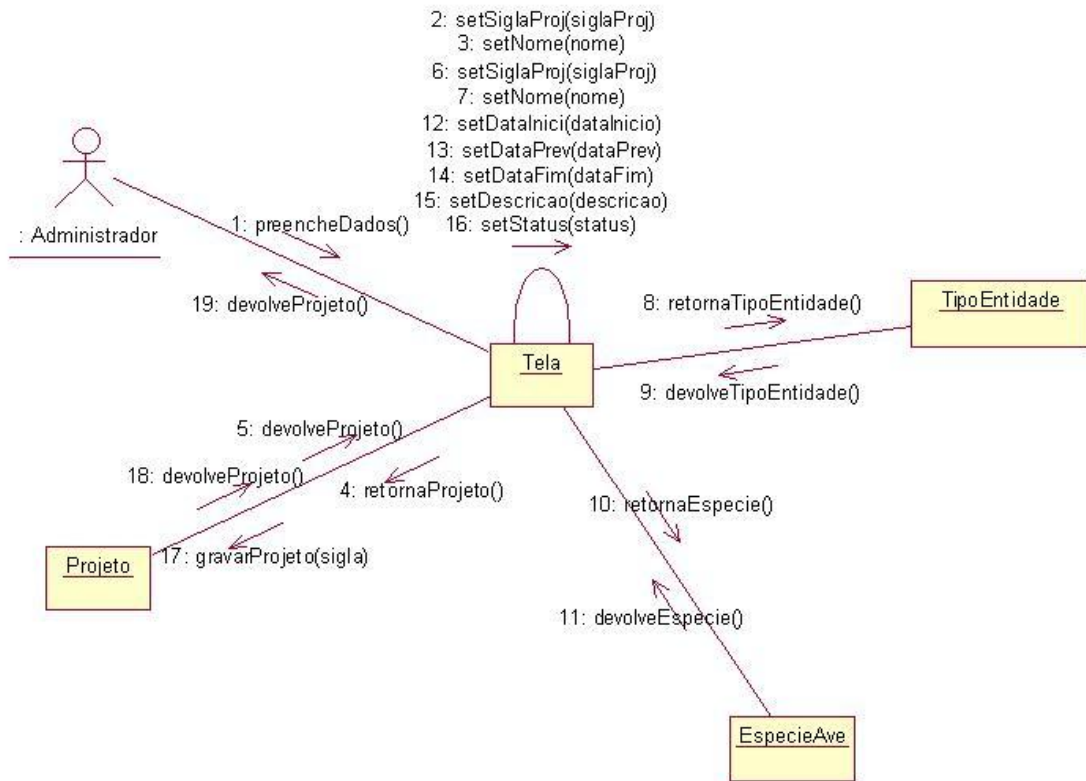
8.3.2.14 Cadastrar Ovo – Novo

FIGURA 74 – CADASTRAR OVO NOVO



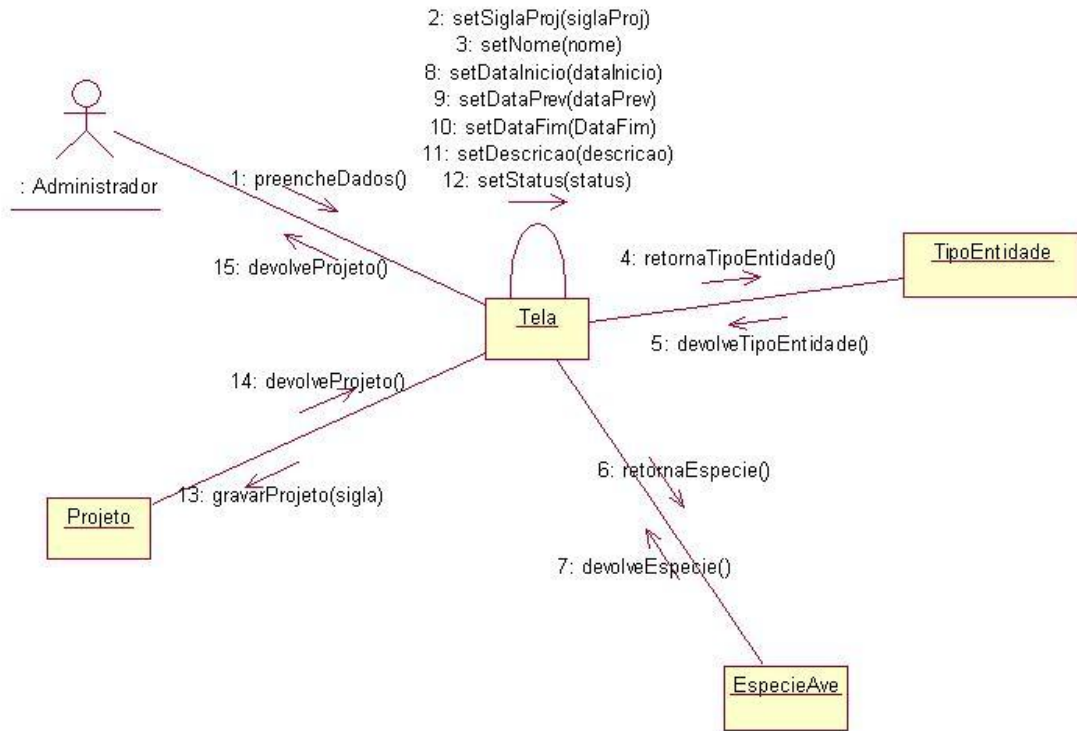
8.3.2.15 Cadastrar Projeto – Alterar

FIGURA 75 – CADASTRAR PROJETO ALTERAR



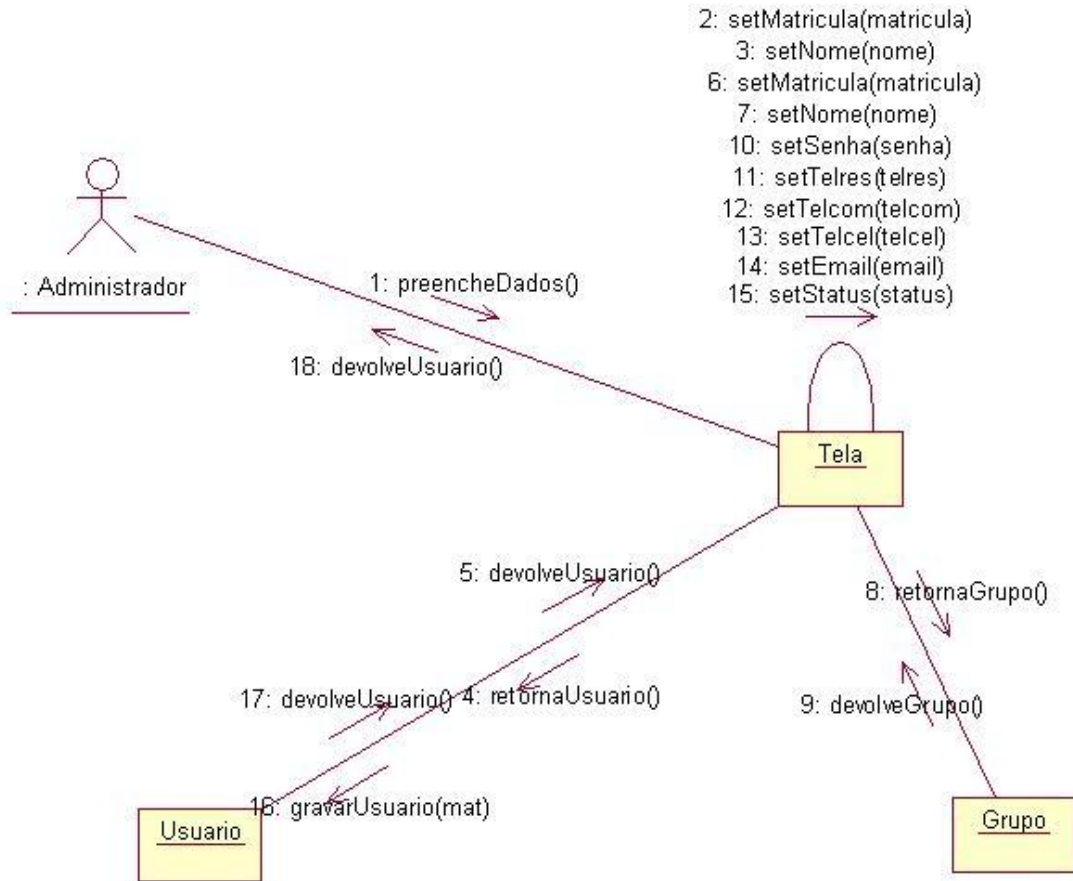
8.3.2.16 Cadastrar Projeto – Novo

FIGURA 76 – CADASTRAR PROJETO NOVO



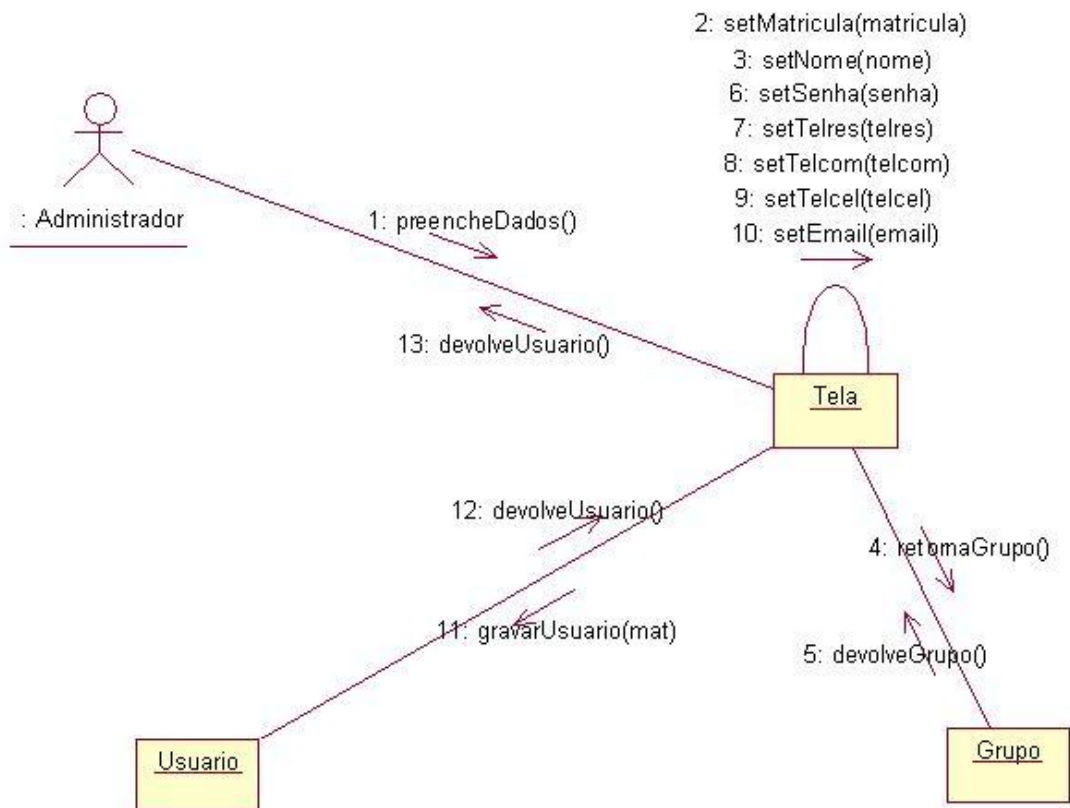
8.3.2.17 Cadastrar Usuário – Alterar

FIGURA 77 – CADASTRAR USUÁRIO ALTERAR



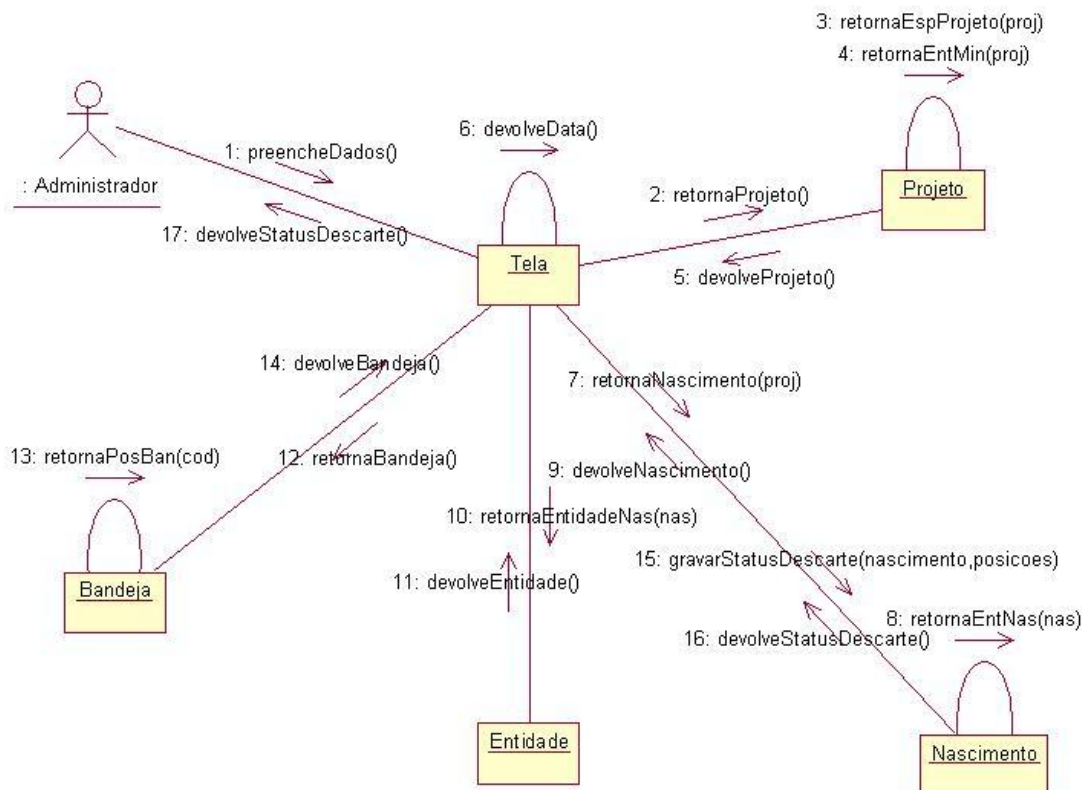
8.3.2.18 Cadastrar Usuário – Novo

FIGURA 78 – CADASTRAR USUÁRIO NOVO



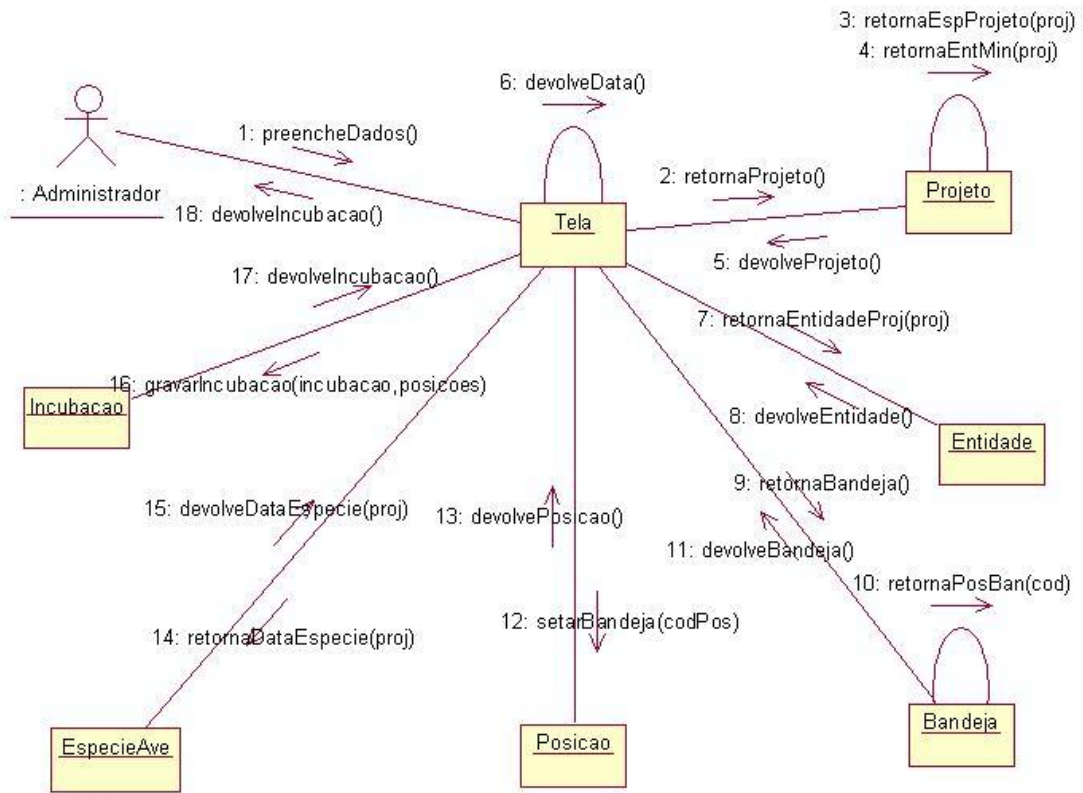
8.3.2.19 Embriodiagnosticar Ovos – Novo

FIGURA 79 – EMBRIODIAGNOSTICAR OVOS NOVO



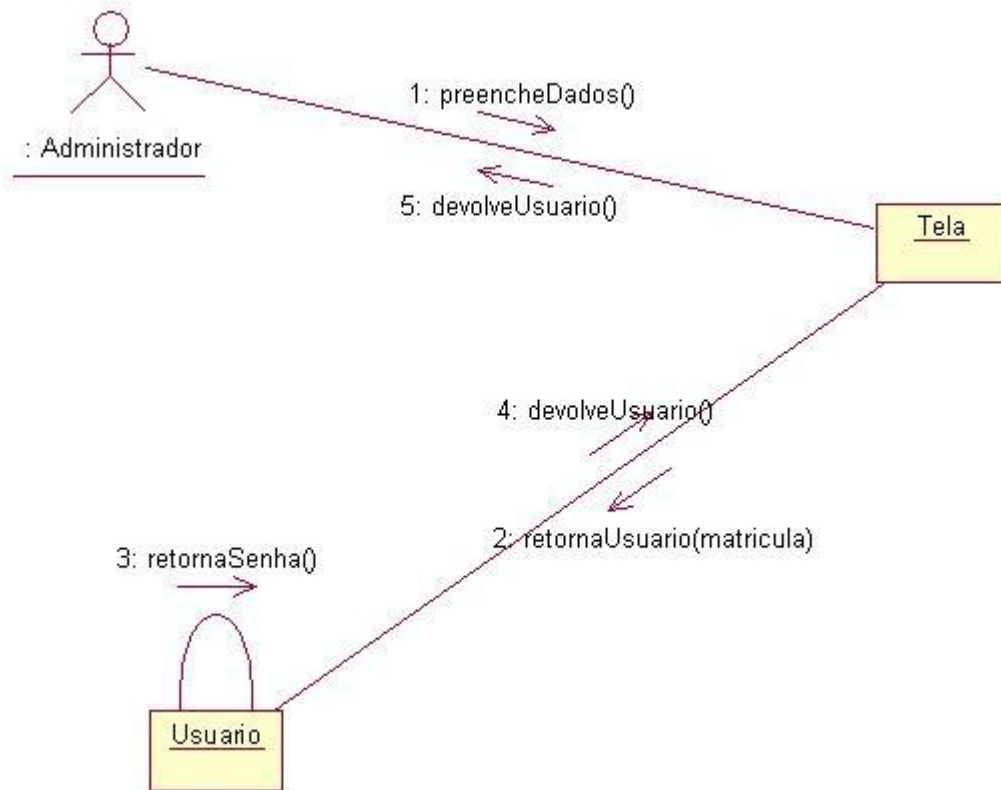
8.3.2.20 Incubar Ovos – Novo

FIGURA 80 – INCUBAR OVOS NOVO



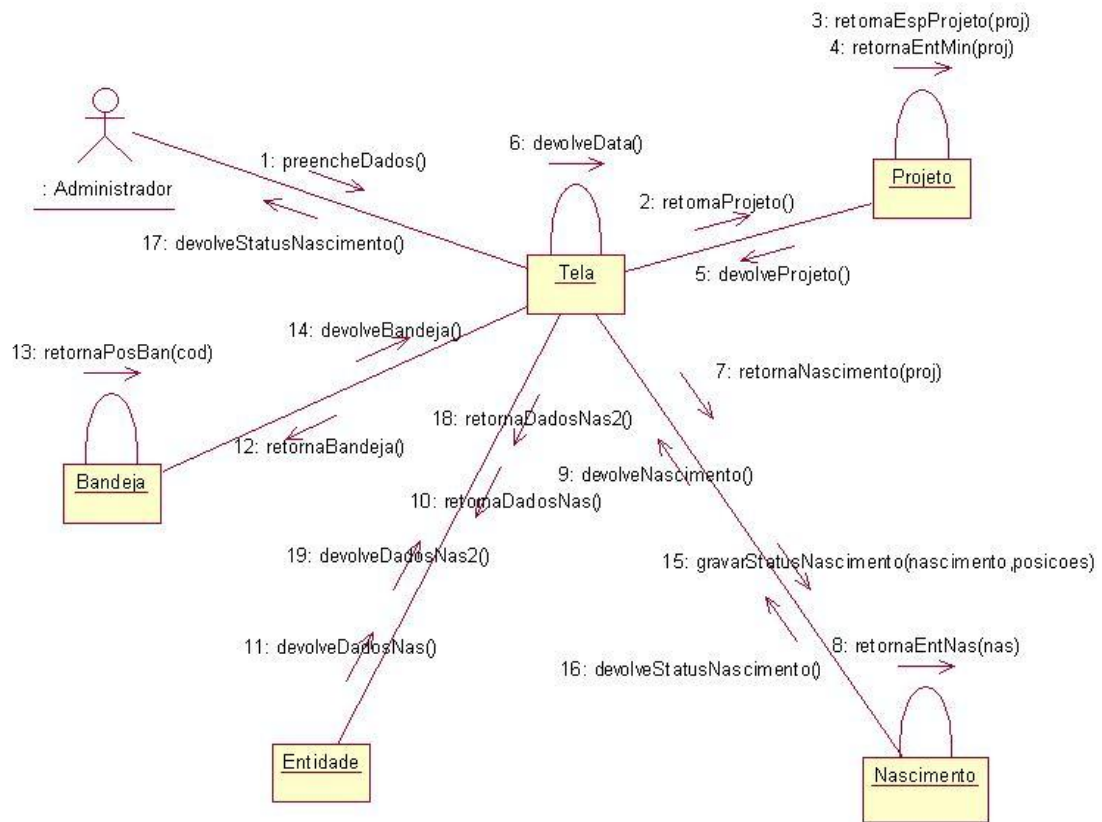
8.3.2.21 Efetuar Login – Novo

FIGURA 81 – EFETUAR LOGIN NOVO



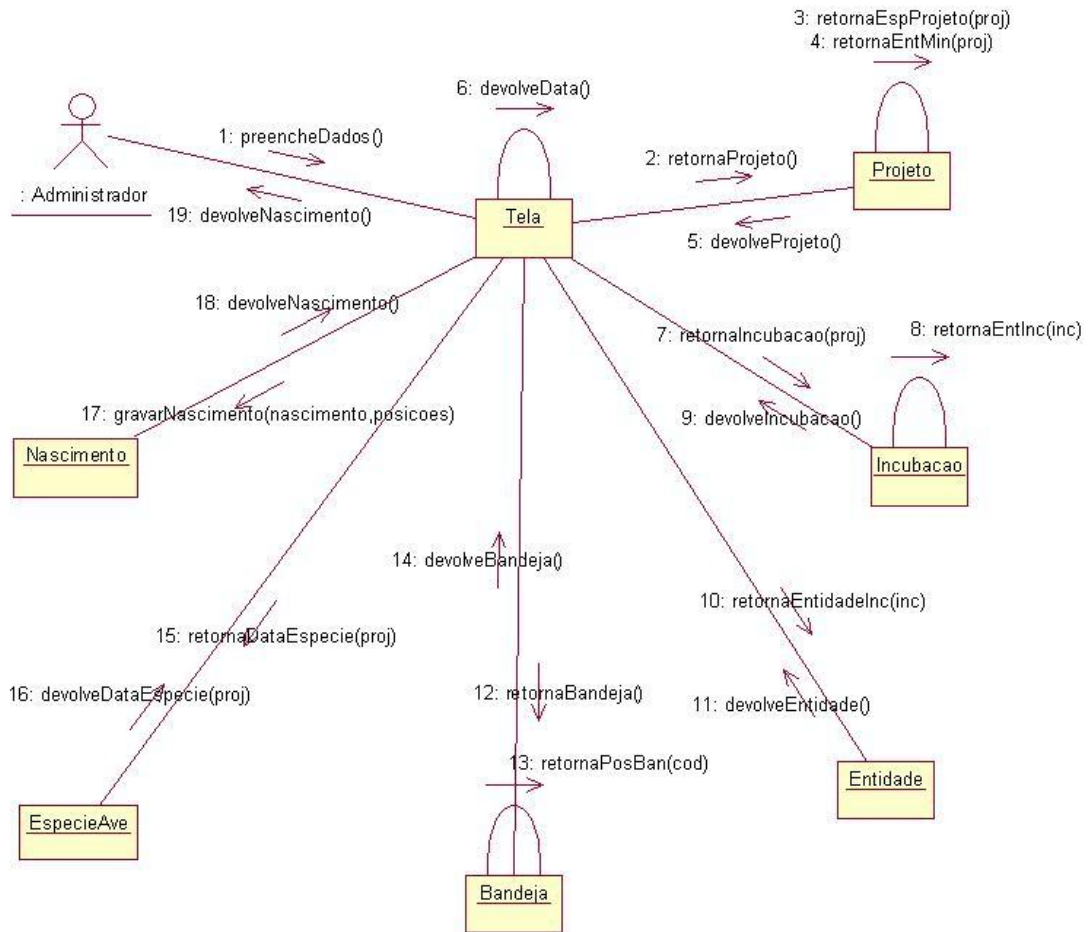
8.3.2.2 Retirar Ovos (Saque) – Novo

FIGURA 82 – RETIRAR OVOS (SAQUE) NOVO



8.3.2.23 Transferir Ovos – Novo

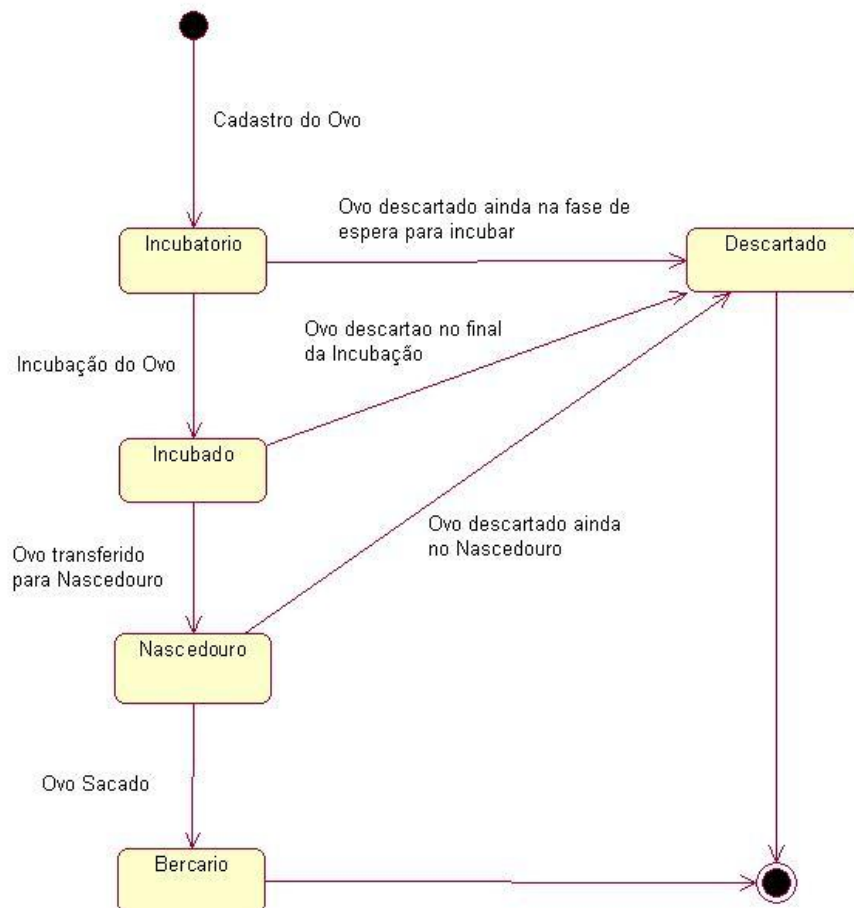
FIGURA 83 – TRANSFERIR OVOS NOVO



8.4 DIAGRAMAS DE ESTADO

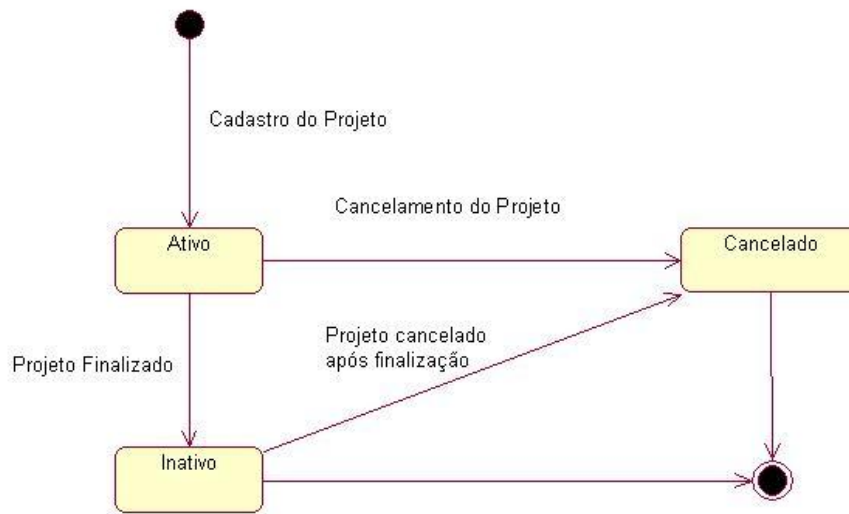
8.4.1 Ovo

FIGURA 84 – D.E. OVO



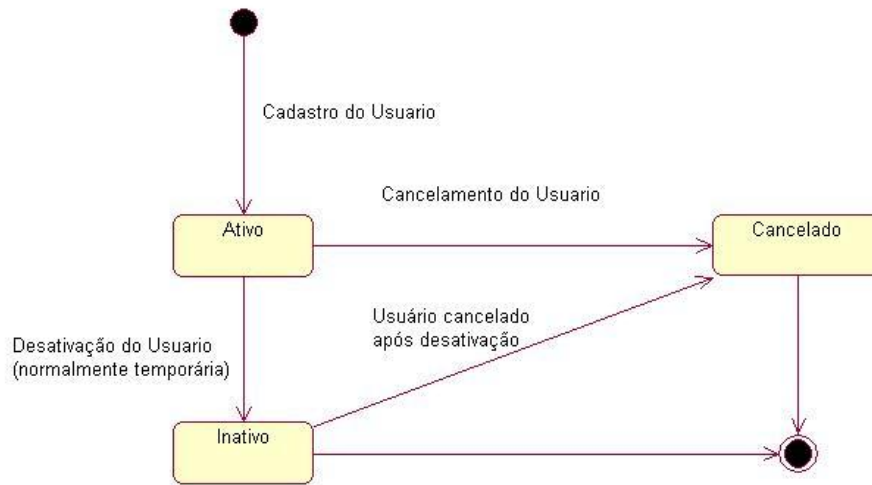
8.4.2 Projeto

FIGURA 85 – D.E. PROJETO



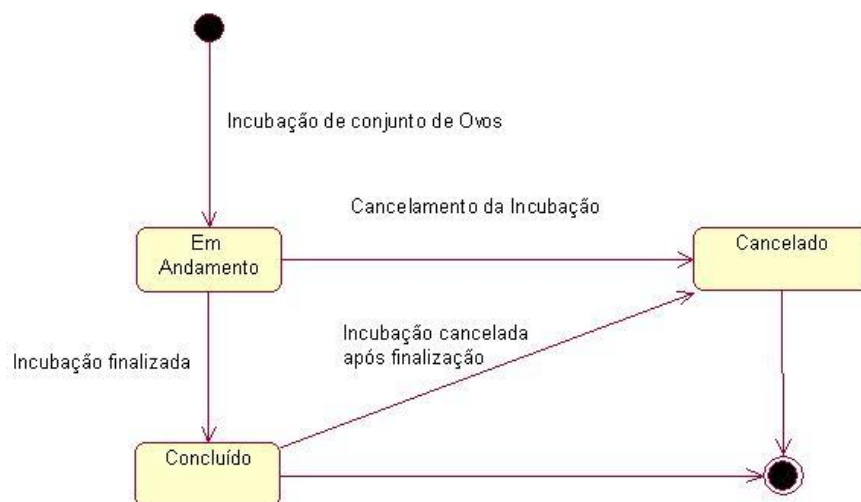
8.4.3 Usuário

FIGURA 86 – D.E. USUÁRIO



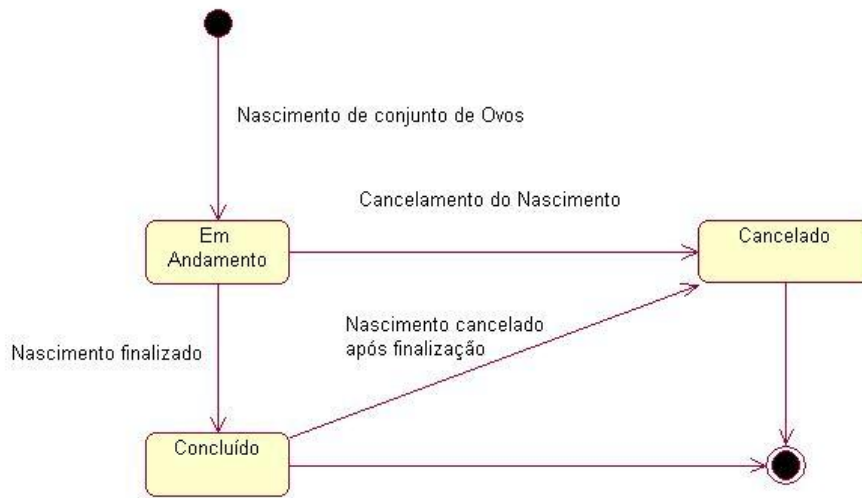
8.4.4 Incubação

FIGURA 87 – D.E. INCUBAÇÃO



8.4.5 Nascimento

FIGURA 88 – D.E. NASCIMENTO



8.6 DICIONÁRIO DE DADOS

TABELA: ANDAR

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
codand	numeric(12,0)	SIM	SIM	Chave da tabela
codpre	numeric(12,0)	SIM	NÃO	Chave estrangeira da tabela Predio

TABELA: AVE

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
codave	numeric(10,0)	SIM	SIM	Chave da tabela
codstatusgeral	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela Statusgeral
codmotmor	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela Motivomorte
codovo	numeric(12,0)	SIM	NÃO	Chave estrangeira da tabela Ovo
codgs	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela Grausangue
codraca	numeric(8,0)	SIM	NÃO	Chave estrangeira da tabela Racaave
codfam	numeric(10,0)	SIM	NÃO	Chave estrangeira da tabela Familia
sexave	char(1)	SIM	NÃO	Sexo da ave
datanasave	date	NÃO	NÃO	Data de Nascimento
datamorave	date	NÃO	NÃO	Data de Morte
motmorave	varchar	NÃO	NÃO	Descrição do motivo da morte

TABELA: BANDEJA

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
codban	numeric(8,0)	SIM	SIM	Chave da tabela
codstatusgeral	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela Statusgeral
tlinban	numeric(2,0)	SIM	NÃO	Número de linhas da bandeja
tcolban	numeric(2,0)	SIM	NÃO	Número de colunas da bandeja
nomeban	varchar(50)	SIM	NÃO	Nome da bandeja

TABELA: BANDEJA

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
codban	numeric(8,0)	SIM	SIM	Chave da tabela
codstatusgeral	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela Statusgeral

TABELA: BLOCO

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
codbloco	numeric(12,0)	SIM	SIM	Chave da tabela
codsala	numeric(12,0)	SIM	NÃO	Chave estrangeira da tabela Sala

TABELA: CELULA

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
---------------	--------------	-------------	-------	-----------

codcel	numeric(12,0)	SIM	SIM	Chave da tabela
codgai	numeric(12,0)	SIM	NÃO	Chave estrangeira da tabela Gaiola

TABELA: CIDADE

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
codcid	numeric(6,0)	SIM	SIM	Chave da tabela
codstatusgeral	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela Statusgeral
coduf	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela Uf
nomecid	varchar(100)	SIM	NÃO	Nome da cidade

TABELA: COROVO

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
codcorovo	numeric(3,0)	SIM	SIM	Chave da tabela
codstatusgeral	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela Statusgeral
codesp	numeric(6,0)	SIM	NÃO	Chave estrangeira da tabela Corovo
nomecorovo	varchar(50)	SIM	NÃO	Nome da cor
descorovo	varchar	NÃO	NÃO	Descrição da cor

TABELA: DESOVO

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
coddesovo	numeric(3,0)	SIM	SIM	Chave da tabela
codstatusgeral	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela Statusgeral
nomedesovo	varchar(50)	SIM	NÃO	Nome da descontaminação
descdesovo	varchar	NÃO	NÃO	Descrição da descontaminação

TABELA: ENTIDADE

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
codent	numeric(12,0)	SIM	SIM	Chave da tabela
codstatusgeral	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela Statusgeral
codtipoent	numeric(1,0)	SIM	NÃO	Chave estrangeira da tabela Tipoentidade
nomeent	varchar(50)	SIM	NÃO	Nome da entidade
descent	varchar	NÃO	NÃO	Descrição da entidade

TABELA: ESPECIEAVE

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
codesp	numeric(6,0)	SIM	SIM	Chave da tabela
codstatusgeral	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela Statusgeral
nomeesp	varchar(50)	SIM	NÃO	Nome da espécie
tempmininc	numeric(6,4)	NÃO	NÃO	Tempo mínimo de incubação dos ovos
tempmaxinc	numeric(6,4)	NÃO	NÃO	Tempo máximo de incubação dos ovos
umimininc	numeric(6,4)	NÃO	NÃO	Umidade mínima de incubação dos ovos

umimaxinc	numeric(6,4)	NÃO	NÃO	Umidade máxima de incubação dos ovos
perinc	numeric(3,0)	SIM	NÃO	Período de incubação
peraviinc	numeric(2,0)	SIM	NÃO	Período de aviso de término da incubação
tempminnas	numeric(6,4)	NÃO	NÃO	Tempo mínimo de nascimento dos ovos
tempmaxnas	numeric(6,4)	NÃO	NÃO	Tempo máximo de nascimento dos ovos
umiminnas	numeric(6,4)	NÃO	NÃO	Umidade mínima d nascimento dos ovos
umimaxnas	numeric(6,4)	NÃO	NÃO	Umidade máxima de nascimento dos ovos
pernas	numeric(3,0)	SIM	NÃO	Período de nascimento
peravinas	numeric(2,0)	SIM	NÃO	Período de aviso de término do nascimento
perrec	numeric(3,0)	SIM	NÃO	Período de recria

TABELA: FAMILIA

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
codfam	numeric(10,0)	SIM	SIM	Chave da tabela
codstatusgeral	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela Statusgeral
codcel	numeric(12,0)	SIM	NÃO	Chave estrangeira da tabela Celula
nomefam	varchar(50)	SIM	NÃO	Nome da família
nummacho	numeric(1,0)	NÃO	NÃO	Número de machos na família
numfemea	numeric(1,0)	NÃO	NÃO	Número de fêmeas na família
datainifam	date	SIM	NÃO	Data de formação da família
datafimfam	date	NÃO	NÃO	Data de término da família

TABELA: FOROVO

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
codforovo	numeric(3,0)	SIM	SIM	Chave da tabela
codstatusgeral	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela Statusgeral
nomeforovo	varchar(50)	SIM	NÃO	Nome do formato do ovo
descforovo	varchar	NÃO	NÃO	Descrição do formato do ovo

TABELA: GAIOLA

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
codgai	numeric(12,0)	SIM	SIM	Chave da tabela
codlinha	numeric(12,0)	SIM	NÃO	Chave estrangeira da tabela Linha

TABELA: GRAUSANGUE

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
codgs	numeric(3,0)	SIM	SIM	Chave da tabela
codstatusgeral	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela Statusgeral
nomegs	varchar(50)	SIM	NÃO	Nome do grau de sangue

descgs	varchar	NÃO	NÃO	Descrição do grau de sangue
--------	---------	-----	-----	-----------------------------

TABELA: GRUPO

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
codgru	numeric(1,0)	SIM	SIM	Chave da tabela
codstatusgeral	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela Statusgeral
nomegru	varchar(100)	SIM	NÃO	Nome do grupo
descgru	varchar	NÃO	NÃO	Descrição do grupo
senhaexp	numeric(3,0)	SIM	NÃO	Período de expiração da senha para usuários associados ao grupo

TABELA: HISTINCUBADORA

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
codhistenc	numeric(20,0)	SIM	SIM	Chave da tabela
codenc	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela Incubadora
datamedenc	timestamp	SIM	NÃO	Data de realização da medição
tempmedenc	numeric(6,4)	NÃO	NÃO	Temperatura medida na incubadora
unimedenc	numeric(6,4)	NÃO	NÃO	Umidade medida na incubadora

TABELA: HISTNASCEDOURO

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
codhistnes	numeric(20,0)	SIM	SIM	Chave da tabela
codnes	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela Nascedouro
datamednes	timestamp	SIM	NÃO	Data de realização da medição
tempmednes	numeric(6,4)	NÃO	NÃO	Temperatura medida no nascedouro
unimednes	numeric(6,4)	NÃO	NÃO	Umidade medida no nascedouro

TABELA: HISTUSUARIO

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
codhistusu	numeric(20,0)	SIM	SIM	Chave da tabela
codusu	numeric(8,0)	SIM	NÃO	Chave estrangeira da tabela Usuario
sessaoini	timestamp	SIM	NÃO	Data de início da última sessão realizada pelo usuário
sessaofim	timestamp	NÃO	NÃO	Data de término da última sessão realizada pelo usuário

TABELA: INCTEMENT

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
codinc	numeric(10,0)	SIM	SIM	Chave estrangeira da tabela Incubacao

codent	numeric(12,0)	SIM	SIM	Chave estrangeira da tabela Entidade
nlinban	numeric(2,0)	SIM	SIM	Chave da tabela, corresponde ao número da linha da bandeja
ncolban	numeric(2,0)	SIM	SIM	Chave da tabela, correspondente ao número da coluna da bandeja
codposbanenc	numeric(10,0)	SIM	SIM	Chave estrangeira da tabela Posbanenc
codstatusgeral	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela Statusgeral
codmotdes	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela Motivodescarte
ovotransf	char(1)	NÃO	NÃO	Indica transferência do ovo

TABELA: INCUBACAO

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
codinc	numeric(10,0)	SIM	SIM	Chave da tabela
codstatusinc	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela StatusIncNas
codproj	numeric(8,0)	SIM	NÃO	Chave estrangeira da tabela Projeto
datainiinc	timestamp	SIM	NÃO	Data de início da incubação
dataprevinc	timestamp	NÃO	NÃO	Data prevista para término da incubação
datafiminc	timestamp	NÃO	NÃO	Data de término da incubação
numovoinii	numeric(3,0)	SIM	NÃO	Número de ovos incubados
numovodesi	numeric(3,0)	NÃO	NÃO	Número de ovos descartados
numovotransf	numeric(3,0)	NÃO	NÃO	Número de ovos transferidos (para nascedouro)
obsstatusinc	varchar	NÃO	NÃO	Observação do status da incubação

TABELA: INCUBADORA

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
codenc	numeric(3,0)	SIM	SIM	Chave da tabela
codstatusgeral	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela Statusgeral
codsala	numeric(12,0)	SIM	NÃO	Chave estrangeira da tabela Sala
nomeenc	varchar(100)	SIM	NÃO	Nome da incubadora
tposenc	numeric(2,0)	NÃO	NÃO	Número total de posições de bandeja disponíveis na incubadora
descenc	varchar	NÃO	NÃO	Descrição da incubadora

TABELA: LINHA

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
codlinha	numeric(12,0)	SIM	SIM	Chave da tabela

codbloco	numeric(12,0)	SIM	NÃO	Chave estrangeira da tabela Bloco
----------	---------------	-----	-----	-----------------------------------

TABELA: MOTIVODESCARTE

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
codmotdes	numeric(3,0)	SIM	SIM	Chave da tabela
codstatusgeral	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela Statusgeral
motdes	varchar(50)	SIM	NÃO	Nome do motivo de descarte
descmotdes	varchar	NÃO	NÃO	Descrição do motivo de descarte

TABELA: MOTIVOMORTE

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
codmotmor	numeric(3,0)	SIM	SIM	Chave da tabela
codstatusgeral	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela Statusgeral
motmor	varchar(50)	SIM	NÃO	Nome do motivo da morte
descmotmor	varchar	NÃO	NÃO	Descrição do motivo a morte

TABELA: NASCEDOURO

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
codnes	numeric(3,0)	SIM	SIM	Chave da tabela
codstatusgeral	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela Statusgeral
codsala	numeric(12,0)	SIM	NÃO	Chave estrangeira da tabela Sala
nomenes	varchar(100)	SIM	NÃO	Nome do nascedouro
tcolnes	numeric(2,0)	NÃO	NÃO	Número total de posições de bandeja disponíveis no nascedouro
descnes	varchar	NÃO	NÃO	Descrição do nascedouro

TABELA: NASCIMENTO

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
codnas	numeric(10,0)	SIM	SIM	Chave da tabela
codinc	numeric(10,0)	SIM	NÃO	Chave estrangeira da tabela Incubadora
codstatusin	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela StatusIncNas
dataininas	timestamp	SIM	NÃO	Data de início do nascimento
dataprevnas	timestamp	NÃO	NÃO	Data prevista de término do nascimento
datafimnas	timestamp	NÃO	NÃO	Data de término do nascimento
numovoinin	numeric(3,0)	SIM	NÃO	Número de ovos inseridos para nascimento
numovodesn	numeric(3,0)	NÃO	NÃO	Número de ovos descartados no nascimento
numovorec	numeric(3,0)	NÃO	NÃO	Número de ovos que geraram recria

obsstatusnas	varchar	NÃO	NÃO	Observação sobre status do nascimento
--------------	---------	-----	-----	---------------------------------------

TABELA: NASTEMENT

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
codent	numeric(12,0)	SIM	SIM	Chave estrangeira da tabela Entidade
codnas	numeric(10,0)	SIM	SIM	Chave estrangeira da tabela Nascimento
nlinban	numeric(2,0)	SIM	SIM	Chave da tabela, corresponde ao número da linha da bandeja
ncolban	numeric(2,0)	SIM	SIM	Chave da tabela, correspondente ao número da coluna da bandeja
codposbannes	numeric(10,0)	SIM	SIM	Chave estrangeira da tabela Posbanenc
codstatusgeral	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela Statusgeral
codmotdes	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela Motivodescarte
ovonasceu	char(1)	NÃO	NÃO	Indica nascimento do ovo

TABELA: OVO

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
codovo	numeric(12,0)	SIM	SIM	Chave da tabela
codbarras	numeric	NÃO	NÃO	Código de barras associado ao ovo
coddesevo	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela Desovo
codstatusovo	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela Statusovo
codforovo	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela Forovo
codcorovo	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela Corovo
codcel	numeric(12,0)	SIM	NÃO	Chave estrangeira da tabela Celula
aquisovo	numeric(12,0)	SIM	NÃO	Chave estrangeira da tabela Propriedade
datacolovo	date	NÃO	NÃO	Data de coleta do vo
aspovo	varchar(50)	NÃO	NÃO	Aspecto do ovo
pesoovo	numeric(6,0)	NÃO	NÃO	Peso do ovo
porovo	numeric(14,0)	NÃO	NÃO	Porosidade do ovo
incubavel	char(1)	NÃO	NÃO	Registra se ovo é incubável

TABELA: POSBANENC

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
codposbanenc	numeric(10,0)	SIM	SIM	Chave da tabela
codstatusgeral	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela Statusgeral
codenc	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela Incubadora

codban	numeric(8,0)	SIM	NÃO	Chave estrangeira da tabela Bandeja
nposenc	numeric(2,0)	SIM	NÃO	Posição da bandeja na incubadora
dataaloenc	timestamp	SIM	NÃO	Data de alocação da bandeja na incubadora
dataretenc	timestamp	NÃO	NÃO	Data de retirada da bandeja da incubadora

TABELA: POSBANNES

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
codposbannes	numeric(10,0)	SIM	SIM	Chave da tabela
codstatusgeral	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela Statusgeral
codnes	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela Nascedouro
codban	numeric(8,0)	SIM	NÃO	Chave estrangeira da tabela Bandeja
nposnes	numeric(2,0)	SIM	NÃO	Posição da bandeja no nascedouro
dataalones	timestamp	SIM	NÃO	Data de alocação da bandeja no nascedouro
dataretnes	timestamp	NÃO	NÃO	Data de retirada da bandeja do nascedouro

TABELA: PREDIO

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
codpre	numeric(12,0)	SIM	SIM	Chave da tabela
codprop	numeric(12,0)	SIM	NÃO	Chave estrangeira da tabela Propriedade

TABELA: PROJETO

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
codproj	numeric(8,0)	SIM	SIM	Chave da tabela
codesp	numeric(6,0)	SIM	NÃO	Chave estrangeira da tabela Especieave
codtipoent	numeric(1,0)	SIM	NÃO	Chave estrangeira da tabela Tipoentidade
codstatusproj	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela Statusprojeto
siglaproj	char(14)	SIM	NÃO	Sigla do projeto
nomeproj	varchar(100)	SIM	NÃO	Nome do projeto
datainiproj	date	SIM	NÃO	data de início do projeto
dataprevproj	date	NÃO	NÃO	Data prevista de término do projeto
datafimproj	date	NÃO	NÃO	Data de término do projeto
descproj	varchar	NÃO	NÃO	Descrição do projeto
numinc	numeric(3,0)	NÃO	NÃO	Número de incubações do projeto

TABELA: PROJTEMUSU

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
codproj	numeric(8,0)	SIM	SIM	Chave estrangeira da tabela Projeto

codusu	numeric(8,0)	SIM	SIM	Chave estrangeira da tabela Usuario
codstatusgeral	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela Statusgeral
aviso	char(1)	SIM	NÃO	Registra se usuário deve receber aviso dentro do projeto
datainc	date	SIM	NÃO	Data de início dos trabalhos do usuário no dado projeto
datater	date	NÃO	NÃO	data de término dos trabalhos do usuário no dado projeto
motalt	varchar	NÃO	NÃO	Motivo da alteração

TABELA: PROPRIIDADE

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
codprop	numeric(12,0)	SIM	SIM	Chave da tabela
codcid	numeric(6,0)	SIM	NÃO	Chave estrangeira da tabela Cidade
endprop	varchar	NÃO	NÃO	Endereço da propriedade

TABELA: RACAAVE

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
codraca	numeric(8,0)	SIM	SIM	Chave da tabela
codstatusgeral	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela Statusgeral
codesp	numeric(6,0)	SIM	NÃO	Chave estrangeira da tabela Especieave
nomeraca	varchar(100)	SIM	NÃO	Nome da raça
descraca	varchar	NÃO	NÃO	Descrição da raça

TABELA: REPETICAO

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
codrep	numeric(12,0)	SIM	SIM	Chave da tabela
codstatusgeral	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela Statusgeral
codent	numeric(12,0)	SIM	NÃO	Chave estrangeira da tabela Entidade
codtrat	numeric(10,0)	SIM	NÃO	Chave estrangeira da tabela tratamento
codtipomed	numeric(8,0)	SIM	NÃO	Chave estrangeira da tabela Tipounimed
datamedrep	timestamp	NÃO	NÃO	Data da medida da repetição
valmedrep	numeric(12,8)	NÃO	NÃO	Valor da medida da repetição
codunimed	numeric(10,0)	SIM	NÃO	Chave estrangeira da tabela Unidamedida

TABELA: SALA

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
codsala	numeric(12,0)	SIM	SIM	Chave da tabela
codand	numeric(12,0)	SIM	NÃO	Chave estrangeira da tabela Andar

TABELA: STATUSGERAL

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
codstatusgeral	numeric(3,0)	SIM	SIM	Chave da tabela
nomestatusgeral	varchar(50)	NÃO	NÃO	Nome do status
descstatusgeral	varchar	NÃO	NÃO	Descrição do status

TABELA: STATUSINCNAS

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
codstatusin	numeric(3,0)	SIM	SIM	Chave da tabela
codstatusgeral	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela Statusgeral
nomestatusin	varchar(50)	SIM	NÃO	Nome do status
descstatusin	varchar	NÃO	NÃO	Descrição do status

TABELA: STATUSOVO

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
codstatuovo	numeric(3,0)	SIM	SIM	Chave da tabela
codstatusgeral	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela Statusgeral
nomestatusovo	varchar(50)	SIM	NÃO	Nome do status
descstatusovo	varchar	NÃO	NÃO	Descrição do status

TABELA: STATUSPROJETO

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
codstatuproj	numeric(3,0)	SIM	SIM	Chave da tabela
codstatusgeral	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela Statusgeral
nomestatusproj	varchar(50)	SIM	NÃO	Nome do status
descstatusproj	varchar	NÃO	NÃO	Descrição do status

TABELA: STATUSUSUARIO

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
codstatuus	numeric(3,0)	SIM	SIM	Chave da tabela
codstatusgeral	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela Statusgeral
nomestatususu	varchar(50)	SIM	NÃO	Nome do status
descstatususu	varchar	NÃO	NÃO	Descrição do status

TABELA: TIPOENTIDADE

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
codtipoent	numeric(1,0)	SIM	SIM	Chave da tabela
codstatusgeral	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela Statusgeral
nometipoent	varchar(50)	SIM	NÃO	Nome do tipo de entidade
desctipoent	varchar	NÃO	NÃO	Descrição do tipo de entidade
tipoentmin	char(1)	SIM	NÃO	Registra se tipo de entidade pode ser utilizada como entidade mínima em um dado projeto

TABELA: TIPOTRATAMENTO

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
---------------	--------------	-------------	-------	-----------

codtipotrat	numeric(8,0)	SIM	SIM	Chave da tabela
codstatusgeral	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela Statusgeral
nometipotrat	varchar(50)	SIM	NÃO	Nome do tipo de tratamento
desctipotrat	varchar	NÃO	NÃO	Descrição do tipo de tratamento

TABELA: TIPOTRATTEMUNIMED

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
codtipotrat	numeric(8,0)	SIM	SIM	Chave estrangeira da tabela Tipotratamento
codtipomed	numeric(8,0)	SIM	SIM	Chave estrangeira da tabela Tipounimed
codstatusgeral	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela Statusgeral

TABELA: TIPOUNIMED

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
codtipomed	numeric(8,0)	SIM	SIM	Chave da tabela
codstatusgeral	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela Statusgeral
nometipomed	varchar(50)	SIM	NÃO	Nome do tipo de medida
desctipomed	varchar	NÃO	NÃO	Descrição do tipo de medida

TABELA: TRATAMENTO

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
codtrat	numeric(10,0)	SIM	SIM	Chave da tabela
codstatusgeral	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela Statusgeral
codproj	numeric(8,0)	SIM	NÃO	Chave estrangeira da tabela Projeto
nometrat	varcahr(255)	SIM	NÃO	Nome do tratamento
desctrat	varchar	NÃO	NÃO	Descrição do tratamento
numrep	numeric(3,0)	NÃO	NÃO	Número de repetições do tratamento

TABELA: TRATTEM TIPOTRAT

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
codtrat	numeric(10,0)	SIM	SIM	Chave estrangeira da tabela Tratamento
codtipotrat	numeric(8,0)	SIM	NÃO	Chave estrangeira da tabela Tipotratamento
codstatusgeral	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela Statusgeral

TABELA: UF

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
coduf	numeric(3,0)	SIM	SIM	Chave da tabela
codstatusgeral	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela Statusgeral
siglauf	char(2)	SIM	NÃO	Sigla da UF
nomeuf	varchar(50)	SIM	NÃO	Nome da UF

TABELA: UNIDADEMEDIDA

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
codunimed	numeric(10,0)	SIM	SIM	Chave da tabela
codstatusgeral	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela Statusgeral
codtipomed	numeric(8,0)	SIM	NÃO	Chave estrangeira da tabela Tipounimed
nomeunimed	varchar(100)	SIM	NÃO	Nome da unidade de medida
descunimed	varchar	NÃO	NÃO	Descrição da unidade de medida

TABELA: USUARIO

Nome do Campo	Tipo/Tamanho	Obrigatório	Chave	Descrição
codusu	numeric(8,0)	SIM	SIM	Chave da tabela
codstatususu	numeric(3,0)	SIM	NÃO	Chave estrangeira da tabela Statususuario
codgru	numeric(1,0)	SIM	NÃO	Chave estrangeira da tabela Grupo
matrusu	char(10)	SIM	NÃO	Matrícula do usuário
nomeusu	varchar(100)	SIM	NÃO	Nome do usuário
sencrt	varchar	SIM	NÃO	Senha do usuário
datainc	timestamp	SIM	NÃO	Data de cadastro do usuário
telres	numeric(10,0)	NÃO	NÃO	Telefone residencial
telcom	numeric(10,0)	NÃO	NÃO	Telefone comercial
telcel	numeric(10,0)	NÃO	NÃO	Telefona celular
email	varchar(50)	SIM	NÃO	E-mail
ultaltsen	timestamp	NÃO	NÃO	Data da última alteração de senha
ultlogon	timestamp	NÃO	NÃO	Data do último logon do usuário

9 CONCLUSÃO

Com a criação do SISINC e os testes iniciais realizados com o cliente foram atendidos os objetivos principais do sistema, que correspondem a:

- Armazenamento das principais características do ambiente do aviário;
- Armazenar informações associadas aos projetos experimentais desenvolvidos no laboratório (aviário);
- Armazenar principais características dos equipamentos do laboratório;
- Armazenar informações dos ovos, nascidos e não-nascidos (descartados);
- Armazenar informações das incubações e nascimentos realizados.

O projeto apresenta possibilidade de melhorias para que sejam contempladas todas as necessidades dos usuários, dentre as principais oportunidades de melhorias podemos citar:

- Medições automáticas (termohigrômetro);
- Aviso de fim de incubação e nascimento;
- Aviso de falta de energia;
- Embriodiagnóstico;
- Tratamentos e repetições;
- Identificação por cores para entidades nas bandejas das telas de Incubação, Nascimento, Saque e Embriodiagnóstico;
- Permitir criação de famílias para rastreamento da origem do ovo.

Além dos ganhos disponibilizados ao cliente com a criação do sistema o projeto permitiu, por parte dos alunos que dele participaram, a consolidação dos estudos realizados durante o curso de Tecnologia de Informática da UFPR.

10 REFERÊNCIAS

Artigos do PostgreSQL, disponíveis no site:

<http://www.postgresql.org.br>.

PMBOK Guide, A Guide to the Project Management Body of Knowledge, Third edition – PMI – Project Management Institute.

VARGAS, Ricardo Viana, Gerenciamento de projetos: estabelecendo diferenciais competitivos / Ricardo Viana Vargas; prefácio de Reeve Harold R. – 6.ed. – Rio de Janeiro: Brasoft, 2005.

CLELAND, David I., Gerência de Projetos / David I. Cleland, Lewis R. Ireland; revisão técnica de Carlos A. C. Salles Jr. – Rio de Janeiro: Reichmann & Afonso, 2002.

BOCH, Grady, UML, guia do usuário / Grady Booch, James Rumbaugh, Ivar Jacobson; tradução de Fábio Freitas da Silva – Rio de Janeiro: Campus, 2000.

BLASCHEK, José Roberto, Análise orientada a objetos com UML / Prof. José Roberto Blaschek e Prof. José A. Rodrigues Neto – Rio de Janeiro: PUC-RJ, 2004.

SOUZA, Marco Aurélio de, SQL, PL/SQL, SQL* Plus: manual de referência completo e objetivo – Rio de Janeiro: Editora Ciência Moderna Ltda., 2004.

DEITEL, H. M., Java, como programar / H. M. Deitel e P. J. Deitel; tradução de Carlos Arthur Lang Lisboa – 4.ed. – Posrto Alegre: Bookman, 2003.

BOMFIM JÚNIOR, Francisco Tarcizo, JSP: A Tecnolgoia Java na Internet – São Paulo: Érica, 2002.

LOZANO, Fernando, Tomcat Fundamental: Arquitetura, instalação e configuração – Revista Java Magazine – Edição 6.

11 GLOSSÁRIO

11.1 TERMOS UTILIZADOS NO BANCO DE DADOS

- Bloco: conjunto de todas as linhas sobrepostas.
- Célula: unidade onde as famílias são constituídas em um período de tempo.
- Família: grupo de animais formados por machos e fêmeas.
- Gaiola: conjunto de cinco células posicionadas em linha horizontal.
- Incubadora: equipamento que realiza o aquecimento dos ovos.
- Linha: conjunto de todas as gaiolas da linha horizontal.
- Nasedouro: equipamento onde os ovos permanecem na fase final da incubação.

11.2 TERMOS UTILIZADOS NA MODELAGEM

- Sacar Ovos: retirada dos ovos do nasedouro.
- Transferir Ovos: retirada dos ovos da incubadora para o nasedouro.

12 CÓDIGO FONTE

12.1 BASE DE DADOS

12.1.1 Script de Criação

```
CREATE TABLE StatusGeral (
  CodStatusGeral NUMERIC(3) NOT NULL DEFAULT nextval('statusgeral_codstatusgeral'),
  NomeStatusGeral VARCHAR(50) NULL,
  DescStatusGeral VARCHAR NULL,
  PRIMARY KEY(CodStatusGeral)
);

INSERT INTO StatusGeral (CodStatusGeral, NomeStatusGeral, DescStatusGeral)
VALUES (DEFAULT, 'Ativo', '');
INSERT INTO StatusGeral (CodStatusGeral, NomeStatusGeral, DescStatusGeral)
VALUES (DEFAULT, 'Inativo', '');
INSERT INTO StatusGeral (CodStatusGeral, NomeStatusGeral, DescStatusGeral)
VALUES (DEFAULT, 'Cancelado', '');

-----
-- Grupos de acesso e definidos para usuários.
-----

CREATE TABLE Grupo (
  CodGru NUMERIC(1) NOT NULL DEFAULT nextval('grupo_codgru_seq'),
  CodStatusGeral NUMERIC(3) NOT NULL,
  NomeGru VARCHAR(100) NOT NULL,
  DescGru VARCHAR NULL,
  SenhaExp NUMERIC(3) NOT NULL,
  PRIMARY KEY(CodGru),
  FOREIGN KEY(CodStatusGeral)
  REFERENCES StatusGeral(CodStatusGeral)
  ON DELETE RESTRICT
  ON UPDATE CASCADE
);

INSERT INTO Grupo (CodGru, NomeGru, DescGru, SenhaExp, CodStatusGeral)
VALUES (DEFAULT, 'Administrador', 'Permite inclusão, alteração e cancelamento de usuários,
usuários em grupos, projetos, entidades físicas, espécies e raças de ave.', 999, 1);
INSERT INTO Grupo (CodGru, NomeGru, DescGru, SenhaExp, CodStatusGeral)
VALUES (DEFAULT, 'Operador', 'Permite inclusão, alteração e cancelamento de entidades que
permitem incubação, incubação e saque de entidades.', 30, 1);
```

```

INSERT INTO Grupo (CodGru, NomeGru, DescGru, SenhaExp, CodStatusGeral)
VALUES (DEFAULT, 'Consultor', 'Permite visualizar status de incubação, gerar relatórios e
notificações de incubações.', 30, 1);

```

```

-----
-- Grau de aplicação da raça de maior influência.
-----

```

```

CREATE TABLE GrauSangue (
  CodGS NUMERIC(3) NOT NULL DEFAULT nextval('grausangue_codgs'),
  CodStatusGeral NUMERIC(3) NOT NULL,
  NomeGS VARCHAR(50) NOT NULL,
  DescGS VARCHAR NULL,
  PRIMARY KEY (CodGS),
  FOREIGN KEY (CodStatusGeral)
  REFERENCES StatusGeral (CodStatusGeral)
  ON DELETE RESTRICT
  ON UPDATE CASCADE
);

```

```

INSERT INTO GrauSangue (CodGS, NomeGS, DescGS, CodStatusGeral) VALUES (DEFAULT, 'Puro', '', 1);
INSERT INTO GrauSangue (CodGS, NomeGS, DescGS, CodStatusGeral) VALUES (DEFAULT, '1/2', '', 1);
INSERT INTO GrauSangue (CodGS, NomeGS, DescGS, CodStatusGeral) VALUES (DEFAULT, '3/4', '', 1);
INSERT INTO GrauSangue (CodGS, NomeGS, DescGS, CodStatusGeral) VALUES (DEFAULT, '7/8', '', 1);
INSERT INTO GrauSangue (CodGS, NomeGS, DescGS, CodStatusGeral)
VALUES (DEFAULT, 'Desconhecido', '', 1);

```

```

-----
-- Tipos de momento atual para projetos.
-----

```

```

CREATE TABLE StatusProjeto (
  CodStatusProj NUMERIC(3) NOT NULL DEFAULT nextval('statusprojeto_codstatusproj'),
  CodStatusGeral NUMERIC(3) NOT NULL,
  NomeStatusProj VARCHAR(50) NOT NULL,
  DescStatusProj VARCHAR NULL,
  PRIMARY KEY (CodStatusProj),
  FOREIGN KEY (CodStatusGeral)
  REFERENCES StatusGeral (CodStatusGeral)
  ON DELETE RESTRICT
  ON UPDATE CASCADE
);

```

```

INSERT INTO StatusProjeto (CodStatusProj, NomeStatusProj, DescStatusProj, CodStatusGeral)
VALUES (DEFAULT, 'Ativo', '', 1);
INSERT INTO StatusProjeto (CodStatusProj, NomeStatusProj, DescStatusProj, CodStatusGeral)
VALUES (DEFAULT, 'Inativo', '', 1);
INSERT INTO StatusProjeto (CodStatusProj, NomeStatusProj, DescStatusProj, CodStatusGeral)
VALUES (DEFAULT, 'Cancelado', '', 1);

```

```
-----  
-- Motivos de descarte de posição de entidade.  
-----
```

```
CREATE TABLE MotivoDescarte (  
  CodMotDes NUMERIC(3) NOT NULL DEFAULT nextval('motivodescarte_codmotdes'),  
  CodStatusGeral NUMERIC(3) NOT NULL,  
  MotDes VARCHAR(50) NOT NULL,  
  DescMotDes VARCHAR NULL,  
  PRIMARY KEY(CodMotDes),  
  FOREIGN KEY(CodStatusGeral)  
    REFERENCES StatusGeral(CodStatusGeral)  
    ON DELETE RESTRICT  
    ON UPDATE CASCADE  
);
```

```
CREATE TABLE MotivoMorte (  
  CodMotMor NUMERIC(3) NOT NULL DEFAULT nextval('motivomorte_codmotmorte'),  
  CodStatusGeral NUMERIC(3) NOT NULL,  
  MotMor VARCHAR(50) NOT NULL,  
  DescMotMor VARCHAR NULL,  
  PRIMARY KEY(CodMotMor),  
  FOREIGN KEY(CodStatusGeral)  
    REFERENCES StatusGeral(CodStatusGeral)  
    ON DELETE RESTRICT  
    ON UPDATE CASCADE  
);
```

```
-----  
-- Tipos de momento atual para incubações e nascimentos.  
-----
```

```
CREATE TABLE StatusIncNas (  
  CodStatusIN NUMERIC(3) NOT NULL DEFAULT nextval('statusincnas_codstatusin'),  
  CodStatusGeral NUMERIC(3) NOT NULL,  
  NomeStatusIN VARCHAR(50) NOT NULL,  
  DescStatusIN VARCHAR NULL,  
  PRIMARY KEY(CodStatusIN),  
  FOREIGN KEY(CodStatusGeral)  
    REFERENCES StatusGeral(CodStatusGeral)  
    ON DELETE RESTRICT  
    ON UPDATE CASCADE  
);
```

```
INSERT INTO StatusIncNas (CodStatusIN, NomeStatusIN, DescStatusIN, CodStatusGeral)  
VALUES (DEFAULT, 'Em Andamento', '', 1);  
INSERT INTO StatusIncNas (CodStatusIN, NomeStatusIN, DescStatusIN, CodStatusGeral)  
VALUES (DEFAULT, 'Concluido', '', 1);
```

```

INSERT INTO StatusIncNas (CodStatusIN, NomeStatusIN, DescStatusIN, CodStatusGeral)
VALUES (DEFAULT, 'Cancelado', '', 1);

```

```

-----
-- Tipos de momento atual para ovos.
-----

```

```

CREATE TABLE StatusOvo (
  CodStatusOvo NUMERIC(3) NOT NULL DEFAULT nextval('statusovo_codstatusovo'),
  CodStatusGeral NUMERIC(3) NOT NULL,
  NomeStatusOvo VARCHAR(50) NOT NULL,
  DescStatusOvo VARCHAR NULL,
  PRIMARY KEY(CodStatusOvo),
  FOREIGN KEY(CodStatusGeral)
  REFERENCES StatusGeral(CodStatusGeral)
  ON DELETE RESTRICT
  ON UPDATE CASCADE
);

```

```

-----
-- Tipos de formato de ovos.
-----

```

```

CREATE TABLE ForOvo (
  CodForOvo NUMERIC(3) NOT NULL DEFAULT nextval('forovo_codforovo'),
  CodStatusGeral NUMERIC(3) NOT NULL,
  NomeForOvo VARCHAR(50) NOT NULL,
  DescForOvo VARCHAR NULL,
  PRIMARY KEY(CodForOvo),
  FOREIGN KEY(CodStatusGeral)
  REFERENCES StatusGeral(CodStatusGeral)
  ON DELETE RESTRICT
  ON UPDATE CASCADE
);

```

```

-----
-- Tipos de medida para unidade.
-----

```

```

CREATE TABLE TipoUniMed (
  CodTipoMed NUMERIC(8) NOT NULL DEFAULT nextval('tipounimed_codtipomed'),
  CodStatusGeral NUMERIC(3) NOT NULL,
  NomeTipoMed VARCHAR(50) NOT NULL,
  DescTipoMed VARCHAR NULL,
  PRIMARY KEY(CodTipoMed),
  FOREIGN KEY(CodStatusGeral)
  REFERENCES StatusGeral(CodStatusGeral)
  ON DELETE RESTRICT
  ON UPDATE CASCADE
);

```

```
-- -----  
-- Tipos de tratamento para pesquisa.  
-- -----
```

```
CREATE TABLE TipoTratamento (  
    CodTipoTrat NUMERIC(8) NOT NULL DEFAULT nextval('tipotrtratamento_codtipotrat'),  
    CodStatusGeral NUMERIC(3) NOT NULL,  
    NomeTipoTrat VARCHAR(50) NOT NULL,  
    DescTipoTrat VARCHAR NULL,  
    PRIMARY KEY(CodTipoTrat),  
    FOREIGN KEY(CodStatusGeral)  
        REFERENCES StatusGeral(CodStatusGeral)  
            ON DELETE RESTRICT  
            ON UPDATE CASCADE  
);
```

```
-- -----  
-- Bandejas e total de linhas e colunas.  
-- -----
```

```
CREATE TABLE Bandeja (  
    CodBan NUMERIC(8) NOT NULL DEFAULT nextval('bandeja_codban'),  
    CodStatusGeral NUMERIC(3) NOT NULL,  
    TLinBan NUMERIC(2) NOT NULL,  
    TColBan NUMERIC(2) NOT NULL,  
    NomeBan VARCHAR(50) NOT NULL,  
    PRIMARY KEY(CodBan),  
    FOREIGN KEY(CodStatusGeral)  
        REFERENCES StatusGeral(CodStatusGeral)  
            ON DELETE RESTRICT  
            ON UPDATE CASCADE  
);
```

```
-- -----  
-- UFs cadastrados no sistema.  
-- -----
```

```
CREATE TABLE UF (  
    CodUF NUMERIC(3) NOT NULL DEFAULT nextval('uf_coduf'),  
    CodStatusGeral NUMERIC(3) NOT NULL,  
    SiglaUF CHAR(2) NOT NULL,  
    NomeUF VARCHAR(50) NOT NULL,  
    PRIMARY KEY(CodUF),  
    FOREIGN KEY(CodStatusGeral)  
        REFERENCES StatusGeral(CodStatusGeral)  
            ON DELETE RESTRICT  
            ON UPDATE CASCADE  
);
```

```
INSERT INTO UF (CodUF, SiglaUF, NomeUF, CodStatusGeral) VALUES(DEFAULT, 'AC', 'Acre', 1);
```

```

INSERT INTO UF (CodUF, SiglaUF, NomeUF, CodStatusGeral) VALUES(DEFAULT, 'AL', 'Alagoas', 1);
INSERT INTO UF (CodUF, SiglaUF, NomeUF, CodStatusGeral) VALUES(DEFAULT, 'AM', 'Amazonas', 1);
INSERT INTO UF (CodUF, SiglaUF, NomeUF, CodStatusGeral) VALUES(DEFAULT, 'AP', 'Amapá', 1);
INSERT INTO UF (CodUF, SiglaUF, NomeUF, CodStatusGeral) VALUES(DEFAULT, 'BH', 'Bahia', 1);
INSERT INTO UF (CodUF, SiglaUF, NomeUF, CodStatusGeral) VALUES(DEFAULT, 'CE', 'Ceará', 1);
INSERT INTO UF (CodUF, SiglaUF, NomeUF, CodStatusGeral) VALUES(DEFAULT, 'DF', 'Distrito
Federal', 1);
INSERT INTO UF (CodUF, SiglaUF, NomeUF, CodStatusGeral) VALUES(DEFAULT, 'ES', 'Espírito
Santo', 1);
INSERT INTO UF (CodUF, SiglaUF, NomeUF, CodStatusGeral) VALUES(DEFAULT, 'GO', 'Goiás', 1);
INSERT INTO UF (CodUF, SiglaUF, NomeUF, CodStatusGeral) VALUES(DEFAULT, 'MA', 'Maranhão', 1);
INSERT INTO UF (CodUF, SiglaUF, NomeUF, CodStatusGeral) VALUES(DEFAULT, 'MG', 'Minas Gerais', 1);
INSERT INTO UF (CodUF, SiglaUF, NomeUF, CodStatusGeral) VALUES(DEFAULT, 'MS', 'Mato Grosso do
Sul', 1);
INSERT INTO UF (CodUF, SiglaUF, NomeUF, CodStatusGeral) VALUES(DEFAULT, 'MT', 'Mato Grosso', 1);
INSERT INTO UF (CodUF, SiglaUF, NomeUF, CodStatusGeral) VALUES(DEFAULT, 'PA', 'Pará', 1);
INSERT INTO UF (CodUF, SiglaUF, NomeUF, CodStatusGeral) VALUES(DEFAULT, 'PB', 'Paraíba', 1);
INSERT INTO UF (CodUF, SiglaUF, NomeUF, CodStatusGeral) VALUES(DEFAULT, 'PE', 'Pernambuco', 1);
INSERT INTO UF (CodUF, SiglaUF, NomeUF, CodStatusGeral) VALUES(DEFAULT, 'PI', 'Piauí', 1);
INSERT INTO UF (CodUF, SiglaUF, NomeUF, CodStatusGeral) VALUES(DEFAULT, 'PR', 'Paraná', 1);
INSERT INTO UF (CodUF, SiglaUF, NomeUF, CodStatusGeral) VALUES(DEFAULT, 'RJ', 'Rio de
Janeiro', 1);
INSERT INTO UF (CodUF, SiglaUF, NomeUF, CodStatusGeral) VALUES(DEFAULT, 'RN', 'Rio Grande do
Norte', 1);
INSERT INTO UF (CodUF, SiglaUF, NomeUF, CodStatusGeral) VALUES(DEFAULT, 'RO', 'Rondônia', 1);
INSERT INTO UF (CodUF, SiglaUF, NomeUF, CodStatusGeral) VALUES(DEFAULT, 'RR', 'Roraima', 1);
INSERT INTO UF (CodUF, SiglaUF, NomeUF, CodStatusGeral) VALUES(DEFAULT, 'RS', 'Rio Grande do
Sul', 1);
INSERT INTO UF (CodUF, SiglaUF, NomeUF, CodStatusGeral) VALUES(DEFAULT, 'SC', 'Santa
Catarina', 1);
INSERT INTO UF (CodUF, SiglaUF, NomeUF, CodStatusGeral) VALUES(DEFAULT, 'SP', 'São Paulo', 1);
INSERT INTO UF (CodUF, SiglaUF, NomeUF, CodStatusGeral) VALUES(DEFAULT, 'SE', 'Sergipe', 1);
INSERT INTO UF (CodUF, SiglaUF, NomeUF, CodStatusGeral) VALUES(DEFAULT, 'TO', 'Tocantins', 1);

```

```

-----
-- Espécie de ave e suas características.
-----

```

```

CREATE TABLE EspécieAve (
    CodEsp NUMERIC(6) NOT NULL DEFAULT nextval('especieave_codesp'),
    CodStatusGeral NUMERIC(3) NOT NULL,
    NomeEsp VARCHAR(50) NOT NULL,
    TempMinInc NUMERIC(6,4) NULL,
    TempMaxInc NUMERIC(6,4) NULL,
    UmiMinInc NUMERIC(6,4) NULL,
    UmiMaxInc NUMERIC(6,4) NULL,
    PerInc NUMERIC(3) NOT NULL,
    PerAviInc NUMERIC(2) NOT NULL,
    TempMinNas NUMERIC(6,4) NULL,
    TempMaxNas NUMERIC(6,4) NULL,

```

```

UmiMinNas NUMERIC(6,4) NULL,
UmiMaxNas NUMERIC(6,4) NULL,
PerNas NUMERIC(3) NOT NULL,
PerAviNas NUMERIC(2) NOT NULL,
PerRec NUMERIC(3) NOT NULL,
PRIMARY KEY(CodEsp),
FOREIGN KEY(CodStatusGeral)
REFERENCES StatusGeral(CodStatusGeral)
ON DELETE RESTRICT
ON UPDATE CASCADE
);

-----
-- Tipos de momento atual para usuários.
-----

CREATE TABLE StatusUsuario (
CodStatusUsu NUMERIC(3) NOT NULL DEFAULT nextval('statususuario_codstatususu_seq'),
CodStatusGeral NUMERIC(3) NOT NULL,
NomeStatusUsu VARCHAR(50) NOT NULL,
DescStatusUsu VARCHAR NULL,
PRIMARY KEY(CodStatusUsu),
FOREIGN KEY(CodStatusGeral)
REFERENCES StatusGeral(CodStatusGeral)
ON DELETE RESTRICT
ON UPDATE CASCADE
);

INSERT INTO StatusUsuario (CodStatusUsu, NomeStatusUsu, DescStatusUsu, CodStatusGeral)
VALUES (DEFAULT, 'Ativo', '', 1);
INSERT INTO StatusUsuario (CodStatusUsu, NomeStatusUsu, DescStatusUsu, CodStatusGeral)
VALUES (DEFAULT, 'Inativo', '', 1);
INSERT INTO StatusUsuario (CodStatusUsu, NomeStatusUsu, DescStatusUsu, CodStatusGeral)
VALUES (DEFAULT, 'Cancelado', '', 1);

-----
-- Tipo de desinfecção do ovo.
-----

CREATE TABLE DesOvo (
CodDesOvo NUMERIC(3) NOT NULL DEFAULT nextval('desovo_coddesovo'),
CodStatusGeral NUMERIC(3) NOT NULL,
NomeDesOvo VARCHAR(50) NOT NULL,
DescDesOvo VARCHAR NULL,
PRIMARY KEY(CodDesOvo),
FOREIGN KEY(CodStatusGeral)
REFERENCES StatusGeral(CodStatusGeral)
ON DELETE RESTRICT
ON UPDATE CASCADE
);

```

```

INSERT INTO DesOvo (CodDesOvo, NomeDesOvo, DescDesOvo, CodStatusGeral)
VALUES (DEFAULT, 'Nenhuma', '', 1);
INSERT INTO DesOvo (CodDesOvo, NomeDesOvo, DescDesOvo, CodStatusGeral)
VALUES (DEFAULT, 'Iodo', '', 1);
INSERT INTO DesOvo (CodDesOvo, NomeDesOvo, DescDesOvo, CodStatusGeral)
VALUES (DEFAULT, 'Fumigado', '', 1);
INSERT INTO DesOvo (CodDesOvo, NomeDesOvo, DescDesOvo, CodStatusGeral)
VALUES (DEFAULT, 'Formol', '', 1);
INSERT INTO DesOvo (CodDesOvo, NomeDesOvo, DescDesOvo, CodStatusGeral) VALUES (DEFAULT, 'Amônia
Quaternária', '', 1);

```

```

-----
-- Tipos de entidade para projetos.
-----

```

```

CREATE TABLE TipoEntidade (
    CodTipoEnt NUMERIC(1) NOT NULL DEFAULT nextval('tipoentidade_codtipoent'),
    CodStatusGeral NUMERIC(3) NOT NULL,
    NomeTipoEnt VARCHAR(50) NOT NULL,
    DescTipoEnt VARCHAR NULL,
    TipoEntMin CHAR(1) NOT NULL,
    PRIMARY KEY(CodTipoEnt),
    FOREIGN KEY(CodStatusGeral)
        REFERENCES StatusGeral(CodStatusGeral)
        ON DELETE RESTRICT
        ON UPDATE CASCADE
);

```

```

INSERT INTO TipoEntidade (CodTipoEnt, NomeTipoEnt, DescTipoEnt, TipoEntMin, CodStatusGeral)
VALUES (DEFAULT, 'Ovo', 'Entidade mínima do projeto permite rastreamento individual por
Ovo.', 'S', 1);
INSERT INTO TipoEntidade (CodTipoEnt, NomeTipoEnt, DescTipoEnt, TipoEntMin, CodStatusGeral)
VALUES (DEFAULT, 'Celula', 'Entidade mínima do projeto permite rastreamento de ovos por
Célula.', 'S', 1);
INSERT INTO TipoEntidade (CodTipoEnt, NomeTipoEnt, DescTipoEnt, TipoEntMin, CodStatusGeral)
VALUES (DEFAULT, 'Gaiola', 'Entidade mínima do projeto permite rastreamento de ovos por
Gaiola.', 'S', 1);
INSERT INTO TipoEntidade (CodTipoEnt, NomeTipoEnt, DescTipoEnt, TipoEntMin, CodStatusGeral)
VALUES (DEFAULT, 'Linha', 'Entidade mínima do projeto permite rastreamento de ovos por
Linha.', 'S', 1);
INSERT INTO TipoEntidade (CodTipoEnt, NomeTipoEnt, DescTipoEnt, TipoEntMin, CodStatusGeral)
VALUES (DEFAULT, 'Bloco', 'Entidade mínima do projeto permite rastreamento de ovos por
Bloco.', 'S', 1);
INSERT INTO TipoEntidade (CodTipoEnt, NomeTipoEnt, DescTipoEnt, TipoEntMin, CodStatusGeral)
VALUES (DEFAULT, 'Sala', 'Entidade do projeto permite rastreamento de ovos por Sala.', 'N', 1);
INSERT INTO TipoEntidade (CodTipoEnt, NomeTipoEnt, DescTipoEnt, TipoEntMin, CodStatusGeral)
VALUES (DEFAULT, 'Andar', 'Entidade do projeto permite rastreamento de ovos por Andar.', 'N', 1);
INSERT INTO TipoEntidade (CodTipoEnt, NomeTipoEnt, DescTipoEnt, TipoEntMin, CodStatusGeral)
VALUES (DEFAULT, 'Predio', 'Entidade do projeto permite rastreamento de ovos por Prédio.', 'N', 1);

```

```

INSERT INTO TipoEntidade (CodTipoEnt, NomeTipoEnt, DescTipoEnt, TipoEntMin, CodStatusGeral)
VALUES (DEFAULT, 'Propriedade', 'Entidade do projeto permite rastreamento de ovos por
Propriedade.', 'N', 1);

```

```

-----
-- Raças de aves por espécie.
-----

```

```

CREATE TABLE RacaAve (
  CodRaca NUMERIC(8) NOT NULL DEFAULT nextval('racaave_codraca'),
  CodStatusGeral NUMERIC(3) NOT NULL,
  CodEsp NUMERIC(6) NOT NULL,
  NomeRaca VARCHAR(100) NOT NULL,
  DescRaca VARCHAR NULL,
  PRIMARY KEY(CodRaca),
  FOREIGN KEY(CodEsp)
    REFERENCES EspecieAve(CodEsp)
    ON DELETE RESTRICT
    ON UPDATE CASCADE,
  FOREIGN KEY(CodStatusGeral)
    REFERENCES StatusGeral(CodStatusGeral)
    ON DELETE RESTRICT
    ON UPDATE CASCADE
);

```

```

-----
-- Usuários com acesso ao sistema.
-----

```

```

CREATE TABLE Usuario (
  CodUsu NUMERIC(8) NOT NULL DEFAULT nextval('usuario_codusu') ,
  CodStatusUsu NUMERIC(3) NOT NULL,
  CodGru NUMERIC(1) NOT NULL,
  MatrUsu CHAR(10) NOT NULL,
  NomeUsu VARCHAR(100) NOT NULL,
  SenCrt VARCHAR NOT NULL,
  DataInc TIMESTAMP NOT NULL DEFAULT CURRENT_DATE,
  TelRes NUMERIC(10) NULL,
  TelCom NUMERIC(10) NULL,
  TelCel NUMERIC(10) NULL,
  Email VARCHAR(50) NOT NULL,
  UltAltSen TIMESTAMP NULL,
  UltLogon TIMESTAMP NULL,
  PRIMARY KEY(CodUsu),
  FOREIGN KEY(CodGru)
    REFERENCES Grupo(CodGru)
    ON DELETE RESTRICT
    ON UPDATE CASCADE,
  FOREIGN KEY(CodStatusUsu)
    REFERENCES StatusUsuario(CodStatusUsu)
);

```

```

        ON DELETE RESTRICT
        ON UPDATE CASCADE
    );

INSERT INTO Usuario (CodUsu, CodStatusUsu, CodGru, MatrUsu, NomeUsu, SenCrt, DataInc, TelRes,
TelCom, TelCel, Email, UltAltSen, UltLogon)
VALUES (DEFAULT,1,1,'admin','Administrador','admin',CURRENT_TIMESTAMP,NULL,NULL,NULL,'admin@sis
inc.com',CURRENT_TIMESTAMP,CURRENT_TIMESTAMP);

```

```

-- -----
-- Unidades de medida para valores de repetições.
-- -----

```

```

CREATE TABLE UnidadeMedida (
    CodUniMed NUMERIC(10) NOT NULL DEFAULT nextval('unidademedida_codunimed'),
    CodStatusGeral NUMERIC(3) NOT NULL,
    CodTipoMed NUMERIC(8) NOT NULL,
    NomeUniMed VARCHAR(100) NOT NULL,
    DescUniMed VARCHAR NULL,
    PRIMARY KEY(CodUniMed),
    FOREIGN KEY(CodTipoMed)
        REFERENCES TipoUniMed(CodTipoMed)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY(CodStatusGeral)
        REFERENCES StatusGeral(CodStatusGeral)
        ON DELETE RESTRICT
        ON UPDATE CASCADE
);

```

```

-- -----
-- Entidades físicas associadas ao projeto e à incubação.
-- -----

```

```

CREATE TABLE Entidade (
    CodEnt NUMERIC(12) NOT NULL DEFAULT nextval('entidade_codent'),
    CodStatusGeral NUMERIC(3) NOT NULL,
    CodTipoEnt NUMERIC(1) NOT NULL,
    NomeEnt VARCHAR(50) NOT NULL,
    DescEnt VARCHAR NULL,
    PRIMARY KEY(CodEnt),
    FOREIGN KEY(CodTipoEnt)
        REFERENCES TipoEntidade(CodTipoEnt)
        ON DELETE RESTRICT
        ON UPDATE CASCADE,
    FOREIGN KEY(CodStatusGeral)
        REFERENCES StatusGeral(CodStatusGeral)
        ON DELETE RESTRICT
        ON UPDATE CASCADE
);

```

-- Tipos de formato de ovos por espécie.

```
CREATE TABLE CorOvo (  
  CodCorOvo NUMERIC(3) NOT NULL DEFAULT nextval('corovo_codcorovo'),  
  CodStatusGeral NUMERIC(3) NOT NULL,  
  CodEsp NUMERIC(6) NOT NULL,  
  NomeCorOvo VARCHAR(50) NOT NULL,  
  DescCorOvo VARCHAR NULL,  
  PRIMARY KEY(CodCorOvo),  
  FOREIGN KEY(CodEsp)  
    REFERENCES EspécieAve(CodEsp)  
    ON DELETE NO ACTION  
    ON UPDATE CASCADE,  
  FOREIGN KEY(CodStatusGeral)  
    REFERENCES StatusGeral(CodStatusGeral)  
    ON DELETE RESTRICT  
    ON UPDATE CASCADE  
);
```

-- Cidades cadastradas no sistema.

```
CREATE TABLE Cidade (  
  CodCid NUMERIC(6) NOT NULL DEFAULT nextval('cidade_codcid'),  
  CodStatusGeral NUMERIC(3) NOT NULL,  
  CodUF NUMERIC(3) NOT NULL,  
  NomeCid VARCHAR(100) NOT NULL,  
  PRIMARY KEY(CodCid),  
  FOREIGN KEY(CodUF)  
    REFERENCES UF(CodUF)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE,  
  FOREIGN KEY(CodStatusGeral)  
    REFERENCES StatusGeral(CodStatusGeral)  
    ON DELETE RESTRICT  
    ON UPDATE CASCADE  
);
```

```
INSERT INTO Cidade (CodCid, CodUF, NomeCid, CodStatusGeral) VALUES(DEFAULT,18,'Curitiba',1);
```

-- Tipos de Tratamentos tem Tipos de Unidades de Medidas.

```
CREATE TABLE TipoTratTemUniMed (  
  CodTipoTrat NUMERIC(8) NOT NULL,
```

```

CodTipoMed NUMERIC(8) NOT NULL,
CodStatusGeral NUMERIC(3) NOT NULL,
PRIMARY KEY(CodTipoTrat, CodTipoMed),
FOREIGN KEY(CodTipoTrat)
  REFERENCES TipoTratamento(CodTipoTrat)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
FOREIGN KEY(CodTipoMed)
  REFERENCES TipoUniMed(CodTipoMed)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
FOREIGN KEY(CodStatusGeral)
  REFERENCES StatusGeral(CodStatusGeral)
    ON DELETE RESTRICT
    ON UPDATE CASCADE
);

-----
-- Projetos por espécie de ave definidos no tempo.
-----

CREATE TABLE Projeto (
  CodProj NUMERIC(8) NOT NULL DEFAULT nextval('projeto_codproj'),
  CodEsp NUMERIC(6) NOT NULL,
  CodTipoEnt NUMERIC(1) NOT NULL,
  CodStatusProj NUMERIC(3) NOT NULL,
  SiglaProj CHAR(14) NOT NULL,
  NomeProj VARCHAR(100) NOT NULL,
  DataIniProj DATE NOT NULL,
  DataPrevProj DATE NULL,
  DataFimProj DATE NULL,
  DescProj VARCHAR NULL,
  NumInc NUMERIC(3) NULL,
  PRIMARY KEY(CodProj),
  FOREIGN KEY(CodTipoEnt)
    REFERENCES TipoEntidade(CodTipoEnt)
      ON DELETE RESTRICT
      ON UPDATE CASCADE,
  FOREIGN KEY(CodEsp)
    REFERENCES EspécieAve(CodEsp)
      ON DELETE RESTRICT
      ON UPDATE CASCADE,
  FOREIGN KEY(CodStatusProj)
    REFERENCES StatusProjeto(CodStatusProj)
      ON DELETE RESTRICT
      ON UPDATE CASCADE
);

-----
-- Histórico de acessos de usuários.
-----

```

```

CREATE TABLE HistUsuario (
  CodHistUsu NUMERIC(20) NOT NULL DEFAULT nextval('histusuario_codhistusu_seq'),
  CodUsu NUMERIC(8) NOT NULL,
  SessaoIni TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  SessaoFim TIMESTAMP NULL,
  PRIMARY KEY(CodHistUsu),
  FOREIGN KEY(CodUsu)
    REFERENCES Usuario(CodUsu)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
);

```

```

-----
-- Propriedade, instituição ou granja de origem.
-----

```

```

CREATE TABLE Propriedade (
  CodProp NUMERIC(12) NOT NULL DEFAULT currval('entidade_codent'),
  CodCid NUMERIC(6) NOT NULL,
  EndProp VARCHAR NULL,
  PRIMARY KEY(CodProp),
  FOREIGN KEY(CodCid)
    REFERENCES Cidade(CodCid)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  FOREIGN KEY(CodProp)
    REFERENCES Entidade(CodEnt)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);

```

```

-----
-- Prédios disponíveis para propriedade.
-----

```

```

CREATE TABLE Predio (
  CodPre NUMERIC(12) NOT NULL DEFAULT currval('entidade_codent'),
  CodProp NUMERIC(12) NOT NULL,
  PRIMARY KEY(CodPre),
  FOREIGN KEY(CodProp)
    REFERENCES Propriedade(CodProp)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  FOREIGN KEY(CodPre)
    REFERENCES Entidade(CodEnt)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);

```

-- Pesquisas pertencentes aos projetos.

```
-----  
CREATE TABLE Tratamento (  
  CodTrat NUMERIC(10) NOT NULL DEFAULT nextval('tratamento_codtrat'),  
  CodStatusGeral NUMERIC(3) NOT NULL,  
  CodProj NUMERIC(8) NOT NULL,  
  NomeTrat VARCHAR(255) NOT NULL,  
  DescTrat VARCHAR NULL,  
  NumRep NUMERIC(3) NULL,  
  PRIMARY KEY(CodTrat),  
  FOREIGN KEY(CodProj)  
    REFERENCES Projeto(CodProj)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE,  
  FOREIGN KEY(CodStatusGeral)  
    REFERENCES StatusGeral(CodStatusGeral)  
    ON DELETE RESTRICT  
    ON UPDATE CASCADE  
);
```

-- Incubação de entidades e suas quantidades de ovos.

```
-----  
CREATE TABLE Incubacao (  
  CodInc NUMERIC(10) NOT NULL DEFAULT nextval('incubacao_codinc'),  
  CodStatusIN NUMERIC(3) NOT NULL,  
  CodProj NUMERIC(8) NOT NULL,  
  DataIniInc TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,  
  DataPrevInc TIMESTAMP NULL,  
  DataFimInc TIMESTAMP NULL,  
  NumOvoIniI NUMERIC(3) NOT NULL,  
  NumOvoDesI NUMERIC(3) NULL,  
  NumOvoTransf NUMERIC(3) NULL,  
  ObsStatusInc VARCHAR NULL,  
  PRIMARY KEY(CodInc),  
  FOREIGN KEY(CodProj)  
    REFERENCES Projeto(CodProj)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE,  
  FOREIGN KEY(CodStatusIN)  
    REFERENCES StatusIncNas(CodStatusIN)  
    ON DELETE RESTRICT  
    ON UPDATE CASCADE  
);
```

-- Andares disponíveis para prédio.

```

CREATE TABLE Andar (
  CodAnd NUMERIC(12) NOT NULL DEFAULT currval('entidade_codent'),
  CodPre NUMERIC(12) NOT NULL,
  PRIMARY KEY(CodAnd),
  FOREIGN KEY(CodPre)
    REFERENCES Predio(CodPre)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  FOREIGN KEY(CodAnd)
    REFERENCES Entidade(CodEnt)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);

-----
-- Nascimento de entidades que foram incubadas.
-----

CREATE TABLE Nascimento (
  CodNas NUMERIC(10) NOT NULL DEFAULT nextval('nascimento_codnas'),
  CodInc NUMERIC(10) NOT NULL,
  CodStatusIN NUMERIC(3) NOT NULL,
  DataIniINas TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  DataPrevNas TIMESTAMP NULL,
  DataFimNas TIMESTAMP NULL,
  NumOvoIniN NUMERIC(3) NOT NULL,
  NumOvoDesN NUMERIC(3) NULL,
  NumOvoRec NUMERIC(3) NULL,
  ObsStatusNas VARCHAR NULL,
  PRIMARY KEY(CodNas),
  FOREIGN KEY(CodStatusIN)
    REFERENCES StatusIncNas(CodStatusIN)
    ON DELETE RESTRICT
    ON UPDATE CASCADE,
  FOREIGN KEY(CodInc)
    REFERENCES Incubacao(CodInc)
    ON DELETE NO ACTION
    ON UPDATE CASCADE
);

-----
-- Tratamentos tem Tipos de Tratamentos.
-----

CREATE TABLE TratTemTipoTrat (
  CodTrat NUMERIC(10) NOT NULL,
  CodTipoTrat NUMERIC(8) NOT NULL,
  CodStatusGeral NUMERIC(3) NOT NULL,
  PRIMARY KEY(CodTrat, CodTipoTrat),
  FOREIGN KEY(CodTrat)
    REFERENCES Tratamento(CodTrat)

```

```

        ON DELETE CASCADE
        ON UPDATE CASCADE,
FOREIGN KEY(CodTipoTrat)
    REFERENCES TipoTratamento(CodTipoTrat)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
FOREIGN KEY(CodStatusGeral)
    REFERENCES StatusGeral(CodStatusGeral)
        ON DELETE RESTRICT
        ON UPDATE CASCADE
);

```

```

-----
-- Projetos compostos por usuários.
-----

```

```

CREATE TABLE ProjTemUsu (
    CodProj NUMERIC(8) NOT NULL,
    CodUsu NUMERIC(8) NOT NULL,
    CodStatusGeral NUMERIC(3) NOT NULL,
    Aviso CHAR(1) NOT NULL,
    DataInc DATE NOT NULL,
    DataTer DATE NULL,
    MotAlt VARCHAR NULL,
    PRIMARY KEY(CodProj, CodUsu),
    FOREIGN KEY(CodProj)
        REFERENCES Projeto(CodProj)
            ON DELETE NO ACTION
            ON UPDATE CASCADE,
    FOREIGN KEY(CodUsu)
        REFERENCES Usuario(CodUsu)
            ON DELETE NO ACTION
            ON UPDATE CASCADE,
    FOREIGN KEY(CodStatusGeral)
        REFERENCES StatusGeral(CodStatusGeral)
            ON DELETE RESTRICT
            ON UPDATE CASCADE
);

```

```

-----
-- Medições pertencentes ao tratamento.
-----

```

```

CREATE TABLE Repeticao (
    CodRep NUMERIC(12) NOT NULL DEFAULT nextval('repeticao_codrep'),
    CodStatusGeral NUMERIC(3) NOT NULL,
    CodEnt NUMERIC(12) NOT NULL,
    CodTrat NUMERIC(10) NOT NULL,
    CodTipoMed NUMERIC(8) NOT NULL,
    DataMedRep TIMESTAMP NULL DEFAULT CURRENT_TIMESTAMP,
    ValMedRep NUMERIC(12,8) NULL,

```

```

CodUniMed NUMERIC(10) NOT NULL,
PRIMARY KEY(CodRep),
FOREIGN KEY(CodTrat)
    REFERENCES Tratamento(CodTrat)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
FOREIGN KEY(CodUniMed)
    REFERENCES UnidadeMedida(CodUniMed)
        ON DELETE NO ACTION
        ON UPDATE CASCADE,
FOREIGN KEY(CodEnt)
    REFERENCES Entidade(CodEnt)
        ON DELETE RESTRICT
        ON UPDATE CASCADE,
FOREIGN KEY(CodTipoMed)
    REFERENCES TipoUniMed(CodTipoMed)
        ON DELETE NO ACTION
        ON UPDATE CASCADE,
FOREIGN KEY(CodStatusGeral)
    REFERENCES StatusGeral(CodStatusGeral)
        ON DELETE RESTRICT
        ON UPDATE CASCADE
);

-----
-- Salas disponíveis para andares.
-----

CREATE TABLE Sala (
    CodSala NUMERIC(12) NOT NULL DEFAULT currval('entidade_codent'),
    CodAnd NUMERIC(12) NOT NULL,
    PRIMARY KEY(CodSala),
    FOREIGN KEY(CodAnd)
        REFERENCES Andar(CodAnd)
            ON DELETE CASCADE
            ON UPDATE CASCADE,
    FOREIGN KEY(CodSala)
        REFERENCES Entidade(CodEnt)
            ON DELETE CASCADE
            ON UPDATE CASCADE
);

-----
-- Equipamentos que realizam o aquecimento dos ovos.
-----

CREATE TABLE Incubadora (
    CodEnc NUMERIC(3) NOT NULL DEFAULT nextval('incubadora_codenc'),
    CodStatusGeral NUMERIC(3) NOT NULL,
    CodSala NUMERIC(12) NOT NULL,
    NomeEnc VARCHAR(100) NOT NULL,

```

```

TPosEnc NUMERIC(2) NULL,
DescEnc VARCHAR NULL,
PRIMARY KEY(CodEnc),
FOREIGN KEY(CodSala)
    REFERENCES Sala(CodSala)
        ON DELETE RESTRICT
        ON UPDATE CASCADE,
FOREIGN KEY(CodStatusGeral)
    REFERENCES StatusGeral(CodStatusGeral)
        ON DELETE RESTRICT
        ON UPDATE CASCADE
);

-----
-- Blocos em salas, possuem linhas em disposição.
-----

CREATE TABLE Bloco (
    CodBloco NUMERIC(12) NOT NULL DEFAULT currval('entidade_codent'),
    CodSala NUMERIC(12) NOT NULL,
    PRIMARY KEY(CodBloco),
    FOREIGN KEY(CodSala)
        REFERENCES Sala(CodSala)
            ON DELETE CASCADE
            ON UPDATE CASCADE,
    FOREIGN KEY(CodBloco)
        REFERENCES Entidade(CodEnt)
            ON DELETE CASCADE
            ON UPDATE CASCADE
);

-----
-- Equipamentos onde os ovos permanecem na fase final de incubação.
-----

CREATE TABLE Nascedouro (
    CodNes NUMERIC(3) NOT NULL DEFAULT nextval('nascedouro_codnes'),
    CodStatusGeral NUMERIC(3) NOT NULL,
    CodSala NUMERIC(12) NOT NULL,
    NomeNes VARCHAR(100) NOT NULL,
    TColNes NUMERIC(2) NULL,
    DescNes VARCHAR NULL,
    PRIMARY KEY(CodNes),
    FOREIGN KEY(CodSala)
        REFERENCES Sala(CodSala)
            ON DELETE RESTRICT
            ON UPDATE CASCADE,
    FOREIGN KEY(CodStatusGeral)
        REFERENCES StatusGeral(CodStatusGeral)
            ON DELETE RESTRICT
            ON UPDATE CASCADE
);

```

```
);
```

```
-----  
-- Filas em blocos, possuem gaiolas em disposição.  
-----
```

```
CREATE TABLE Linha (  
  CodLinha NUMERIC(12) NOT NULL DEFAULT currval('entidade_codent'),  
  CodBloco NUMERIC(12) NOT NULL,  
  PRIMARY KEY(CodLinha),  
  FOREIGN KEY(CodBloco)  
    REFERENCES Bloco(CodBloco)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE,  
  FOREIGN KEY(CodLinha)  
    REFERENCES Entidade(CodEnt)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE  
);
```

```
-----  
-- Registro de posicionamento de bandeja em nascedouro.  
-----
```

```
CREATE TABLE PosBanNes (  
  CodPosBanNes NUMERIC(10) NOT NULL DEFAULT nextval('posbannes_codposbannes'),  
  CodStatusGeral NUMERIC(3) NOT NULL,  
  CodNes NUMERIC(3) NOT NULL,  
  CodBan NUMERIC(8) NOT NULL,  
  NPosNes NUMERIC(2) NOT NULL,  
  DataAloNes TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,  
  DataRetNes TIMESTAMP NULL,  
  PRIMARY KEY(CodPosBanNes),  
  FOREIGN KEY(CodNes)  
    REFERENCES Nascedouro(CodNes)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE,  
  FOREIGN KEY(CodBan)  
    REFERENCES Bandeja(CodBan)  
    ON DELETE NO ACTION  
    ON UPDATE CASCADE,  
  FOREIGN KEY(CodStatusGeral)  
    REFERENCES StatusGeral(CodStatusGeral)  
    ON DELETE RESTRICT  
    ON UPDATE CASCADE  
);
```

```
-----  
-- Registro de posicionamento de bandeja em incubadora.  
-----
```

```

CREATE TABLE PosBanEnc (
  CodPosBanEnc NUMERIC(10) NOT NULL DEFAULT nextval('posbanenc_codposbanenc'),
  CodStatusGeral NUMERIC(3) NOT NULL,
  CodEnc NUMERIC(3) NOT NULL,
  CodBan NUMERIC(8) NOT NULL,
  NPosEnc NUMERIC(2) NOT NULL,
  DataAloEnc TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  DataRetEnc TIMESTAMP NULL,
  PRIMARY KEY(CodPosBanEnc),
  FOREIGN KEY(CodEnc)
    REFERENCES Incubadora(CodEnc)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  FOREIGN KEY(CodBan)
    REFERENCES Bandeja(CodBan)
    ON DELETE NO ACTION
    ON UPDATE CASCADE,
  FOREIGN KEY(CodStatusGeral)
    REFERENCES StatusGeral(CodStatusGeral)
    ON DELETE RESTRICT
    ON UPDATE CASCADE
);

```

-- Posição de incubações em bandejas de incubadoras.

```

CREATE TABLE IncTemEnt (
  CodInc NUMERIC(10) NOT NULL,
  CodEnt NUMERIC(12) NOT NULL,
  NLinBan NUMERIC(2) NOT NULL,
  NColBan NUMERIC(2) NOT NULL,
  CodPosBanEnc NUMERIC(10) NOT NULL,
  CodStatusGeral NUMERIC(3) NOT NULL,
  CodMotDes NUMERIC(3) NOT NULL,
  OvoTransf CHAR(1) NULL,
  PRIMARY KEY(CodInc, CodEnt, NLinBan, NColBan, CodPosBanEnc),
  FOREIGN KEY(CodInc)
    REFERENCES Incubacao(CodInc)
    ON DELETE NO ACTION
    ON UPDATE CASCADE,
  FOREIGN KEY(CodEnt)
    REFERENCES Entidade(CodEnt)
    ON DELETE NO ACTION
    ON UPDATE CASCADE,
  FOREIGN KEY(CodPosBanEnc)
    REFERENCES PosBanEnc(CodPosBanEnc)
    ON DELETE NO ACTION
    ON UPDATE CASCADE,
  FOREIGN KEY(CodMotDes)
    REFERENCES MotivoDescarte(CodMotDes)
);

```

```

        ON DELETE RESTRICT
        ON UPDATE CASCADE,
FOREIGN KEY(CodStatusGeral)
REFERENCES StatusGeral(CodStatusGeral)
        ON DELETE RESTRICT
        ON UPDATE CASCADE
);

```

```

-----
-- Posição de nascimentos em bandejas de nascedouros.
-----

```

```

CREATE TABLE NasTemEnt (
    CodEnt NUMERIC(12) NOT NULL,
    CodNas NUMERIC(10) NOT NULL,
    NLinBan NUMERIC(2) NOT NULL,
    NColBan NUMERIC(2) NOT NULL,
    CodPosBanNes NUMERIC(10) NOT NULL,
    CodStatusGeral NUMERIC(3) NOT NULL,
    CodMotDes NUMERIC(3) NOT NULL,
    OvoNasceu CHAR(1) NULL,
    PRIMARY KEY(CodEnt, CodNas, NLinBan, NColBan, CodPosBanNes),
    FOREIGN KEY(CodNas)
        REFERENCES Nascimento(CodNas)
            ON DELETE NO ACTION
            ON UPDATE CASCADE,
    FOREIGN KEY(CodEnt)
        REFERENCES Entidade(CodEnt)
            ON DELETE NO ACTION
            ON UPDATE NO ACTION,
    FOREIGN KEY(CodPosBanNes)
        REFERENCES PosBanNes(CodPosBanNes)
            ON DELETE NO ACTION
            ON UPDATE CASCADE,
    FOREIGN KEY(CodMotDes)
        REFERENCES MotivoDescarte(CodMotDes)
            ON DELETE RESTRICT
            ON UPDATE CASCADE,
    FOREIGN KEY(CodStatusGeral)
        REFERENCES StatusGeral(CodStatusGeral)
            ON DELETE RESTRICT
            ON UPDATE CASCADE,
    FOREIGN KEY(CodStatusGeral)
        REFERENCES StatusGeral(CodStatusGeral)
            ON DELETE RESTRICT
            ON UPDATE CASCADE
);

```

```

-----
-- Registra medidas de temperatura e umidade de mecanismo externo para incubadora.
-----

```

```

CREATE TABLE HistIncubadora (
  CodHistEnc NUMERIC(20) NOT NULL DEFAULT nextval('histincubadora_codhistenc'),
  CodEnc NUMERIC(3) NOT NULL,
  DataMedEnc TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  TempMedEnc NUMERIC(6,4) NULL,
  UmiMedEnc NUMERIC(6,4) NULL,
  PRIMARY KEY(CodHistEnc),
  FOREIGN KEY(CodEnc)
    REFERENCES Incubadora(CodEnc)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
);

```

```

-----
-- Registra medidas de temperatura e umidade de mecanismo externo para nascedouro
-----

```

```

CREATE TABLE HistNascedouro (
  CodHistNes NUMERIC(20) NOT NULL DEFAULT nextval('histnascedouro_codhistnes'),
  CodNes NUMERIC(3) NOT NULL,
  DataMedNes TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  TempMedNes NUMERIC(6,4) NULL,
  UmiMedNes NUMERIC(6,4) NULL,
  PRIMARY KEY(CodHistNes),
  FOREIGN KEY(CodNes)
    REFERENCES Nascedouro(CodNes)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
);

```

```

-----
-- Filas em gaiolas, possuem células em disposição.
-----

```

```

CREATE TABLE Gaiola (
  CodGai NUMERIC(12) NOT NULL DEFAULT currval('entidade_codent'),
  CodLinha NUMERIC(12) NOT NULL,
  PRIMARY KEY(CodGai),
  FOREIGN KEY(CodLinha)
    REFERENCES Linha(CodLinha)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  FOREIGN KEY(CodGai)
    REFERENCES Entidade(CodEnt)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);

```

```

-----
-- Células em gaiolas, possuem famílias de aves em um determinado momento.

```

```

-----
CREATE TABLE Celula (
  CodCel NUMERIC(12) NOT NULL DEFAULT currval('entidade_codent'),
  CodGai NUMERIC(12) NOT NULL,
  PRIMARY KEY(CodCel),
  FOREIGN KEY(CodGai)
    REFERENCES Gaiola(CodGai)
      ON DELETE CASCADE
      ON UPDATE CASCADE,
  FOREIGN KEY(CodCel)
    REFERENCES Entidade(CodEnt)
      ON DELETE CASCADE
      ON UPDATE CASCADE
);

```

```

-----
-- Ovos produzidos por aves de famílias em células podem ser incubados.
-----

```

```

CREATE TABLE Ovo (
  CodOvo NUMERIC(12) NOT NULL DEFAULT currval('entidade_codent'),
  CodBarras NUMERIC NULL,
  CodDesOvo NUMERIC(3) NOT NULL,
  CodStatusOvo NUMERIC(3) NOT NULL,
  CodForOvo NUMERIC(3) NOT NULL,
  CodCorOvo NUMERIC(3) NOT NULL,
  CodCel NUMERIC(12) NOT NULL,
  AquisOvo NUMERIC(12) NOT NULL,
  DataColOvo DATE NULL DEFAULT CURRENT_DATE,
  AspOvo VARCHAR(50) NULL,
  PesoOvo NUMERIC(6) NULL,
  PorOvo NUMERIC(14) NULL,
  Incubavel CHAR(1) NULL,
  PRIMARY KEY(CodOvo),
  FOREIGN KEY(CodStatusOvo)
    REFERENCES StatusOvo(CodStatusOvo)
      ON DELETE RESTRICT
      ON UPDATE CASCADE,
  FOREIGN KEY(CodOvo)
    REFERENCES Entidade(CodEnt)
      ON DELETE CASCADE
      ON UPDATE CASCADE,
  FOREIGN KEY(CodForOvo)
    REFERENCES ForOvo(CodForOvo)
      ON DELETE RESTRICT
      ON UPDATE CASCADE,
  FOREIGN KEY(CodCel)
    REFERENCES Celula(CodCel)
      ON DELETE NO ACTION
      ON UPDATE CASCADE,
);

```

```

FOREIGN KEY(CodCorOvo)
  REFERENCES CorOvo(CodCorOvo)
  ON DELETE RESTRICT
  ON UPDATE CASCADE,
FOREIGN KEY(CodDesOvo)
  REFERENCES DesOvo(CodDesOvo)
  ON DELETE RESTRICT
  ON UPDATE CASCADE,
FOREIGN KEY(AquisOvo)
  REFERENCES Propriedade(CodProp)
  ON DELETE RESTRICT
  ON UPDATE CASCADE
);

-----
-- Família composta por aves em uma célula.
-----

CREATE TABLE Familia (
  CodFam NUMERIC(10) NOT NULL DEFAULT nextval('familia_codfam'),
  CodStatusGeral NUMERIC(3) NOT NULL,
  CodCel NUMERIC(12) NOT NULL,
  NomeFam VARCHAR(50) NOT NULL,
  NumMacho NUMERIC(1) NULL,
  NumFemea NUMERIC(1) NULL,
  DataIniFam DATE NOT NULL,
  DataFimFam DATE NULL,
  PRIMARY KEY(CodFam),
  FOREIGN KEY(CodCel)
    REFERENCES Celula(CodCel)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  FOREIGN KEY(CodStatusGeral)
    REFERENCES StatusGeral(CodStatusGeral)
    ON DELETE RESTRICT
    ON UPDATE CASCADE
);

-----
-- Aves compõe família em células e geram ovos por famílias.
-----

CREATE TABLE Ave (
  CodAve NUMERIC(10) NOT NULL DEFAULT nextval('ave_codave'),
  CodStatusGeral NUMERIC(3) NOT NULL,
  CodMotMor NUMERIC(3) NOT NULL,
  CodOvo NUMERIC(12) NOT NULL,
  CodGS NUMERIC(3) NOT NULL,
  CodRaca NUMERIC(8) NOT NULL,
  CodFam NUMERIC(10) NOT NULL,
  SexoAve CHAR(1) NOT NULL,

```

```

DataNasAve DATE NULL DEFAULT CURRENT_DATE,
DataMorAve DATE NULL,
MotMorAve VARCHAR NULL,
PRIMARY KEY(CodAve),
FOREIGN KEY(CodFam)
    REFERENCES Familia(CodFam)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
FOREIGN KEY(CodRaca)
    REFERENCES RacaAve(CodRaca)
    ON DELETE RESTRICT
    ON UPDATE CASCADE,
FOREIGN KEY(CodGS)
    REFERENCES GrauSangue(CodGS)
    ON DELETE RESTRICT
    ON UPDATE CASCADE,
FOREIGN KEY(CodOvo)
    REFERENCES Ovo(CodOvo)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
FOREIGN KEY(CodMotMor)
    REFERENCES MotivoMorte(CodMotMor)
    ON DELETE RESTRICT
    ON UPDATE CASCADE,
FOREIGN KEY(CodStatusGeral)
    REFERENCES StatusGeral(CodStatusGeral)
    ON DELETE RESTRICT
    ON UPDATE CASCADE
);

```

12.1.2 Sequências

```

-- -----
-- Sequências para Acesso
-- -----

```

```

CREATE SEQUENCE public.grupo_codgru_seq
    INCREMENT 1 MINVALUE 1
    MAXVALUE 9223372036854775807 START 1
    CACHE 1;
CREATE SEQUENCE public.histusuario_codhistusu_seq
    INCREMENT 1 MINVALUE 1
    MAXVALUE 9223372036854775807 START 1
    CACHE 1;
CREATE SEQUENCE public.statususuario_codstatususu_seq
    INCREMENT 1 MINVALUE 1
    MAXVALUE 9223372036854775807 START 1
    CACHE 1;
CREATE SEQUENCE public.usuario_codusu

```

```

INCREMENT 1 MINVALUE 1
MAXVALUE 9223372036854775807 START 1
CACHE 1;

-----
-- Sequências para Projeto
-----

CREATE SEQUENCE public.projeto_codproj
INCREMENT 1 MINVALUE 1
MAXVALUE 9223372036854775807 START 1
CACHE 1;

CREATE SEQUENCE public.repeticao_codrep
INCREMENT 1 MINVALUE 1
MAXVALUE 9223372036854775807 START 1
CACHE 1;

CREATE SEQUENCE public.statusprojeto_codstatusproj
INCREMENT 1 MINVALUE 1
MAXVALUE 9223372036854775807 START 1
CACHE 1;

CREATE SEQUENCE public.tipoentidade_codtipoent
INCREMENT 1 MINVALUE 1
MAXVALUE 9223372036854775807 START 1
CACHE 1;

CREATE SEQUENCE public.tipotratoamento_codtipotrat
INCREMENT 1 MINVALUE 1
MAXVALUE 9223372036854775807 START 1
CACHE 1;

CREATE SEQUENCE public.tipounimed_codtipomed
INCREMENT 1 MINVALUE 1
MAXVALUE 9223372036854775807 START 1
CACHE 1;

CREATE SEQUENCE public.tratamento_codtrat
INCREMENT 1 MINVALUE 1
MAXVALUE 9223372036854775807 START 1
CACHE 1;

CREATE SEQUENCE public.unidademedida_codunimed
INCREMENT 1 MINVALUE 1
MAXVALUE 9223372036854775807 START 1
CACHE 1;

-----
-- Sequências para Entidade
-----

CREATE SEQUENCE public.cidade_codcid
INCREMENT 1 MINVALUE 1
MAXVALUE 9223372036854775807 START 1
CACHE 1;

CREATE SEQUENCE public.corovo_codcorovo
INCREMENT 1 MINVALUE 1

```

```

MAXVALUE 9223372036854775807 START 1
CACHE 1;
CREATE SEQUENCE public.desovo_coddesovo
INCREMENT 1 MINVALUE 1
MAXVALUE 9223372036854775807 START 1
CACHE 1;
CREATE SEQUENCE public.entidade_codent
INCREMENT 1 MINVALUE 1
MAXVALUE 9223372036854775807 START 1
CACHE 1;
CREATE SEQUENCE public.forovo_codforovo
INCREMENT 1 MINVALUE 1
MAXVALUE 9223372036854775807 START 1
CACHE 1;
CREATE SEQUENCE public.statusovo_codstatusovo
INCREMENT 1 MINVALUE 1
MAXVALUE 9223372036854775807 START 1
CACHE 1;
CREATE SEQUENCE public.uf_coduf
INCREMENT 1 MINVALUE 1
MAXVALUE 9223372036854775807 START 1
CACHE 1;

```

```

-----
-- Sequências para Ave
-----

```

```

CREATE SEQUENCE public.ave_codave
INCREMENT 1 MINVALUE 1
MAXVALUE 9223372036854775807 START 1
CACHE 1;
CREATE SEQUENCE public.especieave_codesp
INCREMENT 1 MINVALUE 1
MAXVALUE 9223372036854775807 START 1
CACHE 1;
CREATE SEQUENCE public.familia_codfam
INCREMENT 1 MINVALUE 1
MAXVALUE 9223372036854775807 START 1
CACHE 1;
CREATE SEQUENCE public.grausangue_codgs
INCREMENT 1 MINVALUE 1
MAXVALUE 9223372036854775807 START 1
CACHE 1;
CREATE SEQUENCE public.motivomorte_codmotmorte
INCREMENT 1 MINVALUE 1
MAXVALUE 9223372036854775807 START 1
CACHE 1;
CREATE SEQUENCE public.racaave_codraca
INCREMENT 1 MINVALUE 1
MAXVALUE 9223372036854775807 START 1
CACHE 1;

```

-- Sequências para Incubação

```
CREATE SEQUENCE public.bandeja_codban
  INCREMENT 1 MINVALUE 1
  MAXVALUE 9223372036854775807 START 1
  CACHE 1;
CREATE SEQUENCE public.incubacao_codinc
  INCREMENT 1 MINVALUE 1
  MAXVALUE 9223372036854775807 START 1
  CACHE 1;
CREATE SEQUENCE public.incubadora_codenc
  INCREMENT 1 MINVALUE 1
  MAXVALUE 9223372036854775807 START 1
  CACHE 1;
CREATE SEQUENCE public.histnascedouro_codhistnes
  INCREMENT 1 MINVALUE 1
  MAXVALUE 9223372036854775807 START 1
  CACHE 1;
CREATE SEQUENCE public.histincubadora_codhistenc
  INCREMENT 1 MINVALUE 1
  MAXVALUE 9223372036854775807 START 1
  CACHE 1;
CREATE SEQUENCE public.nascedouro_codnes
  INCREMENT 1 MINVALUE 1
  MAXVALUE 9223372036854775807 START 1
  CACHE 1;
CREATE SEQUENCE public.nascimento_codnas
  INCREMENT 1 MINVALUE 1
  MAXVALUE 9223372036854775807 START 1
  CACHE 1;
CREATE SEQUENCE public.posbanenc_codposbanenc
  INCREMENT 1 MINVALUE 1
  MAXVALUE 9223372036854775807 START 1
  CACHE 1;
CREATE SEQUENCE public.posbannes_codposbannes
  INCREMENT 1 MINVALUE 1
  MAXVALUE 9223372036854775807 START 1
  CACHE 1;
CREATE SEQUENCE public.motivodescarte_codmotdes
  INCREMENT 1 MINVALUE 1
  MAXVALUE 9223372036854775807 START 1
  CACHE 1;
CREATE SEQUENCE public.statusincnas_codstatusin
  INCREMENT 1 MINVALUE 1
  MAXVALUE 9223372036854775807 START 1
  CACHE 1;
```

-- Sequências para Status

```
-----  
CREATE SEQUENCE public.statusgeral_codstatusgeral  
    INCREMENT 1 MINVALUE 1  
    MAXVALUE 9223372036854775807 START 1  
    CACHE 1;
```

12.1.3 Funções e Triggers

```
CREATE OR REPLACE FUNCTION "public"."fatualizaincubacaonumovoini" () RETURNS trigger AS  
$body$  
BEGIN  
if (TG_OP = 'INSERT') then  
    IF new.codstatusgeral = 1 then update incubacao SET numovoinii = numovoinii + 1 where  
codinc = new.codinc;  
    END IF;  
elseif (TG_OP = 'UPDATE') then  
    if new.codstatusgeral = 1 and old.codstatusgeral <> 1 then update incubacao SET numovoinii  
= numovoinii + 1 where codinc = new.codinc;  
    elseif new.codstatusgeral <> 1 and old.codstatusgeral = 1 then update incubacao SET  
numovoinii = numovoinii - 1 where codinc = new.codinc;  
    end if;  
    if new.codstatusgeral = 1 and new.codmotdes = 1 and old.codmotdes <> 1 then update  
incubacao set numovodesi = numovodesi - 1 where codinc = new.codinc;  
    elseif new.codstatusgeral = 1 and new.codmotdes <> 1 and old.codmotdes = 1 then update  
incubacao set numovodesi = numovodesi + 1 where codinc = new.codinc;  
    end if;  
end if;  
Return new;  
END;  
$body$  
LANGUAGE 'plpgsql' VOLATILE CALLED ON NULL INPUT SECURITY INVOKER;
```

```
CREATE OR REPLACE FUNCTION "public"."fatualizaincubacaonumovotransf" () RETURNS trigger AS  
$body$  
BEGIN  
if (TG_OP = 'INSERT') then  
    IF new.codstatusgeral = 1 then  
        update incubacao SET numovotransf = numovotransf + 1 where incubacao.codinc = (select  
codinc from nascimento where new.codnas = nascimento.codnas);  
        update nascimento SET numovoinin = numovoinin + 1 where nascimento.codnas =  
new.codnas;  
    END IF;  
elseif (TG_OP = 'UPDATE') then  
    if new.codstatusgeral = 1 and old.codstatusgeral <> 1 then  
        update incubacao SET numovotransf = numovotransf + 1 where incubacao.codinc = (select  
codinc from nascimento where new.codnas = nascimento.codnas);
```

```

        update nascimento SET numovoinin = numovoinin + 1 where nascimento.codnas =
new.codnas;
        elsif new.codstatusgeral <> 1 and old.codstatusgeral = 1 then
            update incubacao SET numovotransf = numovotransf - 1 where incubacao.codinc = (select
codinc from nascimento where new.codnas = nascimento.codnas);
            update nascimento SET numovoinin = numovoinin - 1 where nascimento.codnas =
new.codnas;
        end if;
        if new.codstatusgeral = 1 and new.codmotdes <> 1 and old.codmotdes = 1 then
            update nascimento set numovodesn = numovodesn + 1 where nascimento.codnas =
new.codnas;
        elsif new.codstatusgeral = 1 and new.codmotdes = 1 and old.codmotdes <> 1 then
            update nascimento set numovodesn = numovodesn - 1 where nascimento.codnas =
new.codnas;
        end if;
    end if;
Return new;
END;
$body$
LANGUAGE 'plpgsql' VOLATILE CALLED ON NULL INPUT SECURITY INVOKER;

```

```

CREATE OR REPLACE FUNCTION "public"."fatualizaprojetonuminc" () RETURNS trigger AS
$body$
BEGIN
if (TG_OP = 'INSERT') then
    IF new.codstatusin = 1 then
        update projeto SET numinc = numinc + 1 where projeto.codproj = (select codproj from
incubacao where new.codinc = incubacao.codinc);
    END IF;
elsif (TG_OP = 'UPDATE') then
    if new.codstatusin <> 3 and old.codstatusin = 3 then
        update projeto SET numinc = numinc + 1 where projeto.codproj = (select codproj from
incubacao where new.codinc = incubacao.codinc);
    elsif new.codstatusin = 3 and old.codstatusin <> 3 then
        update projeto SET numinc = numinc - 1 where projeto.codproj = (select codproj from
incubacao where new.codinc = incubacao.codinc);
    end if;
end if;
Return new;
END;
$body$
LANGUAGE 'plpgsql' VOLATILE CALLED ON NULL INPUT SECURITY INVOKER;

```

```

CREATE TRIGGER "tIncTemEnt" AFTER INSERT OR UPDATE
ON "public"."inctement" FOR EACH ROW
EXECUTE PROCEDURE "public"."fatualizaincubacaonumovoini"();

```

```

CREATE TRIGGER "tNasTemEnt" AFTER INSERT OR UPDATE
ON "public"."nastement" FOR EACH ROW
EXECUTE PROCEDURE "public"."fatualizaincubacaonumovotransf"();

```

```
CREATE TRIGGER "tIncubacao" AFTER INSERT OR UPDATE
ON "public"."incubacao" FOR EACH ROW
EXECUTE PROCEDURE "public"."fatualizaprojetonuminc"();
```

12.1.4 Índices

```
CREATE UNIQUE INDEX "inposbanenc_1" ON "public"."posbanenc"
USING btree ("codenc", "nposenc", "dataaloenc", "codstatusgeral");
```

```
CREATE UNIQUE INDEX "inposbanenc_2" ON "public"."posbanenc"
USING btree ("codban", "dataaloenc", "codstatusgeral");
```

```
CREATE UNIQUE INDEX "inusuario_matrusu" ON "public"."usuario"
USING btree ("matrusu");
```

```
CREATE UNIQUE INDEX "inprojeto_siglaproj" ON "public"."projeto"
USING btree ("siglaproj");
```

```
CREATE UNIQUE INDEX "inprojeto_nomeproj" ON "public"."projeto"
USING btree ("nomeproj");
```

12.1.5 Views

```
CREATE OR REPLACE VIEW "public"."vEntidadeOvo" (
codent,
nomeent,
descent,
codstatusgeral,
nomestatusgeral,
codtipoent,
nometipoent,
tipoentmin,
codbarras,
coddesovo,
nomedesovo,
codstatusovo,
nomestatusovo,
codforovo,
nomeforovo,
codcorovo,
nomecorovo,
codcel,
quisovo,
datacolovo,
aspovo,
pesoovo,
porovo,
incubavel)
```

```

AS
SELECT
    entidade.codent,
    entidade.nomeent,
    entidade.descent,
    statusgeral.codstatusgeral,
    statusgeral.nomestatusgeral,
    tipoentidade.codtipoent,
    tipoentidade.nometipoent,
    tipoentidade.tipoentmin,
    ovo.codbarras,
    desovo.coddesovo,
    desovo.nomedesovo,
    statusovo.codstatusovo,
    statusovo.nomestatusovo,
    forovo.codforovo,
    forovo.nomeforovo,
    corovo.codcorovo,
    corovo.nomecorovo,
    ovo.codcel,
    ovo.aquisovo,
    ovo.datacolovo,
    ovo.aspovo,
    ovo.pesoovo,
    ovo.porovo,
    ovo.incubavel
FROM
    public.entidade
    INNER JOIN public.statusgeral ON
    (public.entidade.codstatusgeral=public.statusgeral.codstatusgeral)
    INNER JOIN public.tipoentidade ON (public.entidade.codtipoent=public.tipoentidade.codtipoent)
    INNER JOIN public.ovo ON (public.entidade.codent=public.ovo.codovo)
    INNER JOIN public.statusovo ON (public.ovo.codstatusovo=public.statusovo.codstatusovo)
    INNER JOIN public.forovo ON (public.ovo.codforovo=public.forovo.codforovo)
    INNER JOIN public.desovo ON (public.ovo.coddesovo=public.desovo.coddesovo)
    INNER JOIN public.corovo ON (public.ovo.codcorovo=public.corovo.codcorovo)
WHERE
    (public.entidade.codent = public.ovo.codovo) AND
    (public.entidade.codstatusgeral = public.statusgeral.codstatusgeral) AND
    (public.entidade.codtipoent = public.tipoentidade.codtipoent) AND
    (public.ovo.coddesovo = public.desovo.coddesovo) AND
    (public.ovo.codstatusovo = public.statusovo.codstatusovo) AND
    (public.ovo.codforovo = public.forovo.codforovo) AND
    (public.ovo.codcorovo = public.corovo.codcorovo) AND
    (public.entidade.codtipoent = 1)
--
--
-- ***
--
-- Consulta que relaciona a view EntidadeOvo à tabela Célula (permitindo associação de ovos às
respectivas células)

```

```

--
-- SELECT
--   public."vEntidadeOvo".codent as codentovo,
--   public."vEntidadeOvo".nomeent as nomeentovo,
--   public."vEntidadeOvo".descent as descentovo,
--   public.entidade.nomeent as nomeentcel,
--   public.entidade.descent as descentcel
-- FROM
--   public."vEntidadeOvo"
-- INNER JOIN public.entidade ON (public."vEntidadeOvo".codcel=public.entidade.codent)

CREATE OR REPLACE VIEW "public"."vEntidadeCelula" (
  codent,
  nomeent,
  descent,
  codstatusgeral,
  nomestatusgeral,
  codtipoent,
  nometipoent,
  tipoentmin,
  codgai)
AS
SELECT
  entidade.codent,
  entidade.nomeent,
  entidade.descent,
  statusgeral.codstatusgeral,
  statusgeral.nomestatusgeral,
  tipoentidade.codtipoent,
  tipoentidade.nometipoent,
  tipoentidade.tipoentmin,
  celula.codgai
FROM
  public.entidade
  INNER JOIN public.statusgeral ON
  (public.entidade.codstatusgeral=public.statusgeral.codstatusgeral)
  INNER JOIN public.celula ON (public.entidade.codent=public.celula.codcel)
  INNER JOIN public.tipoentidade ON (public.entidade.codtipoent=public.tipoentidade.codtipoent)
WHERE
  (public.entidade.codent = public.celula.codcel) AND
  (public.entidade.codstatusgeral = public.statusgeral.codstatusgeral) AND
  (public.entidade.codtipoent = public.tipoentidade.codtipoent) AND
  (public.entidade.codtipoent = 2)
--
--
-- ***
--
-- Consulta que relaciona a view EntidadeCelula à tabela Gaiola (permitindo associação de
células às respectivas gaiolas)
--

```

```

-- SELECT
--   public."vEntidadeCelula".codent as codentcel,
--   public."vEntidadeCelula".nomeent as nomeentcel,
--   public."vEntidadeCelula".descent as descentcel,
--   public.entidade.nomeent as nomeentgai,
--   public.entidade.descent as descentgai
-- FROM
--   public."vEntidadeCelula"
-- INNER JOIN public.entidade ON (public."vEntidadeCelula".codgai=public.entidade.codent)

CREATE OR REPLACE VIEW "public"."vEntidadeGaiola" (
  codent,
  nomeent,
  descent,
  codstatusgeral,
  nomestatusgeral,
  codtipoent,
  nometipoent,
  tipoentmin,
  codlinha)
AS
SELECT
  entidade.codent,
  entidade.nomeent,
  entidade.descent,
  statusgeral.codstatusgeral,
  statusgeral.nomestatusgeral,
  tipoentidade.codtipoent,
  tipoentidade.nometipoent,
  tipoentidade.tipoentmin,
  gaiola.codlinha
FROM
  public.entidade
  INNER JOIN public.gaiola ON (public.entidade.codent=public.gaiola.codgai)
  INNER JOIN public.statusgeral ON
(public.entidade.codstatusgeral=public.statusgeral.codstatusgeral)
  INNER JOIN public.tipoentidade ON (public.entidade.codtipoent=public.tipoentidade.codtipoent)
WHERE
  (public.entidade.codent = public.gaiola.codgai) AND
  (public.entidade.codstatusgeral = public.statusgeral.codstatusgeral) AND
  (public.entidade.codtipoent = public.tipoentidade.codtipoent) AND
  (public.entidade.codtipoent = 3)
--
--
-- ***
--
-- Consulta que relaciona a view EntidadeGaiola à tabela Linha (permitindo associação de
gaiolas às respectivas linhas)
--
-- SELECT

```

```

-- public."vEntidadeGaiola".codent as codentgai,
-- public."vEntidadeGaiola".nomeent as nomeentgai,
-- public."vEntidadeGaiola".descent as descentgai,
-- public.entidade.nomeent as nomeentlinha,
-- public.entidade.descent as descentlinha
-- FROM
-- public."vEntidadeGaiola"
-- INNER JOIN public.entidade ON (public."vEntidadeGaiola".codlinha=public.entidade.codent)

CREATE OR REPLACE VIEW "public"."vEntidadeLinha" (
    codent,
    nomeent,
    descent,
    codstatusgeral,
    nomestatusgeral,
    codtipoent,
    nometipoent,
    tipoentmin,
    codbloco)
AS
SELECT
    entidade.codent,
    entidade.nomeent,
    entidade.descent,
    statusgeral.codstatusgeral,
    statusgeral.nomestatusgeral,
    tipoentidade.codtipoent,
    tipoentidade.nometipoent,
    tipoentidade.tipoentmin,
    linha.codbloco
FROM
    public.entidade
INNER JOIN public.linha ON (public.entidade.codent=public.linha.codlinha)
INNER JOIN public.statusgeral ON
(public.entidade.codstatusgeral=public.statusgeral.codstatusgeral)
INNER JOIN public.tipoentidade ON (public.entidade.codtipoent=public.tipoentidade.codtipoent)
WHERE
    (public.entidade.codent = public.linha.codlinha) AND
    (public.entidade.codstatusgeral = public.statusgeral.codstatusgeral) AND
    (public.entidade.codtipoent = public.tipoentidade.codtipoent) AND
    (public.entidade.codtipoent = 4)
--
--
--
-- ***
--
-- Consulta que relaciona a view EntidadeLinha à tabela Bloco (permitindo associação de linhas
aos respectivos blocos)
--
-- SELECT

```

```

-- public."vEntidadeLinha".codent as codentlinha,
-- public."vEntidadeLinha".nomeent as nomeentlinha,
-- public."vEntidadeLinha".descent as descentlinha,
-- public.entidade.nomeent as nomeentbloco,
-- public.entidade.descent as descentbloco
-- FROM
-- public."vEntidadeLinha"
-- INNER JOIN public.entidade ON (public."vEntidadeLinha".codbloco=public.entidade.codent)

CREATE OR REPLACE VIEW "public"."vEntidadeBloco" (
    codent,
    nomeent,
    descent,
    codstatusgeral,
    nomestatusgeral,
    codtipoent,
    nometipoent,
    tipoentmin,
    codsala)
AS
SELECT
    entidade.codent,
    entidade.nomeent,
    entidade.descent,
    statusgeral.codstatusgeral,
    statusgeral.nomestatusgeral,
    tipoentidade.codtipoent,
    tipoentidade.nometipoent,
    tipoentidade.tipoentmin,
    bloco.codsala
FROM
    public.entidade
INNER JOIN public.statusgeral ON
(public.entidade.codstatusgeral=public.statusgeral.codstatusgeral)
INNER JOIN public.tipoentidade ON (public.entidade.codtipoent=public.tipoentidade.codtipoent)
INNER JOIN public.bloco ON (public.entidade.codent=public.bloco.codbloco)
WHERE
    (public.entidade.codent = public.bloco.codbloco) AND
    (public.entidade.codstatusgeral = public.statusgeral.codstatusgeral) AND
    (public.entidade.codtipoent = public.tipoentidade.codtipoent) AND
    (public.entidade.codtipoent = 5)
--
--
--
-- ***
--
--
-- Consulta que relaciona a view EntidadeBloco à tabela Sala (permitindo associação de blocos
às respectivas salas)
--

```

```

--
-- SELECT
--   public."vEntidadeBloco".codent as codentbloco,
--   public."vEntidadeBloco".nomeent as nomeentbloco,
--   public."vEntidadeBloco".descent as descentbloco,
--   public.entidade.nomeent as nomeentsala,
--   public.entidade.descent as descentsala
-- FROM
--   public."vEntidadeBloco"
-- INNER JOIN public.entidade ON (public."vEntidadeBloco".codsala=public.entidade.codent)

CREATE OR REPLACE VIEW "public"."vEntidadeSala" (
  codent,
  nomeent,
  descent,
  codstatusgeral,
  nomestatusgeral,
  codtipoent,
  nometipoent,
  tipoentmin,
  codand)
AS
SELECT
  entidade.codent,
  entidade.nomeent,
  entidade.descent,
  statusgeral.codstatusgeral,
  statusgeral.nomestatusgeral,
  tipoentidade.codtipoent,
  tipoentidade.nometipoent,
  tipoentidade.tipoentmin,
  sala.codand
FROM
  public.entidade
  INNER JOIN public.statusgeral ON
  (public.entidade.codstatusgeral=public.statusgeral.codstatusgeral)
  INNER JOIN public.tipoentidade ON (public.entidade.codtipoent=public.tipoentidade.codtipoent)
  INNER JOIN public.sala ON (public.entidade.codent=public.sala.codsala)
WHERE
  (public.entidade.codent = public.sala.codsala) AND
  (public.entidade.codstatusgeral = public.statusgeral.codstatusgeral) AND
  (public.entidade.codtipoent = public.tipoentidade.codtipoent) AND
  (public.entidade.codtipoent = 6)
--
--
--
-- ***
--
-- Consulta que relaciona a view EntidadeSala à tabela Andar (permitindo associação de salas
aos respectivos andares)

```

```

--
--
-- SELECT
--   public."vEntidadeSala".codent as codentsala,
--   public."vEntidadeSala".nomeent as nomeentsala,
--   public."vEntidadeSala".descent as descentsala,
--   public.entidade.nomeent as nomeentand,
--   public.entidade.descent as descentand
-- FROM
--   public."vEntidadeSala"
-- INNER JOIN public.entidade ON (public."vEntidadeSala".codand=public.entidade.codent)

CREATE OR REPLACE VIEW "public"."vEntidadeAndar" (
  codent,
  nomeent,
  descent,
  codstatusgeral,
  nomestatusgeral,
  codtipoent,
  nometipoent,
  tipoentmin,
  codpre)
AS
SELECT
  entidade.codent,
  entidade.nomeent,
  entidade.descent,
  statusgeral.codstatusgeral,
  statusgeral.nomestatusgeral,
  tipoentidade.codtipoent,
  tipoentidade.nometipoent,
  tipoentidade.tipoentmin,
  andar.codpre
FROM
  public.entidade
  INNER JOIN public.statusgeral ON
(public.entidade.codstatusgeral=public.statusgeral.codstatusgeral)
  INNER JOIN public.tipoentidade ON (public.entidade.codtipoent=public.tipoentidade.codtipoent)
  INNER JOIN public.andar ON (public.entidade.codent=public.andar.codand)
WHERE
  (public.entidade.codent = public.andar.codand) AND
  (public.entidade.codstatusgeral = public.statusgeral.codstatusgeral) AND
  (public.entidade.codtipoent = public.tipoentidade.codtipoent) AND
  (public.entidade.codtipoent = 7)
--
--
--
-- ***
--

```

```

-- Consulta que relaciona a view EntidadeAndar à tabela Predio (permitindo associação de
andares aos respectivos predios)
--
--
-- SELECT
--   public."vEntidadeAndar".codent as codentand,
--   public."vEntidadeAndar".nomeent as nomeentand,
--   public."vEntidadeAndar".descent as descentand,
--   public.entidade.nomeent as nomeentpre,
--   public.entidade.descent as descentpre
-- FROM
--   public."vEntidadeAndar"
-- INNER JOIN public.entidade ON (public."vEntidadeAndar".codpre=public.entidade.codent)

CREATE OR REPLACE VIEW "public"."vEntidadePredio" (
  codent,
  nomeent,
  descent,
  codstatusgeral,
  nomestatusgeral,
  codtipoent,
  nometipoent,
  tipoentmin,
  codprop)
AS
SELECT
  entidade.codent,
  entidade.nomeent,
  entidade.descent,
  statusgeral.codstatusgeral,
  statusgeral.nomestatusgeral,
  tipoentidade.codtipoent,
  tipoentidade.nometipoent,
  tipoentidade.tipoentmin,
  predio.codprop
FROM
  public.entidade
  INNER JOIN public.statusgeral ON
(public.entidade.codstatusgeral=public.statusgeral.codstatusgeral)
  INNER JOIN public.tipoentidade ON (public.entidade.codtipoent=public.tipoentidade.codtipoent)
  INNER JOIN public.predio ON (public.entidade.codent=public.predio.codpre)
WHERE
  (public.entidade.codent = public.predio.codpre) AND
  (public.entidade.codstatusgeral = public.statusgeral.codstatusgeral) AND
  (public.entidade.codtipoent = public.tipoentidade.codtipoent) AND
  (public.entidade.codtipoent = 8)
--
--
--
-- ***

```

```

--
-- Consulta que relaciona a view EntidadePredio à tabela Propriedade (permitindo associação de
prédios às respectivas propriedades)
--
--
-- SELECT
--   public."vEntidadePredio".codent as codentpre,
--   public."vEntidadePredio".nomeent as nomeentpre,
--   public."vEntidadePredio".descent as descentpre,
--   public.entidade.nomeent as nomeentprop,
--   public.entidade.descent as descentprop
-- FROM
--   public."vEntidadePredio"
-- INNER JOIN public.entidade ON (public."vEntidadePredio".codprop=public.entidade.codent)

```

```

CREATE OR REPLACE VIEW "public"."vEntidadePropriedade" (
  codent,
  nomeent,
  descent,
  codstatusgeral,
  nomestatusgeral,
  codtipoent,
  nometipoent,
  tipoentmin,
  endprop,
  codcid,
  nomecid,
  coduf,
  siglauf,
  nomeuf)
AS
SELECT
  entidade.codent,
  entidade.nomeent,
  entidade.descent,
  statusgeral.codstatusgeral,
  statusgeral.nomestatusgeral,
  tipoentidade.codtipoent,
  tipoentidade.nometipoent,
  tipoentidade.tipoentmin,
  propriedade.endprop,
  cidade.codcid,
  cidade.nomecid,
  uf.coduf,
  uf.siglauf,
  uf.nomeuf
FROM
  public.entidade
INNER JOIN public.propriedade ON (public.entidade.codent=public.propriedade.codprop)

```

```

INNER JOIN public.statusgeral ON
(public.entidade.codstatusgeral=public.statusgeral.codstatusgeral)
INNER JOIN public.tipoentidade ON (public.entidade.codtipoent=public.tipoentidade.codtipoent)
INNER JOIN public.cidade ON (public.propriedade.codcid=public.cidade.codcid)
INNER JOIN public.uf ON (public.cidade.coduf=public.uf.coduf)
WHERE
(public.entidade.codent = public.propriedade.codprop) AND
(public.propriedade.codcid = public.cidade.codcid) AND
(public.cidade.coduf = public.uf.coduf) AND
(public.entidade.codstatusgeral = public.statusgeral.codstatusgeral) AND
(public.entidade.codtipoent = public.tipoentidade.codtipoent) AND
(public.entidade.codtipoent = 9)
--
--
--
-- ***
--
-- Consulta que relaciona a view EntidadePropriedade à tabela Cidade e Uf (permitindo
associação de propriedades às respectivas cidades e ufs)
--
-- SELECT * FROM public."vEntidadePropriedade"

CREATE OR REPLACE VIEW "public"."vProjetoIncubacao" (
codproj,
siglaproj,
nomeproj,
datainiproj,
dataprevproj,
datafimproj,
descproj,
numinc,
codesp,
codstatusgeral,
nomestatusgeral,
nomeesp,
tempmininc,
tempmaxinc,
umimininc,
umimaxinc,
perinc,
peraviinc,
tempminnas,
tempmaxnas,
umiminnas,
umimaxnas,
pernas,
peravinas,
perrec,
codtipoent,
nometipoent,

```

```

    tipoentmin,
    codstatusproj,
    nomestatusproj,
    codstatusin,
    datainiinc,
    dataprevinc,
    datafiminc,
    numovoinii,
    numovodesi,
    numovotransf,
    obsstatusinc)
AS
SELECT
    projeto.codproj,
    projeto.siglaproj,
    projeto.nomeproj,
    projeto.datainiproj,
    projeto.dataprevproj,
    projeto.datafimproj,
    projeto.descproj,
    projeto.numinc,
    especieave.codesp,
    statusgeral.codstatusgeral,
    statusgeral.nomestatusgeral,
    especieave.nomeesp,
    especieave.tempmininc,
    especieave.tempmaxinc,
    especieave.umimininc,
    especieave.umimaxinc,
    especieave.perinc,
    especieave.peraviinc,
    especieave.tempminnas,
    especieave.tempmaxnas,
    especieave.umiminnas,
    especieave.umimaxnas,
    especieave.pernas,
    especieave.peravinas,
    especieave.perrec,
    tipoentidade.codtipoent,
    tipoentidade.nometipoent,
    tipoentidade.tipoentmin,
    statusprojeto.codstatusproj,
    statusprojeto.nomestatusproj,
    incubacao.codstatusin,
    incubacao.datainiinc,
    incubacao.dataprevinc,
    incubacao.datafiminc,
    incubacao.numovoinii,
    incubacao.numovodesi,
    incubacao.numovotransf,
    incubacao.obsstatusinc

```

```

FROM
  public.projeto
  INNER JOIN public.incubacao ON (public.projeto.codproj=public.incubacao.codproj)
  INNER JOIN public.especieave ON (public.projeto.codesp=public.especieave.codesp)
  INNER JOIN public.tipoentidade ON (public.projeto.codtipoent=public.tipoentidade.codtipoent)
  INNER JOIN public.statusprojeto ON
  (public.projeto.codstatusproj=public.statusprojeto.codstatusproj)
  INNER JOIN public.statusgeral ON
  (public.especieave.codstatusgeral=public.statusgeral.codstatusgeral)
WHERE
  (public.projeto.codproj = public.incubacao.codproj) AND
  (public.projeto.codesp = public.especieave.codesp) AND
  (public.projeto.codtipoent = public.tipoentidade.codtipoent) AND
  (public.especieave.codstatusgeral = public.statusgeral.codstatusgeral) AND
  (public.projeto.codstatusproj = public.statusprojeto.codstatusproj)

```

```

CREATE OR REPLACE VIEW "public"."vProjetoIncubacaoNascimento" (
  codproj,
  siglaproj,
  nomeproj,
  datainiproj,
  dataprevproj,
  datafimproj,
  descproj,
  numinc,
  codesp,
  codstatusgeral,
  nomestatusgeral,
  nomeesp,
  tempmininc,
  tempmaxinc,
  umimininc,
  umimaxinc,
  perinc,
  peraviinc,
  tempminnas,
  tempmaxnas,
  umiminnas,
  umimaxnas,
  pernas,
  peravinas,
  perrec,
  codtipoent,
  nometipoent,
  tipoentmin,
  codstatusproj,
  nomestatusproj,
  codinc,
  datainiinc,
  dataprevinc,

```

```

datafiminc,
numovoinii,
numovodesi,
numovotransf,
obsstatusinc,
codnas,
datainiinas,
dataprevnas,
datafimnas,
numovoinin,
numovodesn,
numovorec,
obsstatusnas,
codstatusin,
nomestatusin)
AS
SELECT
projeto.codproj,
projeto.siglaproj,
projeto.nomeproj,
projeto.datainiproj,
projeto.dataprevproj,
projeto.datafimproj,
projeto.descproj,
projeto.numinc,
especieave.codesp,
statusgeral.codstatusgeral,
statusgeral.nomestatusgeral,
especieave.nomeesp,
especieave.tempmininc,
especieave.tempmaxinc,
especieave.umimininc,
especieave.umimaxinc,
especieave.perinc,
especieave.peraviinc,
especieave.tempminnas,
especieave.tempmaxnas,
especieave.umiminnas,
especieave.umimaxnas,
especieave.pernas,
especieave.peravinas,
especieave.perrec,
tipoentidade.codtipoent,
tipoentidade.nometipoent,
tipoentidade.tipoentmin,
statusprojeto.codstatusproj,
statusprojeto.nomestatusproj,
incubacao.codinc,
incubacao.datainiinc,
incubacao.dataprevinc,
incubacao.datafiminc,

```

```

incubacao.numovoinii,
incubacao.numovodesi,
incubacao.numovotransf,
incubacao.obsstatusinc,
nascimento.codnas,
nascimento.datainiinas,
nascimento.dataprevnas,
nascimento.datafimnas,
nascimento.numovoinin,
nascimento.numovodesn,
nascimento.numovorec,
nascimento.obsstatusnas,
statusincnas.codstatusin,
statusincnas.nomestatusin
FROM
public.projeto
INNER JOIN public.incubacao ON (public.projeto.codproj=public.incubacao.codproj)
INNER JOIN public.nascimento ON (public.incubacao.codinc=public.nascimento.codinc)
INNER JOIN public.statusprojeto ON
(public.projeto.codstatusproj=public.statusprojeto.codstatusproj)
INNER JOIN public.tipoentidade ON (public.projeto.codtipoent=public.tipoentidade.codtipoent)
INNER JOIN public.especieave ON (public.projeto.codesp=public.especieave.codesp)
INNER JOIN public.statusgeral ON
(public.especieave.codstatusgeral=public.statusgeral.codstatusgeral)
INNER JOIN public.statusincnas ON
(public.nascimento.codstatusin=public.statusincnas.codstatusin)
WHERE
(public.projeto.codproj = public.incubacao.codproj) AND
(public.projeto.codesp = public.especieave.codesp) AND
(public.projeto.codtipoent = public.tipoentidade.codtipoent) AND
(public.projeto.codstatusproj = public.statusprojeto.codstatusproj) AND
(public.incubacao.codinc = public.nascimento.codinc) AND
(public.especieave.codstatusgeral = public.statusgeral.codstatusgeral) AND
(public.nascimento.codstatusin = public.statusincnas.codstatusin)

CREATE OR REPLACE VIEW "public"."vProjetoIncubacaoIncubadoraBandeja" (
codproj,
codesp,
siglaproj,
nomeproj,
datainiproj,
dataprevproj,
datafimproj,
descproj,
numinc,
nomeesp,
tempmininc,
tempmaxinc,
umimininc,
umimaxinc,

```

```
perinc,  
peraviinc,  
tempminnas,  
tempmaxnas,  
umiminnas,  
umimaxnas,  
pernas,  
peravinas,  
perrec,  
nometipoent,  
tipoentmin,  
codstatusproj,  
nomestatusproj,  
codinc,  
datainiinc,  
dataprevinc,  
datafiminc,  
numovoinii,  
numovodesi,  
numovotransf,  
obsstatusinc,  
codstatusin,  
nomestatusin,  
nlinban,  
ncolban,  
ovotransf,  
codmotdes,  
motdes,  
codposbanenc,  
nposenc,  
dataaloenc,  
dataretenc,  
codenc,  
nomeenc,  
tposenc,  
descenc,  
codban,  
tlinban,  
tcolban,  
nomeban,  
codent,  
codtipoent,  
nomeent)
```

AS

SELECT

```
projeto.codproj,  
especieave.codesp,  
projeto.siglaproj,  
projeto.nomeproj,  
projeto.datainiproj,  
projeto.dataprevproj,
```

projeto.datafimproj,
projeto.descproj,
projeto.numinc,
especieave.nomeesp,
especieave.tempmininc,
especieave.tempmaxinc,
especieave.umimininc,
especieave.umimaxinc,
especieave.perinc,
especieave.peraviinc,
especieave.tempminnas,
especieave.tempmaxnas,
especieave.umiminnas,
especieave.umimaxnas,
especieave.pernas,
especieave.peravinas,
especieave.perrec,
tipoentidade.nometipoent,
tipoentidade.tipoentmin,
statusprojeto.codstatusproj,
statusprojeto.nomestatusproj,
incubacao.codinc,
incubacao.datainiinc,
incubacao.dataprevinc,
incubacao.datafiminc,
incubacao.numovoinii,
incubacao.numovodesi,
incubacao.numovotransf,
incubacao.obsstatusinc,
statusincnas.codstatusin,
statusincnas.nomestatusin,
inctement.nlinban,
inctement.ncolban,
inctement.ovotransf,
motivodescarte.codmotdes,
motivodescarte.motdes,
posbanenc.codposbanenc,
posbanenc.nposenc,
posbanenc.dataaloenc,
posbanenc.dataretenc,
incubadora.codenc,
incubadora.nomeenc,
incubadora.tposenc,
incubadora.descenc,
bandeja.codban,
bandeja.tlinban,
bandeja.tcolban,
bandeja.nomeban,
entidade.codent,
entidade.codtipoent,
entidade.nomeent

```

FROM
  public.posbanenc
  INNER JOIN public.incubadora ON (public.posbanenc.codenc=public.incubadora.codenc)
  INNER JOIN public.bandeja ON (public.posbanenc.codban=public.bandeja.codban)
  INNER JOIN public.inctement ON (public.posbanenc.codposbanenc=public.inctement.codposbanenc)
  INNER JOIN public.entidade ON (public.inctement.codent=public.entidade.codent)
  INNER JOIN public.motivodescarte ON
(public.inctement.codmotdes=public.motivodescarte.codmotdes)
  INNER JOIN public."vEntidadeSala" ON
(public.incubadora.codsala=public."vEntidadeSala".codent)
  INNER JOIN public.incubacao ON (public.inctement.codinc=public.incubacao.codinc)
  INNER JOIN public.projeto ON (public.incubacao.codproj=public.projeto.codproj)
  INNER JOIN public.tipoentidade ON (public.projeto.codtipoent=public.tipoentidade.codtipoent)
  INNER JOIN public.especieave ON (public.projeto.codesp=public.especieave.codesp)
  INNER JOIN public.statusprojeto ON
(public.projeto.codstatusproj=public.statusprojeto.codstatusproj)
  INNER JOIN public.statusincnas ON
(public.statusincnas.codstatusin=public.incubacao.codstatusin)
WHERE
  (public.inctement.codinc = public.incubacao.codinc) AND
  (public.incubacao.codstatusin = public.statusincnas.codstatusin) AND
  (public.inctement.codmotdes = public.motivodescarte.codmotdes) AND
  (public.inctement.codposbanenc = public.posbanenc.codposbanenc) AND
  (public.inctement.codent = public.entidade.codent) AND
  (public.bandeja.codban = public.posbanenc.codban) AND
  (public.posbanenc.codenc = public.incubadora.codenc) AND
  (public.incubadora.codsala = public."vEntidadeSala".codent) AND
  (public.incubacao.codproj = public.projeto.codproj) AND
  (public.projeto.codtipoent = public.tipoentidade.codtipoent) AND
  (public.projeto.codesp = public.especieave.codesp) AND
  (public.projeto.codstatusproj = public.statusprojeto.codstatusproj)

CREATE OR REPLACE VIEW "public"."vBandejaIncubadoraPosicao" (
  codposbanenc,
  codstatusgeral,
  nomestatusgeral,
  codenc,
  nomeenc,
  tposenc,
  descenc,
  codent,
  codtipoent,
  nomeent,
  codban,
  tlinban,
  tcolban,
  nomeban,
  nposenc,
  dataaloenc,
  dataretenc)

```

```

AS
SELECT
    posbanenc.codposbanenc,
    statusgeral.codstatusgeral,
    statusgeral.nomestatusgeral,
    incubadora.codenc,
    incubadora.nomeenc,
    incubadora.tposenc,
    incubadora.descenc,
    entidade.codent,
    entidade.codtipoent,
    entidade.nomeent,
    bandeja.codban,
    bandeja.tlinban,
    bandeja.tcolban,
    bandeja.nomeban,
    posbanenc.nposenc,
    posbanenc.dataaloenc,
    posbanenc.dataaretenc
FROM
    public.bandeja
    INNER JOIN public.posbanenc ON (public.bandeja.codban=public.posbanenc.codban)
    INNER JOIN public.incubadora ON (public.posbanenc.codenc=public.incubadora.codenc)
    INNER JOIN public.statusgeral ON (public.posbanenc.codstatusgeral=public.statusgeral.codstatusgeral)
    INNER JOIN public.entidade ON (public.incubadora.codsala=public.entidade.codent)
WHERE
    (public.posbanenc.codenc = public.incubadora.codenc) AND
    (public.posbanenc.codban = public.bandeja.codban) AND
    (public.posbanenc.codstatusgeral = public.statusgeral.codstatusgeral) AND
    (public.incubadora.codsala = public.entidade.codent)

CREATE OR REPLACE VIEW "public"."vOvoProjetoIncubacaoNascimentoBandeja" (
    codovo,
    codbarras,
    aquisovo,
    datacolovo,
    aspovo,
    pesoovo,
    porovo,
    incubavel,
    coddesovo,
    nomedesovo,
    codstatusovo,
    nomestatusovo,
    codforovo,
    nomeforovo,
    codcorovo,
    nomecorovo,
    codcel,

```

nomeent,
codproj,
codesp,
nomeesp,
codtipoent,
nometipoent,
codstatusproj,
nomestatusproj,
siglaproj,
nomeproj,
datainiproj,
dataprevproj,
datafimproj,
descproj,
numinc,
codinc,
codstatusin,
nomestatusin,
datainiinc,
dataprevinc,
datafiminc,
numovoinii,
numovodesi,
numovotransf,
obsstatusinc,
nlinbanenc,
ncolbanenc,
codposbanenc,
codmotdesenc,
motdes,
ovotransf,
codenc,
nomeenc,
nposenc,
dataaloenc,
dataretenc,
tposenc,
codnas,
datainiinas,
dataprevnas,
datafimnas,
numovoinin,
numovodesn,
numovorec,
obsstatusnas,
nlinban,
ncolban,
codstatusgeral,
codmotdes,
ovonasceu,
codposbannes,

```

codnes,
nomenes,
tcolnes,
codban,
tlinban,
tcolban,
nomeban,
nposnes,
dataalones,
dataretnes)
AS
SELECT
ovo.codovo,
ovo.codbarras,
ovo.aquisovo,
ovo.datacolovo,
ovo.aspovo,
ovo.pesoovo,
ovo.porovo,
ovo.incubavel,
desovo.coddesovo,
desovo.nomedesovo,
statusovo.codstatusovo,
statusovo.nomestatusovo,
forovo.codforovo,
forovo.nomeforovo,
corovo.codcorovo,
corovo.nomecorovo,
ovo.codcel,
entidade.nomeent,
projeto.codproj,
especieave.codesp,
especieave.nomeesp,
tipoentidade.codtipoent,
tipoentidade.nometipoent,
statusprojeto.codstatusproj,
statusprojeto.nomestatusproj,
projeto.siglaproj,
projeto.nomeproj,
projeto.datainiproj,
projeto.dataprevproj,
projeto.datafimproj,
projeto.descproj,
projeto.numinc,
incubacao.codinc,
statusincnas.codstatusin,
statusincnas.nomestatusin,
incubacao.datainiinc,
incubacao.dataprevinc,
incubacao.datafiminc,
incubacao.numovoinii,

```

```

incubacao.numovodesi,
incubacao.numovotransf,
incubacao.obsstatusinc,
inctement.nlinban,
inctement.ncolban,
posbanenc.codposbanenc,
motivodescarte.codmotdes,
motivodescarte.motdes,
inctement.ovotransf,
incubadora.codenc,
incubadora.nomeenc,
posbanenc.nposenc,
posbanenc.dataaloenc,
posbanenc.dataaretenc,
incubadora.tposenc,
nascimento.codnas,
nascimento.datainiinas,
nascimento.dataprevnas,
nascimento.datafimnas,
nascimento.numovoinin,
nascimento.numovodesn,
nascimento.numovorec,
nascimento.obsstatusnas,
nastement.nlinban,
nastement.ncolban,
nastement.codstatusgeral,
nastement.codmotdes,
nastement.ovonasceu,
posbannes.codposbannes,
nascedoruro.codnes,
nascedoruro.nomenes,
nascedoruro.tcolnes,
bandeja.codban,
bandeja.tlinban,
bandeja.tcolban,
bandeja.nomeban,
posbannes.nposnes,
posbannes.dataalones,
posbannes.dataretnes
FROM
public.ovo
INNER JOIN public.desovo ON (public.ovo.coddesovo=public.desovo.coddesovo)
INNER JOIN public.statusovo ON (public.ovo.codstatusovo=public.statusovo.codstatusovo)
INNER JOIN public.forovo ON (public.ovo.codforovo=public.forovo.codforovo)
INNER JOIN public.corovo ON (public.ovo.codcorovo=public.corovo.codcorovo)
INNER JOIN public.inctement ON (public.ovo.codovo=public.inctement.codent)
INNER JOIN public.nastement ON (public.ovo.codovo=public.nastement.codent)
INNER JOIN public.entidade ON (public.ovo.codcel=public.entidade.codent)
INNER JOIN public.posbannes ON (public.posbannes.codposbannes=public.nastement.codposbannes)
INNER JOIN public.bandeja ON (public.bandeja.codban=public.posbannes.codban)
INNER JOIN public.nascedoruro ON (public.posbannes.codnes=public.nascedoruro.codnes)

```

```

INNER JOIN public.posbanenc ON (public.posbanenc.codposbanenc=public.inctement.codposbanenc)
INNER JOIN public.incubadora ON (public.posbanenc.codenc=public.incubadora.codenc)
INNER JOIN public.incubacao ON (public.incubacao.codinc=public.inctement.codinc)
INNER JOIN public.projeto ON (public.projeto.codproj=public.incubacao.codproj)
INNER JOIN public.statusincnas ON
(public.incubacao.codstatusin=public.statusincnas.codstatusin)
INNER JOIN public.especieave ON (public.projeto.codesp=public.especieave.codesp)
INNER JOIN public.tipoentidade ON (public.projeto.codtipoent=public.tipoentidade.codtipoent)
INNER JOIN public.motivodescarte ON
(public.inctement.codmotdes=public.motivodescarte.codmotdes)
INNER JOIN public.nascimento ON (public.incubacao.codinc=public.nascimento.codinc)
AND (public.nascimento.codnas=public.nastement.codnas)
INNER JOIN public.statusprojeto ON
(public.projeto.codstatusproj=public.statusprojeto.codstatusproj)
WHERE
(public.ovo.coddesovo = public.desovo.coddesovo) AND
(public.ovo.codstatusovo = public.statusovo.codstatusovo) AND
(public.ovo.codforovo = public.forovo.codforovo) AND
(public.ovo.codcorovo = public.corovo.codcorovo) AND
(public.ovo.codcel = public.entidade.codent) AND
(public.projeto.codesp = public.especieave.codesp) AND
(public.projeto.codtipoent = public.tipoentidade.codtipoent) AND
(public.projeto.codstatusproj = public.statusprojeto.codstatusproj) AND
(public.projeto.codproj = public.incubacao.codproj) AND
(public.incubacao.codinc = public.inctement.codinc) AND
(public.ovo.codovo = public.inctement.codinc) AND
(public.inctement.codmotdes = public.motivodescarte.codmotdes) AND
(public.inctement.codposbanenc = public.posbanenc.codposbanenc) AND
(public.posbanenc.codenc = public.incubadora.codenc) AND
(public.incubacao.codstatusin = public.statusincnas.codstatusin) AND
(public.ovo.codovo = public.nastement.codent) AND
(public.nastement.codnas = public.nascimento.codnas) AND
(public.nascimento.codinc = public.incubacao.codinc) AND
(public.nastement.codposbannes = public.posbannes.codposbannes) AND
(public.posbannes.codnes = public.nascedouro.codnes) AND
(public.posbannes.codban = public.bandeja.codban)

```

```

CREATE VIEW "public"."vBandejaNascedouroPosicao" (
    codposbannes,
    codstatusgeral,
    nomestatusgeral,
    codnes,
    nomenes,
    tcolnes,
    descnes,
    codent,
    codtipoent,
    nomeent,
    codban,
    tlinban,

```

```

        tcolban,
        nomeban,
        nposnes,
        dataalones,
        dataretnes)
AS
SELECT posbannes.codposbannes, statusgeral.codstatusgeral,
       statusgeral.nomestatusgeral, nascedouro.codnes, nascedouro.nomenes,
       nascedouro.tcolnes, nascedouro.descnes, entidade.codent,
       entidade.codtipoent, entidade.nomeent, bandeja.codban, bandeja.tlinban,
       bandeja.tcolban, bandeja.nomeban, posbannes.nposnes, posbannes.dataalones,
       posbannes.dataretnes
FROM (((bandeja JOIN posbannes ON ((bandeja.codban = posbannes.codban))) JOIN
      nascedouro ON ((posbannes.codnes = nascedouro.codnes))) JOIN statusgeral ON
      ((posbannes.codstatusgeral = statusgeral.codstatusgeral))) JOIN entidade ON
      ((nascedouro.codsala = entidade.codent)))
WHERE (((posbannes.codnes = nascedouro.codnes) AND (posbannes.codban =
       bandeja.codban)) AND (posbannes.codstatusgeral =
       statusgeral.codstatusgeral)) AND (nascedouro.codsala = entidade.codent));

CREATE VIEW "public"."vOvoIncubacaoNascimento" (
       codovo,
       codbarras,
       coddeseovo,
       nomestatusovo,
       nomeforovo,
       nomecorovo,
       nomeent_cel,
       aquisovo,
       datacolovo,
       aspovo,
       pesoovo,
       porovo,
       incubavel,
       nlinban_inc,
       ncolban_inc,
       codmotdes_inc,
       ovotransf,
       codproj,
       siglaproj,
       nomeproj,
       nomeenc,
       datainiinc,
       nomeesp,
       datainiinas,
       datafimnas,
       obsstatusnas,
       nlinban_nas,
       ncolban_nas,
       nomenes,

```

```

codmotdes_nas)
AS
SELECT
ovo.codovo,
ovo.codbarras,
ovo.coddesovo,
statusovo.nomestatusovo,
forovo.nomeforovo,
corovo.nomecorovo,
"vEntidadeCelula".nomeent AS nomeent_cel,
ovo.aquisovo,
ovo.datacolovo,
ovo.aspovo,
ovo.pesoovo,
ovo.porovo,
ovo.incubavel,
inctement.nlinban AS nlinban_inc,
inctement.ncolban AS ncolban_inc,
inctement.codmotdes AS codmotdes_inc,
inctement.ovotransf,
projeto.codproj,
projeto.siglaproj,
projeto.nomeproj,
incubadora.nomeenc,
incubacao.datainiinc,
especieave.nomeesp,
nascimento.datainiinas,
nascimento.datafimnas,
nascimento.obsstatusnas,
nastement.nlinban AS nlinban_nas,
nastement.ncolban AS ncolban_nas,
nascvoudouro.nomenes,
nastement.codmotdes AS codmotdes_nas
FROM
public.ovo
INNER JOIN public.entidade ON (public.ovo.codovo=public.entidade.codent)
INNER JOIN public.inctement ON (public.ovo.codovo=public.inctement.codent)
INNER JOIN public.nastement ON (public.ovo.codovo=public.nastement.codent)
INNER JOIN public.statusovo ON (public.ovo.codstatusovo=public.statusovo.codstatusovo)
INNER JOIN public.forovo ON (public.ovo.codforovo=public.forovo.codforovo)
INNER JOIN public.corovo ON (public.ovo.codcorovo=public.corovo.codcorovo)
INNER JOIN public."vEntidadeCelula" ON (public.ovo.codcel=public."vEntidadeCelula".codent)
INNER JOIN public.posbanenc ON (public.inctement.codposbanenc=public.posbanenc.codposbanenc)
INNER JOIN public.posbannes ON (public.nastement.codposbannes=public.posbannes.codposbannes)
INNER JOIN public.nascvoudouro ON (public.posbannes.codnes=public.nascvoudouro.codnes)
INNER JOIN public.incubadora ON (public.posbanenc.codenc=public.incubadora.codenc)
INNER JOIN public.nascimento ON (public.nastement.codnas=public.nascimento.codnas)
INNER JOIN public.incubacao ON (public.incubacao.codinc=public.inctement.codinc)
AND (public.nascimento.codinc=public.incubacao.codinc)
INNER JOIN public.projeto ON (public.incubacao.codproj=public.projeto.codproj)
INNER JOIN public.especieave ON (public.projeto.codesp=public.especieave.codesp)

```

WHERE

```
(public.ovo.codovo = public.entidade.codent) AND  
(public.ovo.codovo = public.inctement.codent) AND  
(public.ovo.codovo = public.nastement.codent) AND  
(public.inctement.codinc = public.incubacao.codinc) AND  
(public.incubacao.codproj = public.projeto.codproj) AND  
(public.projeto.codesp = public.especieave.codesp) AND  
(public.incubacao.codinc = public.nascimento.codinc) AND  
(public.nastement.codnas = public.nascimento.codnas) AND  
(public.ovo.codstatusovo = public.statusovo.codstatusovo) AND  
(public.ovo.codforovo = public.forovo.codforovo) AND  
(public.ovo.codcorovo = public.corovo.codcorovo) AND  
(public.ovo.codcel = public."vEntidadeCelula".codent) AND  
(public.inctement.codposbanenc = public.posbanenc.codposbanenc) AND  
(public.posbanenc.codenc = public.incubadora.codenc) AND  
(public.posbannes.codposbannes = public.nastement.codposbannes) AND  
(public.posbannes.codnes = public.nascedouro.codnes)
```

12.1.6 Tela de Configuração

FIGURA 90 – TELA DO EMS POSTGRESQL QUANDO DA CRIAÇÃO DO BANCO DE DADOS

The image shows a 'Register Database' dialog box with the following fields and values:

Field	Value
Host name	localhost
Port	5432
User name	postgres
Password	xxxxxxxx
Database name	sisinc
Database alias	sisinc on localhost
Client encoding	LATIN1
Font charset	DEFAULT_CHARSET

Buttons at the bottom: Test Connect, Copy Alias from..., OK, Cancel, Help.

12.2 PACOTES

12.2.1 broker

12.2.1.1 BrokerConexao

```
package broker;

import java.sql.*;

/**
 * Cria e fecha conexões com o banco de dados.
 * @author Arthur Maçaneiro Magyar
 * @author Fábio Glaucos dos Santos
 * @author Luciano Padilha
 * @version 1.0
 * @see Connection
 */
public class BrokerConexao
{
    // declaração de variáveis para Connection
    private String hostName = "localhost";
    private String userName = "postgres";
    private String password = "sisinc";
    private String jdbcDriver = "org.postgresql.Driver";
    private String dataBaseName = "sisinc";
    private String dataBasePrefix = "jdbc:postgresql://";
    private String dataBasePort = "5432";
    private String url = dataBasePrefix + hostName + ":" + dataBasePort + "/" + dataBaseName +

"/";

    public Connection con = null;

    /**
     * Cria a conexão com o banco de dados.
     * @return url, userName, password
     */
    public Connection criaConexao()
    {
        try
        {
            if (con == null)
            {
                Class.forName(jdbcDriver);
                con = DriverManager.getConnection(url, userName, password);
                System.out.println("Conectou: "+url);
            }
            else if (con.isClosed())
            {
                con = null;
                return DriverManager.getConnection(url, userName, password);
            }
        }
        catch (ClassNotFoundException e)
        {
            e.printStackTrace();
        }
        catch (SQLException e)
        {
            e.printStackTrace();
        }
        return con;
    }

    /**
     * Fecha a conexão com o banco de dados.

```

```

    */
    public void fechaConexao()
    {
        try
        {
            con.close();
            System.out.println("Desconectou");
        }
        catch(SQLException ex)
        {
            ex.printStackTrace();
        }
    }
}

```

12.2.1.2 BrokerConsulta

```

package broker;

import java.sql.*;
import java.util.Vector;
import javaUtil.TransferData;

/**
 *
 * @author Arthur Maçaneiro Magyar
 * @author Fábio Glaucos dos Santos
 * @author Luciano Padilha
 * @version 1.0
 */
public class BrokerConsulta extends BrokerConexao{
    private Connection con;
    private Statement stmt;
    private Vector vetor,vetor2;
    private ResultSet rs;
    private String aux,retorna,entMin,nomeEsp;
    private int retornaData,ret;

    /**
     * RETORNA BANDEJA
     * @return vetor com as bandejas
     */
    public Vector retornaBandeja()
    {
        try
        {
            vetor=new Vector();
            String x="";
            con=criaConexao();
            stmt=con.createStatement();
            aux="SELECT * FROM ";
            aux+="vBandejaIncubadoraPosicao"+" "+" ";
            aux+="ORDER BY CodBan;";
            rs=stmt.executeQuery(aux);
            while(rs.next())
            {
                x=rs.getString("CodBan");
                x+=" - ";
                x+=rs.getString("NomeBan");
                vetor.addElement(x);
            }
            rs.close();
            con.close();
            stmt.close();
        }
        catch(SQLException e)
        {
            System.out.println(e.getMessage());
            e.printStackTrace();
            System.out.println("SQL Exception no retornaBandeja -> "+aux);
        }
        catch(Exception e)

```

```

        {
            System.out.println(e.getMessage());
            e.printStackTrace();
            System.out.println("Saiu do Método retornaBandeja.");
        }
        return vetor;
    }

/**
 * RETORNA BANDEJA ALOCADA EM INCUBADORA
 * @return vetor com bandejas não retiradas de incubadora
 */
public Vector retornaBandejaAlocadaI()
{
    try
    {
        vetor = new Vector();
        String x="";
        con =criaConexao();
        stmt=con.createStatement();
        aux="SELECT Bandeja.CodBan,Bandeja.NomeBan FROM ";
        aux+="Bandeja INNER JOIN PosBanEnc ON ";
        aux+="Bandeja.CodBan=PosBanEnc.CodBan ";
        aux+="WHERE PosBanEnc.DataRetEnc IS NULL;";
        rs=stmt.executeQuery(aux);
        while(rs.next())
        {
            x=rs.getString("CodBan");
            x+=" - ";
            x+=rs.getString("NomeBan");
            vetor.addElement(x);
        }
        rs.close();
        stmt.close();
        con.close();
    }
    catch(SQLException e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("SQL Exception no retornaBandejaAlocadaI -> "+aux);
    }
    catch(Exception e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("Saiu do Método retornaBandejaAlocadaI.");
    }
    return vetor;
}

/**
 * RETORNA BANDEJA ALOCADA EM NASCEDOURO
 * @return vetor com bandejas não retiradas de nascedouro
 */
public Vector retornaBandejaAlocadaN()
{
    try
    {
        vetor = new Vector();
        String x="";
        con = criaConexao();
        stmt=con.createStatement();
        aux="SELECT Bandeja.CodBan,Bandeja.NomeBan FROM ";
        aux+="Bandeja INNER JOIN PosBanNes ON ";
        aux+="Bandeja.CodBan=PosBanNes.CodBan ";
        aux+="WHERE PosBanNes.DataRetNes IS NULL;";
        rs=stmt.executeQuery(aux);
        while(rs.next())
        {
            x=rs.getString("CodBan");
            x+=" - ";
            x+=rs.getString("NomeBan");
            vetor.addElement(x);
        }
        rs.close();
    }
}

```

```

        stmt.close();
        con.close();
    }
    catch(SQLException e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("SQL Exception no retornaBandejaAlocadaN -> "+aux);
    }
    catch(Exception e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("Saiu do Método retornaBandejaAlocadaN.");
    }
    return vetor;
}

/**
 * RETORNA BANDEJA DESALOCADA
 * @return vetor com as bandejas desalocadas
 */
public Vector retornaBandejaDesalocada()
{
    try
    {
        vetor=new Vector();
        String x="";
        con=criaConexao();
        stmt=con.createStatement();
        // pesquisa bandeja retirada de incubadora
        aux="SELECT Bandeja.CodBan,Bandeja.NomeBan FROM ";
        aux+="Bandeja INNER JOIN PosBanEnc ON ";
        aux+="Bandeja.CodBan=PosBanEnc.CodBan WHERE ";
        aux+="DataRetEnc>DataAloEnc;";
        rs=stmt.executeQuery(aux);
        while(rs.next())
        {
            x=rs.getString("CodBan");
            x+=" - ";
            x+=rs.getString("NomeBan");
            vetor.addElement(x);
        }
        rs.close();
        // pesquisa bandeja retirada de nascedouro
        aux="SELECT bandeja.CodBan,bandeja.NomeBan FROM ";
        aux+="Bandeja INNER JOIN PosBanNes ON ";
        aux+="Bandeja.CodBan=PosBanNes.CodBan WHERE ";
        aux+="DataRetNes>DataAloNes;";
        rs=stmt.executeQuery(aux);
        while(rs.next())
        {
            x=rs.getString("CodBan");
            x+=" - ";
            x+=rs.getString("NomeBan");
            vetor.addElement(x);
        }
        rs.close();
        // pesquisa bandeja não alocada em incubadora e nascedouro
        aux="SELECT DISTINCT bandeja.CodBan,bandeja.NomeBan FROM ";
        aux+="Bandeja WHERE ";
        aux+="CodBan NOT IN(";
        aux+="SELECT CodBan FROM PosBanEnc) AND ";
        aux+="CodBan NOT IN(";
        aux+="SELECT CodBan FROM PosBanNes) ";
        aux+="ORDER BY CodBan;";
        rs=stmt.executeQuery(aux);
        while(rs.next())
        {
            x=rs.getString("CodBan");
            x+=" - ";
            x+=rs.getString("NomeBan");
            vetor.addElement(x);
        }
        rs.close();
        stmt.close();
    }
}

```

```

        con.close();
    }
    catch(SQLException e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("SQL Exception no retornaBandejaDesalocada -> "+aux);
    }
    catch(Exception e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("Saiu do Método retornaBandejaDesalocada.");
    }
    return vetor;
}

/**
 * RETORNA BANDEJA EM POSIÇÃO DE NASCEDOURO
 * @return vetor com as bandejas em nascedouro
 */
public Vector retornaBandejaNas ()
{
    try
    {
        vetor=new Vector();
        String x="";
        con=criaConexao();
        stmt=con.createStatement();
        aux="SELECT * FROM ";
        aux+="'+vBandejaNascedouroPosicao+'+'+' ";
        aux+="ORDER BY CodBan;";
        rs=stmt.executeQuery(aux);
        while(rs.next())
        {
            x=rs.getString("CodBan");
            x+=" - ";
            x+=rs.getString("NomeBan");
            vetor.addElement(x);
        }
        rs.close();
        con.close();
        stmt.close();
    }
    catch(SQLException e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("SQL Exception no retornaBandejaNas -> "+aux);
    }
    catch(Exception e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("Saiu do Método retornaBandejaNas.");
    }
    return vetor;
}

/**
 * RETORNA CÉLULA
 * @return vetor com as células ativas
 */
public Vector retornaCelula()
{
    try
    {
        vetor=new Vector();
        String x="";
        con=criaConexao();
        stmt=con.createStatement();
        aux="SELECT * FROM ";
        aux+="Celula INNER JOIN Entidade ";
        aux+="ON Entidade.CodEnt=Celula.CodCel ";
        aux+="WHERE CodStatusGeral=1 ";
        aux+="ORDER BY NomeEnt;";
    }
}

```

```

        rs=stmt.executeQuery(aux);
        while(rs.next())
        {
            x=rs.getString("CodCel");
            x+=" - ";
            x+=rs.getString("NomeEnt");
            vetor.addElement(x);
        }
        rs.close();
        con.close();
        stmt.close();
    }
    catch(SQLException e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("SQL Exception no retornaCelula -> "+aux);
    }
    catch(Exception e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("Saiu do Método retornaCelula.");
    }
    return vetor;
}

/**
 * RETORNA DADOS DO NASCIMENTO COM BANDEJA
 * @param nascimento código do nascimento
 * @return vetor com bandeja, nascedouro e entidade
 */
public Vector retornaDadosNas2(String nascimento)
{
    String x="";
    int i=0;
    try
    {
        vetor = new Vector();
        vetor2 = new Vector();
        con = criaConexao();
        stmt=con.createStatement();
        aux="SELECT DISTINCT ";
        aux+="PosBanNes.CodBan,";
        aux+="PosBanNes.CodNes,";
        aux+="NasTemEnt.CodEnt FROM ";
        aux+="PosBanNes JOIN NasTemEnt ON ";
        aux+="PosBanNes.CodPosBanNes=NasTemEnt.CodPosBanNes ";
        aux+="WHERE NasTemEnt.CodNas="+nascimento+"";
        rs=stmt.executeQuery(aux);
        while(rs.next())
        {
            x=rs.getString("CodBan");
            x+="-";
            x+=rs.getString("CodNes");
            x+=":";
            x+=rs.getString("CodEnt");
            vetor.add(x);
        }
        rs.close();
        stmt.close();

        for(i = 0; i < vetor.size(); i++)
        {
            stmt=con.createStatement();
            String aux1=(String)vetor.elementAt(i);
            String novo=aux1.substring(aux1.indexOf(":")+1);
            aux="SELECT NomeEnt FROM ";
            aux+="Entidade WHERE CodEnt="+novo+"";
            rs=stmt.executeQuery(aux);
            rs.next();
            vetor2.addElement(aux1+"-"+rs.getString("NomeEnt"));
            rs.close();
            stmt.close();
        }
    }
    con.close();
}

```

```

    }
    catch(SQLException e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("SQL Exception no retornaDadosNas2 -> "+aux);
    }
    catch(Exception e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("Saiu do Método retornaDadosNas2.");
    }
    return vetor2;
}

/**
 * RETORNA DADOS DO NASCIMENTO COM INCUBAÇÃO
 * @param nascimento código do nascimento
 * @return incubação e data de término do nascimento
 */
public String retornaDadosNas (String nascimento)
{
    TransferData dt = new TransferData();
    String x="";
    try
    {
        con = criaConexao();
        stmt=con.createStatement();
        aux="SELECT * FROM ";
        aux+="Nascimento WHERE ";
        aux+="CodStatusIN=1 AND ";
        aux+="CodNas="+nascimento+";";
        rs=stmt.executeQuery(aux);
        while(rs.next())
        {
            x=rs.getString("CodInc");
            x+="-";
            x+=dt.DataPApresentar(rs.getString("DataPrevNas"));
        }
        rs.close();
        con.close();
        stmt.close();
        int index=x.indexOf("-");
        String aux1=x.substring(0,index-1);
        con = criaConexao();
        stmt=con.createStatement();
        aux="SELECT * FROM ";
        aux+="Incubacao WHERE ";
        aux+="CodStatusIN=1 AND ";
        aux+="CodInc="+aux1+";";
        rs=stmt.executeQuery(aux);
        while(rs.next())
        {
            x+="|";
            x+=rs.getString("DataIniInc");
        }
        rs.close();
        con.close();
        stmt.close();
    }
    catch(SQLException e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("SQL Exception no retornaDadosNas -> "+aux);
    }
    catch(Exception e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("Saiu do Método retornaDadosNas.");
    }
    return x;
}

```

```

/**
 * RETORNA DADOS PARA PÁGINA DE ENTRADA
 * @return retorna com projetos, incubações e nascimentos ativos
 */
public String retornaDadosGerais()
{
    try
    {
        vetor = new Vector();
        String x="";
        con = criaConexao();
        stmt=con.createStatement();
        // pesquisa todos os projetos
        aux="SELECT COUNT(*) AS ";
        aux+="Projetos FROM Projeto;";
        rs=stmt.executeQuery(aux);
        rs.next();
        x=rs.getString("Projetos");
        x+="-";
        rs.close();
        // pesquisa todos os projetos ativos
        aux="SELECT COUNT(*) AS ";
        aux+="Projetos FROM Projeto ";
        aux+="WHERE CodStatusProj=1;";
        rs=stmt.executeQuery(aux);
        rs.next();
        x+=rs.getString("Projetos");
        x+=":";
        rs.close();
        // pesquisa todos as incubações ativas
        aux="SELECT COUNT(*) AS ";
        aux+="Incubacoes FROM Incubacao ";
        aux+="WHERE CodStatusIN=1;";
        rs=stmt.executeQuery(aux);
        rs.next();
        x+=rs.getString("Incubacoes");
        x+="-";
        rs.close();
        // pesquisa todos os nascimentos ativos
        aux="SELECT COUNT(*) AS ";
        aux+="Nascimentos FROM Nascimento ";
        aux+="WHERE CodStatusIN=1;";
        rs = stmt.executeQuery(aux);
        rs.next();
        x+=rs.getString("Nascimentos");
        rs.close();
        System.out.println(x);
        retorna=x;
        rs.close();
        con.close();
        stmt.close();
    }
    catch(SQLException e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("SQL Exception no retornaDadosGerais -> "+aux);
    }
    catch(Exception e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("Saiu do Método retornaDadosGerais.");
    }
    return retorna;
}

/**
 * RETORNA DATA DE INÍCIO DE INCUBAÇÃO
 * @param inc código da incubação
 * @return data de inicio de uma incubação
 */
public String retornaDataIniInc(String inc)
{
    TransferData dt = new TransferData();
    String x="";

```

```

try
{
    con = criaConexao();
    stmt=con.createStatement();
    aux="SELECT DataIniInc FROM ";
    aux+="Incubacao WHERE CodInc="+inc+"";
    rs=stmt.executeQuery(aux);
    while(rs.next())
    {
        x=rs.getString("DataIniInc");
    }
    rs.close();
    con.close();
    stmt.close();
    x=dt.DataPAPresentar(x);
}
catch(SQLException e)
{
    System.out.println(e.getMessage());
    e.printStackTrace();
    System.out.println("SQL Exception no retornaDataIniInc -> "+aux);
}
catch(Exception e)
{
    System.out.println(e.getMessage());
    e.printStackTrace();
    System.out.println("Saiu do Método retornaDataIniInc.");
}
return x;
}

/**
 * RETORNA DATA PERINC DE AVE DE PROJETO
 * @param proj código do projwto
 * @return data que é o período de incubação da espécie
 */
public int retornaDataEspecie(String proj){
    try
    {
        con = criaConexao();
        stmt=con.createStatement();
        aux="SELECT PerInc FROM ";
        aux+="EspecieAve WHERE ";
        aux+="CodEsp=";
        aux+="SELECT Projeto.CodEsp WHERE ";
        aux+="Projeto.CodProj="+proj+"";
        rs=stmt.executeQuery(aux);
        while(rs.next())
        {
            retornaData=rs.getInt("PerInc");
        }
        rs.close();
        con.close();
        stmt.close();
    }
    catch(SQLException e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("SQL Exception no retornaDataEspecie -> "+aux);
    }
    catch(Exception e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("Saiu do Método retornaDataEspecie.");
    }
    return retornaData;
}

/**
 * RETORNA DATA PERNAS DE AVE DE PROJETO
 * @param proj código do projwto
 * @return data que é o período de nascimento da espécie
 */
public int retornaDataEspecieNas(String proj){

```

```

try
{
    con = criaConexao();
    stmt=con.createStatement();
    aux="SELECT PerNas FROM ";
    aux+="EspecieAve WHERE ";
    aux+="CodEsp=";
    aux+="SELECT Projeto.CodEsp WHERE ";
    aux+="Projeto.CodProj="+proj+";";
    rs=stmt.executeQuery(aux);
    while(rs.next())
    {
        retornaData=rs.getInt("PerNas");
    }
    rs.close();
    con.close();
    stmt.close();
}
catch(SQLException e)
{
    System.out.println(e.getMessage());
    e.printStackTrace();
    System.out.println("SQL Exception no retornaDataEspecieNas -> "+aux);
}
catch(Exception e)
{
    System.out.println(e.getMessage());
    e.printStackTrace();
    System.out.println("Saiu do Método retornaDataEspecieNas.");
}
return retornaData;
}

/**
 * RETORNA EM OVO SEU TIPO DE COR
 * @return vetor com tipos de cor ativos de ovo
 */
public Vector retornaCorOvo()
{
    try
    {
        vetor=new Vector();
        String x="";
        con=criaConexao();
        stmt=con.createStatement();
        aux="SELECT * FROM ";
        aux+="CorOvo WHERE ";
        aux+="CodStatusGeral=1 ";
        aux+="ORDER BY NomeCorOvo;";
        rs=stmt.executeQuery(aux);
        while(rs.next())
        {
            x=rs.getString("CodCorOvo");
            x+=" - ";
            x+=rs.getString("NomeCorOvo");
            vetor.addElement(x);
        }
        rs.close();
        con.close();
        stmt.close();
    }
    catch(SQLException e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("SQL Exception no retornaCorOvo -> "+aux);
    }
    catch(Exception e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("Saiu do Método retornaCorOvo.");
    }
    return vetor;
}
}

```

```

/**
 * RETORNA EM OVO SEU TIPO DE DESINFECÇÃO
 * @return vetor com tipos de desinfecção ativos de ovo
 */
public Vector retornaDesOvo()
{
    try
    {
        vetor=new Vector();
        String x="";
        con=criaConexao();
        stmt=con.createStatement();
        aux="SELECT * FROM ";
        aux+="DesOvo WHERE ";
        aux+="CodStatusGeral=1 ";
        aux+="ORDER BY NomeDesOvo;";
        rs=stmt.executeQuery(aux);
        while(rs.next())
        {
            x=rs.getString("CodDesOvo");
            x+=" - ";
            x+=rs.getString("NomeDesOvo");
            vetor.addElement(x);
        }
        rs.close();
        con.close();
        stmt.close();
    }
    catch(SQLException e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("SQL Exception no retornaDesOvo -> "+aux);
    }
    catch(Exception e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("Saiu do Método retornaDesOvo.");
    }
    return vetor;
}

/**
 * RETORNA EM OVO SEU TIPO DE FORMATO
 * @return vetor com tipos de formato ativos de ovo
 */
public Vector retornaForOvo()
{
    try
    {
        vetor=new Vector();
        String x="";
        con=criaConexao();
        stmt=con.createStatement();
        aux="SELECT * FROM ";
        aux+="ForOvo WHERE ";
        aux+="CodStatusGeral=1 ";
        aux+="ORDER BY NomeForOvo;";
        rs=stmt.executeQuery(aux);
        while(rs.next())
        {
            x=rs.getString("CodForOvo");
            x+=" - ";
            x+=rs.getString("NomeForOvo");
            vetor.addElement(x);
        }
        rs.close();
        con.close();
        stmt.close();
    }
    catch(SQLException e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("SQL Exception no retornaForOvo -> "+aux);
    }
}

```

```

    }
    catch(Exception e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("Saiu do Método retornaForOvo.");
    }
    return vetor;
}

/**
 * RETORNA ENTIDADE
 * @return vetor com entidades
 */
public Vector retornaEntidade()
{
    try
    {
        vetor = new Vector();
        String x="";
        con = criaConexao();
        stmt=con.createStatement();
        aux="SELECT * FROM Entidade;";
        rs=stmt.executeQuery(aux);
        while(rs.next())
        {
            x=rs.getString("CodEnt");
            x+=" - ";
            x+=rs.getString("NomeEnt");
            vetor.addElement(x);
        }
        rs.close();
        con.close();
        stmt.close();
    }
    catch(SQLException e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("SQL Exception no retornaEntidade -> "+aux);
    }
    catch(Exception e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("Saiu do Método retornaEntidade.");
    }
    return vetor;
}

/**
 * RETORNA ENTIDADES DE TIPOS DE ENTIDADE EM INCUBAÇÃO
 * @param inc código da incubação
 * @return código e nome das entidades dos tipos em incubação
 */
public Vector retornaEntidadeInc(String inc)
{
    try
    {
        String auxCodTipo="";
        vetor = new Vector();
        String x="";
        con = criaConexao();
        stmt=con.createStatement();
        aux="SELECT * FROM ";
        aux+="Entidade WHERE ";
        aux+="CodEnt=";
        aux+="SELECT CodEnt FROM "+inc+";";
        rs=stmt.executeQuery(aux);
        while(rs.next())
        {
            auxCodTipo=rs.getString("CodTipoEnt");
        }
        rs.close();
        aux="SELECT * FROM ";
        aux+="Entidade WHERE ";
    }
}

```

```

        aux+="CodTipoEnt="+auxCodTipo+"";
        rs=stmt.executeQuery(aux);
        while(rs.next())
        {
            x=rs.getString("CodEnt");
            x+=" - ";
            x+=rs.getString("NomeEnt");
            vetor.addElement(x);
        }
        rs.close();
        con.close();
        stmt.close();
    }
    catch(SQLException e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("SQL Exception no retornaEntidadeInc -> "+aux);
    }
    catch(Exception e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("Saiu do Método retornaEntidadeInc.");
    }
    return vetor;
}

/**
 * RETORNA ENTIDADES EM INCUBAÇÃO
 * @param inc código da incubação
 * @return código e nome das entidades em incubação
 */
public Vector retornaEntInc(String inc)
{
    try
    {
        vetor = new Vector();
        String x="";
        con = criaConexao();
        stmt=con.createStatement();
        aux="SELECT * FROM ";
        aux+="Entidade WHERE ";
        aux+="Entidade.CodEnt=(";
        aux+="SELECT CodEnt FROM ";
        aux+="Incubacao WHERE ";
        aux+="CodInc="+inc+"";
        rs=stmt.executeQuery(aux);
        while(rs.next())
        {
            x=rs.getString("CodEnt");
            x+=" - ";
            x+=rs.getString("NomeEnt");
            vetor.addElement(x);
        }
        rs.close();
        con.close();
        stmt.close();
    }
    catch(SQLException e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("SQL Exception no retornaTipoEntidade -> "+aux);
    }
    catch(Exception e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("Saiu do Método retornaTipoEntidade.");
    }
    return vetor;
}

/**
 * RETORNA ENTIDADES POR ENTIDADE MÍNIMA DE PROJETO

```

```

* @param projeto código do projeto
* @return vetor com as entidades por tipo de entidade
*/
public Vector retornaEntidadeProj (String projeto)
{
    try
    {
        String auxCodTipo="";
        vetor = new Vector();
        String x="";
        con = criaConexao();
        stmt=con.createStatement();
        aux="SELECT CodTipoEnt FROM ";
        aux+="Projeto WHERE ";
        aux+="CodProj="+projeto+"";
        rs=stmt.executeQuery(aux);
        while(rs.next())
        {
            auxCodTipo=rs.getString("CodTipoEnt");
        }
        rs.close();
        aux="SELECT * FROM ";
        aux+="Entidade WHERE ";
        aux+="CodTipoEnt="+auxCodTipo+"";
        rs=stmt.executeQuery(aux);
        while(rs.next())
        {
            x=rs.getString("CodEnt");
            x+=" - ";
            x+=rs.getString("NomeEnt");
            vetor.addElement(x);
        }
        rs.close();
        con.close();
        stmt.close();
        System.out.println(aux);
    }
    catch(SQLException e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("SQL Exception no retornaEntidadeProj -> "+aux);
    }
    catch(Exception e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("Saiu do Método retornaEntidadeProj.");
    }
    return vetor;
}

/**
* RETORNA ENTIDADE POR TIPO
* @param ent código do tipo de entidade
* @return vetor com as entidades por tipo
*/
public Vector retornaEntidade (String ent)
{
    try
    {
        String pai;
        int filho=Integer.parseInt(ent);
        vetor = new Vector();
        String x="";
        con = criaConexao();
        stmt=con.createStatement();
        if((filho >= 1) && (filho <= 8))
        {
            pai=Integer.toString(filho+1);
        }
        else
        {
            pai="0";
        }

        System.out.println("pai:"+pai+"ent:"+ent);
    }
}

```

```

        aux="SELECT * FROM ";
        aux+="Entidade WHERE ";
        aux+="CodTipoEnt='"+pai+"'";
        rs=stmt.executeQuery(aux);
while(rs.next())
{
    x=rs.getString("CodEnt");
    x+=" - ";
    x+=rs.getString("NomeEnt");
    vetor.addElement(x);
}
rs.close();
con.close();
stmt.close();
}
catch(SQLException e)
{
    System.out.println(e.getMessage());
    e.printStackTrace();
    System.out.println("SQL Exception no retornaEntidade -> "+aux);
}
catch(Exception e)
{
    System.out.println(e.getMessage());
    e.printStackTrace();
    System.out.println("Saiu do Método retornaEntidade.");
}
return vetor;
}

/**
 * RETORNA ESPÉCIE
 * @return vetor com as espécies
 */
public Vector retornaEspecie()
{
    try
    {
        vetor = new Vector();
        int cont=0;
        String x="";
        con = criaConexao();
        stmt=con.createStatement();
        aux="SELECT * FROM ";
        aux+="EspecieAve ORDER BY CodEsp;";
        rs=stmt.executeQuery(aux);
while(rs.next())
{
    x=rs.getString("CodEsp");
    x+=" - ";
    x+=rs.getString("NomeEsp");
    vetor.addElement(x);
    cont++;
}
if(cont==0){
    x="0";
    x+=" - ";
    x+="nenhum";
    vetor.addElement(x);
}
rs.close();
con.close();
stmt.close();
}
catch(SQLException e)
{
    System.out.println(e.getMessage());
    e.printStackTrace();
    System.out.println("SQL Exception no retornaEspecie -> "+aux);
}
catch(Exception e)
{
    System.out.println(e.getMessage());
    e.printStackTrace();
    System.out.println("Saiu do Método retornaEspecie.");
}
}

```

```

        return vetor;
    }

    /**
     * RETORNA ESPÉCIE DE PROJETO
     * @return nome da espécie do projeto
     */
    public String retornaEspProjeto(String projeto)
    {
        try
        {
            String x="";
            con = criaConexao();
            stmt=con.createStatement();
            aux="SELECT * FROM ";
            aux+="EspecieAve JOIN Projeto ";
            aux+="ON EspecieAve.CodEsp=Projeto.CodEsp ";
            aux+="WHERE Projeto.CodProj="+projeto+"";
            rs=stmt.executeQuery(aux);
            while(rs.next())
            {
                x=rs.getString("NomeEsp");
            }
            nomeEsp=x;
            rs.close();
            con.close();
            stmt.close();
        }
        catch(SQLException e)
        {
            System.out.println(e.getMessage());
            e.printStackTrace();
            System.out.println("SQL Exception no retornaEspProjeto -> "+aux);
        }
        catch(Exception e)
        {
            System.out.println(e.getMessage());
            e.printStackTrace();
            System.out.println("Saiu do Método retornaEspProjeto.");
        }
        return nomeEsp;
    }

    /**
     * RETORNA INCUBAÇÃO
     * @param proj código do projeto
     * @return vetor com as incubações e datas de inicio
     */
    public Vector retornaIncubacao(String proj)
    {
        TransferData dt=new TransferData();
        try
        {
            vetor = new Vector();
            String x="";
            con = criaConexao();
            stmt=con.createStatement();
            aux="SELECT * FROM ";
            aux+="Incubacao WHERE ";
            aux+="CodStatusIN=1 AND ";
            aux+="CodProj="+proj+" ";
            aux+="ORDER BY CodInc;";
            rs = stmt.executeQuery(aux);
            while(rs.next())
            {
                x=rs.getString("CodInc");
                x+=" de ";
                x+=dt.DataPApresentar(rs.getString("DataIniInc"));
                vetor.addElement(x);
            }
            rs.close();
            con.close();
            stmt.close();
        }
        catch(SQLException e)
        {

```

```

        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("SQL Exception no retornaIncubacao -> "+aux);
    }
    catch(Exception e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("Saiu do Método retornaIncubacao.");
    }
    return vetor;
}

/**
 * RETORNA INCUBADORA
 * @return vetor com as incubadoras ativas
 */
public Vector retornaIncubadora()
{
    try
    {
        vetor=new Vector();
        String x="";
        con=criaConexao();
        stmt=con.createStatement();
        aux="SELECT * FROM ";
        aux+="Incubadora WHERE ";
        aux+="CodStatusGeral=1 ORDER BY NomeEnc;";
        rs=stmt.executeQuery(aux);
        while(rs.next())
        {
            x=rs.getString("CodEnc");
            x+=" - ";
            x+=rs.getString("NomeEnc");
            vetor.addElement(x);
        }
        rs.close();
        con.close();
        stmt.close();
    }
    catch(SQLException e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("SQL Exception no retornaIncubadora -> "+aux);
    }
    catch(Exception e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("Saiu do Método retornaIncubadora.");
    }
    return vetor;
}

/**
 * RETORNA GRUPO DE USUÁRIO
 * @return vetor com grupos de usuário
 */
public Vector retornaGrupo()
{
    try
    {
        vetor=new Vector();
        String x="";
        con=criaConexao();
        stmt=con.createStatement();
        aux="SELECT * FROM ";
        aux+="Grupo ORDER BY NomeGru;";
        rs=stmt.executeQuery(aux);
        while(rs.next())
        {
            x=rs.getString("CodGru");
            x+=" - ";
            x+=rs.getString("NomeGru");
            vetor.addElement(x);
        }
    }
}

```

```

    }
    rs.close();
    con.close();
    stmt.close();
}
catch(SQLException e)
{
    System.out.println(e.getMessage());
    e.printStackTrace();
    System.out.println("SQL Exception no retornaGrupo -> "+aux);
}
catch(Exception e)
{
    System.out.println(e.getMessage());
    e.printStackTrace();
    System.out.println("Saiu do Método retornaGrupo.");
}
return vetor;
}

/**
 * RETORNA NASCEDOURO
 * @return vetor com os nascedouros ativos
 */
public Vector retornaNascadouro()
{
    try
    {
        vetor = new Vector();
        String x="";
        con = criaConexao();
        stmt=con.createStatement();
        aux="SELECT * FROM Nascadouro ";
        aux+="WHERE CodStatusGeral=1 ";
        aux+="ORDER BY NomeNes;";
        rs=stmt.executeQuery(aux);
        while(rs.next())
        {
            x=rs.getString("CodNes");
            x+=" - ";
            x+=rs.getString("NomeNes");
            vetor.addElement(x);
        }
        rs.close();
        con.close();
        stmt.close();
    }
    catch(SQLException e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("SQL Exception no retornaNascadouro -> "+aux);
    }
    catch(Exception e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("Saiu do Método retornaNascadouro.");
    }
    return vetor;
}

/**
 * RETORNA NASCIMENTO
 * @param proj código do projeto
 * @return vetor com os nascimentos e datas de início
 */
public Vector retornaNascimento(String proj)
{
    TransferData dt = new TransferData();
    try
    {
        vetor = new Vector();
        String x="";
        con = criaConexao();
        stmt=con.createStatement();

```

```

        aux="SELECT * FROM ";
        aux+="Nascimento JOIN Incubacao ON ";
        aux+="Incubacao.CodInc=Nascimento.CodInc WHERE ";
        aux+="Incubacao.CodProj="+proj+" AND ";
        aux+="Nascimento.CodStatusIN=1 ";
        aux+="ORDER BY Nascimento.CodInc;";
        rs = stmt.executeQuery(aux);
        while(rs.next())
        {
            x=rs.getString("CodNas");
            x+=" de ";
            x+=dt.DataPApresentar(rs.getString("DataIniINas"));
            vetor.addElement(x);
        }
        rs.close();
        con.close();
        stmt.close();
    }
    catch(SQLException e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("SQL Exception no retornaIncubacao -> "+aux);
    }
    catch(Exception e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("Saiu do Método retornaIncubacao.");
    }
    return vetor;
}

/**
 * RETORNA POSIÇÃO DE BANDEJA EM INCUBADORA
 * @param cod código da bandeja
 * @return posições de bandeja e incubadora
 */
public String retornaPosBan(String cod){
    try
    {
        String x="";
        con = criaConexao();
        stmt=con.createStatement();
        aux="SELECT * FROM ";
        aux+="'"+"vBandejaIncubadoraPosicao"+"'"+" ";
        aux+="WHERE CodBan="+cod+";";
        rs=stmt.executeQuery(aux);
        while(rs.next())
        {
            x=rs.getString("NomeEnc");
            x+=" - ";
            x+=rs.getString("CodPosBanEnc");
            x+=" : ";
            x+=rs.getString("NPosEnc");
        }
        retorna=x;
        rs.close();
        con.close();
        stmt.close();
    }
    catch(SQLException e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("SQL Exception no retornaPosBan -> "+aux);
    }
    catch(Exception e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("Saiu do Método retornaPosBan.");
    }
    return retorna;
}
}

```

```

/**
 * RETORNA POSIÇÃO DE BANDEJA EM NASCEDOURO
 * @param cod código da bandeja
 * @return posições de bandeja e nascedouro
 */
public String retornaPosBanNas(String cod) {
    try
    {
        String x="";
        con = criaConexao();
        stmt=con.createStatement();
        aux="SELECT * FROM ";
        aux+="vBandejaNascedouroPosicao"+" "+cod+" ";
        aux+="WHERE CodBan="+cod+";";
        rs=stmt.executeQuery(aux);
        while(rs.next())
        {
            x=rs.getString("NomeNes");
            x+=" - ";
            x+=rs.getString("CodPosBanNes");
            x+=" : ";
            x+=rs.getString("NPosNes");
        }
        System.out.println("aux:"+aux);
        retorna=x;
        rs.close();
        con.close();
        stmt.close();
    }
    catch(SQLException e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("SQL Exception no retornaPosBanNas -> "+aux);
    }
    catch(Exception e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("Saiu do Método retornaPosBanNas.");
    }
    return retorna;
}

/**
 * RETORNA POSIÇÃO DE INCUBADORA
 * @return vetor com posições e incubadora ativa
 */
public Vector retornaPosEnc()
{
    try
    {
        vetor=new Vector();
        String x="";
        String n="";
        int i=0;
        con=criaConexao();
        stmt=con.createStatement();
        aux="SELECT * FROM ";
        aux+="Incubadora WHERE ";
        aux+="CodStatusGeral=1 ORDER BY NomeEnc;";
        rs=stmt.executeQuery(aux);
        n=rs.getString("CodEnc");
        n+=" - ";
        n+=rs.getString("TPosEnc");
        while(i <= Integer.parseInt(n))
        {
            x=rs.getString("CodEnc");
            x+=" - ";
            x+=rs.getString(i+"");
            i++;
            vetor.addElement(x);
        }
        rs.close();
        con.close();
        stmt.close();
    }
}

```

```

    }
    catch(SQLException e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("SQL Exception no retornaPosEnc -> "+aux);
    }
    catch(Exception e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("Saiu do Método retornaPosEnc.");
    }
    return vetor;
}

/**
 * RETORNA PROJETO
 * @return vetor com os projetos
 */
public Vector retornaProjeto()
{
    try
    {
        vetor = new Vector();
        String x="";
        con = criaConexao();
        stmt=con.createStatement();
        aux="SELECT * FROM ";
        aux+="Projeto ORDER BY SiglaProj;";
        rs=stmt.executeQuery(aux);
        while(rs.next())
        {
            x=rs.getString("CodProj");
            x+=" - ";
            x+=rs.getString("SiglaProj");
            vetor.addElement(x);
        }
        rs.close();
        con.close();
        stmt.close();
    }
    catch(SQLException e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("SQL Exception no retornaProjeto -> "+aux);
    }
    catch(Exception e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("Saiu do Método retornaProjeto.");
    }
    return vetor;
}

/**
 * RETORNA PROJETO DE INCUBAÇÃO
 * @param inc código da incubação
 * @return código do projeto da incubação
 */
public String retornaProjInc(String inc)
{
    try
    {
        vetor=new Vector();
        String x="";
        con=criaConexao();
        stmt=con.createStatement();
        aux="SELECT * FROM ";
        aux+="Incubacao WHERE ";
        aux+="CodInc="+inc+";";
        rs=stmt.executeQuery(aux);
        while(rs.next())
        {

```

```

        x=rs.getString("CodProj");
        retorna=x;
    }
    rs.close();
    con.close();
    stmt.close();
}
catch(SQLException e)
{
    System.out.println(e.getMessage());
    e.printStackTrace();
    System.out.println("SQL Exception no retornaProjInc -> "+aux);
}
catch(Exception e)
{
    System.out.println(e.getMessage());
    e.printStackTrace();
    System.out.println("Saiu do Método retornaProjInc.");
}
return retorna;
}

/**
 * RETORNA PROPRIEDADE
 * @return vetor com as propriedades ativas
 */
public Vector retornaPropriedade()
{
    try
    {
        vetor=new Vector();
        String x="";
        con=criaConexao();
        stmt=con.createStatement();
        aux="SELECT * FROM ";
        aux+="Propriedade INNER JOIN Entidade ";
        aux+="ON Entidade.CodEnt=Propriedade.CodProp ";
        aux+="WHERE CodStatusGeral=1 ";
        aux+="ORDER BY NomeEnt;";
        rs=stmt.executeQuery(aux);
        while(rs.next())
        {
            x=rs.getString("CodProp");
            x+=" - ";
            x+=rs.getString("NomeEnt");
            vetor.addElement(x);
        }
        rs.close();
        con.close();
        stmt.close();
    }
    catch(SQLException e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("SQL Exception no retornaPropriedade -> "+aux);
    }
    catch(Exception e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("Saiu do Método retornaPropriedade.");
    }
    return vetor;
}

/**
 * RETORNA QUANTIDADE DE ENTIDADE EM INCUBAÇÃO
 * @param incubacao código da incubação
 * @param entidade código da entidade
 * @return vetor com status geral
 */
public String buscarQuantidade(String incubacao,String entidade)
{
    try
    {

```

```

        con = criaConexao();
        stmt=con.createStatement();
        aux="SELECT COUNT(CodInc) FROM ";
        aux+="IncTemEnt WHERE ";
        aux+="CodInc="+incubacao+" AND ";
        aux+="OvoTransf<>'S'";
        rs = stmt.executeQuery(aux);
while(rs.next())
{
    retorna=rs.getString(1);
    retorna+="-";
}
rs.close();
aux="SELECT COUNT(CodInc) FROM ";
aux+="IncTemEnt WHERE ";
aux+="CodInc="+incubacao+" AND ";
aux+="CodEnt="+entidade+" AND ";
aux+="OvoTransf<>'S'";
rs = stmt.executeQuery(aux);
while(rs.next())
{
    retorna+=rs.getString(1);
}
rs.close();
con.close();
stmt.close();
}
catch(SQLException e)
{
    System.out.println(e.getMessage());
    e.printStackTrace();
    System.out.println("SQL Exception no buscarQuantidade -> "+aux);
}
catch(Exception e)
{
    System.out.println(e.getMessage());
    e.printStackTrace();
    System.out.println("Saiu do Método buscarQuantidade.");
}
return retorna;
}

/**
 * RETORNA TAMANHO DA BANDEJA
 * @param cod código da bandeja
 * @return número de linhas e colunas
 */
public String retornaTamBan(String cod)
{
    String posicao="";
    try
    {
        String x="";
        con = criaConexao();
        stmt=con.createStatement();
        aux="SELECT * FROM ";
        aux+="Bandeja WHERE ";
        aux+="CodBan="+cod+"";
        rs=stmt.executeQuery(aux);
        while(rs.next())
        {
            x=rs.getString("tLinBan");
            x+="-";
            x+=rs.getString("tColBan");
            posicao=x;
        }
        rs.close();
        con.close();
        stmt.close();
    }
    catch(SQLException e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("SQL Exception no retornaTamBan -> "+aux);
    }
}

```

```

        catch(Exception e)
        {
            System.out.println(e.getMessage());
            e.printStackTrace();
            System.out.println("Saiu do Método retornaTamBan.");
        }
        return posicao;
    }

/**
 * RETORNA TIPO DE ENTIDADE
 * @return vetor com os tipos de entidade
 */
public Vector retornaTipoEntidade()
{
    try
    {
        vetor = new Vector();
        int cont=0;
        String x="";
        con = criaConexao();
        stmt=con.createStatement();
        aux="SELECT * FROM ";
        aux+="TipoEntidade ORDER BY CodTipoEnt;";
        rs=stmt.executeQuery(aux);
        rs.next();
        while(rs.next())
        {
            x=rs.getString("CodTipoEnt");
            x+=" - ";
            x+=rs.getString("NomeTipoEnt");
            vetor.addElement(x);
            cont++;
        }
        if(cont == 0)
        {
            x="0";
            x+=" - ";
            x+="nenhum";
            vetor.addElement(x);
        }
        rs.close();
        con.close();
        stmt.close();
    }
    catch(SQLException e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("SQL Exception no retornaTipoEntidade -> "+aux);
    }
    catch(Exception e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("Saiu do Método retornaTipoEntidade.");
    }
    return vetor;
}

/**
 * RETORNA TIPO DE ENTIDADE MÍNIMA DO PROJETO
 * @param projeto código do projeto
 * @return código e nome da entidade mínima
 */
public String retornaEntMin(String projeto)
{
    try
    {
        String x="";
        con = criaConexao();
        stmt=con.createStatement();
        aux="SELECT * FROM ";
        aux+="TipoEntidade JOIN Projeto ";
        aux+="ON TipoEntidade.CodTipoEnt=Projeto.CodTipoEnt ";
        aux+="WHERE Projeto.CodProj="+projeto+ " ";
    }

```

```

        rs=stmt.executeQuery(aux);
while(rs.next())
{
    x=rs.getString("CodTipoEnt");
    x+=" - ";
    x+=rs.getString("NomeTipoEnt");
    entMin=x;
}
rs.close();
con.close();
stmt.close();
}
catch(SQLException e)
{
    System.out.println(e.getMessage());
    e.printStackTrace();
    System.out.println("SQL Exception no retornaEntMin -> "+aux);
}
catch(Exception e)
{
    System.out.println(e.getMessage());
    e.printStackTrace();
    System.out.println("Saiu do Método retornaEntMin.");
}
return entMin;
}

/**
 * RETORNA TIPO DE STATUS DE OVO
 * @return vetor com status ativos de ovo
 */
public Vector retornaStatusOvo()
{
    try
    {
        vetor=new Vector();
        String x="";
        con=criaConexao();
        stmt=con.createStatement();
        aux="SELECT * FROM ";
        aux+="StatusOvo WHERE ";
        aux+="CodStatusGeral=1 ";
        aux+="ORDER BY NomeStatusOvo;";
        rs=stmt.executeQuery(aux);
        while(rs.next())
        {
            x=rs.getString("CodStatusOvo");
            x+=" - ";
            x+=rs.getString("NomeStatusOvo");
            vetor.addElement(x);
        }
        rs.close();
        con.close();
        stmt.close();
    }
    catch(SQLException e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("SQL Exception no retornaStatusOvo -> "+aux);
    }
    catch(Exception e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("Saiu do Método retornaStatusOvo.");
    }
    return vetor;
}

/**
 * RETORNA TIPO DE STATUS DE PROJETO
 * @return vetor com status de projeto
 */
public Vector retornaStatusProj()
{

```

```

try
{
    vetor = new Vector();
    String x="";
    con = criaConexao();
    stmt=con.createStatement();
    aux="SELECT * FROM ";
    aux+="StatusProjeto WHERE ";
    aux+="CodStatusProj!=3";
    aux+="ORDER BY CodStatusProj;";
    rs=stmt.executeQuery(aux);
    while(rs.next())
    {
        x=rs.getString("CodStatusProj");
        x+=" - ";
        x+=rs.getString("NomeStatusProj");
        vetor.addElement(x);
    }
    rs.close();
    con.close();
    stmt.close();
}
catch(SQLException e)
{
    System.out.println(e.getMessage());
    e.printStackTrace();
    System.out.println("SQL Exception no retornaStatusProj -> "+aux);
}
catch(Exception e)
{
    System.out.println(e.getMessage());
    e.printStackTrace();
    System.out.println("Saiu do Método retornaStatusProj.");
}
return vetor;
}

/**
 * RETORNA TIPO DE STATUS DE USUÁRIO
 * @return vetor com status de usuário
 */
public Vector retornaStatusUsu()
{
    try
    {
        vetor = new Vector();
        String x="";
        con = criaConexao();
        stmt=con.createStatement();
        aux="SELECT * FROM ";
        aux+="StatusUsuario WHERE ";
        aux+="CodStatusUsu!=3";
        aux+="ORDER BY NomeStatusUsu;";
        rs=stmt.executeQuery(aux);
        while(rs.next())
        {
            x=rs.getString("CodStatusUsu");
            x+=" - ";
            x+=rs.getString("NomeStatusUsu");
            vetor.addElement(x);
        }
        rs.close();
        con.close();
        stmt.close();
    }
    catch(SQLException e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("SQL Exception no retornaStatusUsu -> "+aux);
    }
    catch(Exception e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("Saiu do Método retornaStatusUsu.");
    }
}

```

```

    }
    return vetor;
}

/**
 * RETORNA TIPO DE STATUS GERAL
 * @return vetor com status geral
 */
public Vector retornaStatus()
{
    try
    {
        vetor = new Vector();
        String x="";
        con = criaConexao();
        stmt=con.createStatement();
        aux="SELECT * FROM ";
        aux+="StatusGeral ";
        aux+="ORDER BY CodStatusGeral;";
        rs=stmt.executeQuery(aux);
        while(rs.next())
        {
            x=rs.getString("CodStatusGeral");
            x+=" - ";
            x+=rs.getString("NomeStatusGeral");
            vetor.addElement(x);
        }
        rs.close();
        con.close();
        stmt.close();
    }
    catch(SQLException e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("SQL Exception no retornaStatus -> "+aux);
    }
    catch(Exception e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("Saiu do Método retornaStatus.");
    }
    return vetor;
}

/**
 * RETORNA TOTAL DE POSIÇÕES DE INCUBADORA
 * @param inc código da incubadora
 * @return número total de posições de uma incubadora
 */
public int retornaPosinc(String inc){
    try
    {
        con = criaConexao();
        stmt=con.createStatement();
        aux="SELECT TPosEnc FROM ";
        aux+="Incubadora WHERE CodEnc="+inc+";";
        rs=stmt.executeQuery(aux);
        while(rs.next())
        {
            ret=rs.getInt("TPosEnc");
        }
        rs.close();
        con.close();
        stmt.close();
    }
    catch(SQLException e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("SQL Exception no retornaPosinc -> "+aux);
    }
    catch(Exception e)
    {
        System.out.println(e.getMessage());
    }
}

```

```

        e.printStackTrace();
        System.out.println("Saiu do Método retornaPosinc.");
    }
    return ret;
}

/**
 * RETORNA TOTAL DE POSIÇÕES DE NASCEDOURO
 * @param inc código da incubadora
 * @return número total de posições de um nascedouro
 */
public int retornaPosnas(String nas){
    try
    {
        con = criaConexao();
        stmt=con.createStatement();
        aux="SELECT TColNes FROM ";
        aux+="Nascedouro WHERE CodNes="+nas+"";
        rs=stmt.executeQuery(aux);
        while(rs.next())
        {
            ret=rs.getInt("TColNes");
        }
        rs.close();
        con.close();
        stmt.close();
    }
    catch(SQLException e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("SQL Exception no retornaPosnas -> "+aux);
    }
    catch(Exception e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("Saiu do Método retornaPosnas.");
    }
    return ret;
}
}

```

CÓDIGO FONTE DAS PÁGINAS JSP

EntCad.jsp

```

<%@ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xml:ns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>SISINC - Cadastrar Entidade</title>
<link href="estilos/principal.css" rel="stylesheet" type="text/css" />
</head>
<%
//codigo em java que traz os valores para os

```

```

controladora.ControladoraConsulta cont = new controladora.ControladoraConsulta();

java.util.Vector filho,status,pai;

String filhoX = request.getParameter("filho");
String nomeEnt="";
if (filhoX==null){

    filho=cont.retornaTipoEntidade();
    status=cont.retornaStatus();
    pai=cont.retornaEntidade();
    nomeEnt="";
}else{
    filho=cont.retornaTipoEntidade();
    nomeEnt=(String) request.getAttribute("nomeEnt");
    status=cont.retornaStatus();
    pai=(java.util.Vector) request.getAttribute("ents");
}
%>
<body>
<form id="MainForm" name="MainForm" method="post" action="Entidade">
<table width="100%" border="0" cellspacing="5" cellpadding="0">
    <tr>
        <td colspan="3" class="titulo">Cadastrar Nova Entidade </td>
    </tr>
    <tr>
        <td colspan="3">&nbsp;  </td>
    </tr>
    <tr>
        <td width="20%" height="43" align="left" valign="top"><strong>Tipo de
Entidade:</strong></td>
        <td width="80%" colspan="2" align="left" valign="top">
            <!-- trecho de código que carrega o campo filho-->
            <select name = "filho" >
                <%
String aux = "";
int index = 0;
for(int i = 0 ; i < filho.size() ; i++)
{
    aux = (String) filho.elementAt(i);
    index = aux.indexOf('-');
    %>
                <%if(filhoX!=null){%>
                <%if(aux.substring(0,index-1).compareTo(filhoX)==0){ %>
                <option value="<%=aux.substring(0,index-1) %>"
selected="selected"><%=aux.substring(index +1) %></option>
                <%}else{ %>
                <option value="<%=aux.substring(0,index-1) %>"><%=aux.substring(index +1)
%></option>
                <%} %>
                <%}else{ %>
                <option value="<%=aux.substring(0,index-1) %>"><%=aux.substring(index +1)
%></option>
                <%}%>
                <%}%>
            </select>
            <!-- acaba o select -->
            <!-- Botão que carrega as entidades possíveis "pais" da entidade
escolhida -->
            <input type="submit" class="rodape" name="acao" value="pesquisar"> <br />
            <span class="atencao">ATENÇÃO:</span> <span class="descricao">Selecione o tipo
de Entidade e clique em Pesquisar. </span></td>
    </tr>
    <tr>
        <td align="left" valign="top"><em><strong>que está em:</strong></em></td>
        <td colspan="2" align="left" valign="top">
            <select name = "pai">
                <%
String aux1 = "";
int index1 = 0;
for(int i = 0 ; i < pai.size() ; i++)
{
    aux1 = (String) pai.elementAt(i);
    index1 = aux.indexOf('-');
    %>
                <option value="<%=aux1.substring(0,index1)%>"><%=aux1.substring(index1+1)%></option>

```



```

                String codigo=aux.substring(0,index-1);
            %>
        <tr>
            <td hight="26">
                <a href= "EntidadeAltera?codigo=<%=codigo%>&acao=carregar" ><%=nome%></a>
            </td>
            <td>
                <%=tipo%>
            </td>
        </tr>
    <% } %>
<% } else {%>
<tr>
    <td>
    </td>
    <td>
    </td>
</tr>
<% } %>
</table>
</body>
</html>

```

3-entidaade carregada

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title>projeto carregado</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<%
controladora.ControladoraConsulta cont = new controladora.ControladoraConsulta();
classepersistente.Entidade
auxEntidade=(classepersistente.Entidade) request.getAttribute("dados");
String nome = auxEntidade.getNome();
String entidade= auxEntidade.getEntidade();
String tipoEnt=auxEntidade.getTipoEntidade();
String descricao=auxEntidade.getDescEnt();
String codStatusGeral=auxEntidade.getTipoEntidade();

java.util.Vector ,tipoEntidade;

tipoEntidade=cont.retornaTipoEntidade();
statusProj= cont.retornaStatusProj();
%>

<body>
<form name="Form" method="POST" action="EntidadeAltera" >

<table width="82%" border="0" cellspacing="2" cellpadding="0">
    <tr>
        <th width="17%" align="left" valign="top" scope="row"><div
align="left">Sigla:</div></th>
        <td width="83%" align="left" valign="top"><div align="left">
            <input name="siglaProj" type="text" id="siglaProj" tabindex="1" size="14"
maxlength="14" value=<%=sigla%>/>
        </div></td>
    </tr>
    <tr>
        <th align="left" valign="top" scope="row"><div align="left">Nome:</div></th>
        <td align="left" valign="top"><div align="left">
            <input name="nomeProj" type="text" id="nomeProj" tabindex="2" size="103"
maxlength="100" value=<%=nome%>/>
        </div></td>
    </tr>
    <tr>
        <th align="left" valign="top" scope="row">

        <div align="left">Esp&eacute;cie de Ave:
        <td align="left" valign="top"><div align="left">
            <select name = "especie">

```

```

        String aux = "";
        int index = 0;
        for(int i = 0 ; i < especie.size() ; i++)
        {
            aux = (String) especie.elementAt(i);
            index = aux.indexOf('-');
        }
    }
    <%>
    <%if(aux.substring(0,index-1).compareTo(esp)==0){ %>
    <option value= <%=aux.substring(0,index-1) %> selected><%=aux.substring(index +1) %></option>
    <%
    }else{%>
    <option value= <%=aux.substring(0,index-1) %>><%=aux.substring(index +1) %></option>
    <%} %>
    <%} %>
</select></div>

</tr>
<tr>
    <td align="left" valign="top" scope="row"><div align="left">Entidade Escolhida: <td
    align="left" valign="top"><select name = "tipoentidade">

    <%
    String aux1 = "";
    int index1 = 0;
    for(int i = 0 ; i < tipoEntidade.size() ; i++)
    {
        aux1 = (String) tipoEntidade.elementAt(i);
        index1 = aux1.indexOf('-');
    }
    <%>
    <%if(aux1.substring(0,index1-1).compareTo(entidade)==0){ %>
    <option value= <%=aux1.substring(0,index1-1) %> selected><%=aux1.substring(index1 +1)
    %></option>
    <%
    }else{%>
    <option value= <%=aux1.substring(0,index1-1) %>><%=aux1.substring(index1 +1) %></option>
    <%} %>
    <%} %>
</select>

    </td>
</tr>
<tr>
    <th align="left" valign="top" scope="row"><div align="left">Data de In&iacute;cio:
</div></th>
    <td align="left" valign="top"><div align="left">
        <input name="dataIniProj" type="text" id="dataIniProj" tabindex="5" size="10"
        maxlength="10" value=<%=dataIni%>/>
    </div></td>
</tr>
<tr>
    <th align="left" valign="top" scope="row"><div align="left">Previs&atilde;o de
T&eacute;c&ccedil;&otilde;es: </div></th>
    <td align="left" valign="top">
        <input name="dataPrevProj" type="text"
        id="dataPrevProj" tabindex="6" size="10" maxlength="10" value=<%=dataPrev%>/>
    </td>
</tr>

    <tr valign="top">
        <th align="left" valign="top" scope="row"><div
        align="left">Descri&ccedil;&atilde;o:</div></th>
        <td align="left" valign="top"><div align="left">
            <textarea name="descProj" cols="100" rows="2" id="descProj" tabindex="8"
            ><%=descricao%></textarea>
        </div></td>
    </tr>
    <tr>
        <th align="left" valign="top" scope="row"><div align="left">N&uacute;mero de
        Incuba&ccedil;&otilde;es: </div></th>
        <td align="left" valign="top">
            <input name="numInc" type="text" id="numInc"
            tabindex="9" size="3" maxlength="3" disabled="true" /></td></tr>
    <tr>
        <th align="left" valign="top" scope="row"><div align="left">STATUS do Projeto: <td
        align="left" valign="top">
            <select name = "statusproj">

    <%
    String aux2 = "";

```



```

        index2 = aux.indexOf('|');

        String  codigo=aux.substring(0,index-1);
        String  tipo=aux.substring(index+1,index1-1);
        String  nome=aux.substring(index1+1,index2-1);
        String  pai=aux.substring(index2+1);
    %>

```

```

        <tr>
            <td hight="26">
                <%=nome%>
            </td>
            <td>
                <%=tipo%>
            </td>
            <td>
                <%=pai%>
            </td>

```

```

        </tr>
        <%= %>
    <% } else {%>
    <tr>
        <td>
        </td>
        <td>
        </td>
    </tr>
    <%= %>
</table>
</body>
</html>

```

5-EspCad.jsp

```

<%@ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>SISINC - Cadastrar Esp&eacute;cie de Ave</title>
<link href="estilos/principal.css" rel="stylesheet" type="text/css" />
<!-- verifica se suporta javascript
<script>function lastmod() {}</script>
<!-- validador do formulário
<script language="javascript" type="text/javascript" src="scripts\validator.js"></script>
<!-- caminho botão adicional -->
<script language="javascript" type="text/javascript">
function IrPara() {parent.location='../EntidadeOvoAlterar';}
</script>
</head>
<%
controladora.ControladoraConsulta cont=new controladora.ControladoraConsulta();

java.util.Vector CodDesOvo,CodStatusOvo,CodForOvo,CodCorOvo,CodCel,AquisOvo;
CodDesOvo=cont.retornaDesOvo();
CodStatusOvo=cont.retornaStatusOvo();
CodForOvo=cont.retornaForOvo();
CodCorOvo=cont.retornaCorOvo();
CodCel=cont.retornaCelula();
AquisOvo=cont.retornaPropriedade();
%>
<body>
<form id="MainForm" name="MainForm" method="post" action="EspecieAve">
<table width="100%" border="0" cellspacing="5" cellpadding="0">
    <tr>
        <td colspan="3" class="titulo">Cadastrar Nova Esp&eacute;cie de Ave </td>
    </tr>
    <tr>
        <td colspan="3">&nbsp;</td>
    </tr>
    <tr>
        <td width="20%" align="left" valign="top"><strong>Nome da Esp&eacute;cie:</strong></td>

```

```

        <td width="80%" colspan="2" align="left" valign="top">
            <input name="NomeEsp" type="text" id="NomeEsp" size="50" maxlength="50" />
            <br /></td>
    </tr>
    <tr>
        <td align="left" valign="top">Temp. Min. Incuba&ccedil;&atilde;o: </td>
        <td colspan="2" align="left" valign="top"><input name="TempMinInc" type="text"
id="TempMinInc" size="7" maxlength="7" />
            <span class="descricao">Celsius</span></td>
    </tr>
    <tr>
        <td height="22" align="left" valign="top">Temp. M&aacute;x. Incuba&ccedil;&atilde;o: </td>
        <td colspan="2" align="left" valign="top"><input name="TempMaxInc" type="text"
id="TempMaxInc" size="7" maxlength="7" />
            <span class="descricao">Celsius</span></td>
    </tr>
    <tr>
        <td height="22" align="left" valign="top">Umi. Min. Incuba&ccedil;&atilde;o: </td>
        <td colspan="2" align="left" valign="top"><input name="UmiMinInc" type="text"
id="UmiMinInc" size="7" maxlength="7" />
            <span class="descricao">?</span></td>
    </tr>
    <tr>
        <td height="22" align="left" valign="top">Umi. M&aacute;x. Incuba&ccedil;&atilde;o:</td>
        <td colspan="2" align="left" valign="top"><input name="UmiMaxInc" type="text"
id="UmiMaxInc" size="7" maxlength="7" />
            <span class="descricao">?</span></td>
    </tr>
    <tr>
        <td height="38" align="left" valign="top"><strong>Per&iacute;odo de
Incuba&ccedil;&atilde;o: </strong></td>
        <td colspan="2" align="left" valign="top">
            <input name="PerInc" type="text" id="PerInc" size="3" maxlength="3" />
            <span
class="descricao">dias.<br />
            <br />
            </span></td>
    </tr>
    <tr>
        <td height="22" align="left" valign="top"><strong>Per&iacute;odo Aviso Inc.:
</strong></td>
        <td colspan="2" align="left" valign="top"><input name="PerAviInc" type="text"
id="PerAviInc" size="3" maxlength="2" />
            <span class="descricao">dias.<br />
            <span class="atencao">NOTA:</span> Per&iacute;odo antes de completar para enviar
aviso.</span></td>
    </tr>
    <tr>
        <td align="left" valign="top">Temp. Min. Nascimento: </td>
        <td colspan="2" align="left" valign="top"><input name="TempMinNas" type="text"
id="TempMinNas" size="7" maxlength="7" /> <span class="descricao">Celsius</span></td>
    </tr>
    <tr>
        <td align="left" valign="top">Temp. M&aacute;x. Nascimento: </td>
        <td colspan="2" align="left" valign="top"><input name="TempMaxNas" type="text"
id="TempMaxNas" size="7" maxlength="7" />
            <span class="descricao">Celsius </span></td>
    </tr>
    <tr>
        <td align="left" valign="top">Umi. Min. Nascimento: </td>
        <td colspan="2" align="left" valign="top"><input name="UmiMinNas" type="text"
id="UmiMinNas" size="7" maxlength="7" />
            <span class="descricao">?</span></td>
    </tr>
    <tr>
        <td align="left" valign="top">Umi. M&aacute;x. Nascimento:</td>
        <td colspan="2" align="left" valign="top"><input name="TempMaxNas" type="text"
id="TempMaxNas" size="7" maxlength="7" />
            <span class="descricao">?</span></td>
    </tr>
    <tr>
        <td align="left" valign="top"><strong>Per&iacute;odo de Nascimento: </strong></td>
        <td colspan="2" align="left" valign="top"><label>
            <input name="PerNas" type="text" id="PerNas" size="3" maxlength="3" />
            <span class="descricao">dias.</span></label></td>
    </tr>

```

```

<tr>
  <td align="left" valign="top"><strong>Per&iacute;odo Aviso Nas.: </strong></td>
  <td colspan="2" align="left" valign="top"><input name="PerAviNas" type="text"
id="PerAviNas" size="3" maxlength="2" />
  <span class="descricao">dias.<br />
  <span class="atencao">NOTA:</span> Per&iacute;odo antes de completar para enviar
aviso.</span></td>
</tr>
<tr>
  <td align="left" valign="top"><strong>Per&iacute;odo em Recria:</strong></td>
  <td colspan="2" align="left" valign="top"><input name="PerRec" type="text" id="PerRec"
size="3" maxlength="3" />
  <span class="descricao">dias.</span></td>
</tr>
<tr>
  <td align="left" valign="top">&nbsp;</td>
  <td colspan="2" align="left" valign="top">&nbsp;</td>
</tr>
<tr>
  <td colspan="3" align="left" valign="top">
    <hr align="center" width="100%" size="1" noshade="noshade" />
  </td>
</tr>
</table>
<table width="100%" border="0" cellspacing="0" cellpadding="0">
  <tr>
    <td>
      <input name="Acao" type="submit" class="rodape" id="Acao" onclick="return v.exec()"
value="Gravar" />
    </td>
    <td>
      <input name="Limpar" type="reset" class="rodape" id="Limpar" value="Limpar" />
    </td>
    <td>
      <input name="IrPara" type="button" id="IrPara" onclick="IrPara()"
value="Alterar/Excluir" />
    </td>
  </tr>
</table>
</form>
<jsp:include page="../../include/cadastro.jsp" flush="true"/>
</body>
</html>

```

6-OvoCad.jsp

```

<%@ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>SISINC - Cadastrar Ovo</title>
<link href="estilos/principal.css" rel="stylesheet" type="text/css" />
<!-- verifica se suporta javascript
<script>function lastmod() {}</script>
<!-- validador do formulário
<script language="javascript" type="text/javascript" src="scripts\Validator.js"></script>
<!-- caminho botão adicional -->
<script language="javascript" type="text/javascript">
function IrPara() {parent.location='../..EntidadeOvoAlterar';}
</script>
</head>
<%
controladora.ControladoraConsulta cont=new controladora.ControladoraConsulta();

java.util.Vector CodDesOvo,CodStatusOvo,CodForOvo,CodCorOvo,CodCel,AquisOvo;
CodDesOvo=cont.retornaDesOvo();
CodStatusOvo=cont.retornaStatusOvo();
CodForOvo=cont.retornaForOvo();
CodCorOvo=cont.retornaCorOvo();
CodCel=cont.retornaCelula();
AquisOvo=cont.retornaPropriedade();
%>
<body>
<form id="MainForm" name="MainForm" method="post" action="EntidadeOvo">

```

```

<table width="100%" border="0" cellspacing="5" cellpadding="0">
  <tr>
    <td colspan="3" class="titulo">Cadastrar Nova Entidade: Ovo </td>
  </tr>
  <tr>
    <td colspan="3">&nbsp;&nbsp;&nbsp;</td>
  </tr>
  <tr>
    <td width="20%" align="left" valign="top"><strong>C&ocirc;odigo de Barras:</strong></td>
    <td width="80%" colspan="2" align="left" valign="top">
      <input name="CodBarras" type="text" id="CodBarras" size="14" maxlength="14" />
      <br />
      <span class="descricao"><span class="atencao">NOTA:</span> O c&ocirc;odigo deste
campo &eacute; gerado automaticamente se for deixado em branco.</span> </td>
    </tr>
  <tr>
    <td align="left" valign="top"><strong>C&eacute;lula de Origem: </strong></td>
    <td colspan="2" align="left" valign="top"><select name="CodCel" id="CodCel">
      <%
        String aux = "";
        int index = 0;
        for(int i = 0 ; i < CodCel.size() ; i++)
        {
          aux = (String) CodCel.elementAt(i);
          index = aux.indexOf('-');
        }
      %>
      <option value="<%=aux.substring(0,index-1)%"><%=aux.substring(index+1)%"></option>
      <%=aux.substring(0,index-1)%"><%=aux.substring(index+1)%"></option>
      <%=aux.substring(0,index-1)%"><%=aux.substring(index+1)%"></option>
    </select></td>
  </tr>
  <tr>
    <td height="22" align="left" valign="top"><strong>Cor do Ovo:</strong></td>
    <td colspan="2" align="left" valign="top"><select name="CodCorOvo" id="CodCorOvo">
      <%
        aux = "";
        index = 0;
        for(int i = 0 ; i < CodCorOvo.size() ; i++)
        {
          aux = (String) CodCorOvo.elementAt(i);
          index = aux.indexOf('-');
        }
      %>
      <option value="<%=aux.substring(0,index-1)%"><%=aux.substring(index+1)%"></option>
      <%=aux.substring(0,index-1)%"><%=aux.substring(index+1)%"></option>
      <%=aux.substring(0,index-1)%"><%=aux.substring(index+1)%"></option>
    </select></td>
  </tr>
  <tr>
    <td height="22" align="left" valign="top"><strong>Formato do Ovo:</strong></td>
    <td colspan="2" align="left" valign="top"><select name="CodForOvo" id="CodForOvo">
      <%
        aux = "";
        index = 0;
        for(int i = 0 ; i < CodForOvo.size() ; i++)
        {
          aux = (String) CodForOvo.elementAt(i);
          index = aux.indexOf('-');
        }
      %>
      <option value="<%=aux.substring(0,index-1)%"><%=aux.substring(index+1)%"></option>
      <%=aux.substring(0,index-1)%"><%=aux.substring(index+1)%"></option>
      <%=aux.substring(0,index-1)%"><%=aux.substring(index+1)%"></option>
    </select> </td>
  </tr>
  <tr>
    <td height="22" align="left" valign="top"><strong>Tipo de
Desinfec&ccedil;&atilde;o:</strong></td>
    <td colspan="2" align="left" valign="top"><select name="CodDesOvo" id="CodDesOvo">
      <%
        aux = "";
        index = 0;
        for(int i = 0 ; i < CodDesOvo.size() ; i++)
        {
          aux = (String) CodDesOvo.elementAt(i);
          index = aux.indexOf('-');
        }
      %>
      <option value="<%=aux.substring(0,index-1)%"><%=aux.substring(index+1)%"></option>
      <%=aux.substring(0,index-1)%"><%=aux.substring(index+1)%"></option>
      <%=aux.substring(0,index-1)%"><%=aux.substring(index+1)%"></option>
    </select></td>
  </tr>

```

```

</tr>

<tr>
  <td align="left" valign="top"><strong>Local de Aquisi&ccedil;&atilde;o: </strong></td>
  <td colspan="2" align="left" valign="top"><select name="AquisOvo" id="AquisOvo">
    <%
      aux = "";
      index = 0;
      for(int i = 0 ; i < AquisOvo.size() ; i++)
      {
        aux = (String) AquisOvo.elementAt(i);
        index = aux.indexOf('-');
      }
    %>
    <option value="<%=aux.substring(0,index-1)%"><%=aux.substring(index+1)%"></option>
    <}%>
  </select>
  <br />
  <span class="descricao"><span class="atencao">ATEN&Ccedil;&Atilde;O:</span> Se o ovo
n&atilde;o foi comprado, selecione sua pr&ocute;pria Propriedade.</span> </td>
</tr>
<tr>
  <td height="22" align="left" valign="top">Data da Coleta: </td>
  <td colspan="2" align="left" valign="top"><input name="DataColOvo" type="text"
id="DataColOvo" size="10" maxlength="10" /></td>
</tr>
<tr>
  <td align="left" valign="top">Aspecto:</td>
  <td colspan="2" align="left" valign="top"><span class="descricao">
  <input name="AspOvo" type="text" id="AspOvo" size="50" maxlength="50" />
  </span></td>
</tr>
<tr>
  <td align="left" valign="top">Peso:</td>
  <td colspan="2" align="left" valign="top"><input name="PesoOvo" type="text" id="PesoOvo"
size="6" maxlength="6" />
  <span class="descricao">gr. </span></td>
</tr>
<tr>
  <td align="left" valign="top">Porosidade:</td>
  <td colspan="2" align="left" valign="top"><input name="PorOvo" type="text" id="PorOvo"
size="10" maxlength="10" />
  <span class="descricao">un.</span></td>
</tr>
<tr>
  <td align="left" valign="top"><strong>Incub&aacute;vel:</strong></td>
  <td colspan="2" align="left" valign="top">
  <input name="Incubavel" type="text" id="Incubavel" size="2" maxlength="1" />
  <span class="atencao">(S/N) </span> </td>
</tr>
<tr>
  <td align="left" valign="top">Descri&ccedil;&atilde;o:</td>
  <td colspan="2" align="left" valign="top"><label>
  <textarea name="DescEnt" cols="38" rows="2" id="DescEnt"></textarea>
  </label></td>
</tr>
<tr>
  <td align="left" valign="top">&nbsp;</td>
  <td colspan="2" align="left" valign="top">&nbsp;</td>
</tr>
<tr>
  <td colspan="3" align="left" valign="top">
  <hr align="center" width="100%" size="1" noshade="noshade" /> </td>
</tr>
</table>
<table width="100%" border="0" cellspacing="0" cellpadding="0">
  <tr>
    <td>
      <input name="Acao" type="submit" class="rodape" id="Acao" onclick="return v.exec()"
value="Gravar" />
    </td>
    <td>
      <input name="Limpar" type="reset" class="rodape" id="Limpar" value="Limpar" />
    </td>
    <td>
      <input name="IrPara" type="button" id="IrPara" onclick="IrPara()"
value="Alterar/Excluir" />
    </td>
  </tr>

```

```

        </td>
    </tr>
</table>
</form>
<jsp:include page="../../include/cadastro.jsp" flush="true"/>
</body>
</html>

```

```

7-UsuAlt.jsp
<%@ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<meta name="robots" content="noindex, nofollow" />
<meta name="description" content="Página de alteração/exclusão de usuário." />
<meta name="author" content="Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano
Padilha" />
<title>SISINC - Alterar/Excluir Usuário</title>
<link href="estilos/principal.css" rel="stylesheet" type="text/css" />
<!-- validador do formulário -->
<script language="JavaScript" src="scripts/validador.js"></script>
<!-- limita entrada do formulário -->
<script language="JavaScript" src="scripts/form.js"></script>
<!-- posiciona na primeira linha -->
<script language="JavaScript" src="scripts/focus.js"></script>
<!-- criptografa a senha -->
<script language="JavaScript" src="scripts/cripto.js"></script>
</head>
<%
controladora.ControladoraConsulta cont = new controladora.ControladoraConsulta();
classepersistente.Usuario auxUsuario =
(classepersistente.Usuario)request.getAttribute("dados");

String MatrUsu=auxUsuario.getMatricula();
String NomeUsu=auxUsuario.getNome();
String CodGru=auxUsuario.getGrupo();
String SenCrt=auxUsuario.getSenha();
String TelRes=auxUsuario.getTelres();
String TelCom=auxUsuario.getTelcom();
String TelCel=auxUsuario.getTelcel();
String Email=auxUsuario.getEmail();
String CodStatusUsu=auxUsuario.getStatus();

// acerto de nulos
if (TelRes==null){TelRes="";}
if (TelCom==null){TelCom="";}
if (TelCel==null){TelCel="";}

java.util.Vector vCodGru,vCodStatusUsu;
vCodGru=cont.retornaGrupo();
vCodStatusUsu=cont.retornaStatusUsu();
%>
<body OnLoad="placeFocus()">
<form id="MainForm" name="MainForm" method="post" action="UsuarioAltera" onSubmit="return
v.exec()">
<table width="100%" border="0" cellspacing="5" cellpadding="0">
<tr>
<td colspan="3" class="titulo">Alterar/Excluir Usuário </td>
</tr>
<tr>
<td colspan="3">&nbsp;&nbsp;&nbsp;</td>
</tr>
<tr>
<td width="20%" align="left" valign="top"><strong>Matrícula:</strong></td>
<td width="80%" colspan="2" align="left" valign="top">
<input name="MatrUsu" type="text" class="atform" id="MatrUsu" size="8" maxlength="8"
onKeyUp="doCheckInteger(MainForm.MatrUsu)" value="<%=MatrUsu%>" />
<span class="descricao"> (ex.: 20060101)</span>
</td>
</tr>
<tr>
<td colspan="3"></td>
</tr>

```

```

        <td align="left" valign="top"><strong>Nome Completo:</strong></td>
        <td colspan="2" align="left" valign="top">
            <input name="NomeUsu" type="text" id="NomeUsu" size="50" maxlength="100"
onKeyUp="doCheckAlpha(MainForm.NomeUsu)" value="<%=NomeUsu%" />
        </td>
    </tr>
    <tr>
        <td height="22" align="left" valign="top"><strong>Grupo de Acesso:</strong></td>
        <td colspan="2" align="left" valign="top">
            <select name="CodGru" id="CodGru">
                <%
String aux = "";
int index = 0;
for(int i = 0 ; i < vCodGru.size() ; i++)
{
    aux = (String) vCodGru.elementAt(i);
    index = aux.indexOf('-');
}
                <%
                <if(aux.substring(0,index-1).compareTo(CodGru)==0) { %>
                    <option value="<%=aux.substring(0,index-1)%">
selected><%=aux.substring(index+1)%"></option>
                <%>else{%>
                    <option value="<%=aux.substring(0,index-1)%"><%=aux.substring(index+1)%"></option>
                <%>}%>
                <%>}%>
            </select>
        </td>
    </tr>
    <tr>
        <td height="22" align="left" valign="top"><strong>Senha:</strong></td>
        <td colspan="2" align="left" valign="top">
            <input name="SenCrt" type="text" id="SenCrt" size="8" value="<!--SenCrt--%">
onBlur="preTime(Matrusu.value,SenCrt.value, 0)" />
            <br />
            <span class="descricao"><span class="atencao">ATEN&Ccedil;&Atilde;O:</span> Senha deve
ter entre 6 e 8 caracteres.</span>
        </td>
    </tr>
    <tr>
        <td align="left" valign="top"><strong>Confirmar Senha: </strong></td>
        <td colspan="2" align="left" valign="top">
            <input name="SenCrt1" type="text" id="SenCrt1" size="8" value="<!--SenCrt--%">
onBlur="preTime(Matrusu.value,SenCrt1.value, 1)" />
            <br />
            <span class="descricao"><span class="atencao">ATEN&Ccedil;&Atilde;O:</span> Favor
digitar novamente a Senha.</span>
        </td>
    </tr>
    <tr>
        <td align="left" valign="top">Telefone Residencial: </td>
        <td colspan="2" align="left" valign="top">
            <input name="TelRes" type="text" id="TelRes" size="10" maxlength="10"
onKeyUp="doCheckInteger(MainForm.TelRes)" value="<%=TelRes%" />
            <span class="descricao"> (ex.: XX12345678)</span>
        </td>
    </tr>
    <tr>
        <td align="left" valign="top">Telefone Comercial:</td>
        <td colspan="2" align="left" valign="top">
            <input name="TelCom" type="text" id="TelCom" size="10" maxlength="10"
onKeyUp="doCheckInteger(MainForm.TelCom)" value="<%=TelCom%" />
            <span class="descricao"> (ex.: XX12345678)</span>
        </td>
    </tr>
    <tr>
        <td align="left" valign="top">Telefone Celular:</td>
        <td colspan="2" align="left" valign="top">
            <input name="TelCel" type="text" id="TelCel" size="10" maxlength="10"
onKeyUp="doCheckInteger(MainForm.TelCel)" value="<%=TelCel%" />
            <span class="descricao"> (ex.: XX12345678)</span>
        </td>
    </tr>
    <tr>
        <td align="left" valign="top"><strong>E-mail:</strong></td>
        <td colspan="2" align="left" valign="top">
            <input name="Email" type="text" id="Email" size="50" maxlength="50"
value="<%=Email%" />

```

```

        </td>
</tr>
<tr>
  <td align="left" valign="top"><em>STATUS</em></td>
  <td colspan="2" align="left" valign="top">
    <select name="CodStatusUsu" id="CodStatusUsu">
      <%
        aux = "";
        index = 0;
        for(int i = 0 ; i < vCodStatusUsu.size() ; i++)
        {
          aux = (String) vCodStatusUsu.elementAt(i);
          index = aux.indexOf('-');
        }
      <%
        if(aux.substring(0,index-1).compareTo(CodStatusUsu)==0) { %>
      <option value="<%=aux.substring(0,index-1)%>"
selected><%=aux.substring(index+1)%></option>
      <%
        }else{%>
      <option value="<%=aux.substring(0,index-1)%>"><%=aux.substring(index+1)%></option>
      <%}%>
      </select>
    </td>
</tr>
<tr>
  <td colspan="3" align="left" valign="top"><hr align="center" width="100%" size="1"
noshade="noshade" /></td>
</tr>
</table>
<table width="100%" border="0" cellspacing="0" cellpadding="0">
  <tr>
    <td>
      <input name="Acao" type="submit" class="rodape" id="Acao" value="Alterar" />
    </td>
    <td>
      <input name="Acao" type="submit" class="rodape" id="Acao" value="Excluir"
onClick="return confirm('Tem certeza que deseja excluir este registro?')" />
    </td>
    <td>
      <input name="Voltar" type="reset" class="rodape" id="Voltar" value="Voltar"
onClick="history.back(1)" />
    </td>
    <td>
      <input name="IrPara" type="button" id="IrPara" value="Cadastrar Novo"
onClick="window.open('Usuario','_self')" />
    </td>
  </tr>
</table>
</form>
<!-- chamada da página de rodapé -->
<jsp:include page="../include/gerenciamento.jsp" flush="true"/>
<!-- validação do formulário -->
<script language="JavaScript">
// descrição da estrutura de campos de formulário
var a_fields = {
  'MatrUsu':{'l':'Matrícula','r':true,'f':'integer','t':'MatrUsu','mn':6},
  'NomeUsu':{'l':'Nome Completo','r':true,'f':'alpha','t':'NomeUsu'},
  'CodGru':{'l':'Grupo de Acesso','r':true,'f':'integer','t':'CodGru'},
  'SenCrt':{'l':'Senha','r':true,'f':'alphanum','t':'SenCrt','mn':6,'mx':14},
  'SenCrt1':{'l':'Confirmação da
Senha','r':true,'f':'alphanum','t':'SenCrt1','m':'SenCrt','mn':6,'mx':14},
  'TelRes':{'l':'Telefone Residencial','r':false,'f':'integer','t':'TelRes','mn':7},
  'TelCom':{'l':'Telefone Comercial','r':false,'f':'integer','t':'TelCom','mn':7},
  'TelCel':{'l':'Telefone Celular','r':false,'f':'integer','t':'TelCel','mn':7},
  'Email':{'l':'E-mail','r':true,'f':'email','t':'Email','mn':11}
},
o_config = {
  'to_disable':['Gravar','Alterar'],
  'alert':1
}
// chamada do construtor do validator
var v = new validator('MainForm', a_fields, o_config);
</script>
</body>
</html>

```

8-UsuCad.jsp

```
<%@ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<meta name="robots" content="noindex, nofollow" />
<meta name="description" content="Página de cadastro de usuário." />
<meta name="author" content="Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano
Padilha" />
<title>SISINC - Cadastrar Usuário</title>
<link href="estilos/principal.css" rel="stylesheet" type="text/css" />
<!-- validador do formulário -->
<script language="JavaScript" src="scripts/validador.js"></script>
<!-- limita entrada do formulário -->
<script language="JavaScript" src="scripts/form.js"></script>
<!-- posiciona na primeira linha -->
<script language="JavaScript" src="scripts/focus.js"></script>
<!-- criptografa a senha -->
<script language="JavaScript" src="scripts/cripto.js"></script>
</head>
<%
controladora.ControladoraConsulta cont = new controladora.ControladoraConsulta();
// monta a lista Grupo
java.util.Vector CodGru;
CodGru=cont.retornaGrupo();
%>
<body OnLoad="placeFocus()">
<form id="MainForm" name="MainForm" method="post" action="Usuario" onSubmit="return v.exec()">
<table width="100%" border="0" cellspacing="5" cellpadding="0">
<tr>
<td colspan="3" class="titulo">Cadastrar Novo Usuário</td>
</tr>
<tr>
<td colspan="3">&nbsp;&nbsp;&nbsp;</td>
</tr>
<tr>
<td width="20%" align="left" valign="top"><strong>Matrícula:</strong></td>
<td width="80%" colspan="2" align="left" valign="top">
<input name="MatrUsu" type="text" id="MatrUsu" size="8" maxlength="8"
onKeyUp="doCheckInteger(MainForm.MatrUsu)" />
<span class="descricao"> (ex.: 20060101)</span>
</td>
</tr>
<tr>
<td align="left" valign="top"><strong>Nome Completo:</strong></td>
<td colspan="2" align="left" valign="top">
<input name="NomeUsu" type="text" id="NomeUsu" size="50" maxlength="100"
onKeyUp="doCheckAlpha(MainForm.NomeUsu)" />
</td>
</tr>
<tr>
<td align="left" valign="top"><strong>Grupo de Acesso:</strong></td>
<td colspan="2" align="left" valign="top">
<select name="CodGru" id="CodGru">
<%
String aux = "";
int index = 0;
for(int i = 0 ; i < CodGru.size() ; i++)
{
aux = (String) CodGru.elementAt(i);
index = aux.indexOf('-');
%>
<option value="<%=aux.substring(0,index-1)%%"><%=aux.substring(index+1)%%"></option>
<}%>
</select>
</td>
</tr>
<tr>
<td align="left" valign="top"><strong>Senha:</strong></td>
<td colspan="2" align="left" valign="top">
<input name="SenCrt" type="password" id="SenCrt" size="8"
onBlur="preTime(Matrusu.value,SenCrt.value, 0)" />
</td>
</tr>
</table>
</body>
</html>
```

```

        <br />
        <span class="descricao"><span class="atencao">ATEN&Ccedil;&Atilde;O:</span> Senha deve
ter entre 6 e 8 caracteres.</span> </td>
</tr>
<tr>
<td align="left" valign="top"><strong>Confirmar Senha: </strong></td>
<td colspan="2" align="left" valign="top">
<input name="SenCrt1" type="password" id="SenCrt1" size="8"
onBlur="preTime(Matrusu.value,SenCrt1.value, 1)" />
<br />
<span class="descricao"><span class="atencao">ATEN&Ccedil;&Atilde;O:</span> Favor
digitar novamente a Senha.</span>
</td>
</tr>
<tr>
<td align="left" valign="top">Telefone Residencial: </td>
<td colspan="2" align="left" valign="top">
<input name="TelRes" type="text" id="TelRes" size="10" maxlength="10"
onKeyUp="doCheckInteger(MainForm.TelRes)" />
<span class="descricao"> (ex.: XX12345678)</span>
</td>
</tr>
<tr>
<td align="left" valign="top">Telefone Comercial: </td>
<td colspan="2" align="left" valign="top">
<input name="TelCom" type="text" id="TelCom" size="10" maxlength="10"
onKeyUp="doCheckInteger(MainForm.TelCom)" />
<span class="descricao"> (ex.: XX12345678)</span>
</td>
</tr>
<tr>
<td align="left" valign="top">Telefone Celular: </td>
<td colspan="2" align="left" valign="top">
<input name="TelCel" type="text" id="TelCel" size="10" maxlength="10"
onKeyUp="doCheckInteger(MainForm.TelCel)" />
<span class="descricao"> (ex.: XX12345678)</span>
</td>
</tr>
<tr>
<td align="left" valign="top"><strong>E-mail:</strong></td>
<td colspan="2" align="left" valign="top">
<input name="Email" type="text" id="Email" size="50" maxlength="50" />
</td>
</tr>
<tr>
<td align="left" valign="top">&nbsp;</td>
<td colspan="2" align="left" valign="top">&nbsp;</td>
</tr>
<tr>
<td colspan="3" align="left" valign="top">
<hr align="center" width="100%" size="1" noshade="noshade" />
</td>
</tr>
</table>
<table width="100%" border="0" cellspacing="0" cellpadding="0">
<tr>
<td>
<input name="Acao" type="submit" class="rodape" id="Acao" value="Gravar" />
</td>
<td>
<input name="Limpar" type="reset" class="rodape" id="Limpar" value="Limpar" />
</td>
<td>
<input name="IrPara" type="button" id="IrPara" value="Alterar/Excluir"
onClick="window.open('UsuarioAltera','_self')" />
</td>
</tr>
</table>
</form>
<!-- chamada da página de rodapé -->
<jsp:include page="../include/gerenciamento.jsp" flush="true"/>
<!-- validação do formulário -->
<script language="JavaScript">
// descrição da estrutura de campos para o validador
var a_fields = {
    'Matrusu':{'l':'Matrícula','r':true,'f':'integer','t':'Matrusu','mn':6},

```

```

        'NomeUsu':{'l':'Nome Completo','r':true,'f':'alpha','t':'NomeUsu'},
        'CodGru':{'l':'Grupo de Acesso','r':true,'f':'integer','t':'CodGru'},
        'SenCrt':{'l':'Senha','r':true,'t':'SenCrt','mn':6,'mx':14},
        'SenCrt1':{'l':'Confirmação da
Senha','r':true,'t':'SenCrt1','m':'SenCrt','mn':6,'mx':14},
        'TelRes':{'l':'Telefone Residencial','r':false,'f':'integer','t':'TelRes','mn':7},
        'TelCom':{'l':'Telefone Comercial','r':false,'f':'integer','t':'TelCom','mn':7},
        'TelCel':{'l':'Telefone Celular','r':false,'f':'integer','t':'TelCel','mn':7},
        'Email':{'l':'E-mail','r':true,'f':'email','t':'Email','mn':11}
    },
    o_config = {
        'to_disable':['Gravar','Limpar'],
        'alert':1
    }
}
// chamada do construtor do validator
var v = new validator('MainForm', a_fields, o_config);
</script>
</body>
</html>

```

9-UsuCon.jsp

```

<%@ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<meta name="robots" content="noindex, nofollow" />
<meta name="description" content="Página de consulta de usuário." />
<meta name="author" content="Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano
Padilha" />
<title>SISINC - Consultar Usuário</title>
<link href="estilos/principal.css" rel="stylesheet" type="text/css" />
<!-- validador do formulário -->
<script language="JavaScript" src="scripts/validator.js"></script>
<!-- limita entrada do formulário -->
<script language="JavaScript" src="scripts/form.js"></script>
<!-- posiciona na primeira linha -->
<script language="JavaScript" src="scripts/focus.js"></script>
</head>
<%@ page import="controladora.ControladoraUsuario" %>
<%@ page import="servelet.UsuarioAlterar" %>
<%
String matricula = (String) request.getAttribute("matricula");
String nome = (String) request.getAttribute("nome");
String status = (String) request.getAttribute("status");

controladora.ControladoraConsulta cont = new controladora.ControladoraConsulta();
// monta a lista Status
java.util.Vector vCodStatusUsu;
vCodStatusUsu=cont.retornaStatusUsu();
%>
<body OnLoad="placeFocus()" >
<form id="MainForm" name="MainForm" method="post" action="UsuarioAlterar" onSubmit="return
v.exec()">
<table width="600" border="0" cellspacing="5" cellpadding="0">
<tr>
<td colspan="2">
<span class="titulo">Alterar/Excluir Usuário </span> </td>
</tr>
<tr>
<td colspan="2">&nbsp;</td>
</tr>
<tr>
<td width="15%">Matrícula:</td>
<td width="465%">
<input name="MatrUsu" type="text" id="MatrUsu" size="8" maxlength="8"
onKeyUp="doCheckInteger(MainForm.MatrUsu)" <%=if(matricula!=null){%> value="<%=matricula%>"
<%=}%>/>
<span class="descricao"> (ex.: 20060101)</span>
</td>
</tr>
<tr>
<td>
<td>Nome:</td>

```

```

        <td>
            <input name="NomeUsu" type="text" id="NomeUsu" size="50" maxlength="100"
onKeyUp="doCheckAlpha(MainForm.NomeUsu)" <%if(nome!=null){%> value="<%=nome%>" <%}%>/>
            </td>
        </tr>
        <tr>
            <td><em>STATUS</em></td>
            <td>
                <select name="CodStatusUsu" id="CodStatusUsu">
                    <%
String aux = "";
int index = 0;
for(int i = 0 ; i < vCodStatusUsu.size() ; i++)
{
    aux = (String) vCodStatusUsu.elementAt(i);
    index = aux.indexOf('-');
    <%
    <%if((status!=null) && (aux.substring(0,index-1).compareTo(status)==0)) { %>
        <option value="<%=aux.substring(0,index-1)%>"
selected><%=aux.substring(index+1)%></option>
        <%}else{%>
            <option value="<%=aux.substring(0,index-1)%>"><%=aux.substring(index+1)%></option>
        <%}%>
    <%}%>
                </select>
            </td>
        </tr>
        <tr>
            <td colspan="2">
                <hr align="center" width="100%" size="1" noshade="noshade" />
            </td>
        </tr>
        <tr>
            <td colspan="2"><table width="100%" border="0" cellspacing="0" cellpadding="0">
                <tr>
                    <td>
                        <input name="Acao" type="submit" class="rodape" id="Acao" value="Consultar"
                    </td>
                    <td>
                        <input name="Limpar" type="reset" class="rodape" id="Limpar" value="Limpar"
                    </td>
                    <td align="right">
                        <input name="IrPara" type="button" id="IrPara" value="Cadastrar Novo"
OnClick="window.open('Usuario','_self')" />
                    </td>
                </tr>
                <tr>
                    <td colspan="3">&nbsp;&nbsp;&nbsp;</td>
                </tr>
            </table></td>
        </tr>
    </table>
</form>
<table width="600" border="1" cellpadding="0" cellspacing="0" bordercolor="#000000">
    <tr>
        <td width="15%"><div align="center"><strong>Matr<iacute>cula</strong></div></td>
        <td width="85%"><div align="center"><strong>Nome</strong></div></td>
    </tr>
    <%
java.util.Vector tabela = new java.util.Vector();
tabela=(java.util.Vector) request.getAttribute("tabela");
%>
    <%if((tabela!=null) && (tabela.size()!=0)) {%>
    <%
        aux = "";
        index = 0;
        int index2 = 0;
        for(int i = 0; i < tabela.size(); i++)
        {
            aux = (String) tabela.elementAt(i);
            index = aux.indexOf('-');
            index2 = aux.indexOf(':');
        }
    <%
    <tr>

```

```

        <td>
            <a href="UsuarioAlterar?codigo=<%=aux.substring(0,index-
1)%>&Acao=Carregar"><%=aux.substring(index+2,index2-1)%></a>
        </td>
        <td>
            <a href="UsuarioAlterar?codigo=<%=aux.substring(0,index-
1)%>&Acao=Carregar"><%=aux.substring(index2+1)%></a>
        </td>
    </tr>
    <%=}%>
    <%=}else{%>
<tr>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
</tr>
<%=}%>
</table>
<!-- chamada da página de rodapé -->
<jsp:include page="../../include/gerenciamento.jsp" flush="true"/>
<!-- validação do formulário -->
<script language="JavaScript">
// descrição da estrutura de campos de formulário
var a_fields = {
    'MatrUsu':{'l':'Matrícula','r':false,'f':'integer','t':'MatrUsu'},
    'NomeUsu':{'l':'Nome','r':false,'f':'alpha','t':'NomeUsu'},
},
o_config = {
    'to_disable':['Consultar', 'Limpar'],
    'alert':1
}
// chamada do construtor do validator
var v = new validator('MainForm', a_fields, o_config);
</script>
</body>
</html>

```

10-Cadastro.jsp

```

<table width="100%" border="0" cellpadding="0" cellspacing="0">
    <tr>
        <td colspan="3">&nbsp;</td>
    </tr>
    <tr>
        <td colspan="3">
            <strong>Ir para:</strong>    <a href="jsp/principal/principal.jsp">Principal</a>
<strong>\</strong>    <a href="jsp/menu/administrador/cadastro.jsp">Cadastro</a>    </td>
        </tr>
</table>

```

11-Gerenciamento.jsp

```

<table width="100%" border="0" cellpadding="0" cellspacing="0">
    <tr>
        <td colspan="3">&nbsp;</td>
    </tr>
    <tr>
        <td colspan="3">
            <strong>Ir para:</strong>    <a href="jsp/principal/principal.jsp">Principal</a>
<strong>\</strong>    <a href="jsp/menu/administrador/gerenciamento.jsp">Gerenciamento</a>
        </td>
    </tr>
</table>

```

12-incubacao.jsp

```

<table width="100%" border="0" cellpadding="0" cellspacing="0">
    <tr>
        <td colspan="3">&nbsp;</td>
    </tr>
    <tr>
        <td colspan="3">
            <strong>Ir para:</strong>    <a href="jsp/principal/principal.jsp">Principal</a>
<strong>\</strong>    <a href="jsp/menu/administrador/incubacao.jsp">Incuba&ccedil;&atilde;o</a>    </td>
        </tr>
</table>

```

13-nascimento.jsp

```
<table width="100%" border="0" cellpadding="0" cellspacing="0">
  <tr>
    <td colspan="3">&nbsp;</td>
  </tr>
  <tr>
    <td colspan="3">
      <strong>Ir para:</strong>   <a href="jsp/principal/principal.jsp">Principal</a>
<strong>\</strong>   <a href="jsp/menu/administrador/nascimento.jsp">Nascimento</a>  </td>
    </tr>
</table>
```

14-projeto.jsp

```
<table width="100%" border="0" cellpadding="0" cellspacing="0">
  <tr>
    <td colspan="3">&nbsp;</td>
  </tr>
  <tr>
    <td colspan="3">
      <strong>Ir para:</strong>   <a href="jsp/principal/principal.jsp">Principal</a>
<strong>\</strong>   <a href="jsp/menu/administrador/projeto.jsp">Projeto</a>  </td>
    </tr>
</table>
```

15-rodape.jsp

```
<br />
<div align="center" class="rodape">
  <hr align="center" width="100%" size="1" noshade />
  <table width="100%" border="0" align="center" cellpadding="0" cellspacing="0">
    <tr>
      <td width="300"><div align="center">Curso de Zootecnia</div></td>
      <td width="300"><div align="center">Universidade Federal do
Paraná</div></td>
      <td width="300"><div align="center">Tecnologia em Inform&aacute;tica</div></td>
    </tr>
    <tr>
      <td><div align="center"><a href="http://www.ccz.agrarias.ufpr.br"></a></div></td>
      <td><div align="center"><a href="http://ww.ufpr.br"></a> <br />
2005</div></td>
      <td><div align="center"><a href="http://et.ufpr.br"></a></div>   </td>
    </tr>
  </table>
</div>
```

16-incban.jsp

```
<%@ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>SISINC - Cadastrar Bandeja em Incubadora</title>
<link href="estilos/principal.css" rel="stylesheet" type="text/css" />
<!-- verifica se suporta javascript
<script>function lastmod() {}</script>
<!-- validador do formulário
<script language="javascript" type="text/javascript" src="scripts\validator.js"></script>
<!-- caminho botão adicional -->

<script language="javascript" type="text/javascript">
function IrPara() {parent.location='jsp/incubacao/inccad.jsp';}
</script>
<script>
function mudaEstadoF(){
    document.MainForm.incubadora.disabled=false;

```

```

        document.MainForm.nascedouro.disabled=true;
        document.MainForm.posnas.disabled=true;
        document.MainForm.posinc.disabled=false;
    }
    function mudaEstadoT() {
        document.MainForm.incubadora.disabled=true;
        document.MainForm.nascedouro.disabled=true;
        document.MainForm.posnas.disabled=true;
        document.MainForm.posinc.disabled=true;
    }
    function mudaEstado2F() {
        document.MainForm.incubadora.disabled=true;
        document.MainForm.nascedouro.disabled=false;
        document.MainForm.posnas.disabled=false;
        document.MainForm.posinc.disabled=true;
    }
    function mudaEstado2T() {
        document.MainForm.incubadora.disabled=true;
        document.MainForm.nascedouro.disabled=true;
        document.MainForm.posnas.disabled=true;
        document.MainForm.posinc.disabled=true;
    }
</script>
</head>
<%@ page import="java.lang.*" %>
<%
controladora.ControladoraConsulta cont=new controladora.ControladoraConsulta();

java.util.Vector nas,inc,vetposN,vetposI,CodBan;
int x,var;
String marcaI="",marcaN="";
inc=new java.util.Vector();
nas=new java.util.Vector();
vetposI=new java.util.Vector();
vetposN=new java.util.Vector();

String ban = request.getParameter("Bandes");
String incubadora=request.getParameter("incubadora");
String nascedouro=request.getParameter("nascedouro");

CodBan=cont.retornaBandejaDesalocada();
inc=cont.retornaIncubadora();
nas=cont.retornaNascedouro();

if(incubadora!=null){
    int posinc = cont.retornaPosinc(incubadora);
    for(x=0;x<posinc;x++){
        vetposI.addElement(new Integer(x));

    }
    marcaN="disabled";
}else{
    if(nascedouro!=null){
        int posnas = cont.retornaPosnas(nascedouro);
        for(x=0;x<posnas;x++){
            vetposN.addElement(new Integer(x));
            marcaI="disabled";
        }
    }
}

%>
<body>
<form id="MainForm" name="MainForm" method="post" action="IncubadoraBandeja">
    <table width="100%" border="0" cellspacing="5" cellpadding="0">
        <tr>
            <td colspan="3" class="titulo">Alocar Bandeja em Incubadora/Nascedouro </td>
        </tr>
        <tr>
            <td colspan="3">&nbsp;&nbsp;&nbsp;</td>
        </tr>
        <tr>
    </table>

```

```

        <td colspan="3" align="left" valign="top"><hr align="center" width="100%" size="1"
noshade="noshade" /></td>
    </tr>
    <tr>
        <td colspan="3" align="left" valign="top"><em><strong>ALOCAR:</strong></em></td>
    </tr>
    <tr>
        <td width="20%" height="2" align="left" valign="top"><em><strong>Bandejas
desalocadas:</strong></em></td>
        <td height="2" colspan="2" align="left" valign="top"><select name="Bandes" id="Bandes" >
            <%
                String aux = "";
                int index = 0;
                for(int i = 0 ; i < CodBan.size() ; i++)
                {
                    aux = (String) CodBan.elementAt(i);
                    index = aux.indexOf('-');
                }
            %>
            <%if (ban!=null) { %>
                <%if (aux.substring(0,index-1).compareTo(ban)==0) { %>
                <option value="<%=aux.substring(0,index-1)%%"
selected><%=aux.substring(index+1)%%"></option>
                <% }else{ %>
                <option value="<%=aux.substring(0,index-1)%%"><%=aux.substring(index+1)%%"></option>
                <% } %>
                <%}else{ %>
                <option value="<%=aux.substring(0,index-1)%%"><%=aux.substring(index+1)%%"></option>
                <%} %>
            <%} %>
        </select> </td>
    </tr>
    <tr>
        <td height="2" colspan="3" align="left" valign="top"><hr align="center" width="100%"
size="1" noshade="noshade" /></td>
    </tr>
    <tr>
        <td height="22" align="left" valign="top"><strong><em>PARA:</em></strong></td>
        <td height="22" align="left" valign="top"><strong>Incubadora:</strong></td>
        <td height="22" align="left" valign="top"><strong>Nascedouro:</strong></td>
    </tr>
    <tr>
        <td align="left" valign="top"><strong></strong></td>
        <td width="21%" align="left" valign="top"> <input type="checkbox" name="checkbox2"
value="checkbox" onclick="if(this.checked){mudaEstadoF();}else{mudaEstadoT();}" />
        <!-- trecho de código que carrega o campo filho-->
        <select name="incubadora" id="incubadora" <%=marcaI%>
onchange="document.MainForm.submit();" >
            <%
                String aux1 = "";
                int index1 = 0;
                for(int i = 0 ; i < inc.size() ; i++)
                {
                    aux1 = (String) inc.elementAt(i);
                    index1 = aux1.indexOf('-');
                }
            %>
            <%if (incubadora!=null) { %>
                <%if (aux1.substring(0,index1-1).compareTo(incubadora)==0) { %>
                <option value="<%=aux1.substring(0,index1-1)%%"
selected="selected"><%=aux1.substring(index1+1)%%"></option>
                <%}else{ %>
                <option value="<%=aux1.substring(0,index1-
1)%%"><%=aux1.substring(index1+1)%%"></option>
                <%} %>
                <%}else{ %>
                <option value="<%=aux1.substring(0,index1-
1)%%"><%=aux1.substring(index1+1)%%"></option>
                <%} %>
            <%} %>
        </select>
        <!-- acaba o select -->
    </td>
        <td width="59%" align="left" valign="top"><input type="checkbox" name="checkbox2"
value="checkbox" onclick="if(this.checked){mudaEstado2F();}else{mudaEstado2T();}" />
        <select name="nascedouro" id="nascedouro" <%=marcaN%>
onchange="document.MainForm.submit();" >

```

```

        <%
        aux1 = "";
        index1 = 0;
        for(int i = 0 ; i < nas.size() ; i++)
        {
            aux1 = (String) nas.elementAt(i);
            index1 = aux1.indexOf('-');
            %>
            <%if(nascedouro!=null){%>
            <%if(aux1.substring(0,index1-1).compareTo(nascedouro)==0){ %>
            <option value="<%=aux1.substring(0,index1-1)%>"
selected="selected"><%=aux1.substring(index1+1)%></option>
            <%}else{%>
            <option value="<%=aux1.substring(0,index1-
1)%>"><%=aux1.substring(index1+1)%></option>
            <%}%>
            <%}else{ %>
            <option value="<%=aux1.substring(0,index1-
1)%>"><%=aux1.substring(index1+1)%></option>
            <%}%>
            <%}%>
            </select> </td>
        </tr>
        <tr>
        <td align="left" valign="top"><strong>Posi&ccedil;&atilde;o: </strong></td>
        <td align="left" valign="top">
        <!-- trecho de código que carrega o campo pai-->
        <select name="posinc" id="posinc" <%=marcaI%> >
        <%

        for(int i = 0 ; i < vetposI.size() ; i++)
        {

            Integer varI=(Integer)vetposI.elementAt(i);
            var=varI.intValue();
            var=var+1;

        %>
            <option value="<%=var%>"><%=var%></option>
            <%}%>
        </select>
        <!-- acaba o select -->
        </td>
        <td align="left" valign="top"> <select name="posnas" id="posnas" <%=marcaN%> >
        <%

        for(int i = 0 ; i < vetposN.size() ; i++)
        {

            Integer varI=(Integer)vetposN.elementAt(i);
            var=varI.intValue();
            var=var+1;

        %>
            <option value="<%=var%>"><%=var%></option>
            <%}%>
        </select></td>
    </tr>
    <tr>
    <td colspan="3" align="left" valign="top">&nbsp;&nbsp;&nbsp;</td>
    </tr>
    <tr>
    <td colspan="3" align="left" valign="top"> <hr align="center" width="100%" size="1"
noshade="noshade" />
    </td>
    </tr>
</table>
<table width="100%" border="0" cellspacing="0" cellpadding="0">
<tr>
<td>
    <input name="AcaoA" type="submit" class="rodape" id="AcaoA" onclick="return v.exec()"
value="Alocar" /> </td>
    <td><input name="Limpar" type="reset" class="rodape" id="Limpar" value="Limpar" /></td>
    <td>&nbsp;&nbsp;&nbsp;</td>
    <td>
    <input name="IrPara" type="button" id="IrPara" onclick="IrPara()" value="Voltar para
Incuba&ccedil;&atilde;o" /> </td>

```

```

    </tr>
</table>
</form>
<jsp:include page="../../../include/incubacao.jsp" flush="true"/>
<script>
// descrição da estrutura de campos de formulário
var a_fields = {
    'MatrUsu':{'l':'Matrícula','r':true,'f':'integer','t':'MatrUsu','mn':8},
    'NomeUsu':{'l':'Nome','r':true,'f':'alphaesp','t':'NomeUsu'},
    'CodGru':{'l':'Grupo','r':true,'f':'integer','t':'CodGru'},
    'SenCrt':{'l':'Senha','r':true,'f':'alphanum','t':'SenCrt','mn':6},
    'SenCrt1':{'l':'Confirmação da
Senha','r':true,'f':'alphanum','t':'SenCrt1','m':'SenCrt','mn':6},
    'TelRes':{'l':'Telefone Residencial','r':false,'f':'integer','t':'TelRes','mn':7},
    'TelCom':{'l':'Telefone Comercial','r':false,'f':'integer','t':'TelCom','mn':7},
    'TelCel':{'l':'Telefone Celular','r':false,'f':'integer','t':'TelCel','mn':7},
    'Email':{'l':'E-mail','r':true,'f':'email','t':'Email','mn':11},
},

o_config = {
    'to_disable' : ['Acao', 'Limpar'],
    'alert' : 1
}

// chamada do construtor do validator
var v = new validator('MainForm', a_fields, o_config);

</script>
</body>
</html>

```

17-incbandes.jsp

```

<%@ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>SISINC - Cadastrar Bandeja em Incubadora</title>
<link href="estilos/principal.css" rel="stylesheet" type="text/css" />
<!-- verifica se suporta javascript
<script>function lastmod() {}</script>
<!-- validator do formulário
<script language="javascript" type="text/javascript" src="scripts\Validator.js"></script>
<!-- caminho botão adicional -->
<link rel="shortcut icon" href="frango.jpg">
<script language="javascript" type="text/javascript">
function IrPara() {parent.location='jsp/incubacao/inccad.jsp';}
</script>
<script>
function mudaEstadoIF(){
    document.MainForm.CodBan.disabled=false;
    document.MainForm.Bannas.disabled=true;
    document.MainForm.Acao1.disabled=false;
    document.MainForm.Acao2.disabled=true;
}

function mudaEstadoIT(){
    document.MainForm.CodBan.disabled=true;
    document.MainForm.Bannas.disabled=true;
    document.MainForm.Acao1.disabled=true;
    document.MainForm.Acao2.disabled=true;
}

function mudaEstadoNF(){
    document.MainForm.CodBan.disabled=true;
    document.MainForm.Bannas.disabled=false;
    document.MainForm.Acao1.disabled=true;
    document.MainForm.Acao2.disabled=false;
}

function mudaEstadoNT(){
    document.MainForm.CodBan.disabled=true;
    document.MainForm.Bannas.disabled=true;
    document.MainForm.Acao1.disabled=true;
    document.MainForm.Acao2.disabled=true;
}

```

```

}
</script>
</head>
<%
controladora.ControladoraConsulta cont=new controladora.ControladoraConsulta();

java.util.Vector CodBan,CodBanN,bannas,filho,pai,status;
CodBan=cont.retornaBandejaAlocadaI();
bannas=cont.retornaBandejaAlocadaN();

String filhoX = request.getParameter("filho");

if (filhoX==null){
    filho=cont.retornaIncubadora();
    status=cont.retornaStatus();
    pai=cont.retornaPosEnc();
}else{
    filho=cont.retornaIncubadora();
    status=cont.retornaStatus();
    pai=(java.util.Vector) request.getAttribute("encs");
}
%>
<body>
<form id="MainForm" name="MainForm" method="post" action="IncubadoraBandejaDes">
    <table width="100%" border="0" cellspacing="5" cellpadding="0">
        <tr>
            <td colspan="3" class="titulo">Desalocar Bandeja em Incubadora/Nascedouro
            </td>
        </tr>
        <tr>
            <td colspan="3">&nbsp;</td>
        </tr>
        <tr>
            <td colspan="3" align="left" valign="top"><hr align="center" width="100%" size="1"
noshade="noshade" /></td>
        </tr>
        <tr>
            <td colspan="3" align="left" valign="top"><em><strong>DESALOCAR</strong>
            </em></td>
        </tr>
        <tr>
            <td align="left" valign="top">&nbsp;</td>
            <td align="left" valign="top"><input type="checkbox" name="checkbox" value="checkbox"
onclick="if(this.checked){mudaEstadoIF();}else{mudaEstadoIT();}"/>
            Na Incubadora</td>
            <td align="left" valign="top"><input type="checkbox" name="checkbox2" value="checkbox"
onclick="if(this.checked){mudaEstadoNF();}else{mudaEstadoNT();}"/>
            No Nascedouro</td>
        </tr>
        <tr>
            <td width="20%" align="left" valign="top"><em><strong>Selecione
Bandeja:</strong></em></td>
            <td width="25%" align="left" valign="top"><select name="CodBan" id="CodBan"
disabled="disabled" >
                <%
                String aux = "";
                int index = 0;
                for(int i = 0 ; i < CodBan.size() ; i++)
                {
                    aux = (String) CodBan.elementAt(i);
                    index = aux.indexOf('-');
                }
                %>
                <option value="<%=aux.substring(0,index-1)%>"><%=aux.substring(index+1)%></option>
                <%=aux.substring(0,index-1)%>
            </select>
        </td>
            <td width="55%" align="left" valign="top"> <select name="Bannas" id="Bannas"
disabled="true">
                <%
                aux = "";
                index = 0;
                for(int i = 0 ; i < bannas.size() ; i++)
                {
                    aux = (String) bannas.elementAt(i);

```

```

        index = aux.indexOf('-');
    %>
    <option value="<%=aux.substring(0,index-1)%>"><%=aux.substring(index+1)%></option>
    <%}%>
</select>
</td>
</tr>
<tr>
    <td colspan="3" align="left" valign="top"> <hr align="center" width="100%" size="1"
noshade="noshade" />
    </td>
</tr>
</table>
<table width="100%" border="0" cellspacing="0" cellpadding="0">
    <tr>
        <td><input name="Acao" type="submit" class="rodape" id="Acao" onclick="return v.exec()"
value="Desalocar" />
        </td> <td>&nbsp;&nbsp;&nbsp;</td>
        <td><input name="Limpar" type="reset" class="rodape" id="Limpar" value="Limpar" /></td>
        <td>
            <input name="IrPara" type="button" id="IrPara" onclick="IrPara()" value="Voltar para
Incuba&ccedil;&atilde;o" /> </td>
    </tr>
</table>
</form>
<jsp:include page="../../include/incubacao.jsp" flush="true"/>
<script>
// descrição da estrutura de campos de formulário
var a_fields = {
    'MatrUsu':{'l':'Matrícula','r':true,'f':'integer','t':'MatrUsu','mn':8},
    'NomeUsu':{'l':'Nome','r':true,'f':'alphaesp','t':'NomeUsu'},
    'CodGru':{'l':'Grupo','r':true,'f':'integer','t':'CodGru'},
    'SenCrt':{'l':'Senha','r':true,'f':'alphanum','t':'SenCrt','mn':6},
    'SenCrt1':{'l':'Confirmação da
Senha','r':true,'f':'alphanum','t':'SenCrt1','m':'SenCrt','mn':6},
    'TelRes':{'l':'Telefone Residencial','r':false,'f':'integer','t':'TelRes','mn':7},
    'TelCom':{'l':'Telefone Comercial','r':false,'f':'integer','t':'TelCom','mn':7},
    'TelCel':{'l':'Telefone Celular','r':false,'f':'integer','t':'TelCel','mn':7},
    'Email':{'l':'E-mail','r':true,'f':'email','t':'Email','mn':11},
},

o_config = {
    'to_disable' : ['Acao', 'Limpar'],
    'alert' : 1
}

// chamada do construtor do validator
var v = new validator('MainForm', a_fields, o_config);

</script>
</body>
</html>

```

18-inccad.jsp

```

<%@ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>SISINC - Cadastrar Incubação</title>
<link href="estilos/principal.css" rel="stylesheet" type="text/css" />
<script>
    function lastmod() {}
</script>
<script src="scripts\cancelar.js"></script>
</head>
<!-- JSPs -->
<%
controladora.ControladoraConsulta auxCons = new controladora.ControladoraConsulta();
javaUtil.TransferData transf=new javaUtil.TransferData();
//retorna as listas de projetos e bandejas //
java.util.Vector projeto,bandeja;
projeto = auxCons.retornaProjeto();

```

```

bandeja = auxCons.retornaBandeja();
String cod=request.getParameter("CodBan");
String incb="",a="",incubacao,entmin="",inc="",pos="",codpos="",esp="";
String dataPrev="",date="";

//retorna a posição da bandeja no momento não esta na pagina//
if(cod!=null){
    int ind1,ind2;
    incb=auxCons.retornaPosBan(cod);
    System.out.println("incb:"+incb+"cod:"+cod);
    ind1=incb.indexOf("-");
    ind2=incb.indexOf(":");
    inc=incb.substring(0,ind1-1);
    codpos=incb.substring(ind1+2,ind2-1);
    pos=incb.substring(ind2+2);
    inc=inc.replace( ' ', '_');
}
//*****

//pega o valor de CodProj direto*****//
//do servlet se ele nao for nulo carrega a pagina como estava//
String proj=request.getParameter("CodProj");
incubacao=(String)request.getAttribute("incubacao");

java.util.Vector entidade=new Vector();

if(proj!=null){
    entmin=(String)auxCons.retornaEntMin(proj);
    entmin=entmin.substring(4);
    entidade=auxCons.retornaEntidadeProj(proj);
    dataPrev=auxCons.retornaDataEspecie(proj)+"";
    esp=auxCons.retornaEspProjeto(proj);
}

else{
    a="1 - ";
    entidade.addElement(a);
}

}

%>
<%@page import="java.util.*"%>
<%!
Calendar tData;
int dia;
int mes;
int ano;
String mensagem;
%>
<%

//funcao que joga data no campo data de inicio da incubacao//
tData=Calendar.getInstance();
dia=tData.get(Calendar.DATE);
mes=tData.get(Calendar.MONTH)+1;
ano=tData.get(Calendar.YEAR);
mensagem="";
mensagem=mensagem+dia+"/";
mensagem=mensagem+mes+"/";
mensagem=mensagem+ano;
String msg=(String)request.getAttribute("msg");
%>
<%
int col=0;
int lin=0;

String tam=(String)request.getAttribute("tam");
String aux3 = "";
int marcax=0,marcay=0;

if(tam!=null){

int index3 = 0;

```

```

        aux3 = tam;
        index3 = aux3.indexOf('-');
        System.out.println("tam:"+tam);
        lin=Integer.parseInt(aux3.substring(0,index3));
        col=Integer.parseInt(aux3.substring(index3+1));
    }
    int[][] posmark=new int[lin][col];
    if(tam==null){
        System.out.println("tam é null");
    }else{
        //se tam não nulo
        //bc é o vetor com as posições ocupadas
        Vector bandejaCheia=(Vector)request.getAttribute("bc");
        System.out.println("bandeja cheia"+bandejaCheia+"tam vale:"+tam);

        for(int x=0;x<lin;x++){
            for(int y=0;y<col;y++){
                posmark[x][y]=0;
            }
        }

        //pega todos os elementos do vetor
        for(int i=0;i<bandejaCheia.size();i++){
            aux3=(String)bandejaCheia.elementAt(i);

            int index1=aux3.indexOf("-");
            int index2=aux3.indexOf(":");
            marcax=Integer.parseInt(aux3.substring(index1+1,index2));
            marcay=Integer.parseInt(aux3.substring(index2+1));
            posmark[marcax][marcay]=1;
        }
    }

%>
<!-- fim JSPs do topo -->
<body>

<form name="MainForm" method="post" action="Incubacao">

<input type="hidden" name="cordenada" value=<%=tam%>>
<input type="hidden" name="CodEncBan" value=<%=codpos%>>
<input type="hidden" name="Data" <%if(proj!=null) %> value="<%=dataPrev%>">

<!-- primeira parte -----
->
    <table width="100%" border="0" cellspacing="5" cellpadding="0">
        <tr>
            <td colspan="4" class="titulo" bgcolor="#FFEEEC">Cadastrar Nova Incuba&ccedil;&atilde;o
        </td>
        </tr>
        <tr>
            <td colspan="4">&nbsp;&nbsp;&nbsp;&nbsp;</td>
        </tr>
        <tr >
            <td width="193" align="left" valign="top"><strong>Projeto:</strong></td>
            <td colspan="2" align="left" valign="top"><select name="CodProj"
onchange="document.MainForm.submit();" >
                <%
                String aux = "";
                int index = 0;
                for(int i = 0 ; i < projeto.size() ; i++)
                {
                    aux = (String) projeto.elementAt(i);
                    index = aux.indexOf('-');
                    if(proj==null){ %>
                        <option value=<%=aux.substring(0,index-1) %>><%=aux.substring(index+2) %></option>
                    }else{
                        a=aux.substring(0,index-1);
                        %>
                        <%if(a.compareTo(proj)==0) { %>

```

```

        <option value= <%=aux.substring(0,index-1) %> selected><%=aux.substring(index+2)
%></option>
        <%
                                }else{%>
        <option value= <%=aux.substring(0,index-1) %>><%=aux.substring(index+2) %></option>
        <%} %>
        <% }%>
        <%}%>
    </select> <strong>Especie:</strong> <input name="CodEspecie" type="text"
disabled="disabled" size="12" maxlength="20" value=<%=esp%>>
</td>
<td width="355" rowspan="13" align="center" valign="middle" bgcolor="#EEEEEE" >
    <!-- tabela dinamica de bandejas----->
----->
    <table border="1" cellspacing="0" cellpadding="0">
    <%

        if(request.getParameter("CodBan")!=null && lin>0 )out.println("<b>C&ocirc;e; digo da
Bandeja: </b>"+request.getParameter("CodBan"));
        int x,y;
        String nome="";

        for(x=0;x<lin;x++){
        %>
        <tr>
            <td>
                <%for (y=0;y<col;y++){ nome=x+"-"+y;%>
                <td>
                    <%if (posmark[x][y]==1){%>
                    <input type="checkbox" disabled="disable" checked="checked" name=<%=nome%> >
                    <%}else{%>
                    <input type="checkbox" name=<%=nome%> >
                    <%}%>
                </td>
                <%}%>
                <td bgcolor="#cccccc"><%=x+1 %>&nbsp;&nbsp;&nbsp;</td>
            </tr>
            <%}%>
            <tr>
                <% for (x=0;x<col;x++){%>
                <td bgcolor="#cccccc"><%=x+1 %></td>
                <% } %>
            </tr>
        </table></td>
    </tr>
    <!-- segunda parte----->
    <tr >
        <td align="left" valign="top"><strong>Tipo de Entidade:</strong></td>
        <td colspan="2" align="left" valign="top"><input name="CodTipoEnt" type="text"
disabled="disabled" size="35" maxlength="50"<%if(proj!=null) %> value=<%=entmin%> /></td>
    </tr>
    <tr>
        <td align="left" valign="top"><strong>Incubac&atilde;o:</strong></td>
        <td colspan="2" align="left" valign="top"><strong>
            <input name="CodInc" type="text" size="35" maxlength="50" <%if(incubacao!=null)%>
value=<%=incubacao%> >
</strong></td>
    </tr>
    <tr>
        <td colspan="3" align="left" valign="top"><hr align="center" width="100%" size="1"
noshade="noshade" /></td>
    </tr>
    <tr>
        <td align="left" valign="top"><strong>Data de In&iacute;cio:</strong></td>
        <td colspan="2" align="left" valign="top"><input name="DataIniInc" type="text"
id="DataIniProj2" tabindex="3" disabled="true" size="10" maxlength="10" value=<%=mensagem
%>></td>
    </tr>
    <tr>
        <td align="left" valign="top">Previs&atilde;o de T&eacute;rmino: </td>
        <td colspan="2" align="left" valign="top"><input name="DataPrevInc" type="text"
disabled="true" size="10" maxlength="10"
<%if((String)request.getAttribute("DataPrevInc")!=null) {
date=transf.DataPApresentar((String)request.getAttribute("DataPrevInc"));}%> value=<%=date%>
></td>
    </tr>

```

```

<tr>
  <td colspan="3" align="left" valign="top"><hr align="center" width="100%" size="1"
noshade="noshade" /></td>
</tr>
<tr>
  <td align="left" valign="top"><em><strong>Seleccione Entidade:</strong></em></td>
  <td colspan="2" align="left" valign="top"><select name="CodEnt" >
    <%
      String aux2 = "";
      int index2 = 0;

      for(int i = 0 ; i < entidade.size() ; i++)
      {
        aux2 = (String) entidade.elementAt(i);
        index2 = aux2.indexOf('-');
        if(proj!=null){
          String aux4=aux2.substring(0,index2-1);
          String auxEnt=request.getParameter("CodEnt");
          if(auxEnt==null){
            auxEnt="1 - ";
          }
          if(aux4.compareTo(auxEnt)==0){ %>
            <option value= <%=aux2.substring(0,index2-1) %> selected<%=aux2.substring(index2+2)
%></option>
            <%}else{ %>
            <option value=<%=aux2.substring(0,index2-1) %><%=aux2.substring(index2+2)
%></option>
            <%}
          }else{
            %>
            <option value=<%=aux2.substring(0,index2-1) %><%=aux2.substring(index2 +2)
%></option>
            <%
              } %>
            <%} %>
          </select> </td>
</tr>
<tr>
  <td align="left" valign="top"><em><strong>Seleccione Bandeja:</strong></em></td>
  <td colspan="2" align="left" valign="top"><select name="CodBan"
onchange="document.MainForm.submit();" >
    <%
      String aux1 = "";
      int index1 = 0;

      for(int i = 0 ; i < bandeja.size() ; i++)
      {
        aux1 = (String) bandeja.elementAt(i);
        index1 = aux1.indexOf('-');
        if(proj!=null){

          String b=request.getParameter("CodBan");
          if(b==null){b=""};a=aux1.substring(0,index1-1);
          if(a.compareTo(b)==0){ %>
            <option value=<%=aux1.substring(0,index1-1) %> selected<%=aux1.substring(index1+2)
%></option>
            <%}else{
              %>
            <option value=<%=aux1.substring(0,index1-1) %><%=aux1.substring(index1+2)
%></option>
            <%} %>
            <%}else{%>
            <option value=<%=aux1.substring(0,index1-1) %><%=aux1.substring(index1+2) %>
            <%} %>
            <%} %>
          </select> <input type="submit" class="rodape" name="acao" value="Buscar Bandeja">
        </td>
</tr>
<tr>
  <td height="24" align="left" valign="top"><em><strong>Na Incubadora:</strong></em></td>
  <td colspan="2" align="left" valign="top"><input name="CodEnc" type="text"
disabled="true" size="50" maxlength="100" value=<%=inc%> ></td>
</tr>
<tr>

```

```

        <td colspan="3" align="left" valign="top"><input name="Reposicionar" type="button"
class="rodape" id="Reposicionar" tabindex="7" value="Reposicionar" /></td>
    </tr>
    <tr>
        <td colspan="3" align="left" valign="top"><hr /></td>
    </tr>
    <tr>
        <td colspan="4" align="left" valign="top"></td>
    </tr>
    <tr>
        <td height="24" colspan="4" align="left" valign="top"><table width="100%" border="0"
cellspacing="0" cellpadding="0">
            <tr>
                <td><input name="acao" type="submit" class="rodape" value="Incubar" /></td>
                <td><input name="Limpar" type="reset" class="rodape" value="Limpar" /></td>
            </tr>
        </table></td>
    </tr>
</table>
</form>
</body>
</html>

```

19-login/cabecalho.jsp

```

<%@ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<meta name="robots" content="noindex, nofollow" />
<meta name="description" content="Frame de cabeçalho da página de Login." />
<title>SISINC - Cabe&ccedil;alho</title>
<link href="../../../estilos/cabecalho.css" rel="stylesheet" type="text/css" />
</head>

<body>
<div id="cabecalho">
    <div align="center"></div>
</div>
</body>
</html>

```

20-login/index.jsp

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<meta name="robots" content="noindex, nofollow" />
<meta name="description" content="Frameset de Login do SISINC." />
<title>SISINC - Sistema de Controle de Incuba&ccedil;&atilde;o Experimental</title>
</head>

<frameset rows="80,*" cols="*" frameborder="no" border="0" framespacing="0">
    <frame src="jsp/login/cabecalho.jsp" name="cabecalho" scrolling="No" noresize="noresize"
id="cabecalho" title="cabecalho" />
    <frameset cols="160,*" frameborder="no" border="0" framespacing="0">
        <frame src="jsp/login/menu.jsp" name="menu" scrolling="No" noresize="noresize" id="menu"
title="menu" />
        <frame src="jsp/login/principal.jsp" name="principal" scrolling="Yes" id="principal"
title="principal" />
    </frameset>
</frameset>
<noframes>
</noframes>
<body>
</noframeset>

```



```

</tr>
<tr>
  <td height="138" align="left" valign="top">
    <span class="tag">Matr&iacute;cula: </span>
    <label>
      <input name="MatrUsu" type="text" id="MatrUsu" size="8" maxlength="8"
onKeyUp="doCheckInteger(MainForm.MatrUsu)" <%if(login!=null){%> value="<%=login%>" <%}%> />
    </label>
    <p>
      <span class="tag">Senha: </span>
      <br />
      <input name="SenCrt" type="password" id="SenCrt" size="8"
onBlur="preTime (MatrUsu.value,SenCrt.value, 0)" />
    </p>
    <label>
      <input name="Acao" type="submit" class="tag" id="Acao" value="Entrar" />
    </label>
  </td>
</tr>
</table>
</td>
<td width="502"><table width="100%" border="0" cellspacing="8" cellpadding="8">
  <tr>
    <td align="left" valign="top" class="titulo"><div align="center"
class="titulo">
      <div align="left" class="titulo">Instru&ccedil;&otilde;es</div>
    </div>
  </td>
</tr>
<tr>
  <td height="138" align="left" valign="top">
    <div align="justify">
      <span class="body">Para entrar no SISINC voc&ecirc; deve ter
sido cadastrado, se ainda n&atilde;o possui sua matr&iacute;cula cadastrada favor entrar em
contato com um dos Administradores do Sistema.
      <br />
      <br />
      Clique aqui para obter mais ajuda.<br />
      <br />
      Clique aqui para entrar em contato com o Administrador.</span>
    <br />
  </div>
  </td>
</tr>
</table>
</td>
</tr>
</table>
</form>
</td>
</tr>
</table>
<!-- chamada da p&agrave;gina de rodap&eacute; -->
<jsp:include page="../include/rodape.jsp" flush="true"/>
</body>
</html>

```

```

23-menu/administrador
<%@ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<meta name="robots" content="noindex, nofollow" />
<meta name="description" content="Menu." />
<meta name="author" content="Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano
Padilha" />
<title>SISINC - Menu</title>
<link href="../../estilos/submenu.css" rel="stylesheet" type="text/css" />
<!-- retorna submenu para menu -->
<script language="JavaScript" src="../../scripts/menu.js"></script>
</head>
<body>

```



```

<meta name="robots" content="noindex, nofollow" />
<meta name="description" content="Menu." />
<meta name="author" content="Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano
Padilha" />
<title>SISINC - Menu</title>
<link href="../../estilos/menu.css" rel="stylesheet" type="text/css" />
<!-- retorna submenu para menu -->
<script language="JavaScript" src="../../scripts/menu.js"></script>
</head>

<body>
<br />
<ul id="primary-nav">
  <li class="menuparent"><a href="gerenciamento.jsp"target="principal" >Gerenciamento</a>
    <ul>
    </ul>
  </li>
  <li class="menuparent"><a href="cadastro.jsp"target="principal" >Cadastro</a>
    <ul>
    </ul>
  </li>
  <li class="menuparent"><a href="projeto.jsp"target="principal" >Projeto</a>
    <ul>
    </ul>
  </li>
  <li class="menuparent"><a href="incubacao.jsp"target="principal"
>Incubação</a>
    <ul>
    </ul>
  </li>
  <li class="menuparent"><a href="nascimento.jsp"target="principal" >Nascimento</a>
    <ul>
    </ul>
  </li>
  <li class="menuparent"><a href="relatorio.jsp"target="principal" >Relatório</a>
    <ul>
    </ul>
  </li>
  <!-- <li class="menuparent"><a href="../../sair/sair.jsp"target="_parent" >Sair</a> -->
  <li class="menuparent"><a href="../../sair/sair.jsp"target="principal" >Sair</a>
    <ul>
    </ul>
  </li>
</ul>
<font color="#ffffff"size="1">
<br />
&nbsp;&nbsp;&nbsp;Início do Acesso
<br />
&nbsp;&nbsp;&nbsp;<%= mensagem1%>
<br />
&nbsp;&nbsp;&nbsp;<%= mensagem %>
</font>
</body>
</html>

```

28-menu/administrador/relatórios.jsp

```

<%@ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<meta name="robots" content="noindex, nofollow" />
<meta name="description" content="Menu." />
<meta name="author" content="Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano
Padilha" />
<title>SISINC - Menu</title>
<link href="../../estilos/submenu.css" rel="stylesheet" type="text/css" />
<!-- retorna submenu para menu -->
<script language="JavaScript" src="../../scripts/menu.js"></script>
</head>

<body>

```



```

if(cod!=null){
    int ind1,ind2;
    incb=auxCons.retornaPosBanNas(cod);
    System.out.println("incb:"+incb+"cod:"+cod);
    ind1=incb.indexOf("-");
    ind2=incb.indexOf(":");
    inc=incb.substring(0,ind1-1);
    codpos=incb.substring(ind1+2,ind2-1);
    pos=incb.substring(ind2+2);
    inc=inc.replace( ' ', '_');
}

//pega o valor de CodProj direto*****//
//do servlet se ele nao for nulo carrega a pagina como estava//
String proj=request.getParameter("CodProj");
incubacao=(String)request.getAttribute("incubacao");
String ent=(String)request.getAttribute("CodEnt");

java.util.Vector entidade=new Vector();
java.util.Vector listaInc=new Vector();
codnasc=(String)request.getAttribute("codnasc");
hcodnasc=codnasc;
if(proj!=null){
    entmin=(String)auxCons.retornaEntMin(proj);
    entmin=entmin.substring(4);
    entidade=auxCons.retornaEntidadeProj(proj);
    dataPrev=auxCons.retornaDataEspecieNas(proj)+"";
    esp=auxCons.retornaEspProjeto(proj);
    listaInc=auxCons.retornaIncubacao(proj);
}

}else{

    a=" - ";
    entidade.addElement(a);
    listaInc.addElement(a);
}

//*****//
%>
    <%@page import="java.util.*"%>
    <%!
    Calendar tData;
    int dia;
    int mes;
    int ano;
    String mensagem;
    %>
    <%

    //funcao que joga data no campo data de inicio da incubacao//
    tData=Calendar.getInstance();
    dia=tData.get(Calendar.DATE);
    mes=tData.get(Calendar.MONTH)+1;
    ano=tData.get(Calendar.YEAR);
    mensagem="";
    mensagem=mensagem+dia+"/";
    mensagem=mensagem+mes+"/";
    mensagem=mensagem+ano;
    String msg=(String)request.getAttribute("msg");
%>
<%
//*****//
    int col=0;
    int lin=0;

    String tam=(String)request.getAttribute("tam");
    String aux3 = "";
    int marcax=0,marcay=0;

    if(tam!=null){

        int index3 = 0;

```

```

        aux3 = tam;
        index3 = aux3.indexOf('-');
        System.out.println("tam:"+tam);
        lin=Integer.parseInt(aux3.substring(0,index3));
        col=Integer.parseInt(aux3.substring(index3+1));

    }
    int[][] posmark=new int[lin][col];
    if(tam==null){
        System.out.println("tam é null");
    }else{
        //se tam não nulo
        //bc é o vetor com as posições ocupadas
        Vector bandejaCheiaEnt=(Vector)request.getAttribute("bc");
        System.out.println("bandeja cheia"+bandejaCheiaEnt+"tam vale:"+tam);

        for(int x=0;x<lin;x++){
            for(int y=0;y<col;y++){
                posmark[x][y]=0;
            }
        }

        //pega todos os elementos do vetor
        for(int i=0;i<bandejaCheiaEnt.size();i++){
            aux3=(String)bandejaCheiaEnt.elementAt(i);

            index1=aux3.indexOf("-");
            index2=aux3.indexOf(":");
            marcax=Integer.parseInt(aux3.substring(index1+1,index2));
            marcay=Integer.parseInt(aux3.substring(index2+1));
            posmark[marcax][marcay]=1;

        }
    }

    if(incubacao!=null && ent != null){
        String quantidade=auxCons.buscarQuantidade(incubacao,ent);
        aux3=quantidade;
        index1=aux3.indexOf("-");
        qTotal=aux3.substring(0,index1);
        qEntidade=aux3.substring(index1+1);
        HQtEntidade=qEntidade;
    }
}
%>
<!--*****
*
*fim JSPs do topo
*e inicio do html
*****-->

<body>

<form name="MainForm" method="post" action="Nascimento">

<input type="hidden" name="cordenada" value="<%=tam%>"/>
<input type="hidden" name="CodEncBan" value="<%=codpos%>"/>
<input type="hidden" name="Data" <%if(proj!=null) %> value="<%=dataPrev%>"/>
<input type="hidden" name="qEntidade" value="<%=qEntidade%>"/>
<input type="hidden" name="HQtEntidade" value="<%=HQtEntidade%>"/>
<input type="hidden" name="HCodNascimento" <%if(hcodnasc!=null) %> value="<%=hcodnasc%>" />

<!--*****
*1° parte
*informação dos totais
*por entidade e por incubação
*****--

>

<table width="100%" border="0" cellspacing="5" cellpadding="0">
<tr>

```

```

        <td colspan="4" class="titulo" bgcolor="#EAF5E5">Cadastrar Transferência </td>
    </tr>
    <tr>
        <td colspan="4">&nbsp;</td>
    </tr>
    <tr >
        <td colspan="3" align="left" valign="top">|-----Restam para transferir-----|
        -----|</td>
        <td width="335" rowspan="18" align="center" valign="middle" bgcolor="#EEEEEE">
            <!--*****
            *
            *montagem da tabela dinâmica da bandeja
            *
            *****-->
    >

    <table border="1" cellspacing="0" cellpadding="0">
        <%

            if(request.getParameter("CodBan")!=null && lin>0 )out.println("<b>Código da
Bandeja: </b>"+request.getParameter("CodBan"));
            int x,y;
            String nome="";

            for(x=0;x<lin;x++){
                <%>
                <tr>
                    <%for(y=0;y<col;y++){ nome=x+"0"+y;%>
                    <td>
                        <%if (posmark[x][y]==1){%>
                            <input type="checkbox" disabled="disable" checked name=<%=nome%> >
                        <%}else{%>
                            <input type="checkbox" name=<%=nome%> onclick="if (this.checked
)}if(document.MainForm.QtEntidade.value > 0){

                document.MainForm.QtEntidade.value=document.MainForm.QtEntidade.value-1;

                document.MainForm.QtTotal.value=document.MainForm.QtTotal.value-1;

                document.MainForm.HQtEntidade.value=document.MainForm.HQtEntidade.value-1;
                }
                else{
                    alert('Você já transferiu todos os ovos desta entidade!');
                    this.checked=false;
                }
                } else {almenta();}"/>
                <%}%>
            </td>
            <%}%>
            <td bgcolor="#cccccc"><%=x+1 %>&nbsp;</td>
        </tr>
        <%}%>
        <tr>
            <% for(x=0;x<col;x++){%>
            <td bgcolor="#cccccc"><%=x+1 %></td>
            <%} %>
        </tr>
    </table>
    <!--*****
    *
    *acaba montagem da tabela da bandeja
    *
    *****-->
    >

        </td>
    </tr>
    <tr >
        <td align="left" valign="top" bgcolor="#EEEEED"><strong>Total:</strong></td>
        <td colspan="2" align="left" valign="top"><input name="QtTotal" type="text"
disabled="disabled" size="10" maxlength="20"><%=qTotal%>"/></td>
    </tr>
    <tr >
        <td align="left" valign="top" bgcolor="#EEEEED"><strong>Esta entidade:</strong></td>

```



```

        <td width="31" colspan="2" align="left" valign="top">&nbsp;</td>
    </tr>
    <tr>
        <td align="left" valign="top"><strong>Nascimento:</strong>&nbsp;</td>
        <td colspan="2" align="left" valign="top"><input name="CodNascimento" type="text"
disabled="disabled" size="12" maxlength="20" <%if (codnasc!=null) %> value="<%=codnasc%>"
/></td>
        <td colspan="2" align="left" valign="top">&nbsp;</td>
    </tr>
    <tr>
        <td colspan="3" align="left" valign="top"><hr align="center" width="100%" size="1"
noshade="noshade" />
        <strong> </strong></td>
        <td colspan="2" align="left" valign="top">&nbsp;</td>
    </tr>
    <tr>
        <td align="left" valign="top"><strong>Data de In&iacute;cio:</strong></td>
        <td colspan="2" align="left" valign="top"><input name="DataIniInc" type="text"
id="DataIniProj" tabindex="3" disabled="true" size="10" maxlength="10" value="<%=mensagemem
%>></td>
    </tr>
    <tr>
        <td align="left" valign="top">Previs&atilde;o de T&eacute;rmino: </td>
        <td colspan="2" align="left" valign="top"><input name="DataPrevInc" type="text"
disabled="true" size="10" maxlength="10"
<%if ((String)request.getAttribute("DataPrevInc")!=null) {
date=transf.DataPApresentar((String)request.getAttribute("DataPrevInc"));}& %> value="<%=date%>"
/></td>
    </tr>
    <tr>
        <td colspan="3" align="left" valign="top"><hr align="center" width="100%" size="1"
noshade="noshade" /></td>
    </tr>
    <tr>
        <td align="left" valign="top"><em><strong>Selecione Entidade:</strong></em></td>
        <td colspan="2" align="left" valign="top"><select name="CodEnt" >
            <%
String aux2 = "";
index2 = 0;

for(int i = 0 ; i < entidade.size() ; i++)
{
    aux2 = (String) entidade.elementAt(i);
    index2 = aux2.indexOf('-');
    if(proj!=null){
        String aux4=aux2.substring(0,index2-1);
        String auxEnt=request.getParameter("CodEnt");
        if(auxEnt==null){
            auxEnt=" - ";
        }
        if(aux4.compareTo(auxEnt)==0) { %>
            <option value=" <%=aux2.substring(0,index2-1) %> selected"><%=aux2.substring(index2+2)
%></option>
            <%}else{ %>
            <option value="<%=aux2.substring(0,index2-1) %><%=aux2.substring(index2+2)
%></option>
            <%}

        }else{
            %>
            <option value="<%=aux2.substring(0,index2-1) %><%=aux2.substring(index2 +2)
%></option>
            <%
                } %>
            <%} %>
        </select> </td>
    </tr>
    <tr>
        <td align="left" valign="top"><em><strong>Selecione Bandeja:</strong></em></td>
        <td colspan="2" align="left" valign="top"><select name="CodBan"
onchange="document.MainForm.submit();" >
            <%
aux1 = "";
index1 = 0;

for(int i = 0 ; i < bandeja.size() ; i++)
{

```

```

        aux1 = (String) bandeja.elementAt(i);
        index1 = aux1.indexOf('-');
        if(proj!=null){

            String b=request.getParameter("CodBan");
            if(b==null){b=""};a=aux1.substring(0,index1-1);
            if(a.compareTo(b)==0){ %>
                <option value=<%=aux1.substring(0,index1-1) %> selected><%=aux1.substring(index1+2)
%></option>
                <%>else{

                    %>
                    <option value=<%=aux1.substring(0,index1-1) %>><%=aux1.substring(index1+2)
%></option>
                <%> } %>
                <%>else{ %>
                    <option value=<%=aux1.substring(0,index1-1) %>><%=aux1.substring(index1+2) %>
                    <%> } %>
                <%> } %>
            }
        }
    </select> <input type="submit" class="rodape" name="acao" value="Buscar Bandeja">
</td>
</tr>
<tr>
    <td height="24" align="left" valign="top"><em><strong>No Nascedouro:</strong></em></td>
    <td colspan="2" align="left" valign="top"><input name="CodEnc" type="text"
disabled="true" size="50" maxlength="100" value=<%=inc%> ></td>
</tr>
<tr>
    <td colspan="3" align="left" valign="top"><input name="Reposicionar" type="button"
class="rodape" id="Reposicionar" tabindex="7" value="Reposicionar" /></td>
</tr>
<tr>
    <td colspan="3" align="left" valign="top"><hr /></td>
</tr>
<tr>
    <td colspan="4" align="left" valign="top"></td>
</tr>
<tr>
    <td height="24" colspan="4" align="left" valign="top"><table width="100%" border="0"
cellspacing="0" cellpadding="0">
        <tr>
            <td><input name="acao" type="submit" class="rodape" value="Transferir" /></td>
            <td><input name="Limpar" type="reset" class="rodape" value="Limpar" /></td>
        </tr>
    </table></td>
</tr>
</table>
</form>
</body>
</html>

```

30-principal/cabecalho.jsp

```

<%@ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<meta name="robots" content="noindex, nofollow" />
<meta name="description" content="Frame de cabeçalho da página de Login." />
<meta name="author" content="Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano
Padilha" />
<title>SISINC - Cabe&ccedil;alho</title>
<link href="../../../estilos/cabecalho.css" rel="stylesheet" type="text/css" />
</head>

<body>
<div id="cabecalho">
    <div align="center"><a href="principal.jsp" target="principal"></a></div>
</div>
</body>

```

```

</html>

31- principal/index.jsp
<%@ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
<%@page import="java.util.*"%>
<%
classepersistente.Usuario auxUsuario = new classepersistente.Usuario();
javaUtil.TransferData td = new javaUtil.TransferData();
%>
<%
HttpSession s = request.getSession(false);
if(s != null) { auxUsuario = (classepersistente.Usuario) s.getAttribute("login"); }
%>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<meta name="robots" content="noindex, nofollow" />
<meta name="description" content="Frameset de entrada do SISINC." />
<meta name="author" content="Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano
Padilha" />
<title>SISINC - Sistema de Controle de Incuba&ccedil;&atilde;o Experimental</title>
</head>

<frameset rows="80,*" cols="*" frameborder="no" border="0" framespacing="0">
<frame src="jsp/principal/cabecalho.jsp" name="cabecalho" scrolling="No" noresize="noresize"
id="cabecalho" title="cabecalho" />

<frameset cols="160,*" frameborder="no" border="0" framespacing="0">
<%if(auxUsuario.getGrupo().equals("1")){%>
<frame src="jsp/menu/administrador/menu.jsp" name="menu" scrolling="No"
noresize="noresize" id="menu" title="menu" />
<%}else{%>
<%if(auxUsuario.getGrupo().equals("2")){%>
<frame src="jsp/menu/operador/menu.jsp" name="menu" scrolling="No" noresize="noresize"
id="menu" title="menu" />
<%}else{%>
<%if(auxUsuario.getGrupo().equals("3")){%>
<frame src="jsp/menu/consultor/menu.jsp" name="menu" scrolling="No" noresize="noresize"
id="menu" title="menu" />
<%}else{%>
<frame src="jsp/login/menu.jsp" name="menu" scrolling="No" noresize="noresize" id="menu"
title="menu" />
<%}}}%>

<frame src="jsp/principal/principal.jsp" name="principal" scrolling="Yes" id="principal"
title="principal" />
</frameset>
</frameset>
</noframes>
</noframes>
<body>
<noframeset>
</noframeset>
</body>
</html>

```

32-principal/javascript.jsp

```

<%@ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>SISINC - Validar Javascript</title>
<link href="../../estilos/principal.css" rel="stylesheet" type="text/css" />
</head>

<body>
<script language="JavaScript">

```

```

<!--
window.open("principal.jsp","principal");
// -->
</script>
</noscript>
<table width="100%" border="0" align="center" cellpadding="0" cellspacing="0">
  <tr class="cabecalho">
    <td height="20" align="center" valign="top">
      <div align="center" class="cabecalho">
        Erro: JavaScript Desabilitado </div></td>
    </tr>
  <tr>
    <td align="center" valign="top">&nbsp;</td>
  </tr>
  <tr>
    <td height="1507">
      <table width="600" height="100%" border="1" align="center" cellpadding="0"
cellspacing="0" bordercolor="#666666">
        <tr>
          <td height="41"><table width="100%" border="0" align="center" cellpadding="5"
cellspacing="5">
            <tr>
              <td><div align="center"><span class="titulo">ATEN&Ccedil;&Atilde;O! ISSO CAUSA
</span></div></td>
            </tr>
            <tr bordercolor="#666666">
              <td align="left" valign="middle"><ul>
                <li>N&atilde;o verifica&ccedil;&atilde;o dos dados inseridos nos
formul&aacute;rios.</li>
                <li>Funcionamento anormal e n&atilde;o retorno de erros de
grava&ccedil;&atilde;o. </li>
                <li>Perda de recursos adicionais de edi&ccedil;&atilde;o e
vizualiza&ccedil;&atilde;o. </li>
              </ul>
            </td>
          </tr>
        </table></td>
      </tr>
      <tr>
        <td align="center" valign="middle"><table width="100%" border="0" cellspacing="5"
cellpadding="5">
          <tr>
            <td><div align="center"><span class="titulo">HABILITE O
JAVASCRIPT</span></div></td>
          </tr>
          <tr>
            <td height="620" align="left" valign="middle"><b>Internet Explorer 6.X <br />
</b>
            <ol>
              <li>Selecione <em>Op&ccedil;&otilde;es da Internet (Internet Options)</em>
do menu <em>Ferramentas (Tools)</em>. &nbsp;</li>
              <li> Na caixa de di&aacute;logo <em>Op&ccedil;&otilde;es da Internet</em>
(<em>Internet Options</em>) selecione a barra <em>Seguran&ccedil;a</em>
(<em>Security</em>). &nbsp;</li>
              <li> Clique no bot&atilde;o <em>N&iacute;vel Personalizado (Custom
level)</em>. A caixa de di&aacute;logo com configura&ccedil;&otilde;es de seguran&ccedil;a
(<em>Security Settings</em>) aparece. &nbsp;</li>
              <li> Localize, nesta caixa, a categoria <em>Scripts</em>
(<em>Scripting</em>). &nbsp;</li>
              <li> Habilite as op&ccedil;&otilde;es <em>ativar scripts</em> (<em>active
scripting</em>), <em>permitir opera&ccedil;&otilde;es de colagem atrav&eacute;s de script
(allow paste operations via script)</em> e <em>scripts de miniaplicativos Java (scripting
of java applets)</em>. &nbsp;</li>
              <li> Clique OK na caixa de di&aacute;logo de <em>configura&ccedil;&otilde;es
de seguran&ccedil;a</em> (<em>Security Settings</em>) &nbsp;</li> e clique OK na caixa de
di&aacute;logo <em>Op&ccedil;&otilde;es da Internet (Internet Options)</em>. </li>
            </ol>
            <p><b>Internet Explorer 5.X </b></p>
            <ol>
              <li>Selecione <em>Op&ccedil;&otilde;es da Internet (Internet
Options) &nbsp;</em>do menu <em>Ferramentas</em> (<em>Tools</em>). &nbsp;</li>
              <li> Na caixa de di&aacute;logo <em>Op&ccedil;&otilde;es da Internet</em>
(<em>Internet Options</em>) selecione a barra <em>Seguran&ccedil;a</em>
(<em>Security</em>). &nbsp;</li>
            </ol>

```

```

                <li> Clique no bot&atilde;o <em>N&iacute;vel Personalizado (Custom
level)</em>. A caixa de di&aacute;logo <em>Configura&ccedil;&otilde;es de Seguran&ccedil;a
(Security</em> settings) aparece.&nbsp;</li>
                <li> Localize, nesta caixa, a categoria <em>Scripts</em>
<em>Scripting)</em></li>
                <li> Habilite as op&ccedil;&otilde;es <em>ativar scripts</em> (<em>active
scripting)</em>, <em>permitir opera&ccedil;&otilde;es de colagem atrav&eacute;s de script
(allow paste operations via script)</em> e <em>scripts de miniaplicativos Java (scripting
of java applets)</em>.&nbsp;</li>
                <li> Clique OK na caixa de di&aacute;logo de <em>configura&ccedil;&otilde;es
de seguran&ccedil;a</em> (<em>Security Settings)&nbsp;</em> e clique OK na caixa de
di&aacute;logo <em>Op&ccedil;&otilde;es da Internet (Internet Options)</em>.&nbsp;</li>
            </ol>
            <p><b>Internet Explorer 4.X </b></p>
            <ol>
                <li> Selecione <em>Op&ccedil;&otilde;es da Internet (Internet Options)
</em>do menu&nbsp;<em>Visualizar</em> (<em>View</em>)&nbsp;</li>
                <li> Clique na barra <em>Seguran&ccedil;a</em> (<em>Security)</em></li>
                <li> Clique <em>Personalizar</em> (<em>Custom</em>)</li>
                <li> Clique <em>Configura&ccedil;&otilde;es</em> (<em>Settings</em>)</li>
                <li> Localize a categoria <em>Scripts</em> (<em>Scripting)</em></li>
                <li> Clique <em>Habilitar</em> para&nbsp;<em>Ativar Scripts (Active
Scripting)</em>.</li>
                <li> Clique <em>OK</em>.</li>
                <li> Clique no bot&atilde;o <em>Recarregar</em> (<em>Reload</em>)</li>
            </ol>
            <p><b>Internet Explorer 3.X </b></p>
            <ol>
                <li> Selecione <em>Op&ccedil;&otilde;es </em>do menu <em>Visualizar
(View)</em>.</li>
                <li> Clique na barra&nbsp;<em>Seguran&ccedil;a (Security)</em>.</li>
                <li> Clique <em>Habilitar Programas Java (Enable Java Programs)</em>.</li>
                <li> Clique <em>OK</em>.</li>
                <li> Clique no bot&atilde;o <em>Recarregar (Reload)</em>.</li>
            </ol>
            <p><b>Netscape Navigator 5.X e 4.X </b></p>
            <ol>
                <li> Clique <em>Editar (Edit)</em>.</li>
                <li> Clique <em>Prefer&ecirc;ncias</em> (<em>Preferences</em>)</li>
                <li> Clique <em>Avan&ccedil;ado</em> (<em>Advanced</em>)</li>
                <li> Habilite as op&ccedil;&otilde;es <em>Habilitar Java (Enable Java),
Habilitar JavaScript (Enable JavaScript) </em>e<em> Habilitar JavaScript para mensagens
e not&iacute;cias (Enable JavaScript for Mail and News)</em>.</li>
                <li> Clique OK.</li>
            </ol>
            <p><b>Netscape Navigator 3.X </b></p>
            <ol>
                <li> Selecione <em>Op&ccedil;&otilde;es</em> (<em>Options) </em>do menu
<em>Editar (Edit)</em>.</li>
                <li> Clique na op&ccedil;&atilde;o<em> Prefer&ecirc;ncias de Rede</em>
(<em>Network Preferences)</em>.</li>
                <li> Clique <em>Linguagens</em> (<em>Languages)</em>.</li>
                <li> Habilite ambas as op&ccedil;&otilde;es <em>Habilitar Java (Enable
Java)</em> e <em>Habilitar JavaScript (Enable JavaScript)</em>.</li>
                <li> Clique <em>OK</em>.</li>
            </ol>
        </td>
    </tr>
    <tr>
        <td height="29" align="center" valign="middle">
            <form id="MainForm" name="MainForm" method="post" action=""
target="_self" onSubmit="javascript:history.go(0)">
                <label>
                    <input name="Recarregar" type="submit" class="tag" id="Recarregar"
value="Recarregar" />
                </label>
            </form>
        </td>
    </tr>
</table></td>
</tr>
</table>
</td>
</tr>
</table>
</noscript>
</body>

```

```
</html>
```

```
24-principal.jsp
```

```
<%@ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
<%@page import="java.util.*"%>
<%
classepersistente.Usuario auxUsuario = new classepersistente.Usuario();
controladora.ControladoraConsulta auxCons = new controladora.ControladoraConsulta();
javaUtil.TransferData td = new javaUtil.TransferData();
%>
<%@ include file="../../../scripts/damssg.jsp"%>
<%@ include file="../../../scripts/rdados.jsp"%>
<%
HttpSession s = request.getSession(false);
if(s != null) { auxUsuario = (classepersistente.Usuario) s.getAttribute("login"); }
%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<meta name="robots" content="noindex, nofollow" />
<meta name="description" content="Página de entrada do SISINC." />
<meta name="author" content="Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano
Padilha" />
<title>SISINC - Entrada</title>
<link href="../../../estilos/principal.css" rel="stylesheet" type="text/css" />
</head>

<body>
<table width="100%" border="0" align="center" cellpadding="0" cellspacing="0">
<tr class="cabecalho">
<td height="20" align="center" valign="top">
<div align="center" class="cabecalho">
<%=mensagem%>, Seja Bem-Vindo <%=auxUsuario.getNome() %> </div></td>
</tr>
<tr>
<td align="center" valign="top">&nbsp;</td>
</tr>
<tr>
<td height="19"><form id="login" name="login" method="post" action="principal.jsp">
<table width="600" height="100%" border="1" align="center" cellpadding="0"
cellspacing="0" bordercolor="#666666">
<tr>
<td><table width="100%" border="0" align="center" cellpadding="5" cellspacing="5">
<tr>
<td height="48" align="center" valign="middle"><div align="center">
<p>&Uacute;ltimo Acesso como <strong><%=auxUsuario.getDesGru() %></strong> em
<strong><%=td.DataPApresentar(auxUsuario.getUltLogon()) %></strong><br />
N&uacute;mero de Projetos: <a href="../../../Projeto"
target="principal"><%=aux.substring(0,index1) %></a><br />
</p>
</div></td>
</tr>
</table></td>
</tr>
<tr>
<td align="center" valign="middle"><table width="100%" border="0" cellspacing="5"
cellpadding="5">
<tr>
<td><div align="center"><span class="titulo">Quadro de Avisos</span></div></td>
</tr>
<tr>
<td align="center" valign="middle"><b>EM ANDAMENTO<br />
</b><a href="../../../Projeto"
target="principal"><%=aux.substring(index1+1,index2) %></a> Projetos<br />
<a href="../../../Incubacao"><%=aux.substring(index2+1,index3) %></a>
Incuba&ccedil;&otilde;es<br />
<a href="../../../Nascimento"><%=aux.substring(index3+1) %></a> Nascimentos</td>
</tr>
</table></td>
</tr>
</table>
```

```

        </form></td>
    </tr>
</table>
<!-- chamada da página de rodapé -->
<jsp:include page="../../include/rodape.jsp" flush="true"/>
</body>
</html>

```

25-pojcad.jsp

```

<%@ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
<%@page import="java.util.*"%>
<%@ include file="../../scripts/data.jsp"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<meta name="robots" content="noindex, nofollow" />
<meta name="description" content="Página de cadastro de projeto." />
<meta name="author" content="Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano
Padilha" />
<title>SISINC - Cadastrar Projeto</title>
<link href="estilos/principal.css" rel="stylesheet" type="text/css" />
<!-- validador do formulário -->
<script language="JavaScript" src="scripts/validator.js"></script>
<!-- limita entrada do formulário -->
<script language="JavaScript" src="scripts/form.js"></script>
<!-- máscara de entrada -->
<script language="JavaScript" src="scripts/dFilter.js"></script>
<!-- compara e valida duas datas -->
<script language="JavaScript" src="scripts/data.js"></script>
<!-- posiciona na primeira linha -->
<script language="JavaScript" src="scripts/focus.js"></script>
</head>
<%
controladora.ControladoraConsulta cont = new controladora.ControladoraConsulta();
// monta as listas Especie e Tipo de Entidade
java.util.Vector CodEsp,CodEnt;
CodEsp=cont.retornaEspecie();
CodEnt=cont.retornaTipoEntidade();
%>
<body OnLoad="placeFocus()">
<form id="MainForm" name="MainForm" method="post" action="Projeto" onSubmit="return v.exec()">
<table width="100%" border="0" cellspacing="5" cellpadding="0">
<tr>
<td colspan="3" class="titulo">Cadastrar Novo Projeto </td>
</tr>
<tr>
<td colspan="3">&nbsp;</td>
</tr>
<tr>
<td width="20%" align="left" valign="top"><strong>Sigla do Projeto:</strong></td>
<td width="80%" colspan="2" align="left" valign="top">
<input name="SiglaProj" type="text" id="SiglaProj" size="14" maxlength="14" />
<span class="descricao"> (ex.: 200601-A)</span> </td>
</tr>
<tr>
<td align="left" valign="top"><strong>Nome Completo:</strong></td>
<td colspan="2" align="left" valign="top">
<input name="NomeProj" type="text" id="NomeProj" size="50" maxlength="100"
onKeyUp="doCheckAlpha(MainForm.NomeProj)" />
</td>
</tr>
<tr>
<td align="left" valign="top"><strong>Entidade:</strong></td>
<td colspan="2" align="left" valign="top">
<select name="CodEnt" id="CodEnt">
<%
String aux = "";
int index = 0;
for(int i = 0 ; i < CodEnt.size() ; i++)
{
aux = (String) CodEnt.elementAt(i);
index = aux.indexOf('-');

```

```

        %>
        <option value=<%=aux.substring(0,index-1)%>><%=aux.substring(index+1)%></option>
        <%}%>
    </select>
    <br />
    <span class="atencao">CUIDADO:</span> <span class="descricao">Define a entidade para
    tratamentos, repeti&ccedil;&otilde;es, incuba&ccedil;&otilde;es e nascimentos de
    <strong>TODO</strong> o projeto. </span>
    </td>
</tr>
<tr>
    <td align="left" valign="top"><strong>Esp&eacute;cie de Ave: </strong></td>
    <td colspan="2" align="left" valign="top">
        <select name="CodEsp" id="CodEsp">
            <%
                aux = "";
                index = 0;
                for(int i = 0 ; i < CodEsp.size() ; i++)
                {
                    aux = (String) CodEsp.elementAt(i);
                    index = aux.indexOf('-');
                }
            %>
            <option value="<%=aux.substring(0,index-1)%>"><%=aux.substring(index+1)%></option>
            <%}%>
        </select>
        <br />
        <span class="atencao">CUIDADO:</span> <span class="descricao">Define a esp&eacute;cie
        para tratamentos, repeti&ccedil;&otilde;es, incuba&ccedil;&otilde;es e nascimentos de
        <strong>TODO</strong> o projeto.</span>
    </td>
</tr>
<tr>
    <td align="left" valign="top"><strong>Data de In&iacute;cio: </strong></td>
    <td colspan="2" align="left" valign="top">
        <input name="DataIniProj" type="text" id="DataIniProj" size="10" maxlength="10"
        value="<%=mensagem%>" onKeyDown="return dFilter(event.keyCode,this,'##/##/####') "
        onBlur="if(dataMI(DataPrevProj.value,DataIniProj.value)==false) alFocus(DataIniProj)" /></td>
</tr>
<tr>
    <td align="left" valign="top">Pevis&atilde;o de T&eacute;rmino: </td>
    <td colspan="2" align="left" valign="top">
        <input name="DataPrevProj" type="text" id="DataPrevProj" size="10" maxlength="10"
        onKeyDown="return dFilter(event.keyCode,this,'##/##/####') "
        onBlur="if(dataMI(DataPrevProj.value,DataIniProj.value)==false) alFocus(DataPrevProj)" /></td>
</tr>
<tr>
    <td align="left" valign="top">Descri&ccedil;&atilde;o:</td>
    <td colspan="2" align="left" valign="top">
        <textarea name="DescProj" cols="50" rows="2" id="DescProj"></textarea>
    </td>
</tr>
<tr>
    <td colspan="3" align="left" valign="top">
        <hr align="center" width="100%" size="1" noshade="noshade" />
    </td>
</tr>
</table>
<table width="100%" border="0" cellspacing="0" cellpadding="0">
    <tr>
        <td>
            <input name="Acao" type="submit" class="rodape" id="Acao" value="Criar" />
        </td>
        <td>
            <input name="Acao" type="reset" class="rodape" id="Acao" value="Limpar" />
        </td>
        <td>
            <input name="IrPara" type="button" id="IrPara" value="Alterar/Excluir"
            onClick="window.open('ProjetoAlterar','_self')" />
        </td>
    </tr>
</table>
</form>
<!-- chamada da pgina de rodap -->
<jsp:include page="../include/projeto.jsp" flush="true"/>
<script language="JavaScript">
// descrio da estrutura de campos para o validador

```

```

var a_fields = {
    'SiglaProj':{'l':'Sigla do Projeto','r':true,'f':'alphanum','t':'SiglaProj','mn':8},
    'NomeProj':{'l':'Nome Completo','r':true,'f':'alpha','t':'NomeProj'},
    'DataIniProj':{'l':'Data de Início','r':true,'f':'date','t':'DataIniProj'},
    'DataPrevProj':{'l':'Previsão de Término','r':false,'f':'date','t':'DataPrevProj'},
    'DescProj':{'l':'Descrição','r':false,'f':'alphanum','t':'DescProj'}},
},
o_config = {
    'to_disable':['Criar','Limpar'],
    'alert':1
}
// chamada do construtor do validator
var v = new validator('MainForm', a_fields, o_config);
</script>
</body>
</html>

```

26-sair.jsp

```

<%@ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<meta name="robots" content="noindex, nofollow" />
<meta name="description" content="Página de saída do SISINC." />
<meta name="author" content="Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano
Padilha" />
<title>SISINC - Sair</title>
<link href="../../../estilos/principal.css" rel="stylesheet" type="text/css" />
</head>
<%! classepersistente.Usuario auxUsuario; %>
<%
HttpSession s = request.getSession(false);
if(s != null){ auxUsuario = (classepersistente.Usuario) s.getAttribute("login"); }
%>
<body>
<table width="100%" border="0" align="center" cellpadding="0" cellspacing="0">
<tr class="cabecalho">
<td height="20" align="center" valign="top">
<div align="center" class="cabecalho">Sair do SISINC </div>
</td>
</tr>
<tr>
<td align="center" valign="top">&nbsp;&nbsp;&nbsp;</td>
</tr>
<tr>
<td align="center" valign="top">
<form id="MainForm" name="MainForm" method="post" action="../../../Sair"
target="_parent" >
<table width="600" border="1" align="center" cellpadding="0" cellspacing="0"
bordercolor="#666666">
<tr>
<td height="41"><table width="100%" border="0" align="center" cellpadding="5"
cellspacing="5">
<tr>
<td><div align="center"><span class="titulo"><%=auxUsuario.getNome()%>, deseja
realmente sair? </span></div></td>
</tr>
<tr bordercolor="#666666">
<td align="center" valign="middle">
<table width="100%" border="0" cellspacing="0" cellpadding="0">
<tr>
<td align="center"><input name="Acao" type="submit" class="rodape"
id="Acao" value="Sim" /></td>
<td align="center"><input name="Acao" type="submit" class="rodape"
id="Acao" value="N&atilde;o" /> </td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
</tr>
</table>
</tr>

```

```

        </table>
      </form>
    </td>
  </tr>
</table>
</body>
</html>

```

27-saqcad.jsp

```

<%@ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>SISINC - Cadastrar Incubação</title>
<link href="estilos/principal.css" rel="stylesheet" type="text/css" />
<script>
function lastmod() {}
</script>
<script>
function transferir(nome) {

    if(document.MainForm.QtdEntidade.value > 0){

        document.MainForm.QtdEntidade.value=document.MainForm.QtdEntidade.value-1;

        document.MainForm.QtdTotal.value=document.MainForm.QtdTotal.value-1;

        document.MainForm.HQtdEntidade.value=document.MainForm.HQtdEntidade.value-1;

    }
    else{
        alert('Você já transferiu todos os ovos desta entidade!');
        document.MainForm.nome.checked=false;
        document.MainForm.nome.checked=false;
    }

}

function aumenta(){

    x=0,y=0;
    x=parseInt(document.MainForm.QtdTotal.value);
    y=parseInt(document.MainForm.QtdEntidade.value);
    document.MainForm.QtdTotal.value=x+1;
    document.MainForm.QtdEntidade.value=y+1;
    document.MainForm.HQtdEntidade.value=y+1;

}
</script>
<script src="scripts\cancelar.js"></script>
</head>
<!--*****
*
*JSPs
*
*****-->
<%
controladora.ControladoraConsulta auxCons = new controladora.ControladoraConsulta();
javaUtil.TransferData transf=new javaUtil.TransferData();
//retorna as listas de projetos e bandejas //
java.util.Vector projeto;
projeto = auxCons.retornaProjeto();
String ban=request.getParameter("CodBan");
String incb="",a="",nascimento,entmin="",inc="",pos="",codpos="",esp="",codent="";
String dataPrev="",date="",codInc="",hcodnasc="",DataFimInc="",datainiinc="",tam="";
int index1,index2;
int col=0;
int lin=0;
int[][] posmark=new int[17][26];

```

```

//pega o valor de CodProj direto*****//
//do servlet se ele nao for nulo carrega a pagina como estava//
String proj=request.getParameter("CodProj");
nascimento=request.getParameter("CodNas");
String ent=request.getParameter("CodEnt");

java.util.Vector entidade=new Vector();
java.util.Vector listaNas=new Vector();

hcodnasc=nascimento;
if(proj!=null){
    entmin=(String)auxCons.retornaEntMin(proj);
    entmin=entmin.substring(4);

    dataPrev=auxCons.retornaDataEspecieNas(proj)+"";
    esp=auxCons.retornaEspProjeto(proj);
    listaNas=auxCons.retornaNascimento(proj);

    if ((int)listaNas.size()==1){
        String auxnas=(String)listaNas.elementAt(0);
        nascimento=auxnas.substring(0,auxnas.indexOf(" "));

    }else{

        System.out.println("antes do erro"+nascimento+"--");
        if(nascimento==null){
            nascimento="";
        }
        if(nascimento.equals("") && listaNas.size() >= 1){

            String auxnas=(String)listaNas.elementAt(0);
            nascimento=auxnas.substring(0,auxnas.indexOf(" "));

        }else{
            if(listaNas.size()< 1){
                nascimento="";
            }
        }
    }

}else{

    a=" - ";
    entidade.addElement(a);
    listaNas.addElement(a);

}
//*****//
%>
<%@page import="java.util.*"%>
<%!
Calendar tData;
int dia;
int mes;
int ano;
String mensagem;
%>
<%

//funcao que joga data no campo data de inicio da incubacao//
tData=Calendar.getInstance();
dia=tData.get(Calendar.DATE);
mes=tData.get(Calendar.MONTH)+1;
ano=tData.get(Calendar.YEAR);
mensagem="";
mensagem=mensagem+dia+"/";
mensagem=mensagem+mes+"/";
mensagem=mensagem+ano;
String msg=(String)request.getAttribute("msg");

%>
<%
//*****//

if (nascimento!=null && nascimento!=""){
    String dados=auxCons.retornaDadosNas(nascimento);
}

```

```

entidade=auxCons.retornaDadosNas2(nascimento);
int index=dados.indexOf("-");
codInc=dados.substring(0,index);
DataFimInc=dados.substring(index+1);
datainiinc=auxCons.retornaDataIniInc(codInc);
//if (codent != ""){}
//index=dados2.indexOf("-");
//index2=dados2.indexOf("|");
//codigo para quando só retorna um registro no vetor
if ((int)entidade.size()==1){

    String aux=(String)entidade.elementAt(0);
    ent=aux.substring(aux.indexOf(":")+1,aux.lastIndexOf("-"));
}else{

    if( ent==null){
        String aux=(String)entidade.elementAt(0);
        ent=aux.substring(aux.indexOf(":")+1,aux.lastIndexOf("-"));
    }
}
if (ent!=null && ent!=""){

    int i;int j;
    controladora.ControladoraNascimento auxContNas= new
controladora.ControladoraNascimento();
    for(i=0;i<entidade.size();i++){
        String aux=(String)entidade.elementAt(i);

        String aux1=aux.substring(aux.indexOf(":")+1,aux.lastIndexOf("-"));
        System.out.println("ent:"+ent+"aux1"+aux1);
        if(aux1.equals(ent)){

            ban=aux.substring(0,aux.indexOf("-"));

            //retorna a posição da bandeja *****//
            if(ban!=null){
                int ind1,ind2;
                incb=auxCons.retornaPosBanNas(ban);
                System.out.println("incb:"+incb+"cod:"+ban);
                ind1=incb.indexOf("-");
                ind2=incb.indexOf(":");
                inc=incb.substring(0,ind1-1);
                codpos=incb.substring(ind1+2,ind2-1);
                pos=incb.substring(ind2+2);
                inc=inc.replace( ' ', '_');
                tam=auxCons.retornaTamBan(ban);

                String aux3 = "";
                int marcax=0,marcay=0;

                if(tam!=null){

                    int index3 = 0;

                    aux3 = tam;
                    index3 = aux3.indexOf('-');
                    System.out.println("tam:"+tam);

                lin=Integer.parseInt(aux3.substring(0,index3));

                col=Integer.parseInt(aux3.substring(index3+1));

            }

            if(tam==null){
                System.out.println("tam é null");
            }else{
                //se tam não nulo
                //bc é o vetor com as posições ocupadas

```



```

</tr>
<tr>
  <td colspan="3" rowspan="5" align="left" valign="top" bgcolor="#EEEEEE"><strong>
    <!--*****
    *
    *montagem da tabela dinâmica da bandeja
    *
    *****-->
  </td>
  <td colspan="4" align="left" valign="top"><strong> <table border="1" cellspacing="0" cellpadding="0" >
    <%
      if(request.getParameter("CodBan")!=null && lin>0 )out.println("<b>Código da Bandeja:
</b>"+request.getParameter("CodBan"));
      int x,y;
      String nome="";

      for(x=0;x<lin;x++){
        <%>
        <tr>
          <%for(y=0;y<col;y++){ nome=x+"0"+y;%>
          <td>
            <%if (posmark[x][y]==1){%>
            <input type="checkbox" checked="checked" name="<%=nome%>" />
            <%}else{%>
            <input type="checkbox" disabled="disable" name="<%=nome%>" />
            <%}%>
          </td>
          <%}>
          <td bgcolor="#cccccc"><%=x+1 %>&nbsp;</td>
        </tr>
        <%}>
        <tr>
          <% for(x=0;x<col;x++){%>
          <td bgcolor="#cccccc"><%=x+1 %></td>
          <%} %>
        </tr>
      </table>
    <!--*****
    *
    *acaba montagem da tabela da bandeja
    *
    *****-->
  </td>
  <td height="24" align="left" valign="top"><em><strong>Selecione
Entidade:</strong></em></td>
  <td colspan="4" align="left" valign="top"><select name="CodEnt"
onchange="document.MainForm.submit();">
    <%
      String aux2 = "";
      index2 = 0;

      for(int i = 0 ; i < entidade.size() ; i++)
      {
        aux1 = (String) entidade.elementAt(i);
        aux2=aux1.substring(aux1.indexOf(":")+1);
        index2 = aux2.indexOf('-');
        if(proj!=null){
          String aux4=aux2.substring(0,index2);
          System.out.println("valor de ent para comparação:"+aux4);
          if(ent==null){
            ent=" - ";
          }
          if(aux4.compareTo(ent)==0){ %>
            <option value= <%=aux2.substring(0,index2) %> selected><%=aux2.substring(index2+1)
%></option>
            <%}else{ %>
            <option value=<%=aux2.substring(0,index2) %>><%=aux2.substring(index2+1) %></option>
            <%}
          }else{
            <%>
            <option value=<%=aux2.substring(0,index2-1) %>><%=aux2.substring(index2 +2)
%></option>
            <%

```



```

* Insere dados no banco de dados.
* @return sucesso ou falha SQL
*/
public String desmaterializar()throws SQLException
{
    try
    {
        Connection con = criaConexao();
        stmt=con.createStatement();
        aux2="SELECT * FROM ";
        aux2+="TipoEntidade WHERE ";
        aux2+="CodTipoEnt="+tipoEntidade+";";
        rs=stmt.executeQuery(aux2);
        while(rs.next())
        {
            aux1=rs.getString("NomeTipoEnt");
        }
        // insere na tabela entidade
        aux="INSERT INTO Entidade VALUES(";
        aux+="DEFAULT";
        aux+=","+1";
        aux+=","+tipoEntidade;
        aux+=","+nome+" ";
        aux+=","+descEnt+" ";";
        // insere na tabela específica
        aux+="INSERT INTO "+aux1+" VALUES(";
        aux+="DEFAULT";
        aux+=","+entidade+" ";";
        stmt.execute(aux);
        con.close();
        stmt.close();
    }
    catch(SQLException e)
    {
        System.out.println("SQL Exception no BrokerEntidade: ");
        e.printStackTrace();
        System.out.println("Comando SQL: "+aux);
        return e.getMessage().toString();
    }
    catch(Exception e)
    {
        e.printStackTrace();
        System.out.println("Saiu do Método BrokerEntidade.");
        return e.getMessage().toString();
    }
    return "Entidade";
}

/**
* Atualiza dados no banco de dados.
* @param sigla
* @return sucesso ou falha SQL
*/
public String desmaterializar(String sigla)throws SQLException
{
    try
    {
        Connection con = criaConexao();
        stmt=con.createStatement();
        aux="UPDATE Entidade SET ";
        aux+="CodStatusGeral="+statusGeral;
        aux+=","+CodTipoEnt="+tipoEntidade;
        aux+=","+NomeEnt="'+nome+' ";
        aux+=","+DescEnt="'+descEnt+' ";
        aux+="WHERE NomeEnt="'+nome+' ";";
        stmt.executeUpdate(aux);
        con.close();
        stmt.close();
    }
    catch(SQLException e)
    {
        System.out.println("SQL Exception no BrokerEntidade: ");
        e.printStackTrace();
        System.out.println("Comando SQL: "+aux);
        return e.getMessage().toString();
    }
}

```

```

        catch(Exception e)
        {
            e.printStackTrace();
            System.out.println("Saiu do Método BrokerEntidade.");
            return e.getMessage().toString();
        }
        return "EntidadeAlterar";
    }

/**
 * Atualiza para excluído no banco de dados.
 * @param siglaProj
 * @return sucesso ou falha SQL
 */
public String mudarEntidade(String siglaProj)
{
    try
    {
        Connection con = criaConexao();
        stmt=con.createStatement();
        aux="";
        aux="UPDATE entidade SET   codstatusgeral='"+2+"' where
codent='"+//codent+"'";
        stmt.executeQuery(aux);
        con.close();
        stmt.close();
    }
    catch(SQLException e)
    {
        System.out.println("SQL Exception no BrokerUsuario: ");
        e.printStackTrace();
        System.out.println("Comando SQL: "+aux);
        return e.getMessage().toString();
    }
    catch(Exception e)
    {
        e.printStackTrace();
        System.out.println("Saiu do Método BrokerUsuario.");
        return e.getMessage().toString();
    }
    return "UsuarioExclui";
}

//    metodo retorna conslta de usuarios

public Vector retornaConsulta(String t,String valor)
{
    String x="",aux2,auxent="",auxtipo="";

    try
    {
        consulta = new Vector();
        con = criaConexao();
        stmt=con.createStatement();
        if(t=="d"){
            System.out.println("entrou em d");
            int index = valor.indexOf('-');
            String valor1=valor.substring(0,index-1);
            String valor2=valor.substring(index+1);
            aux="select * from entidade where nomeent like'%" +valor1+"%'and
'%" +valor2+"'";
        }
        else{
            nome dos entidades
            if(t=="n"){
                //pega pelo
                System.out.println("entrou em n");
                aux="select * from entidade where nomeent like '%" +valor+"%'";
            }
            else{
                if(t=="a"){
                    //pega tudo
                    System.out.println("entrou em a");
                }
            }
        }
    }
}

```

```

                                aux="select * from entidade";
                                }else{                                //pega pelo
tipo
                                if(t=="t"){
                                    aux="select * from entidade where codtipoent like
'%" + valor + "%'";
                                }
                                }
                                }
                                rs = stmt.executeQuery(aux);

while(rs.next())
{
    auxent=rs.getString("codent");
    auxtipo=rs.getString("codtipoent");
}

rs.close();

aux2="";
aux2 = "SELECT ";
aux2 += "public."+"'"+"vEntidadeCelula"+"'"+"codent AS codentcel,";
aux2 += "public."+"'"+"vEntidadeCelula"+"'"+"nomeent AS nomeentcel,";
aux2 += "public."+"'"+"vEntidadeCelula"+"'"+"descent AS descentcel,";
aux2 += "public.entidade.descent AS descentgai,";
aux2 += "public.entidade.nomeent";
aux2 += " FROM ";
aux2 += "public."+"'"+"vEntidadeCelula"+"'"+"
aux2 += " INNER JOIN public.entidade ON
(public."+"'"+"vEntidadeCelula"+"'"+"codgai=public.entidade.codent)";
aux2 += "where codentecel=";
System.out.println("sql:"+aux2);
aux2="";
if(auxtipo.equals("2")){
    aux2 = "SELECT ";
    aux2 += "public."+"'"+"vEntidadeCelula"+"'"+"codent AS
codentcel,";
    aux2 += "public."+"'"+"vEntidadeCelula"+"'"+"nomeent AS
nomeentcel,";
    aux2 += "public."+"'"+"vEntidadeCelula"+"'"+"descent AS
descentcel,";
    aux2 += "public.entidade.descent AS descentgai,";
    aux2 += "public.entidade.nomeent";
    aux2 += "FROM";
    aux2 += "public."+"'"+"vEntidadeCelula"+"'"+"
    aux2 += " INNER JOIN public.entidade ON
(public."+"'"+"vEntidadeCelula"+"'"+"codgai=public.entidade.codent)";
    aux2 += "where codentecel="+auxent;
}
if(auxtipo.equals("1")){
    aux2 = "SELECT ";
    aux2 += "public."+"'"+"vEntidadeCelula"+"'"+"codent AS codentcel,";
    aux2 += "public."+"'"+"vEntidadeCelula"+"'"+"nomeent AS nomeentcel,";
    aux2 += "public."+"'"+"vEntidadeCelula"+"'"+"descent AS descentcel,";
    aux2 += "public.entidade.descent AS descentgai,";
    aux2 += "public.entidade.nomeent";
    aux2 += "FROM";
    aux2 += "public."+"'"+"vEntidadeCelula"+"'"+"
    aux2 += " INNER JOIN public.entidade ON
(public."+"'"+"vEntidadeCelula"+"'"+"codgai=public.entidade.codent)";
    aux2 += "where codentecel="+auxent;
}
if(auxtipo.equals("2")){
    aux2 = "SELECT ";
    aux2 += "public."+"'"+"vEntidadeCelula"+"'"+"codent AS codentcel,";
    aux2 += "public."+"'"+"vEntidadeCelula"+"'"+"nomeent AS nomeentcel,";
    aux2 += "public."+"'"+"vEntidadeCelula"+"'"+"descent AS descentcel,";
    aux2 += "public.entidade.descent AS descentgai,";
    aux2 += "public.entidade.nomeent";
    aux2 += "FROM";
    aux2 += "public."+"'"+"vEntidadeCelula"+"'"+"
    aux2 += " INNER JOIN public.entidade ON
(public."+"'"+"vEntidadeCelula"+"'"+"codgai=public.entidade.codent)";
    aux2 += "where codentecel="+auxent;
}
}

```



```

        aux2 += "where codentecel="+auxent;
    }
    if(auxtipo.equals("9")){
        aux2 = "SELECT ";
        aux2 += "public."+"'"+"vEntidadeCelula"+"'"+"codent AS codentcel,";
        aux2 += "public."+"'"+"vEntidadeCelula"+"'"+"nomeent AS nomeentcel,";
        aux2 += "public."+"'"+"vEntidadeCelula"+"'"+"descent AS descentcel,";
        aux2 += "public.entidade.descent AS descentgai,";
        aux2 += "public.entidade.nomeent";
        aux2 += "FROM";
        aux2 += "public."+"'"+"vEntidadeCelula"+"'"+"
        aux2 += " INNER JOIN public.entidade ON
(public."+"'"+"vEntidadeCelula"+"'"+"codgai=public.entidade.codent)";
        aux2 += "where codentecel="+auxent;
    }
    rs = stmt.executeQuery(aux2);
    while(rs.next())
    {
        x = rs.getString("codent");
        x += " - ";
        x += rs.getString("nomeent");
        x += " : ";
        x += rs.getString("tipoent");
        x += " | ";
        x += rs.getString("");
        consulta.addElement(x);
    }
    rs.close();
    con.close();
    stmt.close();
}
catch(SQLException e)
{
    System.out.println (e.getMessage());
    e.printStackTrace();
}
catch(Exception e)
{
    System.out.println (e.getMessage());
    e.printStackTrace();
}
}
return consulta;
}
// método que controio um objeto entidade

public Entidade materializar(String codigo)
{
    Entidade auxEntidade= new Entidade();

    try
    {
        con = criaConexao();
        stmt=con.createStatement();
        aux="select * from entidade where codent='"+codigo+"'";
        rs = stmt.executeQuery(aux);
        while(rs.next())
        {
            auxEntidade.setNome(rs.getString("nomeent"));
            auxEntidade.setEntidade(rs.getString("codent"));
            auxEntidade.setTipoEntidade(rs.getString("codtipoent"));
            auxEntidade.setDescEnt(rs.getString("descent"));
        }
    }
    rs.close();
    con.close();
    stmt.close();
}
catch(SQLException e)
{
    System.out.println (e.getMessage());
}
}

```

```

        e.printStackTrace();
        System.out.println("SQL Exception no metodo desmaterializar BrokerUsuario
Query->" + aux);
    }
    catch(Exception e)
    {
        System.out.println (e.getMessage());
        e.printStackTrace();
        System.out.println("saiu no");}

    return auxEntidade;
}

// beans - gets e sets
public String getEntidade() {
    return entidade;
}
public void setEntidade(String entidade) {
    this.entidade = entidade;
}
public String getTipoEntidade() {
    return tipoEntidade;
}
public void setTipoEntidade(String tipoEntidade) {
    this.tipoEntidade = tipoEntidade;
}
public String getNome() {
    return nome;
}
public void setNome(String nome) {
    this.nome = nome;
}
public String getDescEnt() {
    return descEnt;
}
public void setDescEnt(String descEnt) {
    this.descEnt = descEnt;
}
public String getStatusGeral() {
    return statusGeral;
}
public void setStatusGeral(String statusGeral) {
    this.statusGeral = statusGeral;
}
}
}

```

12.2.1.4 BrokerEntidadeOvo

```

package broker;

import java.sql.*;
import classepersistente.*;
import javaUtil.*;

/*@autor Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano Padilha*/
public class BrokerEntidadeOvo extends BrokerConexao
{
    private String codigobarras;
    private String celula;
    private String cor;
    private String formato;
    private String desinfeccao;
    private String aquisicao;
    private String datacoleta;
    private String aspecto;
    private String peso;
    private String porosidade;
    private String incubavel;
    private String descricao;
    private Statement stmt;
    private String aux;
}

```

```

private ResultSet rs;

//método para desmaterializar uma classe no banco
public int desmaterializar()throws SQLException
{ TransferData data=new TransferData();
  try
  {
      Connection con = criaConexao();
      stmt=con.createStatement();

      aux="INSERT INTO Entidade (CodEnt, CodStatusGeral, CodTipoEnt, NomeEnt,
DescEnt) VALUES (";
      aux+="DEFAULT";
      aux+=","+1";
      aux+=","+1";
      aux+=","+codigobarras+"";
      aux+=","+descricao+""); ";

      aux+="INSERT INTO Ovo (CodOvo, CodBarras, CodDesOvo, CodStatusOvo,
CodForOvo, CodCorOvo, CodCel, AquisOvo, DataColOvo, AspOvo, PesoOvo, PorOvo, Incubavel)
VALUES (";
      aux+="DEFAULT";
      aux+=","+codigobarras;
      aux+=","+desinfeccao;
      aux+=","+1";
      aux+=","+formato;
      aux+=","+cor;
      aux+=","+celula;
      aux+=","+aquisicao;
      aux+=","+data.DataPGravar(datacoleta)+"";
      aux+=","+aspecto+"";
      aux+=","+peso;
      aux+=","+porosidade;
      aux+=","+incubavel+"");";

      System.out.println("comando Sql:"+aux);
      stmt.executeQuery(aux);
      con.close();
      stmt.close();
  }
  catch(SQLException e)
  {
      System.out.println (e.getMessage());
      e.printStackTrace();
      System.out.println("SQL Exception no metodo desmaterializar
BrokerEntidadeOvo Query->" + aux);
  }
  catch(Exception e)
  {
      System.out.println (e.getMessage());
      e.printStackTrace();
      System.out.println("saiu no metodo selectAllEntidadeOvo
BrokerEntidadeOvo");
  }

  return 56;
}

//método para desmaterializar uma classe no banco
public int desmaterializar(String codigo)throws SQLException
{ TransferData data=new TransferData();
  try
  {
      Connection con = criaConexao();
      stmt=con.createStatement();
      aux="";
      aux="UPDATE Ovo INNER JOIN Entidade ON Entidade.CodEnt = Ovo.CodOvo SET
CodBarras="+codigobarras+", CodDesOvo="+desinfeccao+", CodForOvo="+formato+", CodCorOvo="+cor+", C
odCelula="+celula+", AquisOvo="+aquisicao+", DataColOvo="+data.DataPGravar(datacoleta)+"", AspOv
o="+aspecto+"", PesoOvo="+peso+", PorOvo="+porosidade+", DescEnt="+descricao+" WHERE
CodBarras="+codigobarras+";";
      stmt.executeQuery(aux);
  }
}

```

```

        con.close();
        stmt.close();
    }
    catch(SQLException e)
    {
        System.out.println (e.getMessage());
        e.printStackTrace();
        System.out.println("SQL Exception no metodo desmaterializar BrokerEntidadeOvo
Query->" + aux);
    }
    catch(Exception e)
    {
        System.out.println (e.getMessage());
        e.printStackTrace();
        System.out.println("saiu no metodo selectAllEntidadeOvo BrokerEntidadeOvo");
    }

    return 56;

}

//método para construir uma classe no banco
public EntidadeOvo materializar(String codigo)
{
    EntidadeOvo auxEntidadeOvo = new EntidadeOvo();

    try
    {
        con = criaConexao();
        stmt=con.createStatement();
        aux="SELECT * FROM Ovo INNER JOIN Entidade ON Entidade.CodEnt =
Ovo.CodOvo WHERE CodBarras="+codigobarras;
        rs = stmt.executeQuery(aux);
        while(rs.next())
        {
            auxEntidadeOvo.setCodigobarras(rs.getString("CodBarras"));
            auxEntidadeOvo.setCelula(rs.getString("CodCel"));
            auxEntidadeOvo.setCor(rs.getString("CodCorOvo"));
            auxEntidadeOvo.setFormato(rs.getString("CodForOvo"));
            auxEntidadeOvo.setDesinfeccao(rs.getString("CodDesOvo"));
            auxEntidadeOvo.setAquisicao(rs.getString("AquisOvo"));
            auxEntidadeOvo.setDatacoleta(rs.getString("DataColOvo"));
            auxEntidadeOvo.setAspecto(rs.getString("AspOvo"));
            auxEntidadeOvo.setPeso(rs.getString("PesoOvo"));
            auxEntidadeOvo.setPorosidade(rs.getString("PorOvo"));
            auxEntidadeOvo.setIncubavel(rs.getString("Incubavel"));
            auxEntidadeOvo.setDescricao(rs.getString("DescEnt"));
        }
        rs.close();
        con.close();
        stmt.close();
    }
    catch(SQLException e)
    {
        System.out.println (e.getMessage());
        e.printStackTrace();
        System.out.println("SQL Exception no metodo desmaterializar
BrokerEntidadeOvo Query->" + aux);
    }
    catch(Exception e)
    {
        System.out.println (e.getMessage());
        e.printStackTrace();
        System.out.println("saiu no metodo selectAllEntidadeOvo BrokerLogin");
    }
    return auxEntidadeOvo;
}

public String getAspecto() {
    return aspecto;
}

public void setAspecto(String aspecto) {
    this.aspecto = aspecto;
}

```

```

public String getCor() {
    return cor;
}

public void setCor(String cor) {
    this.cor = cor;
}

public String getCelula() {
    return celula;
}

public void setCelula(String celula) {
    this.celula = celula;
}

public String getCodigobarras() {
    return codigobarras;
}

public void setCodigobarras(String codigobarras) {
    this.codigobarras = codigobarras;
}

public String getFormato() {
    return formato;
}

public void setFormato(String formato) {
    this.formato = formato;
}

public String getDatacoleta() {
    return datacoleta;
}

public void setDatacoleta(String datacoleta) {
    this.datacoleta = datacoleta;
}

public String getAquisicao() {
    return aquisicao;
}

public void setAquisicao(String aquisicao) {
    this.aquisicao = aquisicao;
}

public String getDesinfeccao() {
    return desinfeccao;
}

public void setDesinfeccao(String desinfeccao) {
    this.desinfeccao = desinfeccao;
}

public String getPeso() {
    return peso;
}

public void setPeso(String peso) {
    this.peso = peso;
}

public String getPorosidade() {
    return porosidade;
}

public void setPorosidade(String porosidade) {
    this.porosidade = porosidade;
}

public String getIncubavel() {
    return incubavel;
}

```

```

    public void setIncubavel(String incubavel) {
        this.incubavel = incubavel;
    }

    public String getDescricao() {
        return descricao;
    }

    public void setDescricao(String descricao) {
        this.descricao = descricao;
    }
}

```

12.2.1.5 BrokerEspecieAve

```

package broker;

import java.sql.*;
import classepersistente.*;

/*@autor Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano Padilha*/
public class BrokerEspecieAve extends BrokerConexao
{
    private String especie;
    private String tempmininc;
    private String tempmaxinc;
    private String umimininc;
    private String umimaxinc;
    private String perinc;
    private String peraviinc;
    private String tempminnas;
    private String tempmaxnas;
    private String umiminnas;
    private String umimaxnas;
    private String pernas;
    private String peravinas;
    private String perrec;
    private Statement stmt;
    private String aux;
    private ResultSet rs;

    //método para desmaterializar uma classe no banco
    public int desmaterializar()throws SQLException
    {
        try
        {
            Connection con = criaConexao();
            stmt=con.createStatement();

            aux="INSERT INTO EspecieAve (CodEsp, CodStatusGeral, NomeEsp,
TempMinInc, TempMaxInc, UmiMinInc, UmiMaxInc, PerInc, PerAviInc, TempMinNas, TempMaxNas,
UmiMinNas, UmiMaxNas, PerNas, PerAviNas, PerRec) VALUES(";
            aux+="DEFAULT";
            aux+=","+1";
            aux+=","+especie+"";
            aux+=","+tempmininc;
            aux+=","+tempmaxinc;
            aux+=","+umimininc;
            aux+=","+umimaxinc;
            aux+=","+perinc;
            aux+=","+peraviinc;
            aux+=","+umimaxinc;
            aux+=","+tempminnas;

```

```

        aux+=" "+tempmaxnas;
        aux+=" "+umiminnas;
        aux+=" "+umimaxnas;
        aux+=" "+pernas;
        aux+=" "+peravinas;
        aux+=" "+perrec+");";

        System.out.println("comando Sql:"+aux);
        stmt.executeQuery(aux);
        con.close();
        stmt.close();
    }
    catch(SQLException e)
    {
        System.out.println (e.getMessage());
        e.printStackTrace();
        System.out.println("SQL Exception no metodo desmaterializar
BrokerEspecieAve Query->" + aux);
    }
    catch(Exception e)
    {
        System.out.println (e.getMessage());
        e.printStackTrace();
        System.out.println("saiu no metodo selectAllEspecieAve
BrokerEspecieAve");
    }

    return 56;
}

//método para desmaterializar uma classe no banco
public int desmaterializar(String esp)throws SQLException
{
    try
    {
        Connection con = criaConexao();
        stmt=con.createStatement();
        aux="";
        aux="UPDATE EspecieAve SET
NomeEsp='"+especie+"',TempMinInc="+tempmininc+",TempMaxInc="+tempmaxinc+",UmiMinInc="+umimininc+
c+",UmiMaxInc="+umimaxinc+",PerInc="+perinc+",PerAviInc="+peraviinc+",TempMinNas="+tempminnas+
",TempMaxNas="+tempmaxnas+",UmiMinNas="+umiminnas+",UmiMaxNas="+umimaxnas+",PerNas="+pernas+",
PerAviNas="+peravinas+",PerRec="+perrec+" WHERE NomeEsp='"+especie+"';";
        stmt.executeQuery(aux);
        con.close();
        stmt.close();
    }
    catch(SQLException e)
    {
        System.out.println (e.getMessage());
        e.printStackTrace();
        System.out.println("SQL Exception no metodo desmaterializar BrokerEspecieAve
Query->" + aux);
    }
    catch(Exception e)
    {
        System.out.println (e.getMessage());
        e.printStackTrace();
        System.out.println("saiu no metodo selectAllEspecieAve BrokerEspecieAve");
    }

    return 56;
}

//método para construir uma classe no banco
public EspecieAve materializar(String esp)
{
    EspecieAve auxEspecieAve = new EspecieAve();

    try
    {
        con = criaConexao();
        stmt=con.createStatement();
        aux="SELECT * FROM EspecieAve WHERE NomeEsp='"+especie+"';";

```

```

        rs = stmt.executeQuery(aux);
while(rs.next())
{
    auxEspecieAve.setEspecie(rs.getString("NomeEsp"));
    auxEspecieAve.setTempMinInc(rs.getString("TempMinInc"));
    auxEspecieAve.setTempMaxInc(rs.getString("TempMaxInc"));
    auxEspecieAve.setUmiMinInc(rs.getString("UmiMinInc"));
    auxEspecieAve.setUmiMaxInc(rs.getString("UmiMaxInc"));
    auxEspecieAve.setPerInc(rs.getString("PerInc"));
    auxEspecieAve.setPerAviInc(rs.getString("PerAviInc"));
    auxEspecieAve.setTempMinNas(rs.getString("TempMinNas"));
    auxEspecieAve.setTempMaxNas(rs.getString("TempMaxNas"));
    auxEspecieAve.setUmiMinNas(rs.getString("UmiMinNas"));
    auxEspecieAve.setUmiMaxNas(rs.getString("UmiMaxNas"));
    auxEspecieAve.setPerNas(rs.getString("PerNas"));
    auxEspecieAve.setPerAviNas(rs.getString("PerAviNas"));
    auxEspecieAve.setPerRec(rs.getString("PerRec"));
}
rs.close();
con.close();
stmt.close();
}
catch(SQLException e)
{
    System.out.println (e.getMessage());
    e.printStackTrace();
    System.out.println("SQL Exception no metodo desmaterializar
BrokerEspecieAve Query->" + aux);
}
catch(Exception e)
{
    System.out.println (e.getMessage());
    e.printStackTrace();
    System.out.println("saiu no metodo selectAllEspecieAve BrokerLogin");
}
return auxEspecieAve;
}

public String getTempMinNas() {
    return tempminnas;
}

public void setTempMinNas(String tempminnas) {
    this.tempminnas = tempminnas;
}

public String getTempMaxInc() {
    return tempmaxinc;
}

public void setTempMaxInc(String tempmaxinc) {
    this.tempmaxinc = tempmaxinc;
}

public String getTempMinInc() {
    return tempmininc;
}

public void setTempMinInc(String tempmininc) {
    this.tempmininc = tempmininc;
}

public String getEspecie() {
    return especie;
}

public void setEspecie(String especie) {
    this.especie = especie;
}

public String getUmiMinInc() {
    return umimininc;
}
}

```

```

public void setUmiMinInc(String umimininc) {
    this.umimininc = umimininc;
}

public String getPerAviInc() {
    return peraviinc;
}

public void setPerAviInc(String peraviinc) {
    this.peraviinc = peraviinc;
}

public String getPerInc() {
    return perinc;
}

public void setPerInc(String perinc) {
    this.perinc = perinc;
}

public String getUmiMaxInc() {
    return umimaxinc;
}

public void setUmiMaxInc(String umimaxinc) {
    this.umimaxinc = umimaxinc;
}

public String getTempMaxNas() {
    return tempmaxnas;
}

public void setTempMaxNas(String tempmaxnas) {
    this.tempmaxnas = tempmaxnas;
}

public String getUmiMinNas() {
    return umiminnas;
}

public void setUmiMinNas(String umiminnas) {
    this.umiminnas = umiminnas;
}

public String getUmiMaxNas() {
    return umimaxnas;
}

public void setUmiMaxNas(String umimaxnas) {
    this.umimaxnas = umimaxnas;
}

public String getPerAviNas() {
    return peravinas;
}

public void setPerAviNas(String peravinas) {
    this.pernas = peravinas;
}

public String getPerRec() {
    return perrec;
}

public void setPerRec(String perrec) {
    this.pernas = perrec;
}

public String getPerNas() {
    return pernas;
}

public void setPerNas(String pernas) {
    this.pernas = pernas;
}

```

```
}
```

12.2.1.6 BrokerIncubacao

```
package broker;

import java.sql.*;
import java.util.*;
import classepersistente.*;

/*@autor Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano Padilha*/
public class BrokerIncubacao extends BrokerConexao
{
    private String projeto;
    private String datini;
    private String bandeja;
    private String entidade;
    private Statement stmt;
    private String aux,aux1;
    private Vector consulta;
    private ResultSet rs;

    //método para desmaterializar uma classe no banco

    public void mudarIncubacao(String mat)
    {
        try
        {
            System.out.println("estou dentro do broker mat="+mat);
            Connection con = criaConexao();
            stmt=con.createStatement();
            aux="";
            aux="UPDATE Incubacao SET codstatususu='"+2+"' where
matrusu='"+mat+"'";
            stmt.executeQuery(aux);
            con.close();
            stmt.close();
        }
        catch(SQLException e)
        {
            System.out.println (e.getMessage());
            e.printStackTrace();
            System.out.println("SQL Exception no metodo desmaterializar
BrokerIncubacao Query->" + aux);
        }
        catch(Exception e)
        {
            System.out.println (e.getMessage());
            e.printStackTrace();
        }
    }

    public String desmaterializar(ArrayList incubacao,ArrayList posicoes)throws
SQLException
    {
        try
```

```

    {
        int qtde,x;
        Incubacao auxInc=(Incubacao)incubacao.get(0);
        Connection con = criaConexao();
        stmt=con.createStatement();
        System.out.println("inserindo!");

        aux="INSERT INTO incubacao
(codinc,codstatusin,codproj,datainiinc,dataprevinc,numovoinii,numovodesi,numovotransf,obsstatu
sinc) values (default,1,";
        aux+=auxInc.getProjeto();
        aux+=",CURRENT_TIMESTAMP";
        aux+=","+"CURRENT_TIMESTAMP "+auxInc.getData()+"
00:00:00','0','0','0','');";

        qtde=posicoes.size();
        for(x=0;x<qtde;x++){
            stmt=con.createStatement();
            Posicao posicao = (Posicao)posicoes.get(x);
            System.out.println("ent:"+posicao.getEntidade());
            System.out.println("x:"+posicao.getPosicaoX());
            System.out.println("y:"+posicao.getPosicaoY());
            System.out.println("z:"+posicao.getPosicaoZ());
            aux+="INSERT INTO incument
(codinc,codent,nlinban,ncolban,codposbanenc,codstatusgeral,codmotdes,ovotransf)values (";
            aux+="curral('incubacao_codinc'::text),"";
            aux+=posicao.getEntidade();
            aux+=", "+posicao.getPosicaoX();
            aux+=", "+posicao.getPosicaoY();
            aux+=", "+posicao.getPosicaoZ();
            aux+=", '1','1','0')";

        }
        System.out.println("aux"+aux);
        stmt.execute(aux);
        stmt.close();

        stmt=con.createStatement();
        aux="select codinc,dataprevinc from incubacao where codinc=(select
MAX(codinc) as codinc from incubacao);";
        rs=stmt.executeQuery(aux);
        while(rs.next()){
            aux1=rs.getString("codinc");
            aux1+="|";
            aux1+=rs.getString("dataprevinc");
        }
        System.out.println("aux1"+aux1);
    }
    catch(SQLException e)
    {
        System.out.println (e.getMessage());
        e.printStackTrace();
        System.out.println("SQL Exception no metodo desmaterializar
BrokerIncubacao Query->" + aux);
    }
    catch(Exception e)
    {
        System.out.println (e.getMessage());
        e.printStackTrace();
        System.out.println("saiu no metodo selectAllIncubacao
BrokerLogin"+aux);
    }

    return aux1;
}

```

```

public String desmaterializar(ArrayList posicoes) throws SQLException
{
    try
    {
        int qtde,x;

        Connection con = criaConexao();
        qtde=posicoes.size();
        for(x=0;x<qtde;x++){
            stmt=con.createStatement();
            Posicao posicao = (Posicao)posicoes.get(x);
            System.out.println("ent:"+posicao.getEntidade());
            System.out.println("x:"+posicao.getPosicaoX());
            System.out.println("y:"+posicao.getPosicaoY());
            System.out.println("z:"+posicao.getPosicaoZ());
            aux="INSERT INTO incument
(codinc,codent,nlinban,ncolban,codposbanenc,codstatusgeral,codmotdes,ovotransf) values (";
            aux+=posicao.getIncubacao()+" ";
            aux+=posicao.getEntidade();
            aux+=", "+posicao.getPosicaoX();
            aux+=", "+posicao.getPosicaoY();
            aux+=", "+posicao.getPosicaoZ();
            aux+=", '1', '1', '0')";

            stmt.execute(aux);
            stmt.close();

        }

        stmt=con.createStatement();
        aux="select codinc,dataprevinc from incubacao where codinc=(select
MAX(codinc) as codinc from incubacao);";
        rs=stmt.executeQuery(aux);
        while(rs.next()){
            aux1=rs.getString("codinc");
            aux1+="|";
            aux1+=rs.getString("dataprevinc");
        }
    }
    catch(SQLException e)
    {
        System.out.println (e.getMessage());
        e.printStackTrace();
        System.out.println("SQL Exception no metodo desmaterializar
BrokerIncubacao Query->" + aux);
    }
    catch(Exception e)
    {
        System.out.println (e.getMessage());
        e.printStackTrace();
        System.out.println("saiu no metodo selectAllIncubacao
BrokerLogin"+aux);
    }

    return aux1;

}

// metodo retorna conslta de Incubacaos

public Vector retornaConsulta(String campo, String valor)
{
    String x="";

    try
    {
        consulta = new Vector();
        con = criaConexao();
        stmt=con.createStatement();
        aux="select * from Incubacao where "+ campo+" like '%" + valor+"%' order by
"+campo;
        rs = stmt.executeQuery(aux);
        while(rs.next())

```

```

        {
            x = rs.getString("codusu");
            x += " - ";
            x += rs.getString("matrusu");
            x += " : ";
            x += rs.getString("nomeusu");
            consulta.addElement(x);
        }
        rs.close();
        con.close();
        stmt.close();
    }
    catch(SQLException e)
    {
        System.out.println (e.getMessage());
        e.printStackTrace();
    }
    catch(Exception e)
    {
        System.out.println (e.getMessage());
        e.printStackTrace();
    }
    return consulta;
}

//metodo retorna consulta de Incubacao-sobrecarga do anterior, pesquisa com 2 campos.
public Vector retornaConsulta(String campo, String valor,String campo2,String valor2)
{
    String x="";

    try
        //consulta no banco
    {
        consulta = new Vector();
        con = criaConexao();
        stmt=con.createStatement();
        aux="select * from Incubacao where "+ campo+" like '%" + valor+"%'or "+campo2+"
like '%" +valor2+"%' order by "+campo;
        rs = stmt.executeQuery(aux);
        while(rs.next())
        {
            x = rs.getString("codusu");
            x += " - ";
            x += rs.getString("matrusu");
            x += " : ";
            x += rs.getString("nomeusu");
            consulta.addElement(x);
        }
        rs.close();
        con.close();
        stmt.close();
    }
    catch(SQLException e)
    {
        System.out.println (e.getMessage());
        e.printStackTrace();
    }
    catch(Exception e)
    {
        System.out.println (e.getMessage());
        e.printStackTrace();
    }
    return consulta;
}

public Vector retornaConsulta()
{

```

```

String x="";

try
{
    consulta = new Vector();
    con = criaConexao();
    stmt=con.createStatement();
    aux="select * from Incubacao";
    rs = stmt.executeQuery(aux);
    while(rs.next())
    {
        x = rs.getString("codusu");
        x += " - ";
        x += rs.getString("matrusu");
        x += " : ";
        x += rs.getString("nomeusu");
        consulta.addElement(x);
    }
    rs.close();
    con.close();
    stmt.close();
}
catch(SQLException e)
{
    System.out.println (e.getMessage());
    e.printStackTrace();
}
catch(Exception e)
{
    System.out.println (e.getMessage());
    e.printStackTrace();
}

return consulta;
}
// método que constroi uma objeto Incubacao

public Incubacao materializar(String codigo)
{
    Incubacao auxIncubacao= new Incubacao();

    try
    {
        con = criaConexao();
        stmt=con.createStatement();
        aux="select * from Incubacao where codusu="+ codigo;
        rs = stmt.executeQuery(aux);
        while(rs.next())
        {
            //auxIncubacao.setBandeja(rs.getString("nomeusu"));
            auxIncubacao.setDatini(rs.getString("data"));
            auxIncubacao.setProjeto(rs.getString("pro"));
            //auxIncubacao.setEntidade(rs.getString("email"));

        }
        rs.close();
        con.close();
        stmt.close();
    }
    catch(SQLException e)
    {
        System.out.println (e.getMessage());
        e.printStackTrace();
        System.out.println("SQL Exception no metodo desmaterializar BrokerIncubacao
Query->" + aux);
    }
    catch(Exception e)
    {
        System.out.println (e.getMessage());
        e.printStackTrace();
        System.out.println("saiu no metodo selectAllIncubacao BrokerLogin");
    }
}

```

```

        return auxIncubacao;
    }

    public Vector setarBandeja(String bandeja) {
        Vector bandejaCheia=new Vector();
        try
        {
            String x="";
            con = criaConexao();
            stmt=con.createStatement();
            aux="select * from incement where codposbanenc="+ bandeja+" and ovotransf
<>'S'";
            rs = stmt.executeQuery(aux);
            while(rs.next())
            {
                x=rs.getInt("codinc")+"";
                x+="-";
                x+=rs.getInt("nlinban")+"";
                x+=":";
                x+=rs.getInt("ncolban")+"";

                bandejaCheia.addElement(x);

            }
            rs.close();
            con.close();
            stmt.close();
        }
        catch(SQLException e)
        {
            System.out.println (e.getMessage());
            e.printStackTrace();
            System.out.println("SQL Exception no metodo desmaterializar BrokerIncubacao
Query->" + aux);
        }
        catch(Exception e)
        {
            System.out.println (e.getMessage());
            e.printStackTrace();
            System.out.println("saiu no metodo selectAllIncubacao BrokerLogin");
        }

        return bandejaCheia;
    }

    public String getBandeja() {
        return bandeja;
    }

    public void setBandeja(String bandeja) {
        this.bandeja = bandeja;
    }

    public String getDatini() {
        return datini;
    }

    public void setDatini(String datini) {
        this.datini = datini;
    }

    public String getEntidade() {
        return entidade;
    }

    public void setEntidade(String entidade) {
        this.entidade = entidade;
    }

    public String getProjeto() {
        return projeto;
    }
}

```

```

public void setProjeto(String projeto) {
    this.projeto = projeto;
}

}

```

12.2.1.7 BrokerIncubadoraBandeja

```

package broker;

import java.sql.*;
import classepersistente.*;

/*@autor Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano Padilha*/
public class BrokerIncubadoraBandeja extends BrokerConexao
{
    private String codigo;
    private String incubadora;
    private String bandeja;
    private String posicao;
    private String statusGeral;
    private Statement stmt;
    private String aux;
    private ResultSet rs;
    private String var;

    //método para desmaterializar uma classe no banco
    public String desmaterializar(IncubadoraBandeja ib) throws SQLException
    {
        try
        {
            Connection con = criaConexao();
            stmt=con.createStatement();
            aux="SELECT codposbanenc from posbanenc WHERE codban="+ib.getBandeja();
            rs=stmt.executeQuery(aux);

            var="B";
            if(rs.next()){
                var="A";
            }
            if(var.equals("B")){

                aux="INSERT INTO PosBanEnc (CodPosBanEnc, CodStatusGeral,
                CodEnc, CodBan, NPosEnc, DataAloEnc, DataRetEnc) VALUES (";
                aux+="DEFAULT";
                aux+=","+ib.getIncubadora();
                aux+=","+ib.getBandeja();
                aux+=","+ib.getPosinc();
                aux+=","+CURRENT_TIMESTAMP";
                aux+=","+null);";
                System.out.println("comando Sql:"+aux);
                stmt.execute(aux);

            }else{
                System.out.println("to passando aqui incubadora");
                desmaterializarU(ib);
            }
            rs.close();
            con.close();
            stmt.close();
        }
        catch(SQLException e)
        {
            System.out.println (e.getMessage());
            e.printStackTrace();
        }
    }
}

```

```

        System.out.println("SQL Exception no metodo desmaterializar
BrokerUsuario Query->" + aux);
    }
    catch(Exception e)
    {
        System.out.println (e.getMessage());
        e.printStackTrace();
        System.out.println("saiu no metodo selectAllUsuario BrokerLogin");
    }
    return "efetuado";
}

public String desmaterializarN(IncubadoraBandeja ib) throws SQLException
{
    try
    {
        Connection con = criaConexao();
        stmt=con.createStatement();
        aux="SELECT codposbannes from posbannes WHERE codban="+ib.getBandeja();
        rs=stmt.executeQuery(aux);
        var="B";
        if(rs.next()){
            var="A";
        }
        if(var.equals("B")){
            aux="INSERT INTO PosBanNes(CodPosBanNes, CodStatusGeral, CodNes,
CodBan, NPosNes, DataAloNes, DataRetNes) VALUES (";
            aux+="DEFAULT";
            aux+=","+ib.getNascedouro();
            aux+=","+ib.getBandeja();
            aux+=","+ib.getPosnas();
            aux+=","+CURRENT_TIMESTAMP";
            aux+=","+null);";
            System.out.println("comando Sql:"+aux);
            stmt.execute(aux);

        }else{
            System.out.println("to passando aquinascedouro");
            desmaterializarNU(ib);
        }
        rs.close();
        con.close();
        stmt.close();
    }
    catch(SQLException e)
    {
        System.out.println (e.getMessage());
        e.printStackTrace();
        System.out.println("SQL Exception no metodo desmaterializar Query->" +
aux);
    }
    catch(Exception e)
    {
        System.out.println (e.getMessage());
        e.printStackTrace();
        System.out.println("saiu no metodo ");
    }
    return "efetuado";
}

//método de exclusão
public void mudarIncubadoraBandeja(IncubadoraBandeja ib)
{
    try
    {
        System.out.println("estou dentro do broker codigo="+codigo);
        Connection con = criaConexao();
        stmt=con.createStatement();
        aux="";
        aux="UPDATE PosBanEnc SET CodStatusGeral="+2+"
DataRecEnt=CURRENT_TIMESTAMP WHERE Codban="+ib.getBandeja()+";";
        stmt.executeQuery(aux);
        con.close();
        stmt.close();
    }
}

```

```

        catch(SQLException e)
        {
            System.out.println (e.getMessage());
            e.printStackTrace();
            System.out.println("SQL Exception no metodo desmaterializar
BrokerUsuario Query->" + aux);
        }
        catch(Exception e)
        {
            System.out.println (e.getMessage());
            e.printStackTrace();
        }
    }

    //faz update quando a bandeja ja foi alocada em incubadora
    public int desmaterializarU(IncubadoraBandeja ib)throws SQLException
    {
        try
        {
            Connection con = criaConexao();
            stmt=con.createStatement();
            aux="";
            aux="UPDATE PosBanEnc SET dataaloenc=CURRENT_TIMESTAMP,
dataretenc=null,codenc="+ib.getIncubadora()+",nposenc="+ib.getPosinc()+" WHERE
Codban='"+ib.getBandeja()+"'";
            stmt.executeUpdate(aux);
            con.close();
            stmt.close();
        }
        catch(SQLException e)
        {
            System.out.println (e.getMessage());
            e.printStackTrace();
            System.out.println("SQL Exception no metodo desmaterializar
BrokerUsuario Query->" + aux);
        }
        catch(Exception e)
        {
            System.out.println (e.getMessage());
            e.printStackTrace();
            System.out.println("saiu no metodo selectAllUsuario BrokerLogin");
        }
        return 56;
    }

    public int desalocarI(String banI)throws SQLException
    {
        try
        {
            Connection con = criaConexao();
            stmt=con.createStatement();
            aux="";
            aux="UPDATE PosBanEnc SET dataretenc=CURRENT_TIMESTAMP WHERE
Codban='"+banI+"'";
            System.out.println("sql->" +aux);
            stmt.executeUpdate(aux);
            con.close();
            stmt.close();
        }
        catch(SQLException e)
        {
            System.out.println (e.getMessage());
            e.printStackTrace();
            System.out.println("SQL Exception no metodo desmaterializar
BrokerUsuario Query->" + aux);
        }
        catch(Exception e)
        {
            System.out.println (e.getMessage());
            e.printStackTrace();
            System.out.println("saiu no metodo selectAllUsuario BrokerLogin");
        }
        return 56;
    }

    public int desalocarN(String banN)throws SQLException
    {
        try

```

```

        {
            Connection con = criaConexao();
            stmt=con.createStatement();
            aux="";
            aux="UPDATE PosBanNes SET dataretnes=CURRENT_TIMESTAMP WHERE
Codban='"+banN+"'";
            System.out.println("sql->" + aux);
            stmt.executeUpdate(aux);
            con.close();
            stmt.close();
        }
        catch(SQLException e)
        {
            System.out.println (e.getMessage());
            e.printStackTrace();
            System.out.println("SQL Exception no metodo desmaterializar
BrokerUsuario Query->" + aux);
        }
        catch(Exception e)
        {
            System.out.println (e.getMessage());
            e.printStackTrace();
            System.out.println("saiu no metodo selectAllUsuario BrokerLogin");
        }
        return 56;
    }
    //faz update quando a bandeja ja foi alocada em nascedouro
    public int desmaterializarNU(IncubadoraBandeja ib)throws SQLException
    {
        try
        {
            Connection con = criaConexao();
            stmt=con.createStatement();
            aux="";
            aux="UPDATE PosBanNes SET
dataalones=CURRENT_TIMESTAMP,dataretnes=null,codnes="+ib.getNascedouro()+" ,nposnes="+ib.getPos
nas()+" WHERE CodPosBannes='"+ib.getBandeja()+"'";
            stmt.executeUpdate(aux);
            con.close();
            stmt.close();
        }
        catch(SQLException e)
        {
            System.out.println (e.getMessage());
            e.printStackTrace();
            System.out.println("SQL Exception no metodo desmaterializar
BrokerUsuario Query->" + aux);
        }
        catch(Exception e)
        {
            System.out.println (e.getMessage());
            e.printStackTrace();
            System.out.println("saiu no metodo selectAllUsuario BrokerLogin");
        }
        return 56;
    }
}

//método que constroi um objeto incubadorabandeja
public IncubadoraBandeja materializar(String codigo)
{
    IncubadoraBandeja auxIncubadoraBandeja= new IncubadoraBandeja();
    try
    {
        con = criaConexao();
        stmt=con.createStatement();
        aux="SELECT * FROM PosBanEnc WHERE CodPosBanEnc='"+codigo+"'";
        rs = stmt.executeQuery(aux);
        while(rs.next())
        {
            auxIncubadoraBandeja.setIncubadora(rs.getString("CodEnc"));
            auxIncubadoraBandeja.setBandeja(rs.getString("CodBan"));
        }
    }
}

```

```

        //auxIncubadoraBandeja.setPosicao(rs.getString("NPosEnc"));
    }
    rs.close();
    con.close();
    stmt.close();
}
catch(SQLException e)
{
    System.out.println (e.getMessage());
    e.printStackTrace();
    System.out.println("SQL Exception no metodo desmaterializar
BrokerUsuario Query->" + aux);
}
catch(Exception e)
{
    System.out.println (e.getMessage());
    e.printStackTrace();
    System.out.println("saiu no");}

    return auxIncubadoraBandeja;
}

public String getCodigo() {
    return codigo;
}

public void setCodigo(String codigo) {
    this.codigo = codigo;
}

public String getIncubadora() {
    return incubadora;
}

public void setIncubadora(String incubadora) {
    this.incubadora = incubadora;
}

public String getBandeja() {
    return bandeja;
}

public void setBandeja(String bandeja) {
    this.bandeja = bandeja;
}

public String getPosicao() {
    return posicao;
}

public void setPosicao(String posicao) {
    this.posicao = posicao;
}

public String getStatusGeral() {
    return statusGeral;
}

public void setStatusGeral(String statusGeral) {
    this.statusGeral = statusGeral;
}
}

```

12.2.1.8 BrokerNascimento

```

package broker;

import java.sql.*;
import java.util.*;
import classepersistente.*;

/*@autor Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano Padilha*/

```

```

public class BrokerNascimento extends BrokerConexao
{
    private Statement stmt;
    private String aux,aux1;
    private Vector consulta;
    private ResultSet rs;
    private String retorna;
    private int x,y;

    //método para desmaterializar uma classe no banco

    //acertar o array list de nascimento eo de posicoes q sao do novo nascimento a ser
    feito de acordo com os dados da tela parecido com o processo de incubação
    public String transfere(String incubacao,String entidade,int qtde)throws SQLException
    {
        try
        {
            //select que retorna todas as posições com estas características
            //(entidade e incubacao iguais aos parametros passados)
            //e depois armazenado em um vetor que depois sera percorrido
            //alterando o estado para transferido 1 a 1 até o valor chegar
            //no valor de "qtde".
            int[][] posicao=new int[26][17];
            for( x=0;x<26;x++){
                for( y=0;y<17;y++){
                    posicao[x][y]=0;
                }
            }

            Connection con = criaConexao();
            stmt=con.createStatement();
            aux="SELECT * FROM incement WHERE (codinc="+incubacao+" and
codent="+entidade+")and ovotransf=0";
            rs = stmt.executeQuery(aux);
            x=0;
            while((rs.next())&& (x < qtde))
            {
                posicao[rs.getInt("nlinban")][rs.getInt("ncolban")]=1;
                System.out.println(rs.getInt("nlinban")+"-"+rs.getInt("ncolban"));
                x++;
            }

            stmt.close();
            //marca transferido na tabela
            for(x=0;x<26;x++){
                for(y=0;y<17;y++){
                    stmt=con.createStatement();
                    if(posicao[x][y]==1){
                        aux="UPDATE incement SET OvoTransf='S' where
(codinc="+incubacao+" and codent="+entidade+") and (nlinban="+x+" and ncolban="+y+"");
                        System.out.println("comando Sql:"+aux);
                        stmt.executeUpdate(aux);
                    }

                    stmt.close();
                }
            }
            con.close();

        }
        catch(SQLException e)
        {
            System.out.println(e.getMessage());
            e.printStackTrace();
            System.out.println("SQL Exception no metodo desmaterializar
BrokerNascimento Query->" + aux);
        }
        catch(Exception e)

```

```

        {
            System.out.println (e.getMessage());
            e.printStackTrace();
            System.out.println("saiu no metodo ");
        }

        retorna="efetuado";
        return retorna;
    }

    /*public void mudarIncubacao(String mat)
    {
        try
        {
            System.out.println("estou dentro do broker mat="+mat);
            Connection con = criaConexao();
            stmt=con.createStatement();
            aux="";
            aux="UPDATE Incubacao SET codstatususu='"+2+"' where
matrusu='"+mat+"'";
            stmt.executeQuery(aux);
            con.close();
            stmt.close();
        }
        catch(SQLException e)
        {
            System.out.println (e.getMessage());
            e.printStackTrace();
            System.out.println("SQL Exception no metodo desmaterializar
BrokerIncubacao Query->" + aux);
        }
        catch(Exception e)
        {
            System.out.println (e.getMessage());
            e.printStackTrace();
        }
    }
    */
    public String desmaterializar(ArrayList nascimento,ArrayList posicoes,int qtde,String
incubacao,String entidade)throws SQLException
    {
        try
        {

            int y,x;

            transfere( incubacao, entidade, qtde);

            Nascimento auxNas=(Nascimento)nascimento.get(0);
            System.out.println("passei do nascimento");
            Connection con = criaConexao();
            stmt=con.createStatement();
            System.out.println("inserindo!");

            aux="INSERT INTO NASCIMENTO
(codnas,codstatusin,codinc,datainiinas,dataprevnas,numovoinin,numovodesn,numovorec,obsstatusna
s) values (default,1,"
            aux+=auxNas.getIncubacao();
            aux+=","+CURRENT_TIMESTAMP";
            aux+=","+CURRENT_TIMESTAMP +'"+auxNas.getData()+"
00:00:00','0','0','0','')";

            y=posicoes.size();
            for (x=0;x<y;x++){
                stmt=con.createStatement();
                Posicao posicao = (Posicao)posicoes.get(x);
                System.out.println("ent:"+posicao.getEntidade());
                System.out.println("x:"+posicao.getPosicaoX());
                System.out.println("y:"+posicao.getPosicaoY());
                System.out.println("z:"+posicao.getPosicaoZ());
                aux+="INSERT INTO nastement
(codnas,codent,nlinban,ncolban,codposbannes,codstatusgeral,codmotdes,ovonasceu) values ("

```

```

        aux+="currval('nascimento_codnas'::text),";
        aux+=posicao.getEntidade();
        aux+=","+posicao.getPosicaoX();
        aux+=","+posicao.getPosicaoY();
        aux+=","+posicao.getPosicaoZ();
        aux+=",'1','1','0')";

    }
    System.out.println("sql"+aux);
    stmt.execute(aux);
    stmt.close();

    stmt=con.createStatement();
    aux="select codnas,dataprevnas from nascimento where codnas=(select
MAX(codnas) as codnas from nascimento);";
    rs=stmt.executeQuery(aux);
    while(rs.next()){
        aux1=rs.getString("codnas");
        aux1+="|";
        aux1+=rs.getString("dataprevnas");
    }
    System.out.println("aux1"+aux1);
}
catch(SQLException e)
{
    System.out.println (e.getMessage());
    e.printStackTrace();
    System.out.println("SQL Exception no metodo desmaterializar
BrokerIncubacao Query->" + aux);
}
catch(Exception e)
{
    System.out.println (e.getMessage());
    e.printStackTrace();
    System.out.println("saiu no metodo selectAllIncubacao
BrokerLogin"+aux);
}

return aux1;

}

public String desmaterializar(ArrayList posicoes,int qtde,String incubacao,String
entidade)throws SQLException
{
    try
    {
        transfere( incubacao, entidade, qtde);

        int y,x;

        Connection con = criaConexao();
        y=posicoes.size();
        aux="";
        for (x=0;x<y;x++){
            stmt=con.createStatement();
            Posicao posicao = (Posicao)posicoes.get(x);
            System.out.println("ent:"+posicao.getEntidade());
            System.out.println("x:"+posicao.getPosicaoX());
            System.out.println("y:"+posicao.getPosicaoY());
            System.out.println("z:"+posicao.getPosicaoZ());
            aux+="INSERT INTO nastement
(codnas,codent,nlinban,ncolban,codposbannes,codstatusgeral,codmotdes,ovonasceu) values (";
            aux+=posicao.getNascimento()+",";
            aux+=posicao.getEntidade();
            aux+=","+posicao.getPosicaoX();
            aux+=","+posicao.getPosicaoY();

```

```

        aux+=","+posicao.getPosicaoZ();
        aux+=",'1','1','0'");";

    }

    System.out.println("aux="+aux);
    stmt.execute(aux);
    stmt.close();

    stmt=con.createStatement();
    aux="select codnas,dataprevnas from nascimento where codnas=(select
MAX(codnas) as codnas from nascimento);";
    rs=stmt.executeQuery(aux);

    while(rs.next()){
        aux1=rs.getString("codnas");
        aux1+="|";
        aux1+=rs.getString("dataprevnas");

    }System.out.println("*****retorna="+aux1
);
    }
    catch(SQLException e)
    {
        System.out.println (e.getMessage());
        e.printStackTrace();
        System.out.println("SQL Exception no metodo desmaterializar
BrokerIncubacao Query->" + aux);
    }
    catch(Exception e)
    {
        System.out.println (e.getMessage());
        e.printStackTrace();
        System.out.println("saiu no metodo selectAllNascimento
BrokerLogin"+aux);
    }

    return aux1;

}
// metodo retorna

public Vector retornaConsulta(String campo, String valor)
{
    String x="";

    try
    {
        consulta = new Vector();
        con = criaConexao();
        stmt=con.createStatement();
        aux="select * from Incubacao where "+ campo+" like '%" + valor+"%' order by
"+campo;
        rs = stmt.executeQuery(aux);
        while(rs.next())
        {
            x = rs.getString("codusu");
            x += " - ";
            x += rs.getString("matrusu");
            x += " : ";
            x += rs.getString("nomeusu");
            consulta.addElement(x);
        }
        rs.close();
        con.close();
        stmt.close();
    }
    catch(SQLException e)
    {
        System.out.println (e.getMessage());
        e.printStackTrace();
    }
}

```

```

    }
    catch(Exception e)
    {
        System.out.println (e.getMessage());
        e.printStackTrace();
    }

    return consulta;
}

//metodo retorna consulta de Incubacao-sobrecarga do anterior, pesquisa com 2 campos.
public Vector retornaConsulta(String campo, String valor,String campo2,String valor2)
{
    String x="";

    try
        //consulta no banco
    {
        consulta = new Vector();
        con = criaConexao();
        stmt=con.createStatement();
        aux="select * from Incubacao where "+ campo+" like '%" + valor+"%'or "+campo2+"
like '%" +valor2+"%' order by "+campo;
        rs = stmt.executeQuery(aux);
        while(rs.next())
        {
            x = rs.getString("codusu");
            x += " - ";
            x += rs.getString("matrusu");
            x += " : ";
            x += rs.getString("nomeusu");
            consulta.addElement(x);
        }
        rs.close();
        con.close();
        stmt.close();
    }
    catch(SQLException e)
    {
        System.out.println (e.getMessage());
        e.printStackTrace();
    }

    catch(Exception e)
    {
        System.out.println (e.getMessage());
        e.printStackTrace();
    }

    return consulta;
}

public Vector retornaConsulta()
{
    String x="";

    try
    {
        consulta = new Vector();
        con = criaConexao();
        stmt=con.createStatement();
        aux="select * from Incubacao";
        rs = stmt.executeQuery(aux);
        while(rs.next())
        {
            x = rs.getString("codusu");
            x += " - ";
            x += rs.getString("matrusu");
            x += " : ";
            x += rs.getString("nomeusu");

```

```

        consulta.addElement(x);
    }
    rs.close();
    con.close();
    stmt.close();
}
catch(SQLException e)
{
    System.out.println (e.getMessage());
    e.printStackTrace();
}
catch(Exception e)
{
    System.out.println (e.getMessage());
    e.printStackTrace();
}

return consulta;
}
// método que constroi uma objeto Incubacao
public Incubacao materializar(String codigo)
{
    Incubacao auxIncubacao= new Incubacao();

    try
    {
        con = criaConexao();
        stmt=con.createStatement();
        aux="select * from Incubacao where codusu="+ codigo;
        rs = stmt.executeQuery(aux);
        while(rs.next())
        {
            //auxIncubacao.setBandeja(rs.getString("nomeusu"));
            auxIncubacao.setDatini(rs.getString("data"));
            auxIncubacao.setProjeto(rs.getString("pro"));
            //auxIncubacao.setEntidade(rs.getString("email"));
        }
        rs.close();
        con.close();
        stmt.close();
    }
    catch(SQLException e)
    {
        System.out.println (e.getMessage());
        e.printStackTrace();
        System.out.println("SQL Exception no metodo desmaterializar BrokerIncubacao
Query->" + aux);
    }
    catch(Exception e)
    {
        System.out.println (e.getMessage());
        e.printStackTrace();
        System.out.println("saiu no metodo selectAllIncubacao BrokerLogin");
    }

    return auxIncubacao;
}

public Vector setarBandeja(String bandeja){
    Vector bandejaCheia=new Vector();
    try
    {
        String x="";
        con = criaConexao();
        stmt=con.createStatement();
        aux="select * from nastement where codposbannes="+ bandeja+" and (ovonasceu
<>'S' and ovonasceu <>'N')";
        rs = stmt.executeQuery(aux);
        while(rs.next())
        {
            x=rs.getInt("codnas")+"";
            x+="-";

```

```

        x+=rs.getInt("nlinban")+"";
        x+=":";
        x+=rs.getInt("ncolban")+"";

        bandejaCheia.addElement(x);

    }
    rs.close();
    con.close();
    stmt.close();
    }
    catch(SQLException e)
    {
        System.out.println (e.getMessage());
        e.printStackTrace();
        System.out.println("SQL Exception no metodo desmaterializar BrokerNascimento
Query->" + aux);
    }
    catch(Exception e)
    {
        System.out.println (e.getMessage());
        e.printStackTrace();
        System.out.println("saiu no metodo selectAllIncubacao BrokerLogin");
    }

    return bandejaCheia;
}

}

```

12.2.1.9 BrokerProjeto

```

package broker;

import classepersistente.Projeto;
import javaUtil.TransferData;
import java.sql.*;
import java.util.*;

/**
 * Materizaliza e desmaterializa a classe no banco de dados.
 * <p>
 * Efetua inserção, atualização e consulta.
 * @author Arthur Maçaneiro Magyar
 * @author Fábio Glauco dos Santos
 * @author Luciano Padilha
 * @version 1.0
 */
public class BrokerProjeto extends BrokerConexao
{
    // declaração de variáveis
    private String siglaProj;
    private String nome;
    private String entidade;
    private String especie;
    private String dataInicio;
    private String dataPrev;
    private String dataFim;
    private String descricao;
    private String status;
    // variáveis do broker
    private Vector consulta;
    private Statement stmt;
    private String aux;
    private ResultSet rs;

    /**
     * Insere dados no banco de dados.

```

```

* @return sucesso ou falha SQL
*/
public String desmaterializar()throws SQLException
{
    TransferData data = new TransferData();
    try
    {
        Connection con = criaConexao();
        stmt=con.createStatement();
        aux="INSERT INTO Projeto("
        aux+="CodProj";
        aux+="CodEsp";
        aux+="CodTipoEnt";
        aux+="CodStatusProj";
        aux+="SiglaProj";
        aux+="NomeProj";
        aux+="DataIniProj";
        aux+="DataPrevProj";
        aux+="DataFimProj";
        aux+="DescProj";
        aux+="NumInc) ";
        aux+="VALUES(";
        aux+="DEFAULT";
        aux+=","+especie;
        aux+=","+entidade;
        aux+=",1";
        aux+=","'+siglaProj+'";
        aux+=","'+nome+'";
        aux+=","'+data.DataPGravar(dataInicio)+'";
        aux+=","'+data.DataPGravar(dataPrev)+'";
        aux+=",'01-01-9999'";
        aux+=","'+descricao+'";
        aux+=",0);";
        stmt.execute(aux);
        con.close();
        stmt.close();
    }
    catch(SQLException e)
    {
        System.out.println("SQL Exception no BrokerProjeto: ");
        e.printStackTrace();
        System.out.println("Comando SQL: "+aux);
        return e.getMessage().toString();
    }
    catch(Exception e)
    {
        e.printStackTrace();
        System.out.println("Saiu do Método BrokerProjeto.");
        return e.getMessage().toString();
    }
    return "Projeto";
}

/**
* Atualiza dados no banco de dados.
* @param sigla sigla do projeto
* @return sucesso ou falha SQL
*/
public String desmaterializar(String sigla)throws SQLException
{
    TransferData data = new TransferData();
    try
    {
        Connection con = criaConexao();
        stmt=con.createStatement();
        aux="UPDATE Projeto SET ";
        aux+="CodEsp='"+especie+'";
        aux+="CodTipoEnt='"+entidade+'";
        aux+="CodStatusProj='"+status+'";
        aux+="NomeProj='"+nome+'";
        aux+="DataIniProj='"+data.DataPGravar(dataInicio)+'";
        aux+="DataPrevProj='"+data.Prev+'";
        aux+="DescProj='"+descricao+' ";
        aux+="WHERE SiglaProj='"+sigla+'";
        stmt.executeUpdate(aux);
        con.close();
    }
}

```

```

        stmt.close();
    }
    catch(SQLException e)
    {
        System.out.println("SQL Exception no BrokerProjeto: ");
        e.printStackTrace();
        System.out.println("Comando SQL: "+aux);
        return e.getMessage().toString();
    }
    catch(Exception e)
    {
        e.printStackTrace();
        System.out.println("Saiu do Método BrokerProjeto.");
        return e.getMessage().toString();
    }
    return "ProjetoAlterar";
}

/**
 * Atualiza para excluído no banco de dados.
 * @param siglaProj sigla do projeto
 * @return sucesso ou falha SQL
 */
public String mudarProjeto(String siglaProj)
{
    try
    {
        Connection con = criaConexao();
        stmt=con.createStatement();
        aux="UPDATE projeto SET ";
        aux+="CodStatusProj=3 ";
        aux+="WHERE SiglaProj='"+siglaProj+"'";
        stmt.executeQuery(aux);
        con.close();
        stmt.close();
    }
    catch(SQLException e)
    {
        System.out.println("SQL Exception no BrokerProjeto: ");
        e.printStackTrace();
        System.out.println("Comando SQL: "+aux);
        return e.getMessage().toString();
    }
    catch(Exception e)
    {
        e.printStackTrace();
        System.out.println("Saiu do Método BrokerProjeto.");
        return e.getMessage().toString();
    }
    return "ProjetoExclui";
}

/**
 * Consulta no banco de dados.
 * @param t
 * @param valor
 * @param status
 * @return consulta, vetor contendo o resultado
 */
public Vector retornaConsulta(String t, String valor, String status)
{
    TransferData data = new TransferData();
    String x="";
    try
    {
        consulta = new Vector();
        con = criaConexao();
        stmt=con.createStatement();
        aux="SELECT * FROM Projeto WHERE ";
        aux+="CodStatusProj="+status+" AND ";
        // pesquisa por: SiglaProj
        if(t=="s"){
            aux+="UPPER(SiglaProj) LIKE ";
            aux+="UPPER('%"+valor+"%')";
        }
    }
}

```

```

else
{
    // pesquisa por: NomeProj
    if (t=="n"){
        aux+="UPPER(NomeProj) LIKE ";
        aux+="UPPER('%"+valor+"%')";
    }
    else
    {
        // pesquisa por: DataIniProj
        if (t=="d")
        {
            aux+="DataIniProj<='"+data.DataPGGravar(valor)+"'";
        }
        else
        {
            // pesquisa por sem: DataFimProj
            if (t=="a")
            {
                aux+="DataFimProj='9999-01-01'";
            }
            else
            {
                if (t=="x")
                {
                    // pesquisa por: SiglaProj e
                    int index = valor.indexOf('-');
                    String
                    String
                    aux+="(DataIniProj<'+valor2+' AND
                    aux+="DataFimProj>'+valor2+') AND
                    aux+="(SiglaProj LIKE
                }
                else
                {
                    // pesquisa por: NomeProj e
                    if (t=="y"){
                        int index = valor.indexOf('-');
                        String
                        String
                        aux+="(DataIniProj<'+valor2+' AND ";
                        aux+="DataFimProj>'+valor2+') AND ";
                        aux+="(NomeProj LIKE
                    }
                }
            }
        }
    }
}
rs=stmt.executeQuery(aux);
while (rs.next ())
{
    x=rs.getString("SiglaProj");
    x+=" * ";
    x+=rs.getString("NomeProj");
    x+=" : ";
    x+=rs.getString("DataIniProj");
    x+=" | ";
    x+=rs.getString("CodProj");
    consulta.addElement(x);
}
rs.close();
con.close();

```

```

        stmt.close();
    }
    catch(SQLException e)
    {
        System.out.println("SQL Exception no BrokerProjeto: ");
        e.printStackTrace();
        System.out.println("Comando SQL: "+aux);
    }
    catch(Exception e)
    {
        e.printStackTrace();
        System.out.println("Saiu do Método BrokerProjeto.");
    }
    return consulta;
}

/**
 * Constrói objeto especificado.
 * @param codigo código do projeto
 * @return o objeto e seus atributos
 */
public Projeto materializar(String codigo)
{
    Projeto auxProjeto = new Projeto();
    try
    {
        con = criaConexao();
        stmt=con.createStatement();
        aux="SELECT * FROM Projeto ";
        aux+="WHERE CodProj="+codigo+";";
        rs=stmt.executeQuery(aux);
        while(rs.next())
        {
            auxProjeto.setSiglaProj(rs.getString("SiglaProj"));
            auxProjeto.setNome(rs.getString("NomeProj"));
            auxProjeto.setEntidade(rs.getString("CodTipoEnt"));
            auxProjeto.setEspecie(rs.getString("CodEsp"));
            auxProjeto.setDataPrev(rs.getString("DataPrevProj"));
            auxProjeto.setDataInicio(rs.getString("DataIniProj"));
            auxProjeto.setDataFim(rs.getString("DataFimProj"));
            auxProjeto.setDescricao(rs.getString("DescProj"));
            auxProjeto.setStatus(rs.getString("CodStatusProj"));
        }
        rs.close();
        con.close();
        stmt.close();
    }
    catch(SQLException e)
    {
        System.out.println("SQL Exception no BrokerProjeto: ");
        e.printStackTrace();
        System.out.println("Comando SQL: "+aux);
    }
    catch(Exception e)
    {
        e.printStackTrace();
        System.out.println("Saiu do Método BrokerProjeto.");
    }
    return auxProjeto;
}

// beans - gets e sets
public String getSiglaProj() {
    return siglaProj;
}
public void setSiglaProj(String siglaProj) {
    this.siglaProj = siglaProj;
}
public String getNome() {
    return nome;
}
public void setNome(String nome) {
    this.nome = nome;
}
public String getEntidade() {
    return entidade;
}

```

```

    }
    public void setEntidade(String entidade) {
        this.entidade = entidade;
    }
    public String getEspecie() {
        return especie;
    }
    public void setEspecie(String especie) {
        this.especie = especie;
    }
    public String getDataInicio() {
        return dataInicio;
    }
    public void setDataInicio(String dataInicio) {
        this.dataInicio = dataInicio;
    }
    public String getDataPrev() {
        return dataPrev;
    }
    public void setDataPrev(String dataPrev) {
        this.dataPrev = dataPrev;
    }
    public String getDataFim() {
        return dataFim;
    }
    public void setDataFim(String dataFim) {
        this.dataFim = dataFim;
    }
    public String getDescricao() {
        return descricao;
    }
    public void setDescricao(String descricao) {
        this.descricao = descricao;
    }
    public String getStatus() {
        return status;
    }
    public void setStatus(String status) {
        this.status = status;
    }
    //beans do broker
    public String getAux() {
        return aux;
    }
    public void setAux(String aux) {
        this.aux = aux;
    }
    public Vector getConsulta() {
        return consulta;
    }
    public void setConsulta(Vector consulta) {
        this.consulta = consulta;
    }
}
}

```

12.2.1.10 BrokerSaque

```

package broker;

import java.sql.*;
import java.util.*;
import classepersistente.*;

/*@autor Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano Padilha*/
public class BrokerSaque extends BrokerConexao
{

    private Statement stmt;
    private String aux;

```

```

private int x;

public String saque(ArrayList posicoes)
throws SQLException
{
    try
    {

        Connection con = criaConexao();
        stmt=con.createStatement();
        Posicao posicao;
        //marca nasceu na tabela

        for(x=0;x<posicoes.size();x++){

            posicao=(Posicao)posicoes.get(x);
            aux="UPDATE nastement SET
OvoNasceu='"+posicao.getNasceu()+"' where ((nlinban="+posicao.getPosicaoX()+" and
ncolban="+posicao.getPosicaoY()+" ) and codposbannes="+posicao.getPosicaoZ()+" )";
            System.out.println("comando Sql:"+aux);
            stmt.executeUpdate(aux);

            System.out.println("posicao.getNasceu"+posicao.getNasceu());
        }

        stmt.close();
        con.close();

    }catch(SQLException e)
    {

        e.printStackTrace();

    }

    return "sucesso";
}
}

```

12.2.1.11 BrokerUsuario

```

package broker;

import classepersistente.Usuario;
import java.sql.*;
import java.util.*;

/**
 * Materizaliza e desmaterializa a classe no banco de dados.
 * <p>
 * Efetua inserção, atualização e consulta.
 * @author Arthur Maçaneiro Magyar
 * @author Fábio Glauco dos Santos
 * @author Luciano Padilha
 * @version 1.0
 */
public class BrokerUsuario extends BrokerConexao
{
    // declaração de variáveis
    private String matricula;
    private String nome;
    private String grupo;
    private String senha;
    private String telres;
    private String telcom;
    private String telcel;
}

```

```

private String email;
private String status;
// variáveis do broker
private Vector consulta;
private Statement stmt;
private String aux;
private ResultSet rs;

/**
 * Insere dados no banco de dados.
 * @return sucesso ou falha SQL
 */
public String desmaterializar()throws SQLException
{
    try
    {
        Connection con = criaConexao();
        stmt=con.createStatement();
        aux="INSERT INTO Usuario(";
        aux+="CodUsu";
        aux+=","CodStatusUsu";
        aux+=","CodGru";
        aux+=","MatrUsu";
        aux+=","NomeUsu";
        aux+=","SenCrt";
        aux+=","DataInc";
        aux+=","TelRes";
        aux+=","TelCom";
        aux+=","TelCel";
        aux+=","Email";
        aux+=","UltAltSen";
        aux+=","UltLogon) ";
        aux+="VALUES(";
        aux+="DEFAULT";
        aux+=","1";
        aux+=",""+grupo;
        aux+=",""+matricula+""";
        aux+=",""+nome+""";
        aux+=",""+senha+""";
        aux+=","CURRENT_TIMESTAMP";
        aux+=",""+telres;
        aux+=",""+telcom;
        aux+=",""+telcel;
        aux+=",""+email+""";
        aux+=","CURRENT_TIMESTAMP";
        aux+=","CURRENT_TIMESTAMP);";
        stmt.execute(aux);
        con.close();
        stmt.close();
    }
    catch(SQLException e)
    {
        System.out.println("SQL Exception no BrokerUsuario: ");
        e.printStackTrace();
        System.out.println("Comando SQL: "+aux);
        return e.getMessage().toString();
    }
    catch(Exception e)
    {
        e.printStackTrace();
        System.out.println("Saiu do Método BrokerUsuario.");
        return e.getMessage().toString();
    }
    return "Usuario";
}

/**
 * Atualiza dados no banco de dados.
 * @param matricula matricula do usuário
 * @return sucesso ou falha SQL
 */
public String desmaterializar(String matricula)throws SQLException
{
    try
    {
        Connection con = criaConexao();

```

```

        stmt=con.createStatement();
        aux="UPDATE Usuario SET ";
        aux+="CodGru="+grupo;
        aux+="CodStatusUsu="+status;
        aux+="NomeUsu='"+nome+"'";
        aux+="SenCrt='"+senha+"'";
        aux+="TelRes="+telres;
        aux+="TelCom="+telcom;
        aux+="TelCel="+telcel;
        aux+="Email='"+email+"' ";
        aux+="WHERE MatrUsu='"+matricula+"'";
        stmt.executeUpdate(aux);
    con.close();
    stmt.close();
    System.out.println("SQL "+aux);
}
catch(SQLException e)
{
    System.out.println("SQL Exception no BrokerUsuario: ");
    e.printStackTrace();
    System.out.println("Comando SQL: "+aux);
    return e.getMessage().toString();
}
catch(Exception e)
{
    e.printStackTrace();
    System.out.println("Saiu do Método BrokerUsuario.");
    return e.getMessage().toString();
}
return "UsuarioAlterar";
}

/**
 * Atualiza para excluído no banco de dados.
 * @param mat matrícula do usuário
 * @return sucesso ou falha SQL
 */
public String mudarUsuario(String mat)
{
    try
    {
        Connection con = criaConexao();
        stmt=con.createStatement();
        aux="UPDATE Usuario SET ";
        aux+="CodStatusUsu=3 ";
        aux+="WHERE MatrUsu='"+mat+"'";
        stmt.executeUpdate(aux);
        con.close();
        stmt.close();
    }
    catch(SQLException e)
    {
        System.out.println("SQL Exception no BrokerUsuario: ");
        e.printStackTrace();
        System.out.println("Comando SQL: "+aux);
        return e.getMessage().toString();
    }
    catch(Exception e)
    {
        e.printStackTrace();
        System.out.println("Saiu do Método BrokerUsuario.");
        return e.getMessage().toString();
    }
    return "UsuarioExclui";
}

/**
 * Consulta no banco de dados.
 * @param campo registro a ser pesquisado
 * @param valor conteúdo do registro
 * @param status código do status do usuário
 * @return consulta, vetor contendo o resultado
 */
public Vector retornaConsulta(String status)
{
    String x="";

```

```

try
{
    consulta = new Vector();
    con = criaConexao();
    stmt=con.createStatement();
    aux="SELECT * FROM Usuario ";
    aux+="WHERE CodStatusUsu="+status+" ";
    aux+="ORDER BY MatrUsu;";
    rs=stmt.executeQuery(aux);
    while(rs.next())
    {
        x=rs.getString("CodUsu");
        x+=" - ";
        x+=rs.getString("MatrUsu");
        x+=" : ";
        x+=rs.getString("NomeUsu");
        consulta.addElement(x);
    }
    rs.close();
    con.close();
    stmt.close();
}
catch(SQLException e)
{
    System.out.println("SQL Exception no BrokerUsuario: ");
    e.printStackTrace();
    System.out.println("Comando SQL: "+aux);
}
catch(Exception e)
{
    e.printStackTrace();
    System.out.println("Saiu do Método BrokerUsuario.");
}
return consulta;
}

/**
 * Consulta no banco de dados com método de 1 campo.
 * @param campo registro a ser pesquisado
 * @param valor conteúdo do registro
 * @param status código do status do usuário
 * @return consulta, vetor contendo o resultado
 */
public Vector retornaConsulta(String campo, String valor, String status)
{
    String x="";
    try
    {
        consulta = new Vector();
        con = criaConexao();
        stmt=con.createStatement();
        aux="SELECT * FROM Usuario ";
        aux+="WHERE UPPER("+campo+" )";
        aux+="LIKE UPPER('%"+valor+"%') AND ";
        aux+="CodStatusUsu="+status+" ";
        aux+="ORDER BY "+campo+";";
        rs=stmt.executeQuery(aux);
        while(rs.next())
        {
            x=rs.getString("CodUsu");
            x+=" - ";
            x+=rs.getString("MatrUsu");
            x+=" : ";
            x+=rs.getString("NomeUsu");
            consulta.addElement(x);
        }
        rs.close();
        con.close();
        stmt.close();
    }
    catch(SQLException e)
    {
        System.out.println("SQL Exception no BrokerUsuario: ");
        e.printStackTrace();
        System.out.println("Comando SQL: "+aux);
    }
}

```

```

    }
    catch(Exception e)
    {
        e.printStackTrace();
        System.out.println("Saiu do Método BrokerUsuario.");
    }
    return consulta;
}

/**
 * Consulta no banco de dados com método de 2 campos.
 * @param campo registro 1 a ser pesquisado
 * @param valor conteúdo do registro 1
 * @param campo2 registro 2 a ser pesquisado
 * @param valor2 conteúdo do registro 2
 * @param status código do status do usuário
 * @return consulta, vetor contendo o resultado
 */
public Vector retornaConsulta(String campo, String valor, String campo2, String valor2,
String status)
{
    String x="";
    try
    {
        consulta = new Vector();
        con = criaConexao();
        stmt=con.createStatement();
        aux="SELECT * FROM Usuario ";
        aux+="WHERE (" +campo+ " ";
        aux+="LIKE '%" +valor+"%' ";
        aux+="OR " +campo2+ " ";
        aux+="LIKE '%" +valor2+"%' ) AND ";
        aux+="CodStatusUsu="+status+" ";
        aux+="ORDER BY " +campo+";";
        rs=stmt.executeQuery(aux);
        while(rs.next())
        {
            x=rs.getString("CodUsu");
            x+=" - ";
            x+=rs.getString("MatrUsu");
            x+=" : ";
            x+=rs.getString("NomeUsu");
            consulta.addElement(x);
        }
        rs.close();
        con.close();
        stmt.close();
    }
    catch(SQLException e)
    {
        System.out.println("SQL Exception no BrokerUsuario: ");
        e.printStackTrace();
        System.out.println("Comando SQL: "+aux);
    }
    catch(Exception e)
    {
        e.printStackTrace();
        System.out.println("Saiu do Método BrokerUsuario.");
    }
    return consulta;
}

/**
 * Constrói objeto especificado.
 * @param codigo código do usuário
 * @return o objeto e seus atributos
 */
public Usuario materializar(String codigo)
{
    Usuario auxUsuario = new Usuario();
    try
    {
        con = criaConexao();
        stmt=con.createStatement();
        aux="SELECT * FROM Usuario ";
        aux+="WHERE CodUsu="+codigo+";";
    }
}

```

```

        rs=stmt.executeQuery(aux);
        while(rs.next())
        {
            auxUsuario.setMatricula(rs.getString("MatrUsu"));
            auxUsuario.setNome(rs.getString("NomeUsu"));
            auxUsuario.setGrupo(rs.getString("CodGru"));
            auxUsuario.setSenha(rs.getString("SenCrt"));
            auxUsuario.setTelcel(rs.getString("TelCel"));
            auxUsuario.setTelcom(rs.getString("TelCom"));
            auxUsuario.setTelres(rs.getString("TelRes"));
            auxUsuario.setEmail(rs.getString("Email"));
            auxUsuario.setStatus(rs.getString("CodStatusUsu"));
        }
        rs.close();
        con.close();
        stmt.close();
    }
    catch(SQLException e)
    {
        System.out.println("SQL Exception no BrokerUsuario: ");
        e.printStackTrace();
        System.out.println("Comando SQL: "+aux);
    }
    catch(Exception e)
    {
        e.printStackTrace();
        System.out.println("Saiu do Método BrokerUsuario.");
    }
    return auxUsuario;
}

/**
 * Valida login comparando com usuários cadastrados.
 * @param l matrícula do usuário
 * @param s senha do usuário
 * @return dados do usuário com UltLogon atualizado
 */
public Usuario validaLogin(String l, String s)
{
    Usuario auxUsuario = new Usuario();
    try
    {
        int conta = 0;
        con = criaConexao();
        stmt=con.createStatement();
        aux="SELECT * FROM Usuario ";
        aux+="INNER JOIN Grupo ";
        aux+="ON Grupo.CodGru=Usuario.CodGru ";
        aux+="WHERE MatrUsu='"+l+"' ";
        aux+="AND SenCrt='"+s+"' ";
        rs=stmt.executeQuery(aux);
        while(rs.next())
        {
            auxUsuario.setMatricula(rs.getString("MatrUsu"));
            auxUsuario.setNome(rs.getString("NomeUsu"));
            auxUsuario.setGrupo(rs.getString("CodGru"));
            auxUsuario.setSenha(rs.getString("SenCrt"));
            auxUsuario.setTelcel(rs.getString("TelCel"));
            auxUsuario.setTelcom(rs.getString("TelCom"));
            auxUsuario.setTelres(rs.getString("TelRes"));
            auxUsuario.setEmail(rs.getString("Email"));
            auxUsuario.setStatus(rs.getString("CodStatusUsu"));
            auxUsuario.setUltLogon(rs.getString("UltLogon"));
            auxUsuario.setDesGru(rs.getString("NomeGru"));
            conta++;
        }
        if(conta==0)
        {
            auxUsuario.setMatricula("");
            auxUsuario.setSenha("");
            auxUsuario.setGrupo("");
        }
        else
        {
            aux="UPDATE Usuario ";
            aux+="SET UltLogon=CURRENT_TIMESTAMP ";

```

```

        aux+="WHERE MatrUsu='"+auxUsuario.getMatricula()+"'";
        stmt.executeUpdate(aux);
        con.close();
        stmt.close();
    }
}
catch(SQLException e)
{
    System.out.println("SQL Exception no BrokerUsuario: ");
    e.printStackTrace();
    System.out.println("Comando SQL: "+aux);
}
catch(Exception e)
{
    e.printStackTrace();
    System.out.println("Saiu do Método BrokerUsuario.");
}
return auxUsuario;
}

// beans - gets e sets
public String getMatricula() {
    return matricula;
}
public void setMatricula(String matricula) {
    this.matricula = matricula;
}
public String getNome() {
    return nome;
}
public void setNome(String nome) {
    this.nome = nome;
}
public String getGrupo() {
    return grupo;
}
public void setGrupo(String grupo) {
    this.grupo = grupo;
}
public String getSenha() {
    return senha;
}
public void setSenha(String senha) {
    this.senha = senha;
}
public String getTelres() {
    return telres;
}
public void setTelres(String telres) {
    this.telres = telres;
}
public String getTelcom() {
    return telcom;
}
public void setTelcom(String telcom) {
    this.telcom = telcom;
}
public String getTelcel() {
    return telcel;
}
public void setTelcel(String telcel) {
    this.telcel = telcel;
}
public String getEmail() {
    return email;
}
public void setEmail(String email) {
    this.email = email;
}
public String getStatus() {
    return status;
}
public void setStatus(String status) {
    this.status = status;
}
}

```

```
}
```

12.2.2 classepersistente

12.2.2.1 Entidade

```
package classepersistente;

import broker.BrokerEntidade;
import java.sql.*;

/**
 *
 * @author Arthur Maçaneiro Magyar
 * @author Fábio Glauco dos Santos
 * @author Luciano Padilha
 * @version 1.0
 */
public class Entidade
{
    // declaração de variáveis
    private String entidade;
    private String tipoEntidade;
    private String nome;
    private String descEnt;
    private String statusGeral;

    /**
     *
     * @return
     */
    public String gravarEntidade() throws SQLException
    {
        BrokerEntidade auxBroker = new BrokerEntidade();
        auxBroker.setNome(nome);
        auxBroker.setEntidade(entidade);
        auxBroker.setTipoEntidade(tipoEntidade);
        auxBroker.setDescEnt(descEnt);
        return auxBroker.desmaterializar();
    }

    /**
     *
     * @param sigla
     * @return
     */
    public String gravarEntidade(String sigla) throws SQLException
    {
        BrokerEntidade auxBroker = new BrokerEntidade();
        auxBroker.setEntidade(entidade);
        auxBroker.setTipoEntidade(tipoEntidade);
        auxBroker.setNome(nome);
        auxBroker.setDescEnt(descEnt);
        auxBroker.setStatusGeral(statusGeral);
        return auxBroker.desmaterializar(sigla);
    }

    // beans - gets e sets
    public String getEntidade() {
        return entidade;
    }
    public void setEntidade(String entidade) {
        this.entidade = entidade;
    }
    public String getTipoEntidade() {
        return tipoEntidade;
    }
}
```

```

    public void setTipoEntidade(String tipoEntidade) {
        this.tipoEntidade = tipoEntidade;
    }
    public String getNome() {
        return nome;
    }
    public void setNome(String nome) {
        this.nome = nome;
    }
    public String getDescEnt() {
        return descEnt;
    }
    public void setDescEnt(String descEnt) {
        this.descEnt = descEnt;
    }
    public String getStatusGeral() {
        return statusGeral;
    }
    public void setStatusGeral(String statusGeral) {
        this.statusGeral = statusGeral;
    }
}
}

```

12.2.2.2 EntidadeOvo

```

package classepersistente;

import broker.BrokerEntidadeOvo;
import java.sql.*;

/*@autor Arthur Maçaneiro Magyar, Fábio Glaucos dos Santos, Luciano Padilha*/
public class EntidadeOvo
{
    private String codigobarras;
    private String celula;
    private String cor;
    private String formato;
    private String desinfeccao;
    private String aquisicao;
    private String datacoleta;
    private String aspecto;
    private String peso;
    private String porosidade;
    private String incubavel;
    private String descricao;

    public int gravarEntidadeOvo() throws SQLException
    {
        BrokerEntidadeOvo auxBroker = new BrokerEntidadeOvo();
        auxBroker.setCodigobarras(codigobarras);
        auxBroker.setCelula(celula);
        auxBroker.setCor(cor);
        auxBroker.setFormato(formato);
        auxBroker.setDesinfeccao(desinfeccao);
        auxBroker.setAquisicao(aquisicao);
        auxBroker.setDatacoleta(datacoleta);
        auxBroker.setAspecto(aspecto);
        auxBroker.setPeso(peso);
        auxBroker.setPorosidade(porosidade);
        auxBroker.setIncubavel(incubavel);
        auxBroker.setDescricao(descricao);

        return auxBroker.desmaterializar();
    }
}

```

```

public int gravarEntidadeOvo(String codigo) throws SQLException
{
    BrokerEntidadeOvo auxBroker = new BrokerEntidadeOvo();
    auxBroker.setCodigobarras(codigobarras);
    auxBroker.setCelula(celula);
    auxBroker.setCor(cor);
    auxBroker.setFormato(formato);
    auxBroker.setDesinfeccao(desinfeccao);
    auxBroker.setAquisicao(aquisicao);
    auxBroker.setDatacoleta(datacoleta);
    auxBroker.setAspecto(aspecto);
    auxBroker.setPeso(peso);
    auxBroker.setPorosidade(porosidade);
    auxBroker.setIncubavel(incubavel);
    auxBroker.setDescricao(descricao);

    return auxBroker.desmaterializar(codigo);
}

public EntidadeOvo materializar(String codigo)
{
    BrokerEntidadeOvo brok=new BrokerEntidadeOvo();
    return brok.materializar(codigo);
}

public String getAspecto() {
    return aspecto;
}

public void setAspecto(String aspecto) {
    this.aspecto = aspecto;
}

public String getCor() {
    return cor;
}

public void setCor(String cor) {
    this.cor = cor;
}

public String getCelula() {
    return celula;
}

public void setCelula(String celula) {
    this.celula = celula;
}

public String getCodigobarras() {
    return codigobarras;
}

public void setCodigobarras(String codigobarras) {
    this.codigobarras = codigobarras;
}

public String getFormato() {
    return formato;
}

public void setFormato(String formato) {
    this.formato = formato;
}

public String getDatacoleta() {
    return datacoleta;
}

public void setDatacoleta(String datacoleta) {
    this.datacoleta = datacoleta;
}

public String getAquisicao() {
    return aquisicao;
}

```

```

    }

    public void setAquisicao(String aquisicao) {
        this.aquisicao = aquisicao;
    }

    public String getDesinfeccao() {
        return desinfeccao;
    }

    public void setDesinfeccao(String desinfeccao) {
        this.desinfeccao = desinfeccao;
    }

    public String getPeso() {
        return peso;
    }

    public void setPeso(String peso) {
        this.peso = peso;
    }

    public String getPorosidade() {
        return porosidade;
    }

    public void setPorosidade(String porosidade) {
        this.porosidade = porosidade;
    }

    public String getIncubavel() {
        return incubavel;
    }

    public void setIncubavel(String incubavel) {
        this.incubavel = incubavel;
    }

    public String getDescricao() {
        return descricao;
    }

    public void setDescricao(String descricao) {
        this.descricao = descricao;
    }
}

```

12.2.2.3 EspecieAve

```

package classepersistente;

import broker.BrokerEspecieAve;
import java.sql.*;

/*@autor Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano Padilha*/
public class EspecieAve
{
    private String especie;
    private String tempmininc;
    private String tempmaxinc;
    private String umimininc;
    private String umimaxinc;
    private String perinc;
    private String peraviinc;
    private String tempminnas;
    private String tempmaxnas;
    private String umiminnas;

```

```

private String umimaxnas;
private String pernas;
private String peravinas;
private String perrec;

public int gravarEspecieAve() throws SQLException
{
    BrokerEspecieAve auxBroker = new BrokerEspecieAve();
    auxBroker.setEspecie(especie);
    auxBroker.setTempMinInc(tempmininc);
    auxBroker.setTempMaxInc(tempmaxinc);
    auxBroker.setUmiMinInc(umimininc);
    auxBroker.setUmiMaxInc(umimaxinc);
    auxBroker.setPerInc(perinc);
    auxBroker.setPerAviInc(peraviinc);
    auxBroker.setTempMinNas(tempminnas);
    auxBroker.setTempMaxNas(tempmaxnas);
    auxBroker.setUmiMinNas(umiminnas);
    auxBroker.setUmiMaxNas(umimaxnas);
    auxBroker.setPerNas(pernas);
    auxBroker.setPerAviNas(peravinas);
    auxBroker.setPerRec(perrec);
    return auxBroker.desmaterializar();
}

public int gravarEspecieAve(String esp) throws SQLException
{
    BrokerEspecieAve auxBroker = new BrokerEspecieAve();
    auxBroker.setEspecie(especie);
    auxBroker.setTempMinInc(tempmininc);
    auxBroker.setTempMaxInc(tempmaxinc);
    auxBroker.setUmiMinInc(umimininc);
    auxBroker.setUmiMaxInc(umimaxinc);
    auxBroker.setPerInc(perinc);
    auxBroker.setPerAviInc(peraviinc);
    auxBroker.setTempMinNas(tempminnas);
    auxBroker.setTempMaxNas(tempmaxnas);
    auxBroker.setUmiMinNas(umiminnas);
    auxBroker.setUmiMaxNas(umimaxnas);
    auxBroker.setPerNas(pernas);
    auxBroker.setPerAviNas(peravinas);
    auxBroker.setPerRec(perrec);
    return auxBroker.desmaterializar(esp);
}

public EspecieAve materializar(String codigo)
{
    BrokerEspecieAve brok=new BrokerEspecieAve();
    return brok.materializar(codigo);
}

public String getTempMinNas() {
    return tempminnas;
}

public void setTempMinNas(String tempminnas) {
    this.tempminnas = tempminnas;
}

public String getTempMaxInc() {
    return tempmaxinc;
}

public void setTempMaxInc(String tempmaxinc) {
    this.tempmaxinc = tempmaxinc;
}

public String getTempMinInc() {
    return tempmininc;
}

public void setTempMinInc(String tempmininc) {
    this.tempmininc = tempmininc;
}

```

```

}

public String getEspecie() {
    return especie;
}

public void setEspecie(String especie) {
    this.especie = especie;
}

public String getUmiMinInc() {
    return umimininc;
}

public void setUmiMinInc(String umimininc) {
    this.umimininc = umimininc;
}

public String getPerAviInc() {
    return peraviinc;
}

public void setPerAviInc(String peraviinc) {
    this.peraviinc = peraviinc;
}

public String getPerInc() {
    return perinc;
}

public void setPerInc(String perinc) {
    this.perinc = perinc;
}

public String getUmiMaxInc() {
    return umimaxinc;
}

public void setUmiMaxInc(String umimaxinc) {
    this.umimaxinc = umimaxinc;
}

public String getTempMaxNas() {
    return tempmaxnas;
}

public void setTempMaxNas(String tempmaxnas) {
    this.tempmaxnas = tempmaxnas;
}

public String getUmiMinNas() {
    return umiminnas;
}

public void setUmiMinNas(String umiminnas) {
    this.umiminnas = umiminnas;
}

public String getUmiMaxNas() {
    return umimaxnas;
}

public void setUmiMaxNas(String umimaxnas) {
    this.umimaxnas = umimaxnas;
}

public String getPerAviNas() {
    return peravinas;
}

public void setPerAviNas(String peravinas) {
    this.pernas = peravinas;
}

public String getPerRec() {
    return perrec;
}

```

```

    }

    public void setPerRec(String perrec) {
        this.pernas = perrec;
    }

    public String getPerNas() {
        return pernas;
    }

    public void setPerNas(String pernas) {
        this.pernas = pernas;
    }

}

```

12.2.2.4 Incubacao

```

package classepersistente;

/*@autor Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano Padilha*/
public class Incubacao
{
    private String projeto;
    private String datini;
    private String data;

    /*public int gravarIncubacao()throws SQLException
    {
        BrokerIncubacao auxBroker = new BrokerIncubacao();

        auxBroker.setDatini(datini);

        auxBroker.setProjeto(projeto);

        return auxBroker.desmaterializar();
    }*/

    /*public int gravarIncubacao(String mat)throws SQLException
    {
        BrokerIncubacao auxBroker = new BrokerIncubacao();
        auxBroker.setBandeja(bandeja);
        auxBroker.setDatini(datini);
        auxBroker.setEntidade(entidade);
        auxBroker.setProjeto(projeto);

        return auxBroker.desmaterializar(mat);
    }*/

    public String getDatini() {
        return datini;
    }

    public void setDatini(String datini) {
        this.datini = datini;
    }
}

```

```

    public String getProjeto() {
        return projeto;
    }

    public void setProjeto(String projeto) {
        this.projeto = projeto;
    }

    public String getData() {
        return data;
    }

    public void setData(String data) {
        this.data = data;
    }

}

```

12.2.2.5 IncubadoraBandeja

```

package classepersistente;

/*@autor Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano Padilha*/
public class IncubadoraBandeja
{
    private String codigo;
    private String incubadora;
    private String nascedouro;
    private String posinc;
    private String posnas;
    private String bandeja;

    public String getCodigo() {
        return codigo;
    }
    public void setCodigo(String codigo) {
        this.codigo = codigo;
    }
    public String getIncubadora() {
        return incubadora;
    }
    public void setIncubadora(String incubadora) {
        this.incubadora = incubadora;
    }
    public String getNascedouro() {
        return nascedouro;
    }
    public void setNascedouro(String nascedouro) {
        this.nascedouro = nascedouro;
    }
    public String getPosinc() {
        return posinc;
    }
    public void setPosinc(String posinc) {
        this.posinc = posinc;
    }
    public String getPosnas() {
        return posnas;
    }
    public void setPosnas(String posnas) {
        this.posnas = posnas;
    }
    public String getBandeja() {
        return bandeja;
    }
}

```

```

    }
    public void setBandeja(String bandeja) {
        this.bandeja = bandeja;
    }
}

```

12.2.2.6 Nascimento

```

package classepersistente;

/*@autor Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano Padilha*/
public class Nascimento
{
    private String incubacao;
    private String datini;
    private String data;

    public String getDatini() {
        return datini;
    }

    public void setDatini(String datini) {
        this.datini = datini;
    }

    public String getIncubacao() {
        return incubacao;
    }

    public void setIncubacao(String incubacao) {
        this.incubacao = incubacao;
    }

    public String getData() {
        return data;
    }

    public void setData(String data) {
        this.data = data;
    }

}

```

12.2.2.7 Posicao

```

package classepersistente;

/*@autor Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano Padilha*/
public class Posicao
{
    private String entidade;
    private String incubacao;

```

```

private int posicaoX;
private int posicaoY;
private int posicaoZ;
private int nascimento;
private String nasceu;

public String getNasceu() {
    return nasceu;
}
public void setNasceu(String nasceu) {
    this.nasceu = nasceu;
}
public int getPosicaoZ() {
    return posicaoZ;
}
public void setPosicaoZ(int posicaoZ) {
    this.posicaoZ = posicaoZ;
}
public String getEntidade() {
    return entidade;
}
public void setEntidade(String entidade) {
    this.entidade = entidade;
}
public String getIncubacao() {
    return incubacao;
}
public void setIncubacao(String incubacao) {
    this.incubacao = incubacao;
}
public int getPosicaoX() {
    return posicaoX;
}
public void setPosicaoX(int posicaoX) {
    this.posicaoX = posicaoX;
}
public int getPosicaoY() {
    return posicaoY;
}
public void setPosicaoY(int posicaoY) {
    this.posicaoY = posicaoY;
}
public int getNascimento() {
    return nascimento;
}
public void setNascimento(int nascimento) {
    this.nascimento = nascimento;
}
}
}

```

12.2.2.8 Projeto

```

package classepersistente;

import broker.BrokerProjeto;
import java.sql.*;

/**
 *
 * @author Arthur Maçaneiro Magyar
 */

```

```

* @author      Fábio Glauco dos Santos
* @author      Luciano Padilha
* @version     1.0
*/
public class Projeto
{
    // declaração de variáveis
    private String siglaProj;
    private String nome;
    private String entidade;
    private String especie;
    private String dataInicio;
    private String dataPrev;
    private String dataFim;
    private String descricao;
    private String status;

    /**
     *
     * @return
     */
    public String gravarProjeto() throws SQLException
    {
        BrokerProjeto auxBroker = new BrokerProjeto();
        auxBroker.setSiglaProj(siglaProj);
        auxBroker.setNome(nome);
        auxBroker.setEntidade(entidade);
        auxBroker.setEspecie(especie);
        auxBroker.setDataInicio(dataInicio);
        auxBroker.setDataPrev(dataPrev);
        auxBroker.setDataFim(dataFim);
        auxBroker.setDescricao(descricao);
        return auxBroker.desmaterializar();
    }

    /**
     * @param sigla
     * @return
     */
    public String gravarProjeto(String sigla) throws SQLException
    {
        BrokerProjeto auxBroker = new BrokerProjeto();
        auxBroker.setSiglaProj(siglaProj);
        auxBroker.setNome(nome);
        auxBroker.setEntidade(entidade);
        auxBroker.setEspecie(especie);
        auxBroker.setDataInicio(dataInicio);
        auxBroker.setDataPrev(dataPrev);
        auxBroker.setDataFim(dataFim);
        auxBroker.setDescricao(descricao);
        auxBroker.setStatus(status);
        return auxBroker.desmaterializar(sigla);
    }

    // beans - gets e sets
    public String getSiglaProj() {
        return siglaProj;
    }
    public void setSiglaProj(String siglaProj) {
        this.siglaProj = siglaProj;
    }
    public String getNome() {
        return nome;
    }
    public void setNome(String nome) {
        this.nome = nome;
    }
    public String getEntidade() {
        return entidade;
    }
    public void setEntidade(String entidade) {
        this.entidade = entidade;
    }
    public String getEspecie() {
        return especie;
    }
}

```

```

    public void setEspecie(String especie) {
        this.especie = especie;
    }
    public String getDataInicio() {
        return dataInicio;
    }
    public void setDataInicio(String dataInicio) {
        this.dataInicio = dataInicio;
    }
    public String getDataPrev() {
        return dataPrev;
    }
    public void setDataPrev(String dataPrev) {
        this.dataPrev = dataPrev;
    }
    public String getDataFim() {
        return dataFim;
    }
    public void setDataFim(String dataFim) {
        this.dataFim = dataFim;
    }
    public String getDescricao() {
        return descricao;
    }
    public void setDescricao(String descricao) {
        this.descricao = descricao;
    }
    public String getStatus() {
        return status;
    }
    public void setStatus(String status) {
        this.status = status;
    }
}
}

```

12.2.2.9 Usuario

```

package classepersistente;

import broker.BrokerUsuario;
import java.sql.*;

/**
 *
 * @author Arthur Maçaneiro Magyar
 * @author Fábio Glaucos dos Santos
 * @author Luciano Padilha
 * @version 1.0
 */
public class Usuario
{
    // declaração de variáveis
    private String matricula;
    private String nome;
    private String grupo;
    private String senha;
    private String telres;
    private String telcom;
    private String telcel;
    private String email;
    private String status;
    // variáveis para o broker
    private String data;
    private String desgrupo;

    /**
     *
     * @return
     */
    public String gravarUsuario() throws SQLException
    {
        BrokerUsuario auxBroker = new BrokerUsuario();
    }
}

```

```

        auxBroker.setMatricula(matricula);
        auxBroker.setNome(nome);
        auxBroker.setGrupo(grupo);
        auxBroker.setSenha(senha);
        auxBroker.setTelres(telres);
        auxBroker.setTelcom(telcom);
        auxBroker.setTelcel(telcel);
        auxBroker.setEmail(email);
        return auxBroker.desmaterializar();
    }

    /**
     *
     * @param mat
     * @return
     */
    public String gravarUsuario(String mat) throws SQLException
    {
        BrokerUsuario auxBroker = new BrokerUsuario();
        auxBroker.setMatricula(matricula);
        auxBroker.setNome(nome);
        auxBroker.setGrupo(grupo);
        auxBroker.setSenha(senha);
        auxBroker.setTelres(telres);
        auxBroker.setTelcom(telcom);
        auxBroker.setTelcel(telcel);
        auxBroker.setEmail(email);
        auxBroker.setStatus(status);
        return auxBroker.desmaterializar(mat);
    }

    // beans - gets e sets
    public String getMatricula() {
        return matricula;
    }
    public void setMatricula(String matricula) {
        this.matricula = matricula;
    }
    public String getNome() {
        return nome;
    }
    public void setNome(String nome) {
        this.nome = nome;
    }
    public String getGrupo() {
        return grupo;
    }
    public void setGrupo(String grupo) {
        this.grupo = grupo;
    }
    public String getSenha() {
        return senha;
    }
    public void setSenha(String senha) {
        this.senha = senha;
    }
    public String getTelres() {
        return telres;
    }
    public void setTelres(String telres) {
        this.telres = telres;
    }
    public String getTelcom() {
        return telcom;
    }
    public void setTelcom(String telcom) {
        this.telcom = telcom;
    }
    public String getTelcel() {
        return telcel;
    }
    public void setTelcel(String telcel) {
        this.telcel = telcel;
    }
    public String getEmail() {
        return email;
    }

```

```

    }
    public void setEmail(String email) {
        this.email = email;
    }
    public String getStatus() {
        return status;
    }
    public void setStatus(String status) {
        this.status = status;
    }
    // beans para o broker
    public void setUltLogon(String data) {
        this.data = data;
    }
    public String getUltLogon() {
        return data;
    }
    public void setDesGru(String desgrupo) {
        this.desgrupo = desgrupo;
    }
    public String getDesGru() {
        return desgrupo;
    }
}

```

12.2.3 controladora

12.2.3.1 ControladoraConsulta

```

package controladora;

import broker.*;
import java.util.*;

/**
 *
 * @author Arthur Maçaneiro Magyar
 * @author Fábio Glaucos dos Santos
 * @author Luciano Padilha
 * @version 1.0
 */
public class ControladoraConsulta {

    // Bandeja
    public Vector retornaBandeja(){
        BrokerConsulta brok = new BrokerConsulta();
        return brok.retornaBandeja();
    }
    public Vector retornaBandejaAlocadaI(){
        BrokerConsulta brok = new BrokerConsulta();
        return brok.retornaBandejaAlocadaI();
    }
    public Vector retornaBandejaAlocadaN(){
        BrokerConsulta brok = new BrokerConsulta();
        return brok.retornaBandejaAlocadaN();
    }
    public Vector retornaBandejaDesalocada(){
        BrokerConsulta brok = new BrokerConsulta();
        return brok.retornaBandejaDesalocada();
    }
    public Vector retornaBandejaNas(){
        BrokerConsulta brok = new BrokerConsulta();
        return brok.retornaBandejaNas();
    }
    public String retornaPosBan(String cod){
        BrokerConsulta brok = new BrokerConsulta();
        return brok.retornaPosBan(cod);
    }
}

```

```

}
public String retornaPosBanNas(String cod){
    BrokerConsulta brok = new BrokerConsulta();
    return brok.retornaPosBanNas(cod);
}
public String retornaTamBan(String cod){
    BrokerConsulta brok = new BrokerConsulta();
    return brok.retornaTamBan(cod);
}

// Celula
public Vector retornaCelula(){
    BrokerConsulta brok = new BrokerConsulta();
    return brok.retornaCelula();
}

// Entidade
public Vector retornaEntidade(){
    BrokerConsulta brok = new BrokerConsulta();
    return brok.retornaEntidade();
}
public Vector retornaEntidade(String ent){
    BrokerConsulta brok = new BrokerConsulta();
    return brok.retornaEntidade(ent);
}
public String retornaEntMin(String proj){
    BrokerConsulta brok = new BrokerConsulta();
    return brok.retornaEntMin(proj);
}
public Vector retornaTipoEntidade(){
    BrokerConsulta brok = new BrokerConsulta();
    return brok.retornaTipoEntidade();
}

// Especie
public int retornaDataEspecie(String proj){
    BrokerConsulta brok = new BrokerConsulta();
    return brok.retornaDataEspecie(proj);
}
public int retornaDataEspecieNas(String proj){
    BrokerConsulta brok = new BrokerConsulta();
    return brok.retornaDataEspecie(proj);
}
public Vector retornaEspecie(){
    BrokerConsulta brok = new BrokerConsulta();
    return brok.retornaEspecie();
}

// Incubacao
public String buscarQuantidade(String incubacao,String entidade){
    BrokerConsulta brok = new BrokerConsulta();
    return brok.buscarQuantidade(incubacao,entidade);
}
public String retornaDataIniInc(String cod){
    BrokerConsulta brok = new BrokerConsulta();
    return brok.retornaDataIniInc(cod);
}
public Vector retornaIncubacao(String proj){
    BrokerConsulta brok = new BrokerConsulta();
    return brok.retornaIncubacao(proj);
}
public Vector retornaEntidadeInc(String inc){
    BrokerConsulta brok = new BrokerConsulta();
    return brok.retornaEntInc(inc);
}

// Incubadora
public Vector retornaIncubadora(){
    BrokerConsulta brok = new BrokerConsulta();
    return brok.retornaIncubadora();
}
public Vector retornaPosEnc(){
    BrokerConsulta brok = new BrokerConsulta();
    return brok.retornaPosEnc();
}
}
public int retornaPosinc(String cod){

```

```

        BrokerConsulta brok = new BrokerConsulta();
        return brok.retornaPosinc(cod);
    }

    // Ovo
    public Vector retornaCorOvo(){
        BrokerConsulta brok = new BrokerConsulta();
        return brok.retornaCorOvo();
    }
    public Vector retornaDesOvo(){
        BrokerConsulta brok = new BrokerConsulta();
        return brok.retornaDesOvo();
    }
    public Vector retornaForOvo(){
        BrokerConsulta brok = new BrokerConsulta();
        return brok.retornaForOvo();
    }
    public Vector retornaStatusOvo(){
        BrokerConsulta brok = new BrokerConsulta();
        return brok.retornaStatusOvo();
    }

    // Projeto
    public Vector retornaEntidadeProj(String proj){
        BrokerConsulta brok = new BrokerConsulta();
        return brok.retornaEntidadeProj(proj);
    }
    public String retornaEspProjeto(String projeto){
        BrokerConsulta brok = new BrokerConsulta();
        return brok.retornaEspProjeto(projeto);
    }
    public Vector retornaProjeto(){
        BrokerConsulta brok = new BrokerConsulta();
        return brok.retornaProjeto();
    }
    public String retornaProjInc(String inc){
        BrokerConsulta brok = new BrokerConsulta();
        return brok.retornaProjInc(inc);
    }
    public Vector retornaStatusProj(){
        BrokerConsulta brok = new BrokerConsulta();
        return brok.retornaStatusProj();
    }

    // Status Geral
    public Vector retornaStatus(){
        BrokerConsulta brok = new BrokerConsulta();
        return brok.retornaStatus();
    }

    //Usuario
    public Vector retornaGrupo(){
        BrokerConsulta brok = new BrokerConsulta();
        return brok.retornaGrupo();
    }
    public Vector retornaStatusUsu(){
        BrokerConsulta brok = new BrokerConsulta();
        return brok.retornaStatusUsu();
    }

    // Nascledouro
    public Vector retornaNascledouro(){
        BrokerConsulta brok = new BrokerConsulta();
        return brok.retornaNascledouro();
    }

    // Nascimento
    public String retornaDadosNas(String nascimento){
        BrokerConsulta brok = new BrokerConsulta();
        return brok.retornaDadosNas(nascimento);
    }
    public Vector retornaDadosNas2(String nascimento){
        BrokerConsulta brok = new BrokerConsulta();
        return brok.retornaDadosNas2(nascimento);
    }
    public Vector retornaNascimento(String proj){

```

```

        BrokerConsulta brok = new BrokerConsulta();
        return brok.retornaNascimento(proj);
    }
    public int retornaPosnas(String cod){
        BrokerConsulta brok = new BrokerConsulta();
        return brok.retornaPosnas(cod);
    }

    // Principal
    public String retornaDadosGerais(){
        BrokerConsulta brok = new BrokerConsulta();
        return brok.retornaDadosGerais();
    }

    // Propriedade
    public Vector retornaPropriedade(){
        BrokerConsulta brok = new BrokerConsulta();
        return brok.retornaPropriedade();
    }
}

```

12.2.3.2 ControladoraEntidade

```

package controladora;

import classepersistente.Entidade;
import broker.BrokerEntidade;
import java.sql.*;
import java.util.Vector;

/**
 *
 * @author Arthur Maçaneiro Magyar
 * @author Fábio Glaucos dos Santos
 * @author Luciano Padilha
 * @version 1.0
 */
public class ControladoraEntidade
{
    // declaração de variáveis
    private String entidade;
    private String tipoEntidade;
    private String nome;
    private String descEnt;
    private String statusGeral;

    /**
     *
     * @return
     */
    public String gravarEntidade ()throws SQLException
    {
        Entidade auxEntidade = new Entidade();
        auxEntidade.setEntidade(entidade);
        auxEntidade.setTipoEntidade(tipoEntidade);
        auxEntidade.setNome(nome);
        auxEntidade.setDescEnt(descEnt);
        return auxEntidade.gravarEntidade();
    }

    /**
     *
     * @param sigla
     * @return
     */
    public String gravarEntidade (String sigla)throws SQLException
    {
        Entidade auxEntidade = new Entidade();
        auxEntidade.setEntidade(entidade);
        auxEntidade.setTipoEntidade(tipoEntidade);
        auxEntidade.setNome(nome);
    }
}

```

```

        auxEntidade.setDescEnt(descEnt);
        auxEntidade.setStatusGeral(statusGeral);
        return auxEntidade.gravarEntidade(sigla);
    }

    /**
     *
     * @param sigla
     * @return
     */
    public String mudarEntidade(String sigla)
    {
        BrokerEntidade brok = new BrokerEntidade();
        return brok.mudarEntidade(nome) ;
    }

    /**
     * Faz conexão com o broker no método de consulta enviando data ou nenhum (d,n,a).
     * @param t
     * @param valor
     * @return
     */
    public Vector retornaEntidade(String t,String tipo){
        BrokerEntidade brok = new BrokerEntidade();
        return brok.retornaConsulta(t,tipo) ;
    }

    /**
     *
     * @param codigo
     * @return
     */
    public Entidade materializar(String codigo)
    {
        BrokerEntidade brok = new BrokerEntidade();
        return brok.materializar(codigo);
    }

    // beans - gets e sets
    public String getEntidade() {
        return entidade;
    }
    public void setEntidade(String entidade) {
        this.entidade = entidade;
    }
    public String getTipoEntidade() {
        return tipoEntidade;
    }
    public void setTipoEntidade(String tipoEntidade) {
        this.tipoEntidade = tipoEntidade;
    }
    public String getNome() {
        return nome;
    }
    public void setNome(String nome) {
        this.nome = nome;
    }
    public String getDescEnt() {
        return descEnt;
    }
    public void setDescEnt(String descEnt) {
        this.descEnt = descEnt;
    }
    public String getStatusGeral() {
        return statusGeral;
    }
    public void setStatusGeral(String statusGeral) {
        this.statusGeral = statusGeral;
    }
}

```

12.2.3.3 ControladoraEntidadeOvo

```

package controladora;

import broker.*;
import classepersistente.*;
import java.sql.*;

/*@autor Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano Padilha*/
public class ControladoraEntidadeOvo
{
    private String codigobarras;
    private String celula;
    private String cor;
    private String formato;
    private String desinfeccao;
    private String aquisicao;
    private String datacoleta;
    private String aspecto;
    private String peso;
    private String porosidade;
    private String incubavel;
    private String descricao;

    public int gravarEntidadeOvo ()throws SQLException
    {
        EntidadeOvo auxEntidadeOvo = new EntidadeOvo();
        auxEntidadeOvo.setCodigobarras(codigobarras);
        auxEntidadeOvo.setCelula(celula);
        auxEntidadeOvo.setCor(cor);
        auxEntidadeOvo.setFormato(formato);
        auxEntidadeOvo.setDesinfeccao(desinfeccao);
        auxEntidadeOvo.setAquisicao(aquisicao);
        auxEntidadeOvo.setDatacoleta(datacoleta);
        auxEntidadeOvo.setAspecto(aspecto);
        auxEntidadeOvo.setPeso(peso);
        auxEntidadeOvo.setPorosidade(porosidade);
        auxEntidadeOvo.setIncubavel(incubavel);
        auxEntidadeOvo.setDescricao(descricao);

        return auxEntidadeOvo.gravarEntidadeOvo();
    }

    public int gravarEntidadeOvo (String codigo)throws SQLException
    {
        EntidadeOvo auxEntidadeOvo = new EntidadeOvo();
        auxEntidadeOvo.setCodigobarras(codigobarras);
        auxEntidadeOvo.setCelula(celula);
        auxEntidadeOvo.setCor(cor);
        auxEntidadeOvo.setFormato(formato);
        auxEntidadeOvo.setDesinfeccao(desinfeccao);
        auxEntidadeOvo.setAquisicao(aquisicao);
        auxEntidadeOvo.setDatacoleta(datacoleta);
        auxEntidadeOvo.setAspecto(aspecto);
        auxEntidadeOvo.setPeso(peso);
        auxEntidadeOvo.setPorosidade(porosidade);
        auxEntidadeOvo.setIncubavel(incubavel);
        auxEntidadeOvo.setDescricao(descricao);
        return auxEntidadeOvo.gravarEntidadeOvo(codigo);
    }

    public EntidadeOvo materializar(String codigo)
    {
        BrokerEntidadeOvo brok=new BrokerEntidadeOvo();
        return brok.materializar(codigo);
    }

    public String getAspecto() {
        return aspecto;
    }

    public void setAspecto(String aspecto) {
        this.aspecto = aspecto;
    }
}

```

```

public String getCor() {
    return cor;
}

public void setCor(String cor) {
    this.cor = cor;
}

public String getCelula() {
    return celula;
}

public void setCelula(String celula) {
    this.celula = celula;
}

public String getCodigobarras() {
    return codigobarras;
}

public void setCodigobarras(String codigobarras) {
    this.codigobarras = codigobarras;
}

public String getFormato() {
    return formato;
}

public void setFormato(String formato) {
    this.formato = formato;
}

public String getDatacoleta() {
    return datacoleta;
}

public void setDatacoleta(String datacoleta) {
    this.datacoleta = datacoleta;
}

public String getAquisicao() {
    return aquisicao;
}

public void setAquisicao(String aquisicao) {
    this.aquisicao = aquisicao;
}

public String getDesinfeccao() {
    return desinfeccao;
}

public void setDesinfeccao(String desinfeccao) {
    this.desinfeccao = desinfeccao;
}

public String getPeso() {
    return peso;
}

public void setPeso(String peso) {
    this.peso = peso;
}

public String getPorosidade() {
    return porosidade;
}

public void setPorosidade(String porosidade) {
    this.porosidade = porosidade;
}

public String getIncubavel() {
    return incubavel;
}

```

```

    public void setIncubavel(String incubavel) {
        this.incubavel = incubavel;
    }

    public String getDescricao() {
        return descricao;
    }

    public void setDescricao(String descricao) {
        this.descricao = descricao;
    }
}

```

12.2.3.4 ControladoraEspecieAve

```

package controladora;

import broker.*;
import classepersistente.*;
import java.sql.*;

/*@autor Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano Padilha*/
public class ControladoraEspecieAve
{
    private String especie;
    private String tempmininc;
    private String tempmaxinc;
    private String umimininc;
    private String umimaxinc;
    private String perinc;
    private String peraviinc;
    private String tempminnas;
    private String tempmaxnas;
    private String umiminnas;
    private String umimaxnas;
    private String pernas;
    private String peravinas;
    private String perrec;

    public int gravarEspecieAve ()throws SQLException
    {
        EspecieAve auxEspecieAve = new EspecieAve();
        auxEspecieAve.setEspecie(especie);
        auxEspecieAve.setTempMinInc(tempmininc);
        auxEspecieAve.setTempMaxInc(tempmaxinc);
        auxEspecieAve.setUmiMinInc(umimininc);
        auxEspecieAve.setUmiMaxInc(umimaxinc);
        auxEspecieAve.setPerInc(perinc);
        auxEspecieAve.setPerAviInc(peraviinc);
        auxEspecieAve.setTempMinNas(tempminnas);
        auxEspecieAve.setTempMaxNas(tempmaxnas);
        auxEspecieAve.setUmiMinNas(umiminnas);
        auxEspecieAve.setUmiMaxNas(umimaxnas);
        auxEspecieAve.setPerNas(pernas);
        auxEspecieAve.setPerAviNas(peravinas);
        auxEspecieAve.setPerRec(perrec);
        return auxEspecieAve.gravarEspecieAve();
    }

    public int gravarEspecieAve (String codigo)throws SQLException
    {
        EspecieAve auxEspecieAve = new EspecieAve();
        auxEspecieAve.setEspecie(especie);
    }
}

```

```

        auxEspecieAve.setTempMinInc(tempmininc);
        auxEspecieAve.setTempMaxInc(tempmaxinc);
        auxEspecieAve.setUmiMinInc(umimininc);
        auxEspecieAve.setUmiMaxInc(umimaxinc);
        auxEspecieAve.setPerInc(perinc);
        auxEspecieAve.setPerAviInc(peraviinc);
        auxEspecieAve.setTempMinNas(tempminnas);
        auxEspecieAve.setTempMaxNas(tempmaxnas);
        auxEspecieAve.setUmiMinNas(umiminnas);
        auxEspecieAve.setUmiMaxNas(umimaxnas);
        auxEspecieAve.setPerNas(pernas);
        auxEspecieAve.setPerAviNas(peravinass);
        auxEspecieAve.setPerRec(perrec);
        return auxEspecieAve.gravarEspecieAve(codigo);
    }

    public EspecieAve materializar(String codigo)
    {
        BrokerEspecieAve brok=new BrokerEspecieAve();
        return brok.materializar(codigo);
    }

    public String getTempMinNas() {
        return tempminnas;
    }

    public void setTempMinNas(String tempminnas) {
        this.tempminnas = tempminnas;
    }

    public String getTempMaxInc() {
        return tempmaxinc;
    }

    public void setTempMaxInc(String tempmaxinc) {
        this.tempmaxinc = tempmaxinc;
    }

    public String getTempMinInc() {
        return tempmininc;
    }

    public void setTempMinInc(String tempmininc) {
        this.tempmininc = tempmininc;
    }

    public String getEspecie() {
        return especie;
    }

    public void setEspecie(String especie) {
        this.especie = especie;
    }

    public String getUmiMinInc() {
        return umimininc;
    }

    public void setUmiMinInc(String umimininc) {
        this.umimininc = umimininc;
    }

    public String getPerAviInc() {
        return peraviinc;
    }

    public void setPerAviInc(String peraviinc) {
        this.peraviinc = peraviinc;
    }

    public String getPerInc() {
        return perinc;
    }

    public void setPerInc(String perinc) {

```

```

        this.perinc = perinc;
    }

    public String getUmiMaxInc() {
        return umimaxinc;
    }

    public void setUmiMaxInc(String umimaxinc) {
        this.umimaxinc = umimaxinc;
    }

    public String getTempMaxNas() {
        return tempmaxnas;
    }

    public void setTempMaxNas(String tempmaxnas) {
        this.tempmaxnas = tempmaxnas;
    }

    public String getUmiMinNas() {
        return umiminnas;
    }

    public void setUmiMinNas(String umiminnas) {
        this.umiminnas = umiminnas;
    }

    public String getUmiMaxNas() {
        return umimaxnas;
    }

    public void setUmiMaxNas(String umimaxnas) {
        this.umimaxnas = umimaxnas;
    }

    public String getPerAviNas() {
        return peravinas;
    }

    public void setPerAviNas(String peravinas) {
        this.pernas = peravinas;
    }

    public String getPerRec() {
        return perrec;
    }

    public void setPerRec(String perrec) {
        this.pernas = perrec;
    }

    public String getPerNas() {
        return pernas;
    }

    public void setPerNas(String pernas) {
        this.pernas = pernas;
    }

}

```

12.2.3.5 ControladoraIncubacao

```

package controladora;

import broker.*;

```

```

import classepersistente.*;
import java.sql.*;
import java.util.*;

/*@autor Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano Padilha*/
public class ControladoraIncubacao
{
    private String projeto;
    private String datini;
    private String bandeja;
    private String entidade;

    public String gravarIncubacao (ArrayList incubacao,ArrayList posicoes)throws
SQLException
    {
        BrokerIncubacao auxBrokerIncubacao= new BrokerIncubacao();
        return auxBrokerIncubacao.desmaterializar(incubacao,posicoes);
    }
    /*public int gravarIncubacao (String mat)throws SQLException
    {
        Incubacao auxIncubacao = new Incubacao();
        auxIncubacao.setBandeja(bandeja);
        auxIncubacao.setDatini(datini);
        auxIncubacao.setEntidade(entidade);
        auxIncubacao.setProjeto(projeto);
        return auxIncubacao.gravarIncubacao(mat);
    }*/

    public void mudarIncubacao(String mat)
    {
        BrokerIncubacao brok=new BrokerIncubacao();
        brok.mudarIncubacao(mat) ;
    }

    //faz conexão com o broker no método de consulta de 1 campo
    public Vector retornaIncubacao(String campo,String valor){
        BrokerIncubacao brok=new BrokerIncubacao();
        return brok.retornaConsulta(campo,valor) ;
    }

    //faz conexão com o broker no método de consulta de 2 campos
    public Vector retornaIncubacao(String campo,String valor,String campo2,String valor2){
        BrokerIncubacao brok=new BrokerIncubacao();
        return brok.retornaConsulta(campo,valor,campo2,valor2) ;
    }

    // faz conexão com o broker no método de consulta de 2 campos
    public Vector retornaIncubacao(){
        BrokerIncubacao brok=new BrokerIncubacao();
        return brok.retornaConsulta() ;
    }

    public Vector setarBandeja(String bandeja){
        BrokerIncubacao brok=new BrokerIncubacao();
        return brok.setarBandeja(bandeja) ;
    }

    public Incubacao materializar(String codigo)
    {
        BrokerIncubacao brok=new BrokerIncubacao();
        return brok.materializar(codigo);
    }

    public String gravarIncubacao (ArrayList posicoes)throws SQLException
    {
        BrokerIncubacao auxBrokerIncubacao= new BrokerIncubacao();
        return auxBrokerIncubacao.desmaterializar(posicoes);
    }
}

```

```

    public String getBandeja() {
        return bandeja;
    }
    public void setBandeja(String bandeja) {
        this.bandeja = bandeja;
    }
    public String getDatini() {
        return datini;
    }
    public void setDatini(String datini) {
        this.datini = datini;
    }
    public String getEntidade() {
        return entidade;
    }
    public void setEntidade(String entidade) {
        this.entidade = entidade;
    }
    public String getProjeto() {
        return projeto;
    }
    public void setProjeto(String projeto) {
        this.projeto = projeto;
    }
}
}

```

12.2.3.6 ControladoraIncubadoraBandeja

```

package controladora;

import broker.*;
import classepersistente.IncubadoraBandeja;
import java.sql.*;

/*@autor Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano Padilha*/
public class ControladoraIncubadoraBandeja
{
    public String materializar(IncubadoraBandeja ib) throws SQLException{
        BrokerIncubadoraBandeja brok=new BrokerIncubadoraBandeja();
        return brok.desmaterializar(ib);
    }

    public String materializarN(IncubadoraBandeja ib) throws SQLException{
        BrokerIncubadoraBandeja brok=new BrokerIncubadoraBandeja();
        return brok.desmaterializarN(ib);
    }
    public int desalocarN(String banN) throws SQLException{
        BrokerIncubadoraBandeja brok=new BrokerIncubadoraBandeja();
        return brok.desalocarN(banN);
    }
    public int desalocarI(String banI) throws SQLException{
        BrokerIncubadoraBandeja brok=new BrokerIncubadoraBandeja();
        return brok.desalocarI(banI);
    }
}
}

```

12.2.3.7 ControladoraLogin

```

package controladora;

```

```

import classepersistente.Usuario;
import broker.BrokerUsuario;

/**
 *
 * @author Arthur Maçaneiro Magyar
 * @author Fábio Glaucos dos Santos
 * @author Luciano Padilha
 * @version 1.0
 */
public class ControladoraLogin
{
    BrokerUsuario auxBrokerUsuario = new BrokerUsuario();

    /**
     * Faz conexão com o broker validando matrícula e senha.
     * @param lo matrícula do usuário
     * @param se senha do usuário
     * @return dados do usuário e login validado
     */
    public Usuario validaLogin(String lo,String se)
    {
        return auxBrokerUsuario.validaLogin(lo,se);
    }
}

```

12.2.3.8 ControladoraNascimento

```

package controladora;

import broker.*;
import java.sql.*;
import java.util.*;

/*@autor Arthur Maçaneiro Magyar, Fábio Glaucos dos Santos, Luciano Padilha*/
public class ControladoraNascimento
{

    public String transferir (ArrayList posicao,ArrayList nascimento,int qtde,String
incubacao,String entidade)throws SQLException
    {
        BrokerNascimento auxBrokerNascimento= new BrokerNascimento();
        return
auxBrokerNascimento.desmaterializar(posicao,nascimento,qtde,incubacao,entidade);
    }

    public String transferir(ArrayList posicoes,int qtde,String incubacao,String
entidade)throws SQLException
    {
        BrokerNascimento auxBrokerNascimento= new BrokerNascimento();
        return auxBrokerNascimento.desmaterializar(posicoes,qtde,incubacao,entidade);
    }

    public Vector setarBandeja(String bandeja){
        BrokerNascimento brok=new BrokerNascimento();
        return brok.setarBandeja(bandeja) ;
    }

}

```

12.2.3.9 ControladoraProjeto

```

package controladora;

import classepersistente.Projeto;
import broker.BrokerProjeto;
import java.sql.*;
import java.util.Vector;

/**
 *
 * @author Arthur Maçaneiro Magyar
 * @author Fábio Glauco dos Santos
 * @author Luciano Padilha
 * @version 1.0
 */
public class ControladoraProjeto
{
    // declaração de variáveis
    private String nome;
    private String siglaProj;
    private String entidade;
    private String especie;
    private String dataInicio;
    private String dataPrev;
    private String dataFim;
    private String descricao;
    private String status;

    /**
     *
     * @return
     */
    public String gravarProjeto ()throws SQLException
    {
        Projeto auxProjeto = new Projeto();
        auxProjeto.setSiglaProj(siglaProj);
        auxProjeto.setNome(nome);
        auxProjeto.setEntidade(entidade);
        auxProjeto.setEspecie(especie);
        auxProjeto.setDataInicio(dataInicio);
        auxProjeto.setDataPrev(dataPrev);
        auxProjeto.setDataFim(dataFim);
        auxProjeto.setDescricao(descricao);
        return auxProjeto.gravarProjeto();
    }

    /**
     *
     * @param mat
     * @return
     */
    public String gravarProjeto (String sigla)throws SQLException
    {
        Projeto auxProjeto = new Projeto();
        auxProjeto.setSiglaProj(siglaProj);
        auxProjeto.setNome(nome);
        auxProjeto.setEntidade(entidade);
        auxProjeto.setEspecie(especie);
        auxProjeto.setDataInicio(dataInicio);
        auxProjeto.setDataPrev(dataPrev);
        auxProjeto.setDataFim(dataFim);
        auxProjeto.setDescricao(descricao);
        auxProjeto.setStatus(status);
        return auxProjeto.gravarProjeto(sigla);
    }

    /**
     *
     * @param mat
     * @return
     */
    public String mudarProjeto(String sigla)
    {
        BrokerProjeto brok = new BrokerProjeto();
        return brok.mudarProjeto(sigla);
    }
}

```

```

/**
 * Faz conexão com o broker no método de consulta enviando data ou nenhum
(s,n,d,a,x,y).
 * @param t
 * @param valor
 * @return
 */
public Vector retornaProjeto(String t,String valor,String status)
{
    BrokerProjeto brok = new BrokerProjeto();
    return brok.retornaConsulta(t,valor,status) ;
}

/**
 *
 * @param codigo
 * @return
 */
public Projeto materializar(String codigo)
{
    BrokerProjeto brok = new BrokerProjeto();
    return brok.materializar(codigo);
}

// beans - gets e sets
public String getSiglaProj() {
    return siglaProj;
}
public void setSiglaProj(String siglaProj) {
    this.siglaProj = siglaProj;
}
public String getNome() {
    return nome;
}
public void setNome(String nome) {
    this.nome = nome;
}
public String getEntidade() {
    return entidade;
}
public void setEntidade(String entidade) {
    this.entidade = entidade;
}
public String getEspecie() {
    return especie;
}
public void setEspecie(String especie) {
    this.especie = especie;
}
public String getDataInicio() {
    return dataInicio;
}
public void setDataInicio(String dataInicio) {
    this.dataInicio = dataInicio;
}
public String getDataPrev() {
    return dataPrev;
}
public void setDataPrev(String dataPrev) {
    this.dataPrev = dataPrev;
}
public String getDataFim() {
    return dataFim;
}
public void setDataFim(String dataFim) {
    this.dataFim = dataFim;
}
public String getDescricao() {
    return descricao;
}
public void setDescricao(String descricao) {
    this.descricao = descricao;
}
public String getStatus() {
    return status;
}
}

```

```

        public void setStatus(String status) {
            this.status = status;
        }
    }
}

```

12.2.3.10 ControladoraSaque

```

package controladora;

import broker.*;
import java.sql.*;
import java.util.*;

/*@autor Arthur Maçaneiro Magyar, Fábio Glaucos dos Santos, Luciano Padilha*/
public class ControladoraSaque
{
    public String saque(ArrayList posicoes) throws SQLException
    {
        BrokerSaque brok = new BrokerSaque();
        return brok.saque(posicoes) ;
    }
}

```

12.2.3.11 ControladoraUsuario

```

package controladora;

import classepersistente.Usuario;
import broker.BrokerUsuario;
import java.sql.*;
import java.util.Vector;

/**
 *
 * @author Arthur Maçaneiro Magyar
 * @author Fábio Glaucos dos Santos
 * @author Luciano Padilha
 * @version 1.0
 */
public class ControladoraUsuario
{
    // declaração de variáveis
    private String matricula;
    private String nome;
    private String grupo;
    private String senha;
    private String telres;
    private String telcom;
    private String telcel;
    private String email;
    private String status;

    /**
     *
     * @return
     */
    public String gravarUsuario () throws SQLException
    {
        Usuario auxUsuario = new Usuario();
        auxUsuario.setMatricula(matricula);
        auxUsuario.setNome(nome);
        auxUsuario.setGrupo(grupo);
        auxUsuario.setSenha(senha);
        auxUsuario.setTelres(telres);
        auxUsuario.setTelcom(telcom);
        auxUsuario.setTelcel(telcel);
        auxUsuario.setEmail(email);
        return auxUsuario.gravarUsuario();
    }
}

```

```

}

/**
 *
 * @param mat
 * @return
 */
public String gravarUsuario (String mat) throws SQLException
{
    Usuario auxUsuario = new Usuario();
    auxUsuario.setMatricula(matricula);
    auxUsuario.setNome(nome);
    auxUsuario.setGrupo(grupo);
    auxUsuario.setSenha(senha);
    auxUsuario.setTelres(telres);
    auxUsuario.setTelcom(telcom);
    auxUsuario.setTelcel(telcel);
    auxUsuario.setEmail(email);
    auxUsuario.setStatus(status);
    return auxUsuario.gravarUsuario(mat);
}

/**
 *
 * @param mat
 * @return
 */
public String mudarUsuario(String mat)
{
    BrokerUsuario brok = new BrokerUsuario();
    return brok.mudarUsuario(mat);
}

/**
 * Faz conexão com o broker no método de consulta de 1 campo.
 * @param campo
 * @param valor
 * @return
 */
public Vector retornaUsuario(String campo,String valor,String status){
    BrokerUsuario brok = new BrokerUsuario();
    return brok.retornaConsulta(campo,valor,status);
}

/**
 * Faz conexão com o broker no método de consulta de 2 campos.
 * @param campo
 * @param valor
 * @param campo2
 * @param valor2
 * @return
 */
public Vector retornaUsuario(String campo,String valor,String campo2,String
valor2,String status){
    BrokerUsuario brok = new BrokerUsuario();
    return brok.retornaConsulta(campo,valor,campo2,valor2,status) ;
}

/**
 * Faz conexão com o broker no método de consulta de 2 campos.
 * @return
 */
public Vector retornaUsuario(String status){
    BrokerUsuario brok = new BrokerUsuario();
    return brok.retornaConsulta(status);
}

/**
 *
 * @param codigo
 * @return
 */
public Usuario materializar(String codigo)
{
    BrokerUsuario brok = new BrokerUsuario();
    return brok.materializar(codigo);
}

```

```

    }

    // beans - gets e sets
    public String getMatricula() {
        return matricula;
    }
    public void setMatricula(String matricula) {
        this.matricula = matricula;
    }
    public String getNome() {
        return nome;
    }
    public void setNome(String nome) {
        this.nome = nome;
    }
    public String getGrupo() {
        return grupo;
    }
    public void setGrupo(String grupo) {
        this.grupo = grupo;
    }
    public String getSenha() {
        return senha;
    }
    public void setSenha(String senha) {
        this.senha = senha;
    }
    public String getTelres() {
        return telres;
    }
    public void setTelres(String telres) {
        this.telres = telres;
    }
    public String getTelcom() {
        return telcom;
    }
    public void setTelcom(String telcom) {
        this.telcom = telcom;
    }
    public String getTelcel() {
        return telcel;
    }
    public void setTelcel(String telcel) {
        this.telcel = telcel;
    }
    public String getEmail() {
        return email;
    }
    public void setEmail(String email) {
        this.email = email;
    }
    public String getStatus() {
        return status;
    }
    public void setStatus(String status) {
        this.status = status;
    }
}

```

12.2.4 javaUtil

12.2.4.1 TransferData

```

package javaUtil;

/**
 * Formata datas em transferências de dados.

```

```

* @author      Arthur Maçaneiro Magyar
* @author      Fábio Glaucos dos Santos
* @author      Luciano Padilha
* @version     1.0
*/
public class TransferData {
    String newData,aux1,aux2;
    int index1,index2;

    /**
     * Formata a data para ser aceito banco de dados.
     * @param data recebida do formulário
     * @return data formatada para gravar no banco de dados
     */
    public String DataPGravar(String data){
        if (data != ""){
            index1=data.indexOf("/");
            index2=data.lastIndexOf("/");
            String parte1=data.substring(0,index1);
            String parte2=data.substring(index1+1,index2);
            String parte3=data.substring(index2+1);
            String newData=parte3+"-"+parte2+"-"+parte1;
            return newData;
        }
        else {
            return "9999-01-01";
        }
    }

    /**
     * Formata a data do banco de dados para apresentação.
     * @param data recebida do banco de dados
     * @return data formatada para apresentar na página
     */
    public String DataPApresentar(String data){
        index1=data.indexOf(" ");
        data=data.substring(0,index1);
        index1=data.indexOf("-");
        index2=data.lastIndexOf("-");
        String parte1=data.substring(0,index1);
        String parte2=data.substring(index1+1,index2);
        String parte3=data.substring(index2+1);
        String newData=parte3+"/"+parte2+"/"+parte1;
        return newData;
    }
}

```

12.2.5 servelet

12.2.5.1 Embridiagnostico

```

package servelet;

import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpSession;
import javax.servlet.ServletException;
import java.sql.*;
import controladora.*;
import classepersistente.*;
import java.util.*;

/**
 * Servlet Class
 *
 * @web.servlet          name="SISINC"
 *                      display-name="Sistema de Incubação Experimental"
 *                      description="Embridiagnostico"
 */

```

```

* @web.servlet-mapping      url-pattern="/Embriodiagnostico"
* @web.servlet-init-param   name=""
*                             value=""
*/
/*@autor Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano Padilha*/

public class Embriodiagnostico extends HttpServlet {
    // declaração do identificador universal da classe
    private static final long serialVersionUID = 18L;

    public void doGet(javax.servlet.http.HttpServletRequest request,
        javax.servlet.http.HttpServletResponse response) throws javax.servlet.ServletException,
        java.io.IOException
    {
        doPost(request, response);
    }

    public void doPost(javax.servlet.http.HttpServletRequest request,
        javax.servlet.http.HttpServletResponse response)
        throws javax.servlet.ServletException, java.io.IOException
    {
        try
        {
            // valida a sessão por grupo de usuário
            int perm = 2; // nível de acesso
            HttpSession s = request.getSession(false);
            if(s==null){response.sendRedirect("Login");}
            else{classepersistente.Usuario auxUsuario=(classepersistente.Usuario)
s.getAttribute("login");
            if((auxUsuario==null)|| (auxUsuario.getMatricula()==null))
            {response.sendRedirect("Login");}
            else{if(!auxUsuario.getStatus().equals("1"))

            {request.getRequestDispatcher("scripts/status.jsp").forward(request, response);
            System.out.println("Acesso Não Autorizado. Status não está Ativo.");}
            else{if(Integer.parseInt(auxUsuario.getGrupo())>perm)
            {request.setAttribute("retorno", Integer.toString(perm));

            request.getRequestDispatcher("scripts/perm.jsp").forward(request, response);
            System.out.println("Acesso Não Autorizado. Necessário ao menos Nível
"+perm+".");}

            else{

            ControladoraSaque auxContSaq= new ControladoraSaque();
            //ControladoraConsulta auxCons= new ControladoraConsulta();

            String acao = request.getParameter("acao");
            System.out.println("Ação ->" + acao);

            if (acao != null)
            {
                String tam=null;

                //pega as variaveis da pagina//
                String projeto=request.getParameter("CodProj");
                String bandeja=request.getParameter("CodBan");
                String entidade=request.getParameter("CodEnt");
                String dataInicio=request.getParameter("DataIniInc");
                String dataPrev=request.getParameter("DataPrevInc");
                String nascedouro=request.getParameter("CodEnc");
                String codIncubacao=request.getParameter("CodInc");
                String codPos=request.getParameter("CodEncBan");
                //String data=request.getParameter("Data");
                String qFinal=request.getParameter("qEntidade");
                String qInicial=request.getParameter("HQtdEntidade");
                //String qTotal=request.getParameter("QtdTotal");
                String codNascimento=request.getParameter("CodNas");
                //imprime valores q vem da pagina//
                System.out.println("***esta em nascimento os dados da tela são em
cima****");

                System.out.println("projeto:"+projeto);

```

```

System.out.println("bandeja:"+bandeja);
System.out.println("entidade:"+entidade);
System.out.println("data inicio:"+dataInicio);
System.out.println("nascido:"+"nascido");
System.out.println("incubacao:"+codIncubacao);
System.out.println("*codPos:"+codPos);
System.out.println("HQtdeEntidadeInicio:"+qInicial);
System.out.println("*QtdeEntidade:"+qFinal);
System.out.println("nascimento"+codNascimento);

//sacando//

if(acao.equals("Confirmar")){
    //classepersistente.Nascimento auxNascimento=new
classepersistente.Nascimento();

    int x,y,lin,col,index;
    //int qtde;
    String pos;
    ArrayList posicoes=new ArrayList();

    String cordenada=request.getParameter("cordenada");
    index=cordenada.indexOf("-");
    lin=Integer.parseInt(cordenada.substring(0,index));
    col=Integer.parseInt(cordenada.substring(index+1));
    String nasceu=request.getParameter("nasceu");

    //se o codigo do nascimento na pagina for branco
eu
    //gravo com um método que grava nascimento

    for(x=0;x<lin;x++){
        for(y=0;y<col;y++){

            pos=x+"0"+y;
            String

caixa=request.getParameter(pos);
            System.out.println("caixa é:
"+caixa);
            if(caixa==null){
                int

z=Integer.parseInt(request.getParameter("CodEncBan"));
                Posicao posicao=new
Posicao();

                posicao.setPosicaoX(x);

                posicao.setPosicaoY(y);

                posicao.setPosicaoZ(z);

                posicao.setNasceu(nasceu);

                posicoes.add(posicao);

            }

        }

    }

//deve retornar nascimento

try{

    tam=lin+"-"+col;
    //int index1,index2;
    auxContSaq.saque(posicoes);

}
catch(SQLException e){
    System.out.println(e);
}

}

```

```

//setar as variaveis para devolver a página
request.setAttribute("CodProj",projeto);
request.setAttribute("CodBan",bandeja);
request.setAttribute("CodEnt",entidade);
request.setAttribute("DataPrevInc",dataPrev);
request.setAttribute("CodEnc",nascudouro);
request.setAttribute("codInc",codIncubacao);
request.setAttribute("tam",tam);
request.setAttribute("codNascimento",codNascimento);
request.setAttribute("CodEncBan",codPos);
//request.setAttribute("DataPrevInc",data);

System.out.println("***esta em nascimento os dados da tela são-- em
baixo****");

System.out.println("projeto:"+projeto);
System.out.println("bandeja:"+bandeja);
System.out.println("entidade:"+entidade);
System.out.println("data inicio:"+dataInicio);
System.out.println("data prev:"+dataPrev);
System.out.println("nascudouro:"+nascudouro);
System.out.println("incubacao:"+codIncubacao);
System.out.println("*codPos:"+codPos);
System.out.println("*codnasc:"+codNascimento);

request.getRequestDispatcher("jsp/saque/emcad.jsp").forward(request,response);

    }else{

request.getRequestDispatcher("jsp/saque/emcad.jsp").forward(request,response);
    }

    }}}}
    catch(ServletException e)
    {
        System.out.println("Saiu da Classe ");
        System.out.println (e.getMessage());
        e.printStackTrace();
    }

}

}

```

12.2.5.2 Entidade

```

package servelet;

import controladora.ControladoraEntidade;
import controladora.ControladoraConsulta;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpSession;
import javax.servlet.ServletException;
import java.util.Vector;

```

```

/**
 * Servlet Class
 *
 * @web.servlet          name="SISINC"
 *                      display-name="Sistema de Incubação Experimental"
 *                      description="Cadastrar Entidade"
 * @web.servlet-mapping url-pattern="/Entidade"
 * @web.servlet-init-param name=""
 *                          value=""
 */

/**
 * Efetua cadastro de nova entidade e retorna mensagem.
 * <p>
 * Recebe as variáveis da página do respectivo formulário, seta as
 * variáveis da controladora direcionando para uma página de falha
 * ou sucesso dependendo do resultado de retorno.
 * @author      Arthur Maçaneiro Magyar
 * @author      Fábio Glaucos dos Santos
 * @author      Luciano Padilha
 * @version     1.0
 */
public class Entidade extends HttpServlet
{
    // declaração do identificador universal da classe
    private static final long serialVersionUID = 12L;

    public void doGet(javax.servlet.http.HttpServletRequest request,
    javax.servlet.http.HttpServletResponse response) throws javax.servlet.ServletException,
    java.io.IOException
    {
        doPost(request, response);
    }

    public void doPost(javax.servlet.http.HttpServletRequest request,
    javax.servlet.http.HttpServletResponse response)
    throws javax.servlet.ServletException, java.io.IOException
    {
        try
        {
            // valida a sessão por grupo de usuário
            int perm = 2; // nível de acesso
            HttpSession s = request.getSession(false);
            if(s==null){response.sendRedirect("Login");}
            else{classepersistente.Usuario auxUsuario=(classepersistente.Usuario)
s.getAttribute("login");
            if((auxUsuario==null)|| (auxUsuario.getMatricula()==null))
            {response.sendRedirect("Login");}
            else{if(!auxUsuario.getStatus().equals("1"))

            {request.getRequestDispatcher("scripts/status.jsp").forward(request, response);
            System.out.println("Acesso Não Autorizado. Status não está Ativo.");}
            else{if(Integer.parseInt(auxUsuario.getGrupo())>perm)
            {request.setAttribute("retorno", Integer.toString(perm));

            request.getRequestDispatcher("scripts/perm.jsp").forward(request, response);
            System.out.println("Acesso Não Autorizado. Necessário ao menos Nível
"+perm+".");}
            else{

            ControladoraEntidade auxControladoraEntidade = new
ControladoraEntidade();
            String acao=request.getParameter("Acao");
            String nomeEnt=request.getParameter("NomeEnt");
            System.out.println("Ação -> "+acao);
            System.out.println("Nome -> "+nomeEnt);
            String retorno="";

            if (acao != null)
            {
                // efetua pesquisa por entidades pai
                if(acao.equals("Pesquisar"))
                {
                    Vector ents;
                    String filho=request.getParameter("filho");

```

```

ControladoraConsulta cont = new ControladoraConsulta();
ents=cont.retornaEntidade(filho);
request.setAttribute("ents",ents);
request.setAttribute("filho",filho);
request.setAttribute("nomeEnt",nomeEnt);

request.getRequestDispatcher("jsp/cadastro/entcad.jsp").forward(request,response);
}
else
{
    // carrega as variáveis da página
    String tipoEntidade=request.getParameter("filho");
    String entidade=request.getParameter("pai");
    String descEnt=request.getParameter("DescEnt");

    // atribui as variáveis na controladora
    auxControladoraEntidade.setTipoEntidade(tipoEntidade);
    auxControladoraEntidade.setEntidade(entidade);
    auxControladoraEntidade.setNome(nomeEnt);
    auxControladoraEntidade.setDescEnt(descEnt);

    // envia para a controladora o método gravar com parâmetro do
servelet
    try
    {
        retorno=auxControladoraEntidade.gravarEntidade();
        System.out.println("Enviou -> "+nomeEnt);
        // retorna falha ou sucesso ao gravar no banco de dados
        if(retorno == "Entidade")
        {
            System.out.println("Gravou -> "+nomeEnt);
            request.setAttribute("retorno",retorno);

request.getRequestDispatcher("scripts/sucesso.jsp").forward(request,response);
        }
        else
        {
            System.out.println("Não gravou -> "+retorno);
            request.setAttribute("retorno",retorno);

request.getRequestDispatcher("scripts/falha.jsp").forward(request,response);
        }
    }
    catch(Exception e)
    {
        System.out.println("Não enviou -> "+e.getMessage());
    }
}
else
{
request.getRequestDispatcher("jsp/cadastro/entcad.jsp").forward(request,response);
}

}}}}
catch(ServletException e)
{
    System.out.println(e.getMessage());
    e.printStackTrace();
    System.out.println("Saiu da Classe.");
}
}
}
}
}
}

```

12.2.5.3 EntidadeAltera

```

package servelet;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpSession;

```

```

import classepersistente.Entidade;
import java.util.*;
import controladora.ControladoraEntidade;

/**
 * Servlet Class
 *
 * @web.servlet          name="SISINC"
 *                      display-name="Sistema de Incubação Experimental"
 *                      description="Alterar Entidade"
 * @web.servlet-mapping url-pattern="/EntidadeAlterar"
 * @web.servlet-init-param name=""
 *                          value=""
 */
/*@autor Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano Padilha*/

public class EntidadeAlterar extends HttpServlet {
    // declaração do identificador universal da classe
    private static final long serialVersionUID = 42L;

    public void doGet(javax.servlet.http.HttpServletRequest request,
        javax.servlet.http.HttpServletResponse response) throws javax.servlet.ServletException,
        java.io.IOException
    {
        doPost(request, response);
    }

    /* (non-Javadoc)
     * @see javax.servlet.http.HttpServlet#doPost(javax.servlet.http.HttpServletRequest,
     javax.servlet.http.HttpServletResponse)
     */
    public void doPost(javax.servlet.http.HttpServletRequest request,
        javax.servlet.http.HttpServletResponse response)
        throws javax.servlet.ServletException, java.io.IOException
    {
        Vector retorno=new Vector();
        try
        {
            // valida a sessão por grupo de usuário
            int perm = 2; // nível de acesso
            HttpSession s = request.getSession(false);
            if(s==null){response.sendRedirect("Login");}
            else{classepersistente.Usuario auxUsuario=(classepersistente.Usuario)
s.getAttribute("login");
            if((auxUsuario==null)|| (auxUsuario.getMatricula()==null))
            {response.sendRedirect("Login");}
            else{if(!auxUsuario.getStatus().equals("1"))

            {request.getRequestDispatcher("scripts/status.jsp").forward(request, response);
            System.out.println("Acesso Não Autorizado. Status não está Ativo.");}
            else{if(Integer.parseInt(auxUsuario.getGrupo())>perm)
            {request.setAttribute("retorno", Integer.toString(perm));

            request.getRequestDispatcher("scripts/perm.jsp").forward(request, response);
            System.out.println("Acesso Não Autorizado. Necessário ao menos Nível
"+perm+".");}
            else{

            ControladoraEntidade auxControladoraEntidade = new
ControladoraEntidade();
            String acao = request.getParameter("acao");

            System.out.println("Ação ->" + acao);

            if (acao != null){
                if(acao.equals("Consultar")){
                    String nome =request.getParameter("nomeEnt");
                    String tipo = request.getParameter("tipo");
                    String t;

                    //"t"para o tipo da transação

```

```

        if ("".compareTo(nome)!=0) {
            System.out.println("nome:"+nome);
            if ("".compareTo(tipo)!=0) { //pesquisa port nome e tipo
                t="d";
                String valor=nome+" - "+tipo;
                System.out.println("valor:"+valor);

                retorno=auxControladoraEntidade.retornaEntidade(t, valor);

                request.setAttribute("tabela", retorno);

                request.getRequestDispatcher("jsp/entidadeconsulta.jsp").forward(request, response);
            }
            else { //pesquisa por nome
                t="n";

                retorno=auxControladoraEntidade.retornaEntidade(t, nome);

                request.setAttribute("tabela", retorno);

                request.getRequestDispatcher("jsp/entidadeconsulta.jsp").forward(request, response);
            }
        }
        else {
            }
            else {
                if ("".compareTo(tipo)!=0) { //pesquisa por tipo
                    t="t";

                    retorno=auxControladoraEntidade.retornaEntidade(t, tipo);

                    request.setAttribute("tabela", retorno);

                    request.getRequestDispatcher("jsp/entidadeconsulta.jsp").forward(request, response);
                }
                else { //retorna tabela com todos os entidades
                    t="a";
                    nome="";

                    retorno=auxControladoraEntidade.retornaEntidade(t, nome);
                    request.setAttribute("tabela", retorno);

                    request.getRequestDispatcher("jsp/entidadeconsulta.jsp").forward(request, response);
                }
            }
        }
    }
}

} else { //fecha o consultar
    if (acao.equals("carregar")) {
        String codigo =
        request.getParameter("codigo");
        Entidade auxEntidade =
        auxControladoraEntidade.materializar(codigo);
        request.setAttribute("dados", auxEntidade);

        request.getRequestDispatcher("jsp/entidadecarregado.jsp").forward(request, response);
    }
    else {
        if (acao.equals("alterar")) {
        }
        else {
            if (acao.equals("excluir")) {
                String sigla=
                request.getParameter("siglaProj");

                try
                {

```

```

        auxControladoraEntidade.mudarEntidade(sigla); //passa para a controladora o método de
mudar estado de usuário.

    }
    catch(Exception e){

        System.out.println("nao esta excluindo!!");

    }
    //uma tela de aviso
de exclusão
    }
}

    }

        //} //fecha o não consultar
//} //fecha o acao q nao é nulo

    }else
    {

        request.setAttribute("tabela", retorno);

        request.getRequestDispatcher("jsp/entidadealterar.jsp").forward(request, response);
    }

    }}}}
    catch(ServletException e)
    {
        System.out.println (e.getMessage());
        e.printStackTrace();
    }

}

}
}

```

12.2.5.4 EntidadeOvo

```

package servelet;

import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpSession;
import javax.servlet.ServletException;
import java.sql.*;
import controladora.ControladoraEntidadeOvo;

/**
 * Servlet Class
 *
 * @web.servlet          name="SISINC"
 *                      display-name="Sistema de Incubação Experimental"
 *                      description="Cadastrar Usuário"
 * @web.servlet-mapping url-pattern="/EntidadeOvo"
 * @web.servlet-init-param name=""
 *                          value=""
 */
/*@autor Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano Padilha*/

public class EntidadeOvo extends HttpServlet {
    // declaração do identificador universal da classe
    private static final long serialVersionUID = 19L;

```

```

        public void doGet(javax.servlet.http.HttpServletRequest request,
javax.servlet.http.HttpServletResponse response) throws javax.servlet.ServletException,
java.io.IOException
        {
            doPost(request, response);
        }

        public void doPost(javax.servlet.http.HttpServletRequest request,
javax.servlet.http.HttpServletResponse response)
        throws javax.servlet.ServletException, java.io.IOException
        {
            try
            {
                // valida a sessão por grupo de usuário
                int perm = 2; // nível de acesso
                HttpSession s = request.getSession(false);
                if (s==null) {response.sendRedirect("Login");}
                else {classepersistente.Usuario auxUsuario=(classepersistente.Usuario)
s.getAttribute("login");
                if ((auxUsuario==null) || (auxUsuario.getMatricula()==null))
                {response.sendRedirect("Login");}
                else {if (!auxUsuario.getStatus().equals("1"))

                {request.getRequestDispatcher("scripts/status.jsp").forward(request, response);
                System.out.println("Acesso Não Autorizado. Status não está Ativo.");}
                else {if (Integer.parseInt(auxUsuario.getGrupo()) > perm)
                {request.setAttribute("retorno", Integer.toString(perm));

                request.getRequestDispatcher("scripts/perm.jsp").forward(request, response);
                System.out.println("Acesso Não Autorizado. Necessário ao menos Nível
"+perm+".");}
                else {

                ControladoraEntidadeOvo auxControladoraEntidadeOvo = new
ControladoraEntidadeOvo();
                String acao = request.getParameter("Acao");
                String codbarras = request.getParameter("CodBarras");

                System.out.println("Ação ->" + acao);
                System.out.println("Código ->" + codbarras);

                if (acao != null)
                {

                //pegando as variaveis da pagina
                String codigobarras=request.getParameter("CodBarras");
                String celula=request.getParameter("CodCel");
                String cor=request.getParameter("CodCorOvo");
                String formato=request.getParameter("CodForOvo");
                String desinfeccao=request.getParameter("CodDesOvo");
                String aquisicao=request.getParameter("AquisOvo");
                String datacoleta=request.getParameter("DataColOvo");
                String aspecto=request.getParameter("AspOvo");
                String peso=request.getParameter("PesoOvo");
                String porosidade=request.getParameter("PorOvo");
                String incubavel=request.getParameter("Incubavel");
                String descricao=request.getParameter("DescEnt");

                //acerto de nulos
                if (peso.equals("")) {peso="NULL";}
                if (porosidade.equals("")) {porosidade="NULL";}

                //setando as variaveis na controladora
                auxControladoraEntidadeOvo.setCodigobarras(codigobarras);
                auxControladoraEntidadeOvo.setCelula(celula);
                auxControladoraEntidadeOvo.setCor(cor);
                auxControladoraEntidadeOvo.setFormato(formato);
                auxControladoraEntidadeOvo.setDesinfeccao(desinfeccao);
                auxControladoraEntidadeOvo.setAquisicao(aquisicao);
                auxControladoraEntidadeOvo.setDatacoleta(datacoleta);
                auxControladoraEntidadeOvo.setAspecto(aspecto);

```



```

        public void doGet(javax.servlet.http.HttpServletRequest request,
javax.servlet.http.HttpServletResponse response) throws javax.servlet.ServletException,
java.io.IOException
        {
            doPost(request, response);
        }

        public void doPost(javax.servlet.http.HttpServletRequest request,
javax.servlet.http.HttpServletResponse response)
        throws javax.servlet.ServletException, java.io.IOException
        {
            try
            {
                // valida a sessão por grupo de usuário
                int perm = 1; // nível de acesso
                HttpSession s = request.getSession(false);
                if (s==null) {response.sendRedirect("Login");}
                else {classepersistente.Usuario auxUsuario=(classepersistente.Usuario)
s.getAttribute("login");
                if ((auxUsuario==null) || (auxUsuario.getMatricula()==null))
                {response.sendRedirect("Login");}
                else {if (!auxUsuario.getStatus().equals("1"))

                {request.getRequestDispatcher("scripts/status.jsp").forward(request, response);
                System.out.println("Acesso Não Autorizado. Status não está Ativo.");}
                else {if (Integer.parseInt(auxUsuario.getGrupo())>perm)
                {request.setAttribute("retorno", Integer.toString(perm));

                request.getRequestDispatcher("scripts/perm.jsp").forward(request, response);
                System.out.println("Acesso Não Autorizado. Necessário ao menos Nível
"+perm+".");}
                else {

                ControladoraEspecieAve auxControladoraEspecieAve = new
ControladoraEspecieAve();
                String acao = request.getParameter("Acao");
                String especie = request.getParameter("NomeEsp");

                System.out.println("Ação ->" + acao);
                System.out.println("Nome ->" + especie);

                if (acao != null)
                {

                //pegando as variaveis da pagina
                String tempmininc=request.getParameter("TempMinInc");
                String tempmaxinc=request.getParameter("TempMaxInc");
                String umimininc=request.getParameter("UmiMinInc");
                String umimaxinc=request.getParameter("UmiMaxInc");
                String perinc=request.getParameter("PerInc");
                String peraviinc=request.getParameter("PerAviInc");
                String tempminnas=request.getParameter("TempMinNas");
                String tempmaxnas=request.getParameter("TempMaxNas");
                String umiminnas=request.getParameter("UmiMinNas");
                String umimaxnas=request.getParameter("UmiMaxNas");
                String pernas=request.getParameter("PerNas");
                String peravinas=request.getParameter("PerAviNas");
                String perrec=request.getParameter("PerRec");

                //acerto de nullos
                if (tempmininc.equals("")) {tempmininc="NULL";}
                if (tempmaxinc.equals("")) {tempmaxinc="NULL";}
                if (umimininc.equals("")) {umimininc="NULL";}
                if (umimaxinc.equals("")) {umimaxinc="NULL";}
                if (tempminnas.equals("")) {tempminnas="NULL";}
                if (tempmaxnas.equals("")) {tempmaxnas="NULL";}
                if (umiminnas.equals("")) {umiminnas="NULL";}
                if (umimaxnas.equals("")) {umimaxnas="NULL";}

                //setando as variaveis na controladora
                auxControladoraEspecieAve.setEspecie(especie);

```



```

*                                     value=""
*/
/*@autor Luciano Padilha*/

public class Incubacao extends HttpServlet {
    // declaração do identificador universal da classe
    private static final long serialVersionUID = 15L;

    public void doGet(javax.servlet.http.HttpServletRequest request,
        javax.servlet.http.HttpServletResponse response) throws javax.servlet.ServletException,
        java.io.IOException
    {
        doPost(request, response);
    }

    public void doPost(javax.servlet.http.HttpServletRequest request,
        javax.servlet.http.HttpServletResponse response)
        throws javax.servlet.ServletException, java.io.IOException
    {
        try
        {
            // valida a sessão por grupo de usuário
            int perm = 2; // nível de acesso
            HttpSession s = request.getSession(false);
            if (s==null) {response.sendRedirect("Login");}
            else {classepersistente.Usuario auxUsuario=(classepersistente.Usuario)
s.getAttribute("login");
                if ((auxUsuario==null) || (auxUsuario.getMatricula()==null))
                    {response.sendRedirect("Login");}
                else {if (!auxUsuario.getStatus().equals("1"))

                    {request.getRequestDispatcher("scripts/status.jsp").forward(request, response);
                    System.out.println("Acesso Não Autorizado. Status não está Ativo.");}
                    else {if (Integer.parseInt(auxUsuario.getGrupo())>perm)
                        {request.setAttribute("retorno", Integer.toString(perm));

                            request.getRequestDispatcher("scripts/perm.jsp").forward(request, response);
                            System.out.println("Acesso Não Autorizado. Necessário ao menos Nível
"+perm+".");}

                        else{

                            ControladoraIncubacao auxContIncuba= new ControladoraIncubacao();
                            ControladoraConsulta auxCons= new ControladoraConsulta();

                            String acao = request.getParameter("acao");
                            System.out.println("Ação ->" + acao);

                            if (acao != null)
                            {
                                String tam=null;

                                //pega as variaveis da pagina//
                                String projeto=request.getParameter("CodProj");
                                String entidadeMin=request.getParameter("CodTipoEnt");
                                String bandeja=request.getParameter("CodBan");
                                String entidade=request.getParameter("CodEnt");
                                String dataInicio=request.getParameter("DataIniInc");
                                String dataPrev=request.getParameter("DataPrevInc");
                                String incubadora=request.getParameter("CodEnc");
                                String codIncubacao=request.getParameter("CodInc");
                                String codPos=request.getParameter("CodEncBan");
                                String data=request.getParameter("Data");

                                System.out.println("***esta em incubação os dados da tela são em
cima****");

                                System.out.println("projeto:"+projeto);
                                System.out.println("entmin:"+entidadeMin);
                                System.out.println("bandeja:"+bandeja);
                                System.out.println("entidade:"+entidade);
                                System.out.println("data inicio:"+dataInicio);
                                System.out.println("incubadora:"+incubadora);

```

```

System.out.println("incubacao:"+codIncubacao);
System.out.println("*codPos:"+codPos);

//          recolocar valores da bandeja.

String incb=auxCons.retornaPosBan(bandeja);
int ind1=incb.indexOf("-");
int ind2=incb.indexOf(":");
incubadora=incb.substring(0,ind1-1);
codPos=incb.substring(ind1+2,ind2-1);

System.out.println("servelet-incubaçã--codinc:"+codIncubacao);

//buscando uma bandeja//
if (acao.equals("Buscar Bandeja")){
    ControladoraConsulta cont = new ControladoraConsulta();
    tam=cont.retornaTamBan(bandeja);

}

else{

//incubando ovos de uma entidade//
//incubando posições.
if(acao.equals("Incubar")){

    classepersistente.Incubacao auxIncubacao=new

    int x,y,lin,col,index;
    String pos;
    ArrayList incubacao=new ArrayList();
    ArrayList posicoes=new ArrayList();
    String cordenada=request.getParameter("cordenada");

    index=cordenada.indexOf("-");
    lin=Integer.parseInt(cordenada.substring(0,index));
    col=Integer.parseInt(cordenada.substring(index+1));
    if(codIncubacao.equals("")){

        //se o codigo da incubação na pagina for branco eu
        //gravo com uma método que grava incubação

        for(x=0;x<lin;x++){
            for(y=0;y<col;y++){

                pos=x+"-"+y;
                String

                if(caixa!=null){
                    int

                    Posicao posicao=new

                caixa=request.getParameter(pos);

                z=Integer.parseInt(request.getParameter("CodEncBan"));

                Posicao();

                posicao.setEntidade(entidade);

                posicao.setPosicaoX(x);

                posicao.setPosicaoY(y);

                posicao.setPosicaoZ(z);

                auxIncubacao.setProjeto(projeto);

                auxIncubacao.setDatini(dataInicio);

                auxIncubacao.setData(data);

```

```

                                                                    System.out.println("
no servelet: posbanenc"+codPos+" ent:"+entidade+" caixa:"+caixa+"
proj:"+projeto+"datIni"+dataInicio+"pos:"+pos+"z"+z);

    posicoes.add(posicao);

    incubacao.add(auxIncubacao);
                                                                    }
                                                                    }
                                                                    }

//deve retornar incubacao
                                                                    try{
                                                                    tam=lin+"-"+col;
                                                                    int index1;

                                                                    //passa para o método os objetos
"incubacao" e "posicoes" para gravar nas duas tabelas "incubacao" e "inctempos"

                                                                    String
inc=auxContIncuba.gravarIncubacao(incubacao,posicoes);
                                                                    System.out.println("inc:"+inc);
                                                                    index1=inc.indexOf("|");

                                                                    codIncubacao=inc.substring(0,index1);
                                                                    data=inc.substring(index+2);

                                                                    System.out.println("inc:"+inc+"codinc:"+codIncubacao+"data"+data);
                                                                    dataPrev=data;
                                                                    }
                                                                    catch(SQLException e){
                                                                    System.out.println(e);
                                                                    }

                                                                    }else{

//    incubacao nao nula

                                                                    for(x=0;x<lin;x++){
                                                                    for(y=0;y<col;y++){

                                                                    pos=x+"-"+y;
                                                                    String
caixa=request.getParameter(pos);
                                                                    if(caixa!=null){
                                                                    int
z=Integer.parseInt(request.getParameter("CodEncBan"));
                                                                    Posicao posicao=new
Posicao();

                                                                    posicao.setEntidade(entidade);
                                                                    posicao.setPosicaoX(x);
                                                                    posicao.setPosicaoY(y);
                                                                    posicao.setPosicaoZ(z);
                                                                    posicao.setIncubacao(codIncubacao);
                                                                    auxIncubacao.setProjeto(projeto);
                                                                    auxIncubacao.setDatini(dataInicio);
                                                                    auxIncubacao.setData(data);
                                                                    System.out.println("
ent:"+entidade+" caixa:"+caixa+" proj:"+projeto+"datIni"+dataInicio+"pos:"+pos);

                                                                    posicoes.add(posicao);

```

```

        }
    }
}

//deve retornar incubacao

try{

    tam=lin+"-"+col;

    String
    inc=auxContIncuba.gravarIncubacao(posicoes);

    System.out.println("inc:"+inc);
    int index2=inc.indexOf("|");
    codIncubacao=inc.substring(0,index2);
    data=inc.substring(index+2);

    System.out.println("inc:"+inc+"codinc:"+codIncubacao+"data"+data);
    dataPrev=data;
    }
    catch(SQLException e){
        System.out.println(e);
    }

}

}

Vector bandejaCheia=new Vector();
//setar as váriaveis para devolver a página
request.setAttribute("CodProj",projeto);
request.setAttribute("CodBan",bandeja);
request.setAttribute("CodEnt",entidade);
request.setAttribute("CodTipoEnt",entidadeMin);
request.setAttribute("DataPrevInc",dataPrev);
request.setAttribute("CodEnc",incubadora);
request.setAttribute("CodInc",codIncubacao);
request.setAttribute("incubacao",codIncubacao);
request.setAttribute("tam",tam);
request.setAttribute("CodEncBan",codPos);
//request.setAttribute("DataPrevInc",data);

bandejaCheia=auxContIncuba.setarBandeja(codPos);
request.setAttribute("bc",bandejaCheia);

System.out.println("***esta em incubação os dados da tela são-- em
baixo****");

System.out.println("projeto:"+projeto);
System.out.println("entmin:"+entidadeMin);
System.out.println("bandeja:"+bandeja);
System.out.println("entidade:"+entidade);
System.out.println("data inicio:"+dataInicio);
System.out.println("data prev:"+dataPrev);
System.out.println("incubadora:"+incubadora);
System.out.println("incubacao:"+codIncubacao);
System.out.println("*codPos:"+codPos);

request.getRequestDispatcher("jsp/incubacao/inccad.jsp").forward(request,response);

}

request.getRequestDispatcher("jsp/incubacao/inccad.jsp").forward(request,response);
}

```

```

        }}}}
        catch(ServletException e)
        {
            System.out.println("Saiu da Classe ");
            System.out.println (e.getMessage());
            e.printStackTrace();
        }
    }
}
}

```

12.2.5.7 IncubadoraBandeja

```

package servelet;

import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpSession;
import javax.servlet.ServletException;
import java.sql.*;
import controladora.*;

/**
 * Servlet Class
 *
 * @web.servlet          name="SISINC"
 *                      display-name="Sistema de Incubação Experimental"
 *                      description="Cadastrar Bandeja em Incubadora"
 * @web.servlet-mapping url-pattern="/IncubadoraBandeja"
 * @web.servlet-init-param name=""
 *                          value=""
 */
/*@autor Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano Padilha*/

public class IncubadoraBandeja extends HttpServlet {
    // declaração do identificador universal da classe
    private static final long serialVersionUID = 13L;

    public void doGet(javax.servlet.http.HttpServletRequest request,
        javax.servlet.http.HttpServletResponse response) throws javax.servlet.ServletException,
        java.io.IOException
    {
        doPost(request, response);
    }

    public void doPost(javax.servlet.http.HttpServletRequest request,
        javax.servlet.http.HttpServletResponse response)
        throws javax.servlet.ServletException, java.io.IOException
    {
        try
        {
            // valida a sessão por grupo de usuário
            int perm = 2; // nível de acesso
            HttpSession s = request.getSession(false);
            if (s==null) {response.sendRedirect("Login");}
            else {classepersistente.Usuario auxUsuario=(classepersistente.Usuario)
s.getAttribute("login");
            if ((auxUsuario==null) || (auxUsuario.getMatricula()==null))
            {response.sendRedirect("Login");}
            else {if (!auxUsuario.getStatus().equals("1"))
            {request.getRequestDispatcher("scripts/status.jsp").forward(request, response);
            System.out.println("Acesso Não Autorizado. Status não está Ativo.");}
            else {if (Integer.parseInt(auxUsuario.getGrupo())>perm)
            {request.setAttribute("retorno", Integer.toString(perm));}

```

```

        request.getRequestDispatcher("scripts/perm.jsp").forward(request, response);
        System.out.println("Acesso Não Autorizado. Necessário ao menos Nível
"+perm+".");}
        else{

                ControladoraIncubadoraBandeja auxControladoraIncubadoraBandeja = new
ControladoraIncubadoraBandeja();
                String acao = request.getParameter("AcaoA");
                System.out.println("Ação -> " + acao);

                if (acao != null)
                {

                        if(acao.equals("Alocar")){

                                //pegando as variaveis da pagina
                                //para gravar um registro
                                String incubadora=request.getParameter("incubadora");
                                String nascedouro=request.getParameter("nascedouro");
                                String posnas=request.getParameter("posnas");
                                String posinc=request.getParameter("posinc");
                                String bandeja=request.getParameter("Bandes");

                                classepersistente.IncubadoraBandeja incubadorabandeja=new
classepersistente.IncubadoraBandeja();
                                try{

                                        if(nascedouro!=null){
                                                incubadorabandeja.setNascedouro(nascedouro);
                                                incubadorabandeja.setPosnas(posnas);
                                                incubadorabandeja.setBandeja(bandeja);

                                auxControladoraIncubadoraBandeja.materializarN(incubadorabandeja);
                                        }else{
                                                if (incubadora!=null){

                                incubadorabandeja.setIncubadora(incubadora);
                                                incubadorabandeja.setPosinc(posinc);
                                                incubadorabandeja.setBandeja(bandeja);

                                auxControladoraIncubadoraBandeja.materializar(incubadorabandeja);
                                                }
                                        }

                                request.getRequestDispatcher("jsp/incubacao/incban.jsp").forward(request, response);

                                }catch(SQLException e){
                                        System.out.println(e);
                                }

                                }

                                else
                                {
                                        //retorna valores para a pagina

                                request.getRequestDispatcher("jsp/incubacao/incban.jsp").forward(request, response);
                                }

                                }}}}
                                catch(ServletException e)
                                {
                                        System.out.println("Saiu da Classe ");
                                        System.out.println(e.getMessage());
                                        e.printStackTrace();
                                }
                                }
}

```

12.2.5.8 IncubadoraBandejaDes

```
package servelet;

import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpSession;
import javax.servlet.ServletException;
import java.sql.*;
import controladora.*;

/**
 * Servlet Class
 *
 * @web.servlet          name="SISINC"
 *                      display-name="Sistema de Incubação Experimental"
 *                      description="Cadastrar Bandeja em Incubadora"
 * @web.servlet-mapping url-pattern="/IncubadoraBandeja"
 * @web.servlet-init-param name=""
 *                          value=""
 */
/*@autor Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano Padilha*/

public class IncubadoraBandejaDes extends HttpServlet {
    // declaração do identificador universal da classe
    private static final long serialVersionUID = 14L;

    public void doGet(javax.servlet.http.HttpServletRequest request,
        javax.servlet.http.HttpServletResponse response) throws javax.servlet.ServletException,
        java.io.IOException
    {
        doPost(request, response);
    }

    public void doPost(javax.servlet.http.HttpServletRequest request,
        javax.servlet.http.HttpServletResponse response)
        throws javax.servlet.ServletException, java.io.IOException
    {
        try
        {
            // valida a sessão por grupo de usuário
            int perm = 2; // nível de acesso
            HttpSession s = request.getSession(false);
            if (s==null){response.sendRedirect("Login");}
            else{classepersistente.Usuario auxUsuario=(classepersistente.Usuario)
s.getAttribute("login");
            if((auxUsuario==null)|| (auxUsuario.getMatricula()==null))
            {response.sendRedirect("Login");}
            else{if(!auxUsuario.getStatus().equals("1"))

            {request.getRequestDispatcher("scripts/status.jsp").forward(request, response);
            System.out.println("Acesso Não Autorizado. Status não está Ativo.");}
            else{if (Integer.parseInt(auxUsuario.getGrupo())>perm)
            {request.setAttribute("retorno", Integer.toString(perm));

            request.getRequestDispatcher("scripts/perm.jsp").forward(request, response);
            System.out.println("Acesso Não Autorizado. Necessário ao menos Nível
            "+perm+".");}

            else{

            }
            ControladoraIncubadoraBandeja auxControladoraIncubadoraBandeja = new
            ControladoraIncubadoraBandeja();
            String acao = request.getParameter("Acao");
            System.out.println("Ação ->" + acao);

            if (acao != null)
            {

            if(acao.equals("Desalocar")){

            //pegando as variaveis da pagina
            //para gravar um registro
            String bandejaI=request.getParameter("CodBan");
            String bandejaN=request.getParameter("Bannas");
            System.out.println("bandejas:"+bandejaN+bandejaI);
```

```

        try{

            if (bandejaN!=null){

auxControladoraIncubadoraBandeja.desalocarN(bandejaN);
            }else{
                if (bandejaI!=null){

                    System.out.println("bandejas:"+bandejaN+bandejaI);
auxControladoraIncubadoraBandeja.desalocarI(bandejaI);
                }
            }

            request.getRequestDispatcher("jsp/incubacao/incbandes.jsp").forward(request,response);

        }catch(SQLException e){
            System.out.println(e);
        }

        }
        else
        {
            //retorna valores para a pagina

            request.getRequestDispatcher("jsp/incubacao/incbandes.jsp").forward(request,response);
        }

        }}}
        catch(ServletException e)
        {
            System.out.println("Saiu da Classe ");
            System.out.println (e.getMessage());
            e.printStackTrace();
        }
    }
}

```

12.2.5.9 Login

```

package servelet;

import controladora.ControladoraLogin;
import classepersistente.Usuario;
import javax.servlet.http.*;
import javax.servlet.http.HttpServlet;
import javax.servlet.ServletException;

/**
 * Servlet Class
 *
 * @web.servlet          name="SISINC"
 *                      display-name="Sistema de Incubação Experimental"
 *                      description="Validar Login"
 * @web.servlet-mapping url-pattern="/Login"
 * @web.servlet-init-param name=""
 *                          value=""
 */

/**
 * Valida login e senha com cadastro de usuário.
 * <p>
 * Recebe as variáveis da página do respectivo formulário, seta as
 * variáveis da controladora direcionando para uma página de falha
 * ou sucesso dependendo do resultado de retorno.
 * @author      Arthur Maçaneiro Magyar
 * @author      Fábio Glauco dos Santos

```

```

* @author      Luciano Padilha
* @version     1.0
*/
public class Login extends HttpServlet
{
    // declaração do identificador universal da classe
    private static final long serialVersionUID = 01L;

    /**
     *
     */
    public void doGet(javax.servlet.http.HttpServletRequest request,
        javax.servlet.http.HttpServletResponse response) throws javax.servlet.ServletException,
        java.io.IOException
    {
        doPost(request, response);
    }

    /**
     *
     */
    public void doPost(javax.servlet.http.HttpServletRequest request,
        javax.servlet.http.HttpServletResponse response)
        throws javax.servlet.ServletException, java.io.IOException
    {
        try
        {
            ControladoraLogin auxControladoraLogin = new ControladoraLogin();
            String acao=request.getParameter("Acao");
            String login=request.getParameter("MatrUsu");
            String senha=request.getParameter("SenCrt");
            System.out.println("Ação -> "+acao);
            System.out.println("Login -> "+login);
            String retorno="";

            if( acao != null)
            {
                // se for preenchido: MatrUsu
                if("".compareTo(login)!=0)
                {
                    // se for preenchido: SenCrt
                    if("".compareTo(senha)!=0)
                    {
                        Usuario
                        auxUsuario=auxControladoraLogin.validaLogin(login,senha);
                        System.out.println("Permissão ->
                        "+auxUsuario.getGrupo());

                        // envia valores para: HttpSession
                        if("".compareTo(auxUsuario.getMatricula())!= 0 &&
                        auxUsuario.getGrupo() != null)
                        {
                            System.out.println("Acesso Autorizado.");
                            HttpSession s = request.getSession(true);
                            s.setAttribute("login",auxUsuario);
                            response.sendRedirect("Principal");
                        }
                        // retorna falha de login
                        else
                        {
                            System.out.println("Não Autorizado.");
                            retorno="Login ou Senha Inválido.";
                            request.setAttribute("login",login);
                            request.setAttribute("retorno", retorno);
                        }

                        request.getRequestDispatcher("jsp/login/index.jsp").forward(request, response);
                    }
                }
                // retorna sem senha
                else
                {
                    retorno="O campo Senha deve ser preenchido!";
                    request.setAttribute("login",login);
                    request.setAttribute("retorno", retorno);
                }

                request.getRequestDispatcher("jsp/login/index.jsp").forward(request, response);
            }
        }
    }
}

```

```

    }
    // retorna sem login
    else
    {
        retorno="O campo Login deve ser preenchido!";
        request.setAttribute("retorno", retorno);

request.getRequestDispatcher("jsp/login/index.jsp").forward(request, response);
    }
}
else
{

request.getRequestDispatcher("jsp/login/index.jsp").forward(request, response);
}

}
catch(ServletException e)
{
    System.out.println(e.getMessage());
    e.printStackTrace();
    System.out.println("Saiu da Classe.");
}
}
}
}

```

12.2.5.10 Nascimento

```

package servelet;

import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpSession;
import javax.servlet.ServletException;
import java.sql.*;
import controladora.*;
import classepersistente.*;
import java.util.*;

/**
 * Servlet Class
 *
 * @web.servlet          name="SISINC"
 *                      display-name="Sistema de Incubação Experimental"
 *                      description="Cadastrar Nascimento"
 * @web.servlet-mapping url-pattern="/Nascimento"
 * @web.servlet-init-param name=""
 *                          value=""
 */
/*@autor Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano Padilha*/

public class Nascimento extends HttpServlet {
    // declaração do identificador universal da classe
    private static final long serialVersionUID = 16L;

    public void doGet(javax.servlet.http.HttpServletRequest request,
javax.servlet.http.HttpServletResponse response) throws javax.servlet.ServletException,
java.io.IOException
    {
        doPost(request, response);
    }

    public void doPost(javax.servlet.http.HttpServletRequest request,
javax.servlet.http.HttpServletResponse response)
throws javax.servlet.ServletException, java.io.IOException
    {
        try
        {
            // valida a sessão por grupo de usuário
            int perm = 2; // nível de acesso
            HttpSession s = request.getSession(false);
            if (s==null) {response.sendRedirect ("Login");}

```

```

        else{classepersistente.Usuario auxUsuario=(classepersistente.Usuario)
s.getAttribute("login");
        if((auxUsuario==null)|| (auxUsuario.getMatricula()==null))
        {response.sendRedirect("Login");}
        else{if(!auxUsuario.getStatus().equals("1"))

{request.getRequestDispatcher("scripts/status.jsp").forward(request,response);
        System.out.println("Acesso Não Autorizado. Status não está Ativo.");}
        else{if(Integer.parseInt(auxUsuario.getGrupo())>perm)
        {request.setAttribute("retorno",Integer.toString(perm));

        request.getRequestDispatcher("scripts/perm.jsp").forward(request,response);
        System.out.println("Acesso Não Autorizado. Necessário ao menos Nível
"+perm+".");}

        else{

        ControladoraNascimento auxContNas= new ControladoraNascimento();
        ControladoraConsulta auxCons= new ControladoraConsulta();

        String acao = request.getParameter("acao");
        System.out.println("Ação ->" + acao);

        if (acao != null)
        {
            String tam=null;

            //pega as variaveis da pagina//
            String projeto=request.getParameter("CodProj");
            String bandeja=request.getParameter("CodBan");
            String entidade=request.getParameter("CodEnt");
            String dataInicio=request.getParameter("DataIniInc");
            String dataPrev=request.getParameter("DataPrevInc");
            String nascedouro=request.getParameter("CodEnc");
            String codIncubacao=request.getParameter("CodInc");
            String codPos=request.getParameter("CodEncBan");
            String data=request.getParameter("Data");
            String qFinal=request.getParameter("qEntidade");
            String qInicial=request.getParameter("HQtEntidade");
            //String qTotal=request.getParameter("QtTotal");
            String codNascimento=request.getParameter("HCodNascimento");
            //imprime valores q vem da pagina//
            System.out.println("***esta em nascimento os dados da tela são em
cima****");

            System.out.println("projeto:"+projeto);
            System.out.println("bandeja:"+bandeja);
            System.out.println("entidade:"+entidade);
            System.out.println("data inicio:"+dataInicio);
            System.out.println("nascedouro:"+nascedouro);
            System.out.println("incubacao:"+codIncubacao);
            System.out.println("*codPos:"+codPos);
            System.out.println("HQtEntidadeInicio:"+qInicial);
            System.out.println("*QtEntidade:"+qFinal);
            System.out.println("nascimento"+codNascimento);
            //
            posicao e nascedouro//

            String incb=auxCons.retornaPosBanNas(bandeja);
            int ind1=incb.indexOf("-");
            int ind2=incb.indexOf(":");
            nascedouro=incb.substring(0,ind1-1);
            codPos=incb.substring(ind1+2,ind2-1);

            //buscando uma bandeja//
            if (acao.equals("Buscar Bandeja")){
                ControladoraConsulta cont = new ControladoraConsulta();

```

```

        tam=cont.retornaTamBan(bandeja);

    }else{

        //transferindo ovos de uma entidade//
        //transferindo posições.
        if(acao.equals("Transferir")){

            classepersistente.Nascimento auxNascimento=new
classepersistente.Nascimento();

            int x,y,lin,col,index,qtde;
            String pos;
            ArrayList nascimento=new ArrayList();
            ArrayList posicoes=new ArrayList();
            String cordenada=request.getParameter("cordenada");

            qtde=Integer.parseInt(qFinal)-Integer.parseInt(qInicial);

            index=cordenada.indexOf("-");
            lin=Integer.parseInt(cordenada.substring(0,index));
            col=Integer.parseInt(cordenada.substring(index+1));
            if(codNascimento.equals("")){

                //se o codigo do nascimento na pagina for branco
                //gravo com um método que grava nascimento
                for(x=0;x<lin;x++){
                    for(y=0;y<col;y++){

                        pos=x+"0"+y;
                        String
caixa=request.getParameter(pos);
                        System.out.println("caixa é:
"+caixa);
                        if(caixa!=null){
                            int
z=Integer.parseInt(request.getParameter("CodEncBan"));
                            Posicao posicao=new
Posicao();
                            posicao.setEntidade(entidade);

                            posicao.setPosicaoX(x);
                            posicao.setPosicaoY(y);
                            posicao.setPosicaoZ(z);
                            auxNascimento.setIncubacao(codIncubacao);
                            auxNascimento.setDatini(dataInicio);
                            auxNascimento.setData(data);
                            posicoes.add(posicao);
                            nascimento.add(auxNascimento);
                        }
                    }
                }

                //deve retornar nascimento
                try{
                    tam=lin+"-"+col;
                    int index1;

                    //passa para o método os objetos
"incubacao" e "posicoes" para gravar nas duas tabelas "incubacao" e "inctempos"

```

```

String
nas=auxContNas.transferir(nascimento, posicoes, qtde, codIncubacao, entidade);
System.out.println("nas:"+nas);
index1=nas.indexOf("|");

codNascimento=nas.substring(0, index1);

data=nas.substring(index+2);
dataPrev=data;
}
catch(SQLException e){
System.out.println(e);
}

}

}else{
System.out.println("Estou passando por nao
nulo*****");

// nascimento nao nulo//

for(x=0;x<lin;x++){
for(y=0;y<col;y++){

pos=x+"0"+y;
String
caixa=request.getParameter(pos);

if(caixa!=null){
int
z=Integer.parseInt(request.getParameter("CodEncBan"));

Posicao posicao=new
Posicao();

posicao.setEntidade(entidade);
posicao.setPosicaoX(x);
posicao.setPosicaoY(y);
posicao.setPosicaoZ(z);
posicao.setNascimento(Integer.parseInt(codNascimento));
auxNascimento.setIncubacao(codIncubacao);
auxNascimento.setDatini(dataInicio);
auxNascimento.setData(data);

System.out.println("
ent:"+entidade+" caixa:"+caixa+" proj:"+codIncubacao+" datIni"+dataInicio+"pos:"+pos);

posicoes.add(posicao);

}
}

}

//deve retornar nascimento

try{

tam=lin+"-"+col;

String
nas=auxContNas.transferir(posicoes, qtde, codIncubacao, entidade);
System.out.println("nas:"+nas);
int index2=nas.indexOf("|");
codNascimento=nas.substring(0, index2);
data=nas.substring(index+2);

System.out.println("nas:"+nas+"codnas:"+codNascimento+"data"+data);
dataPrev=data;
}
catch(SQLException e){

```

```

        System.out.println(e);
    }

    }

    }

    }

    Vector bandejaCheia=new Vector();
//setar as váriaveis para devolver a página
request.setAttribute("CodProj",projeto);
request.setAttribute("CodBan",bandeja);
request.setAttribute("CodEnt",entidade);
request.setAttribute("DataPrevInc",dataPrev);
request.setAttribute("CodEnc",nascvoudouro);
request.setAttribute("CodInc",codIncubacao);
request.setAttribute("incubacao",codIncubacao);
request.setAttribute("tam",tam);
request.setAttribute("codnasc",codNascimento);
request.setAttribute("CodEncBan",codPos);
//request.setAttribute("DataPrevInc",data);

bandejaCheia=auxContNas.setarBandeja(codPos);
request.setAttribute("bc",bandejaCheia);

baixo*****);

System.out.println("***esta em nascimento os dados da tela são-- em
baixo*****");

System.out.println("projeto:"+projeto);
System.out.println("bandeja:"+bandeja);
System.out.println("entidade:"+entidade);
System.out.println("data inicio:"+dataInicio);
System.out.println("data prev:"+dataPrev);
System.out.println("nascvoudouro:"+nascvoudouro);
System.out.println("incubacao:"+codIncubacao);
System.out.println("*codPos:"+codPos);
System.out.println("*codnasc:"+codNascimento);

request.getRequestDispatcher("jsp/nascimento/nascad.jsp").forward(request,response);

    }else{

request.getRequestDispatcher("jsp/nascimento/nascad.jsp").forward(request,response);
    }

    }}}}
    catch(ServletException e)
    {
        System.out.println("Saiu da Classe ");
        System.out.println (e.getMessage());
        e.printStackTrace();
    }
}

```

```

    }
}

```

12.2.5.11 Principal

```

package servelet;

import classepersistente.Usuario;
import javax.servlet.http.HttpSession;
import javax.servlet.http.HttpServlet;

/**
 * Servlet Class
 *
 * @web.servlet          name="SISINC"
 *                      display-name="Sistema de Incubação Experimental"
 *                      description="Principal"
 * @web.servlet-mapping url-pattern="/Principal"
 * @web.servlet-init-param name=""
 *                          value=""
 */

/**
 * Exibe a página de entrada.
 * <p>
 * Recebe as variáveis da página do respectivo formulário, seta as
 * variáveis da controladora direcionando para uma página de falha
 * ou sucesso dependendo do resultado de retorno.
 * @author      Arthur Maçaneiro Magyar
 * @author      Fábio Glauco dos Santos
 * @author      Luciano Padilha
 * @version     1.0
 */
public class Principal extends HttpServlet
{
    // declaração do identificador universal da classe
    private static final long serialVersionUID = 03L;

    /**
     *
     */
    public void doGet(javax.servlet.http.HttpServletRequest request,
        javax.servlet.http.HttpServletResponse response) throws javax.servlet.ServletException,
        java.io.IOException
    {
        doPost(request, response);
    }

    /**
     *
     */
    public void doPost(javax.servlet.http.HttpServletRequest request,
        javax.servlet.http.HttpServletResponse response)
        throws javax.servlet.ServletException, java.io.IOException
    {
        // valida a sessão por grupo de usuário
        int perm = 3; // nível de acesso
        HttpSession s = request.getSession(false);
        if(s == null)
        {
            response.sendRedirect("Login");
        }
        else
        {
            Usuario auxUsuario = (Usuario) s.getAttribute("login");
            if((auxUsuario == null) || (auxUsuario.getMatricula() == null))
            {
                response.sendRedirect("Login");
            }
        }
    }
}

```

```

    }
    else
    {
        if(! auxUsuario.getStatus().equals("1"))
        {
            request.getRequestDispatcher("scripts/status.jsp").forward(request,response);
            System.out.println("Acesso Não Autorizado. Status não
está Ativo.");
        }
        else
        {
            if(Integer.parseInt(auxUsuario.getGrupo()) > perm)
            {
                request.setAttribute("retorno", Integer.toString(perm));
                request.getRequestDispatcher("scripts/perm.jsp").forward(request,response);
                System.out.println("Acesso Não Autorizado.
Necessário ao menos Nível "+perm+".");
            }
            else
            {
                request.getRequestDispatcher("jsp/principal/index.jsp").forward(request,response);
            }
        }
    }
}
}
}
}

```

12.2.5.12 Projeto

```

package servelet;

import classepersistente.Usuario;
import controladora.ControladoraProjeto;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpSession;
import javax.servlet.ServletException;

/**
 * Servlet Class
 *
 * @web.servlet          name="SISINC"
 *                      display-name="Sistema de Incubação Experimental"
 *                      description="Cadastrar Projeto"
 * @web.servlet-mapping url-pattern="/Projeto"
 * @web.servlet-init-param name=""
 *                          value=""
 */

/**
 * Efetua cadastro de novo projeto e retorna mensagem.
 * <p>
 * Recebe as variáveis da página do respectivo formulário, seta as
 * variáveis da controladora direcionando para uma página de falha
 * ou sucesso dependendo do resultado de retorno.
 * @author      Arthur Maçaneiro Magyar
 * @author      Fábio Glaucos dos Santos
 * @author      Luciano Padilha
 * @version     1.0
 */
public class Projeto extends HttpServlet
{
    // declaração do identificador universal da classe
    private static final long serialVersionUID = 11L;

    /**
     *
     */
}

```

```

        public void doGet(javax.servlet.http.HttpServletRequest request,
javax.servlet.http.HttpServletResponse response) throws javax.servlet.ServletException,
java.io.IOException
        {
            doPost(request,response);
        }

/**
 *
 */
public void doPost(javax.servlet.http.HttpServletRequest request,
javax.servlet.http.HttpServletResponse response)
throws javax.servlet.ServletException, java.io.IOException
{
    try
    {
        // valida a sessão por grupo de usuário
        int perm = 1; // nível de acesso
        HttpSession s = request.getSession(false);
        if(s==null){response.sendRedirect("Login");}
        else{Usuario auxUsuario=(Usuario) s.getAttribute("login");
        if((auxUsuario==null)|| (auxUsuario.getMatricula()==null))
        {response.sendRedirect("Login");}
        else{if(! auxUsuario.getStatus().equals("1"))

{request.getRequestDispatcher("scripts/status.jsp").forward(request,response);
System.out.println("Acesso Não Autorizado. Status não está Ativo.");}
else{if(Integer.parseInt(auxUsuario.getGrupo())>perm)
{request.setAttribute("retorno", Integer.toString(perm));

request.getRequestDispatcher("scripts/perm.jsp").forward(request,response);
System.out.println("Acesso Não Autorizado. Necessário ao menos Nível
"+perm+".");}

else{

ControladoraProjeto auxControladoraProjeto = new ControladoraProjeto();
String acao=request.getParameter("Acao");
String nome=request.getParameter("NomeProj");
System.out.println("Ação -> "+acao);
System.out.println("Nome -> "+nome);
String retorno="";

if(acao != null)
{
    // carrega as variáveis da página
    String sigla =request.getParameter("SiglaProj");
    String entidade=request.getParameter("CodEnt");
    String especie=request.getParameter("CodEsp");
    String dataInicio=request.getParameter("DataIniProj");
    String dataPrev=request.getParameter("DataPrevProj");
    String descricao=request.getParameter("DescProj");

    // atribui vaziao a valores nulos
    if(dataPrev.equals("")||dataPrev.equals(" //")){dataPrev="01/01/9999";}
    if(descricao.equals("")){descricao="NULL";}

    // atribui as variáveis na controladora
    auxControladoraProjeto.setNome(nome);
    auxControladoraProjeto.setSiglaProj(sigla);
    auxControladoraProjeto.setEntidade(entidade);
    auxControladoraProjeto.setEspecie(especie);
    auxControladoraProjeto.setDataInicio(dataInicio);
    auxControladoraProjeto.setDataPrev(dataPrev);
    auxControladoraProjeto.setDescricao(descricao);

    // envia para a controladora o método gravar com parâmetro do servelet
    try
    {
        retorno=auxControladoraProjeto.gravarProjeto();
        System.out.println("Enviou -> "+nome);
        // retorna falha ou sucesso ao gravar no banco de dados
        if(retorno == "Projeto")
        {
            System.out.println("Gravou -> "+nome);
            request.setAttribute("retorno", retorno);

```

```

        request.getRequestDispatcher("scripts/sucesso.jsp").forward(request, response);
    }
    else
    {
        System.out.println("Não gravou -> "+retorno);
        // retorna falha personalizada
        if (retorno.equals("ERROR: duplicate key violates unique
constraint '+'+'projeto_siglaproj_key'+'''))
        {
            retorno = "a sigla informada está sendo utilizada
por outro projeto.";
        }
        request.setAttribute("retorno", retorno);

        request.getRequestDispatcher("scripts/falha.jsp").forward(request, response);
    }
    catch (Exception e)
    {
        System.out.println("Não enviou -> "+e.getMessage());
    }
    else
    {
        request.getRequestDispatcher("jsp/projeto/projcad.jsp").forward(request, response);
    }
    }
    }
    }
    catch (ServletException e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("Saiu da Classe.");
    }
}
}

```

12.2.5.13 ProjetoAlterar

```

package servelet;

import controladora.ControladoraProjeto;
import classepersistente.Projeto;
import classepersistente.Usuario;

import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpSession;
import javax.servlet.ServletException;
import java.util.*;

/**
 * Servlet Class
 *
 * @web.servlet          name="SISINC"
 *                      display-name="Sistema de Incubação Experimental"
 *                      description="Alterar Projeto"
 * @web.servlet-mapping url-pattern="/ProjetoAlterar"
 * @web.servlet-init-param name=""
 *                          value=""
 */

/**
 * Efetua consulta, alteração e marca projeto como excluído.
 * <p>
 * Recebe as variáveis da página do respectivo formulário, seta as
 * variáveis da controladora direcionando para uma página de falha
 * ou sucesso dependendo do resultado de retorno.
 * @author      Arthur Maçaneiro Magyar
 * @author      Fábio Glaucos dos Santos
 * @author      Luciano Padilha
 */

```

```

* @version      1.0
*/
public class ProjetoAlterar extends HttpServlet
{
    // declaração do identificador universal da classe
    private static final long serialVersionUID = 41L;

    /**
     *
     */
    public void doGet(javax.servlet.http.HttpServletRequest request,
        javax.servlet.http.HttpServletResponse response) throws javax.servlet.ServletException,
        java.io.IOException
    {
        doPost(request, response);
    }

    /**
     *
     */
    public void doPost(javax.servlet.http.HttpServletRequest request,
        javax.servlet.http.HttpServletResponse response)
        throws javax.servlet.ServletException, java.io.IOException
    {
        try
        {
            // valida a sessão por grupo de usuário
            int perm = 1; // nível de acesso
            HttpSession s = request.getSession(false);
            if (s == null) { response.sendRedirect("Login"); }
            else { Usuario auxUsuario = (Usuario) s.getAttribute("login");
                if ((auxUsuario == null) || (auxUsuario.getMatricula() == null))
                { response.sendRedirect("Login"); }
                else { if (! auxUsuario.getStatus().equals("1"))

                    { request.getRequestDispatcher("scripts/status.jsp").forward(request, response);
                        System.out.println("Acesso Não Autorizado. Status não está Ativo.");
                    }
                    else { if (Integer.parseInt(auxUsuario.getGrupo()) > perm)
                        { request.setAttribute("retorno", Integer.toString(perm));

                            request.getRequestDispatcher("scripts/perm.jsp").forward(request, response);
                            System.out.println("Acesso Não Autorizado. Necessário ao menos Nível
                                "+perm+".");
                        }
                    else {

                        ControladoraProjeto auxControladoraProjeto = new ControladoraProjeto();
                        String acao = request.getParameter("Acao");
                        String sigla = request.getParameter("SiglaProj");
                        String nome = request.getParameter("NomeProj");
                        String dataInicio = request.getParameter("DataIniProj");
                        String status = request.getParameter("CodStatusProj");
                        System.out.println("Ação -> "+acao);
                        System.out.println("Sigla -> "+sigla);
                        request.setAttribute("sigla", sigla);
                        request.setAttribute("nome", nome);
                        request.setAttribute("dataInicio", dataInicio);
                        request.setAttribute("status", status);
                        Vector resposta = new Vector();
                        String retorno = "";
                        String t = "";

                        if (acao != null)
                        {
                            // efetua consulta
                            if (acao.equals("Consultar"))
                            {
                                // se for preenchido: NomeProj ou NomeProj
                                if ("".compareTo(sigla) != 0 || "".compareTo(nome) != 0)
                                {
                                    // se for preenchido: SiglaProj
                                    if ("".compareTo(sigla) != 0)
                                    {
                                        // se for preenchido: SiglaProj e DataIniProj
                                        if ("".compareTo(dataInicio) != 0)
                                        {
                                            t = "x";
                                            String valor = sigla + " - " + dataInicio;
                                        }
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}

```

```

resposta=auxControladoraProjeto.retornaProjeto(t,valor,status);
        request.setAttribute("tabela", resposta);

request.getRequestDispatcher("jsp/projeto/projcon.jsp").forward(request,response);
    }
    // retorna a pesquisa por: SiglaProj
    else
    {
        t="s";

resposta=auxControladoraProjeto.retornaProjeto(t,sigla,status);
        request.setAttribute("tabela", resposta);

request.getRequestDispatcher("jsp/projeto/projcon.jsp").forward(request,response);
    }
    }
    else
    {
        // se for preenchido: NomeProj e DataIniProj
        if ("".compareTo(dataInicio)!=0)
        {
            t="y";
            String valor=nome+" - "+dataInicio;

resposta=auxControladoraProjeto.retornaProjeto(t,valor,status);
        request.setAttribute("tabela", resposta);

request.getRequestDispatcher("jsp/projeto/projcon.jsp").forward(request,response);
        }
        // retorna a pesquisa por: NomeProj
        else
        {
            t="n";

resposta=auxControladoraProjeto.retornaProjeto(t,nome,status);
        request.setAttribute("tabela", resposta);

request.getRequestDispatcher("jsp/projeto/projcon.jsp").forward(request,response);
        }
    }
    }
    else
    {
        // retorna a pesquisa por: DataIniProj
        if ("".compareTo(dataInicio)!=0)
        {
            t="d";

resposta=auxControladoraProjeto.retornaProjeto(t,dataInicio,status);
        request.setAttribute("tabela", resposta);

request.getRequestDispatcher("jsp/projeto/projcon.jsp").forward(request,response);
        }
        // se nada for preenchido
        else
        {
            t="a";
            dataInicio="";

resposta=auxControladoraProjeto.retornaProjeto(t,dataInicio,status);
        request.setAttribute("tabela", resposta);

request.getRequestDispatcher("jsp/projeto/projcon.jsp").forward(request,response);
        }
    }
    }
    else
    {
        // envia valores para: projalt
        if (acao.equals("Carregar"))
        {
            String codigo=request.getParameter("codigo");
            Projeto auxProjeto =
auxControladoraProjeto.materializar(codigo);
            request.setAttribute("dados",auxProjeto);

request.getRequestDispatcher("jsp/projeto/projalt.jsp").forward(request,response);
        }
    }
    }

```

```

// efetua alteração
if(acao.equals("Alterar"))
{
    // carrega as variáveis da página
    String entidade=request.getParameter("CodEnt");
    String especie=request.getParameter("CodEsp");
    String
dataPrev=request.getParameter("DataPrevProj");
    String descricao=request.getParameter("DescProj");

    // atribui vazios a valores nulos
    if(dataPrev.equals("")){dataPrev="01/01/9999";}
    if(descricao.equals("")){descricao="NULL";}

    // atribui as variáveis na controladora
    auxControladoraProjeto.setNome(nome);
    auxControladoraProjeto.setSiglaProj(sigla);
    auxControladoraProjeto.setEntidade(entidade);
    auxControladoraProjeto.setEspecie(especie);
    auxControladoraProjeto.setDataInicio(dataInicio);
    auxControladoraProjeto.setDataPrev(dataPrev);
    auxControladoraProjeto.setDescricao(descricao);

    // envia para a controladora o método gravar com
parâmetro do servlet
    try
    {
        retorno=auxControladoraProjeto.gravarProjeto(sigla);
        t="n";

        resposta=auxControladoraProjeto.retornaProjeto(t,nome,status);
        request.setAttribute("tabela", resposta);
        // retorna falha ou sucesso ao gravar no
banco de dados
        if(retorno == "ProjetoAlterado")
        {
            System.out.println("Alterou ->
"+sigla);
            request.setAttribute("retorno", retorno);
            request.getRequestDispatcher("scripts/sucesso.jsp").forward(request, response);
        }
        else
        {
            System.out.println("Não alterou ->
"+retorno);
            // retorna falha personalizada
            if(retorno.equals("ERROR: duplicate
key violates unique constraint "+""+"projeto_siglaproj_key"+''))
            {
                retorno = "a sigla informada
está sendo utilizada por outro projeto.";
            }

            request.setAttribute("retorno", retorno);
            request.getRequestDispatcher("scripts/falha.jsp").forward(request, response);
        }
    }
    catch(Exception e)
    {
        System.out.println("Não enviou ->
"+e.getMessage());
    }
}
else
{
    // efetua exclusão
    if(acao.equals("Excluir"))
    {
        // envia para a controladora o método
gravar com parâmetro do servlet
    }
}
}

```

```

        try
        {

            retorno=auxControladoraProjeto.mudarProjeto(sigla);
            // retorna falha ou sucesso ao
            gravar no banco de dados
            if(retorno == "ProjetoExclui")
            {
                System.out.println("Excluiu
-> "+sigla);

                request.setAttribute("retorno", retorno);

                request.getRequestDispatcher("scripts/sucesso.jsp").forward(request, response);
            }
            else
            {
                System.out.println("Não
excluiu -> "+retorno);

                request.setAttribute("retorno", retorno);

                request.getRequestDispatcher("scripts/falha.jsp").forward(request, response);
            }
        }
        catch(Exception e)
        {
            System.out.println("Não enviou ->
"+e.getMessage());
        }
    }
}
else
{
    request.getRequestDispatcher("jsp/projeto/projcon.jsp").forward(request, response);
}
}}}}}
catch(ServletException e)
{
    System.out.println(e.getMessage());
    e.printStackTrace();
    System.out.println("Saiu da Classe.");
}
}
}

```

12.2.5.14 Sair

```

package servelet;

import classepersistente.Usuario;
import javax.servlet.http.HttpSession;
import javax.servlet.http.HttpServlet;
import javax.servlet.ServletException;

/**
 * Servlet Class
 *
 * @web.servlet          name="SISINC"
 *                      display-name="Sistema de Incubação Experimental"
 *                      description="Cancelar Login"
 * @web.servlet-mapping url-pattern="/Sair"
 * @web.servlet-init-param name=""
 *                          value=""
 */

```

```

/**
 * Efetua logout de usuário.
 * <p>
 * Recebe as variáveis da página do respectivo formulário, seta as
 * variáveis da controladora direcionando para uma página de falha
 * ou sucesso dependendo do resultado de retorno.
 * @author Arthur Maçaneiro Magyar
 * @author Fábio Glauco dos Santos
 * @author Luciano Padilha
 * @version 1.0
 */
public class Sair extends HttpServlet
{
    // declaração do identificador universal da classe
    private static final long serialVersionUID = 02L;

    /**
     *
     */
    public void doGet(javax.servlet.http.HttpServletRequest request,
    javax.servlet.http.HttpServletResponse response) throws javax.servlet.ServletException,
    java.io.IOException
    {
        doPost(request, response);
    }

    /**
     *
     */
    public void doPost(javax.servlet.http.HttpServletRequest request,
    javax.servlet.http.HttpServletResponse response)
    throws javax.servlet.ServletException, java.io.IOException
    {
        try
        {
            // valida a sessão por grupo de usuário
            int perm = 3; // nível de acesso
            HttpSession s = request.getSession(false);
            if(s==null){response.sendRedirect("Login");}
            else{Usuario auxUsuario=(Usuario) s.getAttribute("login");
            if((auxUsuario==null)|| (auxUsuario.getMatricula()==null))
            {response.sendRedirect("Login");}
            else{if(! auxUsuario.getStatus().equals("1"))

            {request.getRequestDispatcher("scripts/status.jsp").forward(request, response);
            System.out.println("Acesso Não Autorizado. Status não está Ativo.");}
            else{if (Integer.parseInt(auxUsuario.getGrupo())>perm)
            {request.setAttribute("retorno", Integer.toString(perm));

            request.getRequestDispatcher("scripts/perm.jsp").forward(request, response);
            System.out.println("Acesso Não Autorizado. Necessário ao menos Nível
            "+perm+".");}

            else{

            String acao=request.getParameter("Acao");

            if(acao != null)
            {
                // se for selecionado Não
                if(acao.equals("Não"))
                {
                    response.sendRedirect("Principal");
                }
                // se for selecionado Sim
                else
                {
                    System.out.println("Acesso Finalizado.");
                    response.sendRedirect("Login");
                    s = request.getSession(false);
                    s.invalidate();
                }
            }
            else
            {

            request.getRequestDispatcher("jsp/sair/sair.jsp").forward(request, response);

```

```

    }
    }}}}
    catch(ServletException e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("Saiu da Classe.");
    }
}
}

```

12.2.5.15 Saque

```

package servelet;

import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpSession;
import javax.servlet.ServletException;
import java.sql.*;
import controladora.*;
import classepersistente.*;
import java.util.*;

/**
 * Servlet Class
 *
 * @web.servlet          name="SISINC"
 *                      display-name="Sistema de Incubação Experimental"
 *                      description="Cadastrar Incubação"
 * @web.servlet-mapping url-pattern="/Incubacao"
 * @web.servlet-init-param name=""
 *                          value=""
 */
/*@autor Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano Padilha*/

public class Saque extends HttpServlet {
    // declaração do identificador universal da classe
    private static final long serialVersionUID = 17L;

    public void doGet(javax.servlet.http.HttpServletRequest request,
        javax.servlet.http.HttpServletResponse response) throws javax.servlet.ServletException,
        java.io.IOException
    {
        doPost(request, response);
    }

    public void doPost(javax.servlet.http.HttpServletRequest request,
        javax.servlet.http.HttpServletResponse response)
        throws javax.servlet.ServletException, java.io.IOException
    {
        try
        {
            // valida a sessão por grupo de usuário
            int perm = 2; // nível de acesso
            HttpSession s = request.getSession(false);
            if(s==null){response.sendRedirect("Login");}
            else{classepersistente.Usuario auxUsuario=(classepersistente.Usuario)
s.getAttribute("login");
            if((auxUsuario==null)|| (auxUsuario.getMatricula()==null))
            {response.sendRedirect("Login");}
            else{if(!auxUsuario.getStatus().equals("1"))

            {request.getRequestDispatcher("scripts/status.jsp").forward(request, response);
            System.out.println("Acesso Não Autorizado. Status não está Ativo.");}
            else{if(Integer.parseInt(auxUsuario.getGrupo())>perm)
            {request.setAttribute("retorno", Integer.toString(perm));

            request.getRequestDispatcher("scripts/perm.jsp").forward(request, response);

```

```

        System.out.println("Acesso Não Autorizado. Necessário ao menos Nível
"+perm+".");}

        else{

            ControladoraSaque auxContSaq= new ControladoraSaque();
            //ControladoraConsulta auxCons= new ControladoraConsulta();

            String acao = request.getParameter("acao");
            System.out.println("Ação ->" + acao);

            if (acao != null)
            {
                String tam=null;

                //pega as variaveis da pagina//
                String projeto=request.getParameter("CodProj");
                String bandeja=request.getParameter("CodBan");
                String entidade=request.getParameter("CodEnt");
                String dataInicio=request.getParameter("DataIniInc");
                String dataPrev=request.getParameter("DataPrevInc");
                String nascedouro=request.getParameter("CodEnc");
                String codIncubacao=request.getParameter("CodInc");
                String codPos=request.getParameter("CodEncBan");
                //String data=request.getParameter("Data");
                String qFinal=request.getParameter("qEntidade");
                String qInicial=request.getParameter("HQtEntidade");
                //String qTotal=request.getParameter("QtdTotal");
                String codNascimento=request.getParameter("CodNas");
                //imprime valores q vem da pagina//
                System.out.println("***esta em nascimento os dados da tela são em
cima****");

                System.out.println("projeto:"+projeto);
                System.out.println("bandeja:"+bandeja);
                System.out.println("entidade:"+entidade);
                System.out.println("data inicio:"+dataInicio);
                System.out.println("nascedouro:"+nascedouro);
                System.out.println("incubacao:"+codIncubacao);
                System.out.println("*codPos:"+codPos);
                System.out.println("HQtEntidadeInicio:"+qInicial);
                System.out.println("*QtdEntidade:"+qFinal);
                System.out.println("nascimento"+codNascimento);

                //sacando//

                if(acao.equals("Confirmar")){
                    //classepersistente.Nascimento auxNascimento=new
classepersistente.Nascimento();

                    int x,y,lin,col,index;
                    //int qtde;
                    String pos;
                    ArrayList posicoes=new ArrayList();

                    String cordenada=request.getParameter("cordenada");
                    index=cordenada.indexOf("-");
                    lin=Integer.parseInt(cordenada.substring(0,index));
                    col=Integer.parseInt(cordenada.substring(index+1));
                    String nasceu=request.getParameter("nasceu");

                    //se o codigo do nascimento na pagina for branco
                    //gravo com um método que grava nascimento

                    for(x=0;x<lin;x++){
                        for(y=0;y<col;y++){

                            pos=x+"0"+y;
                            String
caixa=request.getParameter(pos);

                            System.out.println("caixa é:
"+caixa);

                            if(caixa==null){

```

```

                                                                    int
z=Integer.parseInt(request.getParameter("CodEncBan"));
                                                                    Posicao posicao=new
Posicao();

    posicao.setPosicaoX(x);
    posicao.setPosicaoY(y);
    posicao.setPosicaoZ(z);
    posicao.setNasceu(nasceu);
    posicoes.add(posicao);
                                                                    }
                                                                    }
                                                                    }

//deve retornar nascimento
                                                                    try{
                                                                    tam=lin+"-"+col;
                                                                    //int index1,index2;
                                                                    auxContSaq.saque(posicoes);
                                                                    }
                                                                    catch(SQLException e){
                                                                    System.out.println(e);
                                                                    }
                                                                    }

                                                                    }

//setar as váriaveis para devolver a página
    request.setAttribute("CodProj",projeto);
    request.setAttribute("CodBan",bandeja);
    request.setAttribute("CodEnt",entidade);
    request.setAttribute("DataPrevInc",dataPrev);
    request.setAttribute("CodEnc",nasedouro);
    request.setAttribute("codInc",codIncubacao);
    request.setAttribute("tam",tam);
    request.setAttribute("codNascimento",codNascimento);
    request.setAttribute("CodEncBan",codPos);
    //request.setAttribute("DataPrevInc",data);

    System.out.println("***esta em nascimento os dados da tela são-- em
baixo*****");

    System.out.println("projeto:"+projeto);
    System.out.println("bandeja:"+bandeja);
    System.out.println("entidade:"+entidade);
    System.out.println("data inicio:"+dataInicio);
    System.out.println("data prev:"+dataPrev);
    System.out.println("nasedouro:"+nasedouro);
    System.out.println("incubacao:"+codIncubacao);
    System.out.println("*codPos:"+codPos);
    System.out.println("*codnasc:"+codNascimento);

    request.getRequestDispatcher("jsp/saque/saqcad.jsp").forward(request,response);

    }else{

    request.getRequestDispatcher("jsp/saque/saqcad.jsp").forward(request,response);

```

```

    }

    }}}}
    {
        catch(ServletException e)
        {
            System.out.println("Saiu da Classe ");
            System.out.println (e.getMessage());
            e.printStackTrace();
        }
    }
}
}
}

```

12.2.5.16 Usuario

```

package servelet;

import controladora.ControladoraUsuario;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpSession;
import javax.servlet.ServletException;

/**
 * Servlet Class
 *
 * @web.servlet          name="SISINC"
 *                      display-name="Sistema de Incubação Experimental"
 *                      description="Cadastrar Usuário"
 * @web.servlet-mapping url-pattern="/Usuario"
 * @web.servlet-init-param name=""
 *                          value=""
 */

/**
 * Efetua cadastro de novo usuário e retorna mensagem.
 * <p>
 * Recebe as variáveis da página do respectivo formulário, seta as
 * variáveis da controladora direcionando para uma página de falha
 * ou sucesso dependendo do resultado de retorno.
 * @author      Arthur Maçaneiro Magyar
 * @author      Fábio Glaucos dos Santos
 * @author      Luciano Padilha
 * @version     1.0
 */
public class Usuario extends HttpServlet
{
    // declaração do identificador universal da classe
    private static final long serialVersionUID = 10L;

    /**
     *
     */
    public void doGet(javax.servlet.http.HttpServletRequest request,
        javax.servlet.http.HttpServletResponse response) throws javax.servlet.ServletException,
        java.io.IOException
    {
        doPost (request, response);
    }

    /**
     *
     */
    public void doPost(javax.servlet.http.HttpServletRequest request,
        javax.servlet.http.HttpServletResponse response)

```

```

throws javax.servlet.ServletException, java.io.IOException
{
    try
    {
        // valida a sessão por grupo de usuário
        int perm = 1; // nível de acesso
        HttpSession s = request.getSession(false);
        if (s==null) {response.sendRedirect("Login");}
        else {classepersistente.Usuario auxUsuario=(classepersistente.Usuario)
s.getAttribute("login");
        if ((auxUsuario==null) || (auxUsuario.getMatricula()==null))
        {response.sendRedirect("Login");}
        else {if (!auxUsuario.getStatus().equals("1"))

{request.getRequestDispatcher("scripts/status.jsp").forward(request, response);
        System.out.println("Acesso Não Autorizado. Status não está Ativo.");}
        else {if (Integer.parseInt(auxUsuario.getGrupo())>perm)
        {request.setAttribute("retorno", Integer.toString(perm));

request.getRequestDispatcher("scripts/perm.jsp").forward(request, response);
        System.out.println("Acesso Não Autorizado. Necessário ao menos Nível
"+perm+".");}

        else{

            ControladoraUsuario auxControladoraUsuario = new ControladoraUsuario();
            String acao=request.getParameter("Acao");
            String nome=request.getParameter("NomeUsu");
            System.out.println("Ação -> "+acao);
            System.out.println("Nome -> "+nome);
            String retorno="";

            if (acao != null)
            {
                // carrega as variáveis da página
                String matricula=request.getParameter("MatrUsu");
                String grupo=request.getParameter("CodGru");
                String senha=request.getParameter("SenCrt");
                String telres=request.getParameter("TelRes");
                String telcom=request.getParameter("TelCom");
                String telcel=request.getParameter("TelCel");
                String email=request.getParameter("Email");

                // atribui vazios a valores nulos
                if (telres.equals("")) {telres="NULL";}
                if (telcom.equals("")) {telcom="NULL";}
                if (telcel.equals("")) {telcel="NULL";}

                // atribui as variáveis na controladora
                auxControladoraUsuario.setMatricula(matricula);
                auxControladoraUsuario.setNome(nome);
                auxControladoraUsuario.setGrupo(grupo);
                auxControladoraUsuario.setSenha(senha);
                auxControladoraUsuario.setTelres(telres);
                auxControladoraUsuario.setTelcom(telcom);
                auxControladoraUsuario.setTelcel(telcel);
                auxControladoraUsuario.setEmail(email);

                // envia para a controladora o método gravar com parâmetro do servelet
                try
                {
                    retorno=auxControladoraUsuario.gravarUsuario();
                    System.out.println("Enviou -> "+nome);
                    // retorna falha ou sucesso ao gravar no banco de dados
                    if (retorno == "Usuario")
                    {
                        System.out.println("Gravou -> "+nome);
                        request.setAttribute("retorno", retorno);

request.getRequestDispatcher("scripts/sucesso.jsp").forward(request, response);
                    }
                    else
                    {
                        System.out.println("Não gravou -> "+retorno);
                        // retorna falha personalizada
                        if (retorno.equals("ERROR: duplicate key violates unique
constraint '"+''+"usuario_matrusu_key'+'''))

```

```

        {
            retorno = "a matrícula informada está sendo
utilizada por outro usuário.";
        }
        request.setAttribute("retorno", retorno);

        request.getRequestDispatcher("scripts/falha.jsp").forward(request, response);
    }
    catch(Exception e)
    {
        System.out.println("Não enviou -> "+e.getMessage());
    }
    else
    {
        request.getRequestDispatcher("jsp/gerenciamento/usucad.jsp").forward(request, response);
    }

    }
    catch(ServletException e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("Saiu da Classe.");
    }
}
}

```

12.2.5.17 UsuarioAlterar

```

package servelet;

import controladora.ControladoraUsuario;
import classepersistente.Usuario;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpSession;
import javax.servlet.ServletException;
import java.util.*;

/**
 * Servlet Class
 *
 * @web.servlet          name="SISINC"
 *                      display-name="Sistema de Incubação Experimental"
 *                      description="Alterar Usuário"
 * @web.servlet-mapping url-pattern="/UsuarioAlterar"
 * @web.servlet-init-param name=""
 *                          value=""
 */

/**
 * Efetua consulta, alteração e marca usuário como excluído.
 * <p>
 * Recebe as variáveis da página do respectivo formulário, seta as
 * variáveis da controladora direcionando para uma página de falha
 * ou sucesso dependendo do resultado de retorno.
 * @author      Arthur Maçaneiro Magyar
 * @author      Fábio Glauco dos Santos
 * @author      Luciano Padilha
 * @version     1.0
 */

public class UsuarioAlterar extends HttpServlet
{
    // declaração do identificador universal da classe
    private static final long serialVersionUID = 40L;

    /**
     *
     */
}

```



```

else
{
    // se for preenchido: NomeUsu
    if ("".compareTo(nome) != 0)
    {
        resposta=auxControladoraUsuario.retornaUsuario("NomeUsu", nome, status);
        request.setAttribute("tabela", resposta);

        request.getRequestDispatcher("jsp/gerenciamento/usucon.jsp").forward(request, response);
    }
    // se nada for preenchido
    else
    {
        resposta=auxControladoraUsuario.retornaUsuario(status);
        request.setAttribute("tabela", resposta);

        request.getRequestDispatcher("jsp/gerenciamento/usucon.jsp").forward(request, response);
    }
}
else
{
    // envia valores para: usualt
    if (acao.equals("Carregar"))
    {
        String codigo=request.getParameter("codigo");
        auxUsuario = auxControladoraUsuario.materializar(codigo);
        request.setAttribute("dados", auxUsuario);

        request.getRequestDispatcher("jsp/gerenciamento/usualt.jsp").forward(request, response);
    }
    else
    {
        // efetua alteração
        if (acao.equals("Alterar"))
        {
            // carrega as variáveis da página
            String grupo=request.getParameter("CodGru");
            String senha=request.getParameter("SenCrt");
            String telres=request.getParameter("TelRes");
            String telcom=request.getParameter("TelCom");
            String telcel=request.getParameter("TelCel");
            String email=request.getParameter("Email");

            // atribui vazios a valores nulos
            if (telres.equals("")) {telres="NULL";}
            if (telcom.equals("")) {telcom="NULL";}
            if (telcel.equals("")) {telcel="NULL";}

            // atribui as variáveis na controladora
            auxControladoraUsuario.setMatricula(matricula);
            auxControladoraUsuario.setNome(nome);
            auxControladoraUsuario.setGrupo(grupo);
            auxControladoraUsuario.setSenha(senha);
            auxControladoraUsuario.setTelres(telres);
            auxControladoraUsuario.setTelcom(telcom);
            auxControladoraUsuario.setTelcel(telcel);
            auxControladoraUsuario.setEmail(email);
            auxControladoraUsuario.setStatus(status);

            // envia para a controladora o método gravar com
            parâmetro do servelet
            try
            {
                retorno=auxControladoraUsuario.gravarUsuario(matricula);

                resposta=auxControladoraUsuario.retornaUsuario("NomeUsu", nome, status);
                request.setAttribute("tabela", resposta);
                // retorna falha ou sucesso ao gravar no
                banco de dados
                if (retorno == "UsuarioAlterar")

```

```

        {
            System.out.println("Alterou ->
"+nome);
            request.setAttribute("retorno", retorno);
            request.getRequestDispatcher("scripts/sucesso.jsp").forward(request, response);
        }
        else
        {
            System.out.println("Não alterou ->
"+retorno);
            // retorna falha personalizada
            if(retorno.equals("ERROR: duplicate
key violates unique constraint '+'+"usuario_matrusu_key"+""))
            {
                retorno = "a matrícula
informada está sendo utilizada por outro usuário.";
            }

            request.setAttribute("retorno", retorno);
            request.getRequestDispatcher("scripts/falha.jsp").forward(request, response);
        }
        catch(Exception e)
        {
            System.out.println("Não enviou ->
"+e.getMessage());
        }
    }
    else
    {
        // efetua exclusão
        if(acao.equals("Excluir"))
        {
            // envia para a controladora o método
            try
            {
                retorno=auxControladoraUsuario.mudarUsuario(matricula);
                // retorna falha ou sucesso ao
                gravar no banco de dados
                if(retorno == "UsuarioExclui")
                {
                    System.out.println("Excluiu
-> "+nome);
                    request.setAttribute("retorno", retorno);
                    request.getRequestDispatcher("scripts/sucesso.jsp").forward(request, response);
                }
                else
                {
                    System.out.println("Não
excluiu -> "+retorno);
                    request.setAttribute("retorno", retorno);
                    request.getRequestDispatcher("scripts/falha.jsp").forward(request, response);
                }
            }
            catch(Exception e)
            {
                System.out.println("Não enviou ->
"+e.getMessage());
            }
        }
    }
}
}
else
{

```

```
request.getRequestDispatcher("jsp/gerenciamento/usucon.jsp").forward(request, response);
    }
    }}}}
    catch(ServletException e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        System.out.println("Saiu da Classe.");
    }
}
}
```

12.3 RELATÓRIOS

12.3.1 EntidadeBloco

EntidadeBloco.mdb

Table: public_entidade

Page: 1

Columns

Name	Type	Size
codent	Decimal	16
codstatusgeral	Decimal	16
codtipoent	Decimal	16
nomeent	Text	50
descent	Text	254

Table Indexes

Name	Number of Fields
entidade_pkey	1
Fields:	
codent	Ascending

EntidadeBloco.mdb

Table: public_vEntidadeBloco

Page: 2

Columns

Name	Type	Size
codent	Decimal	16
nomeent	Text	50
descent	Text	254
codstatusgeral	Decimal	16
nomestatusgeral	Text	50
codtipoent	Decimal	16
nometipoent	Text	50
tipoentmin	Text	1
codsala	Decimal	16

EntidadeBloco.mdb

Query: EntidadeBloco

Page: 3

SQL

```
SELECT public_vEntidadeBloco.codent, public_vEntidadeBloco.nomeent,
public_vEntidadeBloco.descent,
public_vEntidadeBloco.codstatusgeral, public_vEntidadeBloco.nomestatusgeral,
public_vEntidadeBloco.codtipoent, public_vEntidadeBloco.nometipoent,
public_vEntidadeBloco.tipoentmin,
public_vEntidadeBloco.codsala, public_entidade.codstatusgeral,
public_entidade.codtipoent,
public_entidade.nomeent, public_entidade.descent
FROM public_vEntidadeBloco INNER JOIN public_entidade ON
public_vEntidadeBloco.codsala=public_entidade.codent;
```

Columns

Name	Type	Size
codent	: Update Query	16
public_vEntidadeBloco.nomeent	Text	50
public_vEntidadeBloco.descent	Text	254
public_vEntidadeBloco.codstatusgeral	: Update Query	16
nomestatusgeral	Text	50

public_vEntidadeBloco.codtipoent	: Update Query	16
nometipoent	Text	50
tipoentmin	Text	1
codsala	: Update Query	16
public_entidade.codstatusgeral	: Update Query	16
public_entidade.codtipoent	: Update Query	16
public_entidade.nomeent	Text	50
public_entidade.descent	Text	254

Table Indexes

Name	Number of Fields
entidade_pkey	1

EntidadeBloco.mdb

Database: C:\Program Files\Apache Software Foundation\Tomcat

Page: 4

Properties

AccessVersion:	08.50	ANSI Query Mode:	0
Build:	566	CollatingOrder:	General
ProjVer:	35	QueryTimeout:	60
RecordsAffected:	0	Themed Form Controls:	1
Transactions:	True	Updatable:	True
Version:	4.0		

User Permissions

admin

Group Permissions

Admins
Users

12.3.2 EntidadeCelula

EntidadeCelula.mdb

Table: public_entidade

Page: 1

Properties

Attributes:	Linked ODBC	Connect:	ODBC;DSN=PostgreSQL;DATABASE=sisinc;SERVER=localhost;PORT=5432;A6=;A7=100;A8=8192;B0=254;B1=8190;BI=0;C2=dd_;CX=1b503a9
DateCreated:	1/10/2006 11:09:51 PM	DefaultView:	Datasheet
GUID:	{guid {F4E405CF-06F2-45BA-884A-E3D4E705EA5D}}	LastUpdated:	1/10/2006 11:09:52 PM
NameMap:	Long binary data	OrderByOn:	False
Orientation:	Left-to-Right	RecordCount:	-1
SourceTableName:	public.entidade	Updatable:	False

Columns

Name	Type	Size
codent	Decimal	16
codstatusgeral	Decimal	16
codtipoent	Decimal	16
nomeent	Text	50
descent	Text	254

Table Indexes

Name	Number of Fields
entidade_pkey	1

Fields:
codent Ascending

EntidadeCelula.mdb

Table: public_vEntidadeCelula

Page: 2

Properties

Attributes:	Linked ODBC	Connect:	ODBC;DSN=PostgreSQL;DATABASE=sisinc;SERVER=localhost;PORT=5432;A6=;A7=100;A8=8192;B0=254;B1=8190;BI=0;C2=dd_;CX=1b503a9
DateCreated:	1/10/2006 11:09:52 PM	DefaultView:	Datasheet
GUID:	{guid {06D1B983-C24C-47D3-87AA-3F030202A628}}	LastUpdated:	1/10/2006 11:09:52 PM
NameMap:	Long binary data	OrderByOn:	False
Orientation:	Left-to-Right	RecordCount:	-1
SourceTableName:	public.vEntidadeCelula	Updatable:	False

Columns

Name	Type	Size
codent	Decimal	16
nomeent	Text	50
descent	Text	254
codstatusgeral	Decimal	16
nomestatusgeral	Text	50
codtipoent	Decimal	16
nometipoent	Text	50
tipoentmin	Text	1
codgai	Decimal	16

EntidadeCelula.mdb

Query: EntidadeCelula

Page: 3

SQL

```
SELECT public_vEntidadeCelula.codent, public_vEntidadeCelula.nomeent,
public_vEntidadeCelula.codstatusgeral, public_vEntidadeCelula.nomestatusgeral,
public_vEntidadeCelula.codtipoent, public_vEntidadeCelula.nometipoent,
public_vEntidadeCelula.tipoentmin, public_vEntidadeCelula.codgai,
public_entidade.nomeent
FROM public_vEntidadeCelula INNER JOIN public_entidade ON
public_vEntidadeCelula.codgai=public_entidade.codent;
```

Columns

Name	Type	Size
codent	: Update Query	16
public_vEntidadeCelula.nomeent	Text	50
codstatusgeral	: Update Query	16
nomestatusgeral	Text	50
codtipoent	: Update Query	16
nometipoent	Text	50
tipoentmin	Text	1
codgai	: Update Query	16
public_entidade.nomeent	Text	50

Table Indexes

Name	Number of Fields
entidade_pkey	1

Properties

AccessVersion:	08.50	ANSI Query Mode:	0
Build:	566	CollatingOrder:	General
ProjVer:	35	QueryTimeout:	60
RecordsAffected:	0	Themed Form Controls:	1
Transactions:	True	Updatable:	True
Version:	4.0		

User Permissions

admin

Group Permissions

Admins
Users

12.3.3 EntidadeGaiola

Columns

Name	Type	Size
codent	Decimal	16
codstatusgeral	Decimal	16
codtipoent	Decimal	16
nomeent	Text	50
descent	Text	254

Table Indexes

Name	Number of Fields
entidade_pkey	1
Fields: codent	Ascending

Columns

Name	Type	Size
codent	Decimal	16
nomeent	Text	50
descent	Text	254
codstatusgeral	Decimal	16
nomestatusgeral	Text	50
codtipoent	Decimal	16
nometipoent	Text	50
tipoentmin	Text	1
codlinha	Decimal	16

SQL

```

SELECT public_vEntidadeGaiola.codent, public_vEntidadeGaiola.nomeent,
public_vEntidadeGaiola.descent,
    public_vEntidadeGaiola.codstatusgeral, public_vEntidadeGaiola.nomestatusgeral,
    public_vEntidadeGaiola.codtipoent, public_vEntidadeGaiola.nometipoent,
    public_vEntidadeGaiola.tipoentmin, public_vEntidadeGaiola.codlinha,
public_entidade.codstatusgeral,
    public_entidade.codtipoent, public_entidade.nomeent, public_entidade.descent
FROM public_vEntidadeGaiola INNER JOIN public_entidade ON
public_vEntidadeGaiola.codlinha =
    public_entidade.codent;

```

Columns

Name	Type	Size
codent	: Update Query	16
public_vEntidadeGaiola.nomeent	Text	50
public_vEntidadeGaiola.descent	Text	254
public_vEntidadeGaiola.codstatusgeral	: Update Query	16
nomestatusgeral	Text	50
public_vEntidadeGaiola.codtipoent	: Update Query	16
nometipoent	Text	50
tipoentmin	Text	1
codlinha	: Update Query	16
public_entidade.codstatusgeral	: Update Query	16
public_entidade.codtipoent	: Update Query	16
public_entidade.nomeent	Text	50
public_entidade.descent	Text	254

Table Indexes

Name	Number of Fields
entidade_pkey	1

EntidadeGaiola.mdb

Database: C:\Program Files\Apache Software Foundation\Tomcat

Page: 4

Properties

AccessVersion:	08.50	ANSI Query Mode:	0
Build:	566	CollatingOrder:	General
ProjVer:	35	QueryTimeout:	60
RecordsAffected:	0	Themed Form Controls:	1
Transactions:	True	Updatable:	True
Version:	4.0		

User Permissions

admin

Group Permissions

Admins
Users

12.3.4 EntidadeLinha

EntidadeLinha.mdb

Table: public_entidade

Page: 1

Columns

Name	Type	Size
codent	Decimal	16
codstatusgeral	Decimal	16
codtipoent	Decimal	16
nomeent	Text	50
descent	Text	254

Table Indexes

Name	Number of Fields
entidade_pkey	1
Fields:	
codent	Ascending

EntidadeLinha.mdb

Table: public_vEntidadeLinha

Page: 2

Columns

Name	Type	Size
codent	Decimal	16
nomeent	Text	50
descent	Text	254
codstatusgeral	Decimal	16
nomestatusgeral	Text	50
codtipoent	Decimal	16
nometipoent	Text	50
tipoentmin	Text	1
codbloco	Decimal	16

EntidadeLinha.mdb

Query: EntidadeLinha

Page: 3

SQL

```
SELECT public_vEntidadeLinha.codent, public_vEntidadeLinha.nomeent,
public_vEntidadeLinha.descent,
public_vEntidadeLinha.codstatusgeral, public_vEntidadeLinha.nomestatusgeral,
public_vEntidadeLinha.codtipoent, public_vEntidadeLinha.nometipoent,
public_vEntidadeLinha.tipoentmin,
public_vEntidadeLinha.codbloco, public_entidade.codstatusgeral,
public_entidade.codtipoent,
public_entidade.nomeent, public_entidade.descent
FROM public_entidade INNER JOIN public_vEntidadeLinha ON public_entidade.codent =
public_vEntidadeLinha.codbloco;
```

Columns

Name	Type	Size
codent	: Update Query	16
public_vEntidadeLinha.nomeent	Text	50
public_vEntidadeLinha.descent	Text	254
public_vEntidadeLinha.codstatusgeral	: Update Query	16
nomestatusgeral	Text	50
public_vEntidadeLinha.codtipoent	: Update Query	16
nometipoent	Text	50
tipoentmin	Text	1
codbloco	: Update Query	16
public_entidade.codstatusgeral	: Update Query	16
public_entidade.codtipoent	: Update Query	16
public_entidade.nomeent	Text	50
public_entidade.descent	Text	254

Table Indexes

Name	Number of Fields
entidade_pkey	1

Properties

AccessVersion:	08.50	ANSI Query Mode:	0
Build:	566	CollatingOrder:	General
ProjVer:	35	QueryTimeout:	60
RecordsAffected:	0	Themed Form Controls:	1
Transactions:	True	Updatable:	True
Version:	4.0		

User Permissions

admin

Group Permissions

Admins
Users

12.3.5 ProjetoIncubacao

Columns

Name	Type	Size
codproj	Decimal	16
siglaproj	Text	14
nomeproj	Text	100
datainiproj	Date/Time	8
dataprevproj	Date/Time	8
datafimproj	Date/Time	8
descproj	Text	254
numinc	Decimal	16
codesp	Decimal	16
codstatusgeral	Decimal	16
nomestatusgeral	Text	50
nomeesp	Text	50
tempmininc	Decimal	16
tempmaxinc	Decimal	16
umimininc	Decimal	16
umimaxinc	Decimal	16
perinc	Decimal	16
peraviinc	Decimal	16
tempminnas	Decimal	16
tempmaxnas	Decimal	16
umiminnas	Decimal	16
umimaxnas	Decimal	16
pernas	Decimal	16
peravinas	Decimal	16
perrec	Decimal	16
codtipoent	Decimal	16
nometipoent	Text	50
tipoentmin	Text	1
codstatusproj	Decimal	16
nomestatusproj	Text	50
codstatusin	Decimal	16
datainiinc	Date/Time	8
dataprevinc	Date/Time	8
datafiminc	Date/Time	8
numovoinii	Decimal	16

numovodesi	Decimal	16
numovotransf	Decimal	16
obsstatusinc	Text	254

ProjetoIncubacao.mdb

Query: public_vProjetoIncubacao Consulta

Page: 2

SQL

```
SELECT public_vProjetoIncubacao.codproj, public_vProjetoIncubacao.siglaproj,
public_vProjetoIncubacao.nomeproj, public_vProjetoIncubacao.datainiproj,
public_vProjetoIncubacao.dataprevproj, public_vProjetoIncubacao.datafimproj,
public_vProjetoIncubacao.descproj, public_vProjetoIncubacao.numinc,
public_vProjetoIncubacao.codesp,
public_vProjetoIncubacao.codstatusgeral, public_vProjetoIncubacao.nomestatusgeral,
public_vProjetoIncubacao.nomeesp, public_vProjetoIncubacao.temppmininc,
public_vProjetoIncubacao.temppmaxinc, public_vProjetoIncubacao.umimininc,
public_vProjetoIncubacao.umimaxinc, public_vProjetoIncubacao.perinc,
public_vProjetoIncubacao.peraviinc, public_vProjetoIncubacao.temppminnas,
public_vProjetoIncubacao.temppmaxnas, public_vProjetoIncubacao.umiminnas,
public_vProjetoIncubacao.umimaxnas, public_vProjetoIncubacao.pernas,
public_vProjetoIncubacao.peravinas, public_vProjetoIncubacao.perrec,
public_vProjetoIncubacao.codtipoent, public_vProjetoIncubacao.nometipoent,
public_vProjetoIncubacao.tipoentmin, public_vProjetoIncubacao.codstatusproj,
public_vProjetoIncubacao.nomestatusproj, public_vProjetoIncubacao.codstatusin,
public_vProjetoIncubacao.datainiinc, public_vProjetoIncubacao.dataprevinc,
public_vProjetoIncubacao.datafiminc, public_vProjetoIncubacao.numovoinii,
public_vProjetoIncubacao.numovodesi, public_vProjetoIncubacao.numovotransf,
```

Columns

Name	Type	Size
codproj	: Update Query	16
siglaproj	Text	14
nomeproj	Text	100
datainiproj	Date/Time	8
dataprevproj	Date/Time	8
datafimproj	Date/Time	8
descproj	Text	254
numinc	: Update Query	16
codesp	: Update Query	16
codstatusgeral	: Update Query	16
nomestatusgeral	Text	50
nomeesp	Text	50
temppmininc	: Update Query	16
temppmaxinc	: Update Query	16
umimininc	: Update Query	16
umimaxinc	: Update Query	16
perinc	: Update Query	16
peraviinc	: Update Query	16
temppminnas	: Update Query	16
temppmaxnas	: Update Query	16
umiminnas	: Update Query	16
umimaxnas	: Update Query	16
pernas	: Update Query	16
peravinas	: Update Query	16
perrec	: Update Query	16
codtipoent	: Update Query	16
nometipoent	Text	50
tipoentmin	Text	1
codstatusproj	: Update Query	16
nomestatusproj	Text	50
codstatusin	: Update Query	16
datainiinc	Date/Time	8

ProjetoIncubacao.mdb

Query: public_vProjetoIncubacao Consulta

Page: 3

dataprevinc	Date/Time	8
datafiminc	Date/Time	8
numovoinii	: Update Query	16
numovodesi	: Update Query	16
numovotransf	: Update Query	16

ProjetoIncubacao.mdb

Database: C:\Program Files\Apache Software Foundation\Tomcat

Page: 4

Properties

AccessVersion:	08.50	ANSI Query Mode:	0
Build:	566	CollatingOrder:	General
ProjVer:	35	QueryTimeout:	60
RecordsAffected:	0	Themed Form Controls:	1
Transactions:	True	Updatable:	True
Version:	4.0		

User Permissions

admin

Group PermissionsAdmins
Users

12.3.6 ProjetoIncubacaoIncubadoraBandeja

ProjetoIncubacaoIncubadoraBandeja.mdb

Table: public_vProjetoIncubacaoIncubadoraBandeja

Page: 1

Columns

Name	Type	Size
codproj	Decimal	16
codesp	Decimal	16
siglaproj	Text	14
nomeproj	Text	100
datainiproj	Date/Time	8
dataprevproj	Date/Time	8
datafimproj	Date/Time	8
descproj	Text	254
numinc	Decimal	16
nomeesp	Text	50
tempmininc	Decimal	16
tempmaxinc	Decimal	16
umimininc	Decimal	16
umimaxinc	Decimal	16
perinc	Decimal	16
peraviinc	Decimal	16
tempminnas	Decimal	16
tempmaxnas	Decimal	16
umiminnas	Decimal	16
umimaxnas	Decimal	16
pernas	Decimal	16
peravinas	Decimal	16
perrec	Decimal	16
nometipoent	Text	50
tipoentmin	Text	1
codstatusproj	Decimal	16
nomestatusproj	Text	50
codinc	Decimal	16
datainiinc	Date/Time	8
dataprevinc	Date/Time	8
datafiminc	Date/Time	8
numovoinii	Decimal	16
numovodesi	Decimal	16

numovotransf	Decimal	16
obsstatusinc	Text	254
codstatusin	Decimal	16
nomestatusin	Text	50
nlinban	Decimal	16
ncolban	Decimal	16
ovotransf	Text	1
codmotdes	Decimal	16
motdes	Text	50
codposbanenc	Decimal	16
nposenc	Decimal	16
dataaloenc	Date/Time	8
dataretenc	Date/Time	8
codenc	Decimal	16
nomeenc	Text	100
tposenc	Decimal	16

ProjetoIncubacaoIncubadoraBandeja.mdb

Table: public_vProjetoIncubacaoIncubadoraBandeja		Page: 2
descenc	Text	254
codban	Decimal	16
tlinban	Decimal	16
tcolban	Decimal	16
nomeban	Text	50
codent	Decimal	16
codtipoent	Decimal	16
nomeent	Text	50

ProjetoIncubacaoIncubadoraBandeja.mdb

Database: C:\Program Files\Apache Software Foundation\Tomcat Page: 3

Properties

AccessVersion:	08.50	ANSI Query Mode:	0
Build:	566	CollatingOrder:	General
ProjVer:	35	QueryTimeout:	60
RecordsAffected:	0	Themed Form Controls:	1
Transactions:	True	Updatable:	True
Version:	4.0		

User Permissions

admin

Group Permissions

Admins
Users

12.3.7 ProjetoIncubacaoNascimento

ProjetoIncubacaoNascimento.mdb

Table: public_vProjetoIncubacaoNascimento Page: 1

Columns

Name	Type	Size
codproj	Decimal	16
siglaproj	Text	14
nomeproj	Text	100
datainiproj	Date/Time	8
dataprevproj	Date/Time	8
datafimproj	Date/Time	8

descproj	Text	254
numinc	Decimal	16
codesp	Decimal	16
codstatusgeral	Decimal	16
nomestatusgeral	Text	50
nomeesp	Text	50
tempmininc	Decimal	16
tempmaxinc	Decimal	16
umimininc	Decimal	16
umimaxinc	Decimal	16
perinc	Decimal	16
peraviinc	Decimal	16
tempminnas	Decimal	16
tempmaxnas	Decimal	16
umiminnas	Decimal	16
umimaxnas	Decimal	16
pernas	Decimal	16
peravinas	Decimal	16
perrec	Decimal	16
codtipoent	Decimal	16
nometipoent	Text	50
tipoentmin	Text	1
codstatusproj	Decimal	16
nomestatusproj	Text	50
codinc	Decimal	16
datainiinc	Date/Time	8
dataprevinc	Date/Time	8
datafiminc	Date/Time	8
numovoinii	Decimal	16
numovodesi	Decimal	16
numovotransf	Decimal	16
obsstatusinc	Text	254
codnas	Decimal	16
datainiinas	Date/Time	8
dataprevnas	Date/Time	8
datafimnas	Date/Time	8
numovoinin	Decimal	16
numovodesn	Decimal	16
numovorec	Decimal	16
obsstatusnas	Text	254
codstatusin	Decimal	16
nomestatusin	Text	50

ProjetoIncubacaoNascimento.mdb

Database: C:\Program Files\Apache Software Foundation\Tomcat

Page: 2

Properties

AccessVersion:	08.50	ANSI Query Mode:	0
Build:	566	CollatingOrder:	General
ProjVer:	35	QueryTimeout:	60
RecordsAffected:	0	Themed Form Controls:	1
Transactions:	True	Updatable:	True
Version:	4.0		

User Permissions

admin

Group Permissions

Admins
Users

12.3.8 SalaUF

Properties

Attributes:	Linked ODBC	Connect:	ODBC;DSN=PostgreSQL;DATABASE=sisinc;SERVER=localhost;PORT=5432;A6=;A7=100;A8=8192;B0=254;B1=8190;BI=0;C2=dd_;CX=1b503a9
DateCreated:	1/10/2006 11:20:16 PM	DefaultView:	Datasheet
GUID:	{guid {97E42839-DEB7-4EC3-84AF-3A757BC3C183}}	LastUpdated:	1/10/2006 11:20:16 PM
NameMap:	Long binary data	OrderByOn:	False
Orientation:	Left-to-Right	RecordCount:	-1
SourceTableName:	public.andar	Updatable:	False

Columns

Name	Type	Size
codand	Decimal	16
codpre	Decimal	16

Table Indexes

Name	Number of Fields
andar_pkey	1
Fields:	
codand	Ascending

Properties

Attributes:	Linked ODBC	Connect:	ODBC;DSN=PostgreSQL;DATABASE=sisinc;SERVER=localhost;PORT=5432;A6=;A7=100;A8=8192;B0=254;B1=8190;BI=0;C2=dd_;CX=1b503a9
DateCreated:	1/10/2006 11:20:16 PM	DefaultView:	Datasheet
GUID:	{guid {83F6E509-5287-4358-BCF4-1D134384DF72}}	LastUpdated:	1/10/2006 11:20:16 PM
NameMap:	Long binary data	OrderByOn:	False
Orientation:	Left-to-Right	RecordCount:	-1
SourceTableName:	public.cidade	Updatable:	False

Columns

Name	Type	Size
codcid	Decimal	16
codstatusgeral	Decimal	16
coduf	Decimal	16
nomecid	Text	100

Table Indexes

Name	Number of Fields
cidade_pkey	1
Fields:	
codcid	Ascending

Properties

Attributes: Linked ODBC Connect: ODBC;DSN=PostgreSQL;DATABASE=sisinc;SERVER=localhost;PORT=5432;A6=;A7=100;A8=8192;B0=254;B1=8190;BI=0;C2=dd_;CX=1b503a9

DateCreated: 1/10/2006 11:20:16 PM DefaultView: Datasheet
 GUID: {guid {21989E11-040C-4412-11:20:16 PM 8832-A73D3EB1B175}} LastUpdated: 1/10/2006

NameMap: Long binary data OrderByOn: False
 Orientation: Left-to-Right RecordCount: -1
 SourceTableName: public.entidade Updatable: False

Columns

Name	Type	Size
codent	Decimal	16
codstatusgeral	Decimal	16
codtipoent	Decimal	16
nomeent	Text	50
descent	Text	254

Table Indexes

Name	Number of Fields
entidade_pkey	1
Fields: codent	Ascending

SalaUF.mdb

Table: public_predio

Page: 4

Properties

Attributes: Linked ODBC Connect: ODBC;DSN=PostgreSQL;DATABASE=sisinc;SERVER=localhost;PORT=5432;A6=;A7=100;A8=8192;B0=254;B1=8190;BI=0;C2=dd_;CX=1b503a9

DateCreated: 1/10/2006 11:20:16 PM DefaultView: Datasheet
 GUID: {guid {1EA68338-9235-4EA1-11:20:16 PM 9F1B-537948D72CC0}} LastUpdated: 1/10/2006

NameMap: Long binary data OrderByOn: False
 Orientation: Left-to-Right RecordCount: -1
 SourceTableName: public.predio Updatable: False

Columns

Name	Type	Size
codpre	Decimal	16
codprop	Decimal	16

Table Indexes

Name	Number of Fields
predio_pkey	1
Fields: codpre	Ascending

SalaUF.mdb

Table: public_propriedade

Page: 5

Properties

Attributes: Linked ODBC Connect: ODBC;DSN=PostgreSQL;DA

TABASE=sisinc;SERVER=loc
 alhost;PORT=5432;A6=;A7=
 100;A8=8192;B0=254;B1=81
 90;BI=0;C2=dd_;CX=1b503a9
 Datasheet

DateCreated: 1/10/2006 11:20:16 PM DefaultView:
 GUID: {guid {ED42CE2B-1CA7-4C94-ADE7- LastUpdated: 1/10/2006 11:20:16 PM
 NameMap: Long binary data OrderByOn: False
 Orientation: Left-to-Right RecordCount: -1
 SourceTableName: public.propriedade Updatable: False

Columns

Name	Type	Size
codprop	Decimal	16
codcid	Decimal	16
endprop	Text	254

Table Indexes

Name	Number of Fields
propriedade_pkey	1
Fields: codprop	Ascending

SalaUF.mdb
 Table: public_sala

Page: 6

Properties

Attributes: Linked ODBC Connect: ODBC;DSN=PostgreSQL;DA
 TABASE=sisinc;SERVER=loc
 alhost;PORT=5432;A6=;A7=
 100;A8=8192;B0=254;B1=81
 90;BI=0;C2=dd_;CX=1b503a9
 Datasheet
 LastUpdated: 1/10/2006
 9404-DEE29793C28C}}

DateCreated: 1/10/2006 11:20:16 PM DefaultView:
 GUID: {guid {E03D777B-867E-4990-11:20:16 PM
 NameMap: Long binary data OrderByOn: False
 Orientation: Left-to-Right RecordCount: -1
 SourceTableName: public.sala Updatable: False

Columns

Name	Type	Size
codsala	Decimal	16
codand	Decimal	16

Table Indexes

Name	Number of Fields
sala_pkey	1
Fields: codsala	Ascending

SalaUF.mdb
 Table: public_uf

Page: 7

Properties

Attributes: Linked ODBC Connect: ODBC;DSN=PostgreSQL;DA
 TABASE=sisinc;SERVER=loc
 alhost;PORT=5432;A6=;A7=
 100;A8=8192;B0=254;B1=81
 90;BI=0;C2=dd_;CX=1b503a9
 Datasheet

DateCreated: 1/10/2006 11:20:16 PM DefaultView:
 GUID: {guid {E03D777B-867E-4990-11:20:16 PM
 NameMap: Long binary data OrderByOn: False
 Orientation: Left-to-Right RecordCount: -1
 SourceTableName: public.sala Updatable: False

GUID: {guid {535A47D4-AC01-4CDB-AB9A- LastUpdated: 1/10/2006 11:20:16 PM
 NameMap: Long binary data OrderByOn: False
 Orientation: Left-to-Right RecordCount: -1
 SourceTableName: public.uf Updatable: False

Columns

Name	Type	Size
coduf	Decimal	16
codstatusgeral	Decimal	16
siglauf	Text	2
nomeuf	Text	50

Table Indexes

Name	Number of Fields
uf_pkey	1
Fields:	
coduf	Ascending

SalaUF.mdb

Table: public_vEntidadeAndar

Page: 8

Properties

Attributes: Linked ODBC Connect: ODBC;DSN=PostgreSQL;DATABASE=sisinc;SERVER=localhost;PORT=5432;A6=;A7=100;A8=8192;B0=254;B1=8190;BI=0;C2=dd_;CX=1b503a9
 DateCreated: 1/10/2006 11:20:16 PM DefaultView: Datasheet
 GUID: {guid {A18B07DF-8110-4520-11:20:17 PM LastUpdated: 1/10/2006
 B693-EE963761A678}}
 NameMap: Long binary data OrderByOn: False
 Orientation: Left-to-Right RecordCount: -1
 SourceTableName: public.vEntidadeAndar Updatable: False

Columns

Name	Type	Size
codent	Decimal	16
nomeent	Text	50
descent	Text	254
codstatusgeral	Decimal	16
nomestatusgeral	Text	50
codtipoent	Decimal	16
nometipoent	Text	50
tipoentmin	Text	1
codpre	Decimal	16

SalaUF.mdb

Table: public_vEntidadeBloco

Page: 9

Properties

Attributes: Linked ODBC Connect: ODBC;DSN=PostgreSQL;DATABASE=sisinc;SERVER=localhost;PORT=5432;A6=;A7=100;A8=8192;B0=254;B1=8190;BI=0;C2=dd_;CX=1b503a9
 DateCreated: 1/10/2006 11:20:17 PM DefaultView: Datasheet
 GUID: {guid {560411CC-9C42-4FFD-AAB4-F735100E55CE}} LastUpdated: 1/10/2006 11:20:17 PM
 NameMap: Long binary data OrderByOn: False
 Orientation: Left-to-Right RecordCount: -1
 SourceTableName: public.vEntidadeBloco Updatable: False

Columns

Name	Type	Size
codent	Decimal	16
nomeent	Text	50
descent	Text	254
codstatusgeral	Decimal	16
nomestatusgeral	Text	50
codtipoent	Decimal	16
nometipoent	Text	50
tipoentmin	Text	1
codsala	Decimal	16

SalaUF.mdb

Table: public_vEntidadeCelula

Page: 10

Properties

Attributes: Linked ODBC Connect: ODBC;DSN=PostgreSQL;DATABASE=sisinc;SERVER=localhost;PORT=5432;A6=;A7=100;A8=8192;B0=254;B1=8190;BI=0;C2=dd_;CX=1b503a9

DateCreated: 1/10/2006 11:20:17 PM DefaultView: Datasheet
GUID: {guid {16FED385-71F7-475C-11:20:17 PM
11:20:17 PM 8179-32C36F9EB994}}

NameMap: Long binary data OrderByOn: False
Orientation: Left-to-Right RecordCount: -1
SourceTableName: public.vEntidadeCelula Updatable: False

Columns

Name	Type	Size
codent	Decimal	16
nomeent	Text	50
descent	Text	254
codstatusgeral	Decimal	16
nomestatusgeral	Text	50
codtipoent	Decimal	16
nometipoent	Text	50
tipoentmin	Text	1
codgai	Decimal	16

SalaUF.mdb

Table: public_vEntidadeGaiola

Page: 11

Properties

Attributes: Linked ODBC Connect: ODBC;DSN=PostgreSQL;DATABASE=sisinc;SERVER=localhost;PORT=5432;A6=;A7=100;A8=8192;B0=254;B1=8190;BI=0;C2=dd_;CX=1b503a9

DateCreated: 1/10/2006 11:20:17 PM DefaultView: Datasheet
GUID: {guid {4F931B9C-06A3-4B77-11:20:17 PM
11:20:17 PM 81AB-F7935DEC4222}}

NameMap: Long binary data OrderByOn: False
Orientation: Left-to-Right RecordCount: -1
SourceTableName: public.vEntidadeGaiola Updatable: False

Columns

Name	Type	Size
codent	Decimal	16
nomeent	Text	50
descent	Text	254
codstatusgeral	Decimal	16
nomestatusgeral	Text	50

codtipoent	Decimal	16
nometipoent	Text	50
tipoentmin	Text	1
codlinha	Decimal	16

SalaUF.mdb

Table: public_vEntidadeLinha

Page: 12

Properties

Attributes:	Linked ODBC	Connect:	ODBC;DSN=PostgreSQL;DATABASE=sisinc;SERVER=localhost;PORT=5432;A6=;A7=100;A8=8192;B0=254;B1=8190;BI=0;C2=dd_;CX=1b503a9
DateCreated:	1/10/2006 11:20:17 PM	DefaultView:	Datasheet
GUID:	{guid {8CC15959-4FA9-49C1-9BF0-FC695556668F}}	LastUpdated:	1/10/2006 11:20:17 PM
NameMap:	Long binary data	OrderByOn:	False
Orientation:	Left-to-Right	RecordCount:	-1
SourceTableName:	public.vEntidadeLinha	Updatable:	False

Columns

Name	Type	Size
codent	Decimal	16
nomeent	Text	50
descent	Text	254
codstatusgeral	Decimal	16
nomestatusgeral	Text	50
codtipoent	Decimal	16
nometipoent	Text	50
tipoentmin	Text	1
codbloco	Decimal	16

SalaUF.mdb

Table: public_vEntidadeOvo

Page: 13

Properties

Attributes:	Linked ODBC	Connect:	ODBC;DSN=PostgreSQL;DATABASE=sisinc;SERVER=localhost;PORT=5432;A6=;A7=100;A8=8192;B0=254;B1=8190;BI=0;C2=dd_;CX=1b503a9
DateCreated:	1/10/2006 11:20:17 PM	DefaultView:	Datasheet
GUID:	{guid {31C115B8-18D6-411E-B2F9-36140821B938}}	LastUpdated:	1/10/2006
NameMap:	Long binary data	OrderByOn:	False
Orientation:	Left-to-Right	RecordCount:	-1
SourceTableName:	public.vEntidadeOvo	Updatable:	False

Columns

Name	Type	Size
codent	Decimal	16
nomeent	Text	50
descent	Text	254
codstatusgeral	Decimal	16
nomestatusgeral	Text	50
codtipoent	Decimal	16
nometipoent	Text	50
tipoentmin	Text	1
codbarras	Decimal	16
coddesovo	Decimal	16
nomedesovo	Text	50
codstatusovo	Decimal	16
nomestatusovo	Text	50
codforovo	Decimal	16

nomeforovo	Text	50
codcorovo	Decimal	16
nomecorovo	Text	50
codcel	Decimal	16
aquisovo	Decimal	16
datacolovo	Date/Time	8
aspovo	Text	50
pesoovo	Decimal	16
porovo	Decimal	16
incubavel	Text	1

SalaUF.mdb

Table: public_vEntidadePredio

Page: 14

Properties

Attributes: Linked ODBC Connect: ODBC;DSN=PostgreSQL;DATABASE=sisinc;SERVER=localhost;PORT=5432;A6=;A7=100;A8=8192;B0=254;B1=8190;BI=0;C2=dd_;CX=1b503a9

DateCreated: 1/10/2006 11:20:17 PM DefaultView: Datasheet
GUID: {guid {8722C1C5-80F7-417E-11:20:17 PM
89C8-881601015BC9}} LastUpdated: 1/10/2006

NameMap: Long binary data OrderByOn: False
Orientation: Left-to-Right RecordCount: -1
SourceTableName: public.vEntidadePredio Updatable: False

Columns

Name	Type	Size
codent	Decimal	16
nomeent	Text	50
descent	Text	254
codstatusgeral	Decimal	16
nomestatusgeral	Text	50
codtipoent	Decimal	16
nometipoent	Text	50
tipoentmin	Text	1
codprop	Decimal	16

SalaUF.mdb

Table: public_vEntidadePropriedade

Page: 15

Properties

Attributes: Linked ODBC Connect: ODBC;DSN=PostgreSQL;DATABASE=sisinc;SERVER=localhost;PORT=5432;A6=;A7=100;A8=8192;B0=254;B1=8190;BI=0;C2=dd_;CX=1b503a9

DateCreated: 1/10/2006 11:20:17 PM DefaultView: Datasheet
GUID: {guid {D83558BE-8F76-48FA-8838- LastUpdated: 1/10/2006 11:20:17 PM

NameMap: Long binary data OrderByOn: False
Orientation: Left-to-Right RecordCount: -1
SourceTableName: public.vEntidadePropriedade Updatable: False

Columns

Name	Type	Size
codent	Decimal	16
nomeent	Text	50
descent	Text	254
codstatusgeral	Decimal	16
nomestatusgeral	Text	50
codtipoent	Decimal	16
nometipoent	Text	50
tipoentmin	Text	1

endprop	Text	254
codcid	Decimal	16
nomecid	Text	100
coduf	Decimal	16
siglauf	Text	2
nomeuf	Text	50

SalaUF.mdb

Table: public_vEntidadeSala

Page: 16

Properties

Attributes:	Linked ODBC	Connect:	ODBC;DSN=PostgreSQL;DATABASE=sisinc;SERVER=localhost;PORT=5432;A6=;A7=100;A8=8192;B0=254;B1=8190;BI=0;C2=dd_;CX=1b503a9
DateCreated:	1/10/2006 11:20:17 PM	DefaultView:	Datasheet
GUID:	{guid {78F6A95E-B492-4B39-11:20:17 PM	LastUpdated:	1/10/2006
	BEA8-9A7944DC1219}}		
NameMap:	Long binary data	OrderByOn:	False
Orientation:	Left-to-Right	RecordCount:	-1
SourceTableName:	public.vEntidadeSala	Updatable:	False

Columns

Name	Type	Size
codent	Decimal	16
nomeent	Text	50
descent	Text	254
codstatusgeral	Decimal	16
nomestatusgeral	Text	50
codtipoent	Decimal	16
nometipoent	Text	50
tipoentmin	Text	1
codand	Decimal	16

SalaUF.mdb

Table: public_vProjetoIncubacao

Page: 17

Properties

Attributes:	Linked ODBC	Connect:	ODBC;DSN=PostgreSQL;DATABASE=sisinc;SERVER=localhost;PORT=5432;A6=;A7=100;A8=8192;B0=254;B1=8190;BI=0;C2=dd_;CX=1b503a9
DateCreated:	1/10/2006 11:20:17 PM	DefaultView:	Datasheet
GUID:	{guid {E891AD46-E580-4BD9-A2FB-1/10/2006 11:20:17 PM	LastUpdated:	1/10/2006 11:20:17 PM
	4BD9-A2FB-		
NameMap:	Long binary data	OrderByOn:	False
Orientation:	Left-to-Right	RecordCount:	-1
SourceTableName:	public.vProjetoIncubacao	Updatable:	False

Columns

Name	Type	Size
codproj	Decimal	16
siglaproj	Text	14
nomeproj	Text	100
datainiproj	Date/Time	8
dataprevproj	Date/Time	8
datafimproj	Date/Time	8
descproj	Text	254
numinc	Decimal	16
codesp	Decimal	16
codstatusgeral	Decimal	16
nomestatusgeral	Text	50

nomeesp	Text	50
tempmininc	Decimal	16
tempmaxinc	Decimal	16
umimininc	Decimal	16
umimaxinc	Decimal	16
perinc	Decimal	16
peraviinc	Decimal	16
tempminnas	Decimal	16
tempmaxnas	Decimal	16
umiminnas	Decimal	16
umimaxnas	Decimal	16
pernas	Decimal	16
peravinas	Decimal	16
perrec	Decimal	16
codtipoent	Decimal	16
nometipoent	Text	50
tipoentmin	Text	1
codstatusproj	Decimal	16
nomestatusproj	Text	50
codstatusin	Decimal	16
datainiinc	Date/Time	8
dataprevinc	Date/Time	8
datafiminc	Date/Time	8
numovoinii	Decimal	16
numovodesi	Decimal	16

SalaUF.mdb

Table: public_vProjetoIncubacao		Page: 18
numovotransf	Decimal	16
obsstatusinc	Text	254

SalaUF.mdb

Table: public_vProjetoIncubacaoNascimento		Page: 19
---	--	----------

Properties

Attributes:	Linked ODBC	Connect:	ODBC;DSN=PostgreSQL;DATABASE=sisinc;SERVER=localhost;PORT=5432;A6=;A7=100;A8=8192;B0=254;B1=8190;BI=0;C2=dd_;CX=1b503a9
DateCreated:	1/10/2006 11:20:17 PM	DefaultView:	Datasheet
GUID:	{guid {8E9309A2-A7EA-46C6-9937-8AB6226C58FA}}	LastUpdated:	1/10/2006 11:20:17 PM
NameMap:	Long binary data	OrderByOn:	False
Orientation:	Left-to-Right	RecordCount:	-1
SourceTableName:	public.vProjetoIncubacaoNascimento	Updatable:	False

Columns

Name	Type	Size
codproj	Decimal	16
siglaproj	Text	14
nomeproj	Text	100
datainiproj	Date/Time	8
dataprevproj	Date/Time	8
datafimproj	Date/Time	8
descproj	Text	254
numinc	Decimal	16
codesp	Decimal	16
codstatusgeral	Decimal	16
nomestatusgeral	Text	50
nomeesp	Text	50
tempmininc	Decimal	16
tempmaxinc	Decimal	16
umimininc	Decimal	16
umimaxinc	Decimal	16
perinc	Decimal	16
peraviinc	Decimal	16
tempminnas	Decimal	16
tempmaxnas	Decimal	16

umiminna	Decimal	16
umimaxna	Decimal	16
pernas	Decimal	16
peravinas	Decimal	16
perrec	Decimal	16
codtipoent	Decimal	16
nometipoent	Text	50
tipoentmin	Text	1
codstatusproj	Decimal	16
nomestatusproj	Text	50
codinc	Decimal	16
datainiinc	Date/Time	8
dataprevinc	Date/Time	8
datafiminc	Date/Time	8
numovoinii	Decimal	16

SalaUF.mdb

Table: public_vProjetoIncubacaoNascimento

Page: 20

numovodesi	Decimal	16
numovotransf	Decimal	16
obsstatusinc	Text	254
codnas	Decimal	16
datainiinas	Date/Time	8
dataprevnas	Date/Time	8
datafimnas	Date/Time	8
numovoinin	Decimal	16
numovodesn	Decimal	16
numovorec	Decimal	16
obsstatusnas	Text	254
codstatusin	Decimal	16
nomestatusin	Text	50

SalaUF.mdb

Query: SalaCidade

Page: 21

SQL

```

SELECT public_vEntidadeSala.codent, public_vEntidadeSala.nomeent,
public_vEntidadeSala.descent,
public_vEntidadeSala.codstatusgeral, public_vEntidadeSala.nomestatusgeral,
public_vEntidadeSala.codtipoent, public_vEntidadeSala.nometipoent,
public_vEntidadeSala.tipoentmin,
public_vEntidadeSala.codand, public_vEntidadeAndar.nomeent,
public_vEntidadeAndar.descent,
public_vEntidadeAndar.codstatusgeral, public_vEntidadeAndar.nomestatusgeral,
public_vEntidadeAndar.codtipoent, public_vEntidadeAndar.nometipoent,
public_vEntidadeAndar.tipoentmin,
public_vEntidadeAndar.codpre, public_vEntidadePredio.nomeent,
public_vEntidadePredio.descent,
public_vEntidadePredio.codstatusgeral, public_vEntidadePredio.nomestatusgeral,
public_vEntidadePredio.codtipoent, public_vEntidadePredio.nometipoent,
public_vEntidadePredio.tipoentmin, public_vEntidadePredio.codprop,
public_vEntidadePropriedade.nomeent, public_vEntidadePropriedade.descent,
public_vEntidadePropriedade.codstatusgeral,
public_vEntidadePropriedade.nomestatusgeral,
public_vEntidadePropriedade.codtipoent, public_vEntidadePropriedade.nometipoent,
public_vEntidadePropriedade.tipoentmin, public_vEntidadePropriedade.endprop,
public_vEntidadePropriedade.codcid, public_vEntidadePropriedade.nomecid,
public_vEntidadePropriedade.coduf, public_vEntidadePropriedade.siglauf,
public_vEntidadePropriedade.nomeuf
FROM ((public_vEntidadeSala INNER JOIN public_vEntidadeAndar ON
public_vEntidadeSala.codand=public_vEntidadeAndar.codent) INNER JOIN
public_vEntidadePredio ON

```

Columns

Name	Type	Size
codent	: Update Query	16
public_vEntidadeSala.nomeent	Text	50
public_vEntidadeSala.descent	Text	254
public_vEntidadeSala.codstatusgeral	: Update Query	16
public_vEntidadeSala.nomestatusgeral	Text	50

public_vEntidadeSala.codtipoent	: Update Query	16
public_vEntidadeSala.nometipoent	Text	50
public_vEntidadeSala.tipoentmin	Text	1
codand	: Update Query	16
public_vEntidadeAndar.nomeent	Text	50
public_vEntidadeAndar.descent	Text	254
public_vEntidadeAndar.codstatusgeral	: Update Query	16
public_vEntidadeAndar.nomestatusgeral	Text	50
public_vEntidadeAndar.codtipoent	: Update Query	16
public_vEntidadeAndar.nometipoent	Text	50
public_vEntidadeAndar.tipoentmin	Text	1
codpre	: Update Query	16
public_vEntidadePredio.nomeent	Text	50
public_vEntidadePredio.descent	Text	254
public_vEntidadePredio.codstatusgeral	: Update Query	16
public_vEntidadePredio.nomestatusgeral	Text	50
public_vEntidadePredio.codtipoent	: Update Query	16
public_vEntidadePredio.nometipoent	Text	50
public_vEntidadePredio.tipoentmin	Text	1
codprop	: Update Query	16
public_vEntidadePropriedade.nomeent	Text	50
public_vEntidadePropriedade.descent	Text	254
public_vEntidadePropriedade.codstatusgeral	: Update Query	16
public_vEntidadePropriedade.nomestatusgeral	Text	50
public_vEntidadePropriedade.codtipoent	: Update Query	16
public_vEntidadePropriedade.nometipoent	Text	50

SalaUF.mdb

Query: SalaCidade		Page: 22
public_vEntidadePropriedade.tipoentmin	Text	1
endprop	Text	254
codcid	: Update Query	16
nomecid	Text	100
coduf	: Update Query	16
siglauf	Text	2
nomeuf	Text	50

SalaUF.mdb

Database: C:\Program Files\Apache Software Foundation\Tomcat Page: 23

Properties

AccessVersion:	08.50	ANSI Query Mode:	0
Build:	566	CollatingOrder:	General
ProjVer:	35	QueryTimeout:	60
RecordsAffected:	0	Themed Form Controls:	1
Transactions:	True	Updatable:	True
Version:	4.0		

User Permissions

admin

Group Permissions

Admins
Users


```

        </td>
        <td>
            <%=tipo%>
        </td>

    </tr>
    <%} %>
<% } else {%>
<tr>
    <td>
    </td>
    <td>
    </td>
</tr>
<%} %>
</table>
</body>
</html>

```

12.4.1.1.2 entcad

```

<%@ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<meta name="robots" content="noindex, nofollow" />
<meta name="description" content="Página de cadastro de entidade." />
<meta name="author" content="Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano
Padilha" />
<title>SISINC - Cadastrar Projeto</title>
<link href="estilos/principal.css" rel="stylesheet" type="text/css" />
<!-- validador do formulário -->
<script language="JavaScript" src="scripts/validator.js"></script>
<!-- posiciona na primeira linha -->
<script language="JavaScript" src="scripts/focus.js"></script>
</head>
<%
controladora.ControladoraConsulta cont = new controladora.ControladoraConsulta();
java.util.Vector filho,status,pai;
String filhoX = request.getParameter("filho");
String nomeEnt="";

if (filhoX==null)
{
    filho=cont.retornaTipoEntidade();
    status=cont.retornaStatus();
    pai=cont.retornaEntidade();
    nomeEnt="";
}
else
{
    filho=cont.retornaTipoEntidade();
    nomeEnt=(String) request.getAttribute("nomeEnt");
    status=cont.retornaStatus();
    pai=(java.util.Vector) request.getAttribute("ents");
}
%>
<body>
<form id="MainForm" name="MainForm" method="post" action="Entidade" onSubmit="return
v.exec()">
<table width="100%" border="0" cellspacing="5" cellpadding="0">
    <tr>
        <td colspan="3" class="titulo">Cadastrar Nova Entidade </td>
    </tr>
    <tr>
        <td colspan="3">&nbsp;&nbsp;&nbsp;</td>
    </tr>
    <tr>

```

```

        <td width="20%" height="43" align="left" valign="top"><strong>Tipo de
Entidade:</strong></td>
        <td width="80%" colspan="2" align="left" valign="top">
            <!-- trecho de código que carrega o campo filho-->
            <select name = "filho" >
                <%
                String aux = "";
                int index = 0;
                for(int i = 0 ; i < filho.size() ; i++)
                {
                    aux = (String) filho.elementAt(i);
                    index = aux.indexOf('-');
                    <%
                    <if(filhoX!=null){%>
                        <if(aux.substring(0,index-1).compareTo(filhoX)==0){%>
                            <option value="<%=aux.substring(0,index-
selected><%=aux.substring(index+1)%></option>
                            <%}else{%>
                                <option value="<%=aux.substring(0,index-
1)%>"><%=aux.substring(index+1)%></option>
                                <%}%>
                            <%}else{%>
                                <option value="<%=aux.substring(0,index-
1)%>"><%=aux.substring(index+1)%></option>
                                <%}%>
                            <%}%>
                        </select>
                        <!-- acaba o select -->
                        <!-- efetua pesquisa por entidades pai -->
                        <input name="Acao" type="submit" class="rodape" id="Acao" value="Pesquisar">
                        <br /><span class="atencao">ATEN&Ccedil;&Atilde;O:</span> <span
class="descricao">Selecione o tipo de Entidade e clique em Pesquisar. </span></td>
                    </tr>
                    <tr>
                        <td align="left" valign="top"><em><strong>que est&aacute; em:</strong></em></td>
                        <td colspan="2" align="left" valign="top">
                            <select name = "pai">
                                <%
                                String aux1 = "";
                                int index1 = 0;
                                for(int i = 0 ; i < pai.size() ; i++)
                                {
                                    aux1 = (String) pai.elementAt(i);
                                    index1 = aux.indexOf('-');
                                    <%
                                    <option value="<%=aux1.substring(0,index1)%>"><%=aux1.substring(index1+2)%></option>
                                    <%}%>
                                </select>
                                </td>
                            </tr>
                            <tr>
                                <td align="left" valign="top"><strong>Nome da Entidade:</strong></td>
                                <td colspan="2" align="left" valign="top">
                                    <input name="NomeEnt" type="text" size="50" maxlength="50" <if(pai!=null)%>
value="<%=nomeEnt%" /></td>
                                </tr>
                                <tr>
                                    <td align="left" valign="top">Descri&ccedil;&atilde;o:</td>
                                    <td colspan="2" align="left" valign="top">
                                        <textarea name="DescEnt" cols="50" rows="2" id="DescEnt"></textarea>
                                    </td>
                                </tr>
                                <tr>
                                    <td colspan="3" align="left" valign="top"><hr align="center" width="100%" size="1"
noshade="noshade" /></td>
                                </tr>
                            </table>
                            <table width="100%" border="0" cellspacing="0" cellpadding="0">
                                <tr>
                                    <td><input name="Acao" type="submit" class="rodape" id="Acao" value="Gravar" /></td>
                                    <td><input name="Acao" type="reset" class="rodape" id="Acao" value="Limpar" /></td>
                                    <td>&nbsp;&nbsp;&nbsp;</td>
                                </tr>
                            </table>
                        </form>

```



```

        String codigo=aux.substring(0,index-1);
        String tipo=aux.substring(index+1,index1-1);
        String nome=aux.substring(index1+1,index2-1);
        String pai=aux.substring(index2+1);
    %>

    <tr>
        <td hight="26">
            <%=nome%>
        </td>
        <td>
            <%=tipo%>
        </td>
        <td>
            <%=pai%>
        </td>

    </tr>
    <%> %>
<% } else {%>
<tr>
    <td>
</td>
<td>
</td>
</tr>
<%> %>
</table>
</body>
</html>

```

12.4.1.1.4 espcad

```

<%@ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>SISINC - Cadastrar Esp&eacute;cie de Ave</title>
<link href="estilos/principal.css" rel="stylesheet" type="text/css" />
<!-- verifica se suporta javascript
<script>function lastmod() {}</script>
<!-- validador do formulário
<script language="javascript" type="text/javascript" src="scripts\validator.js"></script>
<!-- caminho botão adicional -->
<script language="javascript" type="text/javascript">
function IrPara() {parent.location='../..//EntidadeOvoAlterar';}
</script>
</head>
<%
controladora.ControladoraConsulta cont=new controladora.ControladoraConsulta();

java.util.Vector CodDesOvo,CodStatusOvo,CodForOvo,CodCorOvo,CodCel,AquisOvo;
CodDesOvo=cont.retornaDesOvo();
CodStatusOvo=cont.retornaStatusOvo();
CodForOvo=cont.retornaForOvo();
CodCorOvo=cont.retornaCorOvo();
CodCel=cont.retornaCelula();
AquisOvo=cont.retornaPropriedade();
%>
<body>
<form id="MainForm" name="MainForm" method="post" action="EspecieAve">
<table width="100%" border="0" cellspacing="5" cellpadding="0">
    <tr>
        <td colspan="3" class="titulo">Cadastrar Nova Esp&eacute;cie de Ave </td>
    </tr>
    <tr>
        <td colspan="3">&nbsp;</td>
    </tr>

```

```

<tr>
  <td width="20%" align="left" valign="top"><strong>Nome da Esp&eacute;cie:</strong></td>
  <td width="80%" colspan="2" align="left" valign="top">
    <input name="NomeEsp" type="text" id="NomeEsp" size="50" maxlength="50" />
    <br /></td>
</tr>
<tr>
  <td align="left" valign="top">Temp. Min. Incuba&ccedil;&atilde;o: </td>
  <td colspan="2" align="left" valign="top"><input name="TempMinInc" type="text"
id="TempMinInc" size="7" maxlength="7" />
  <span class="descricao">Celsius</span></td>
</tr>
<tr>
  <td height="22" align="left" valign="top">Temp. M&aacute;x. Incuba&ccedil;&atilde;o: </td>
  <td colspan="2" align="left" valign="top"><input name="TempMaxInc" type="text"
id="TempMaxInc" size="7" maxlength="7" />
  <span class="descricao">Celsius</span></td>
</tr>
<tr>
  <td height="22" align="left" valign="top">Umi. Min. Incuba&ccedil;&atilde;o: </td>
  <td colspan="2" align="left" valign="top"><input name="UmiMinInc" type="text"
id="UmiMinInc" size="7" maxlength="7" />
  <span class="descricao">?</span></td>
</tr>
<tr>
  <td height="22" align="left" valign="top">Umi. M&aacute;x. Incuba&ccedil;&atilde;o:</td>
  <td colspan="2" align="left" valign="top"><input name="UmiMaxInc" type="text"
id="UmiMaxInc" size="7" maxlength="7" />
  <span class="descricao">?</span></td>
</tr>
<tr>
  <td height="38" align="left" valign="top"><strong>Per&iacute;odo de
Incuba&ccedil;&atilde;o: </strong></td>
  <td colspan="2" align="left" valign="top">
    <input name="PerInc" type="text" id="PerInc" size="3" maxlength="3" />
    <span
class="descricao">dias.<br />
    <br />
</span></td>
</tr>
<tr>
  <td height="22" align="left" valign="top"><strong>Per&iacute;odo Aviso Inc.:
</strong></td>
  <td colspan="2" align="left" valign="top"><input name="PerAviInc" type="text"
id="PerAviInc" size="3" maxlength="2" />
  <span class="descricao">dias.<br />
  <span class="atencao">NOTA:</span> Per&iacute;odo antes de completar para enviar
aviso.</span></td>
</tr>
<tr>
  <td align="left" valign="top">Temp. Min. Nascimento: </td>
  <td colspan="2" align="left" valign="top"><input name="TempMinNas" type="text"
id="TempMinNas" size="7" maxlength="7" /> <span class="descricao">Celsius</span></td>
</tr>
<tr>
  <td align="left" valign="top">Temp. M&aacute;x. Nascimento: </td>
  <td colspan="2" align="left" valign="top"><input name="TempMaxNas" type="text"
id="TempMaxNas" size="7" maxlength="7" />
  <span class="descricao">Celsius </span></td>
</tr>
<tr>
  <td align="left" valign="top">Umi. Min. Nascimento: </td>
  <td colspan="2" align="left" valign="top"><input name="UmiMinNas" type="text"
id="UmiMinNas" size="7" maxlength="7" />
  <span class="descricao">?</span></td>
</tr>
<tr>
  <td align="left" valign="top">Umi. M&aacute;x. Nascimento:</td>
  <td colspan="2" align="left" valign="top"><input name="TempMaxNas" type="text"
id="TempMaxNas" size="7" maxlength="7" />
  <span class="descricao">?</span></td>
</tr>
<tr>
  <td align="left" valign="top"><strong>Per&iacute;odo de Nascimento: </strong></td>
  <td colspan="2" align="left" valign="top"><label>
    <input name="PerNas" type="text" id="PerNas" size="3" maxlength="3" />
  </td>
</tr>

```

```

        <span class="descricao">dias.</span></label></td>
    </tr>
    <tr>
        <td align="left" valign="top"><strong>Per&iacute;odo Aviso Nas.: </strong></td>
        <td colspan="2" align="left" valign="top"><input name="PerAviNas" type="text"
id="PerAviNas" size="3" maxlength="2" />
        <span class="descricao">dias.<br />
        <span class="atencao">NOTA:</span> Per&iacute;odo antes de completar para enviar
aviso.</span></td>
    </tr>
    <tr>
        <td align="left" valign="top"><strong>Per&iacute;odo em Recria:</strong></td>
        <td colspan="2" align="left" valign="top"><input name="PerRec" type="text" id="PerRec"
size="3" maxlength="3" />
        <span class="descricao">dias.</span></td>
    </tr>
    <tr>
        <td align="left" valign="top">&nbsp;</td>
        <td colspan="2" align="left" valign="top">&nbsp;</td>
    </tr>
    <tr>
        <td colspan="3" align="left" valign="top">
        <hr align="center" width="100%" size="1" noshade="noshade" /> </td>
    </tr>
</table>
<table width="100%" border="0" cellspacing="0" cellpadding="0">
    <tr>
        <td>
            <input name="Acao" type="submit" class="rodape" id="Acao" onclick="return v.exec()"
value="Gravar" />
        </td>
        <td>
            <input name="Limpar" type="reset" class="rodape" id="Limpar" value="Limpar" />
        </td>
        <td>
            <input name="IrPara" type="button" id="IrPara" onclick="IrPara()"
value="Alterar/Excluir" />
        </td>
    </tr>
</table>
</form>
<jsp:include page="../../include/cadastro.jsp" flush="true"/>
</body>
</html>

```

12.4.1.1.5 ovocad

```

<%@ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>SISINC - Cadastrar Ovo</title>
<link href="estilos/principal.css" rel="stylesheet" type="text/css" />
<!-- verifica se suporta javascript
<script>function lastmod() {}</script>
<!-- validador do formulário
<script language="javascript" type="text/javascript" src="scripts/validator.js"></script>
<!-- caminho botão adicional -->
<script language="javascript" type="text/javascript">
function IrPara() {parent.location='../..EntidadeOvoAlterar';}
</script>
</head>
<%
controladora.ControladoraConsulta cont=new controladora.ControladoraConsulta();

java.util.Vector CodDesOvo,CodStatusOvo,CodForOvo,CodCorOvo,CodCel,AquisOvo;
CodDesOvo=cont.retornaDesOvo();
CodStatusOvo=cont.retornaStatusOvo();
CodForOvo=cont.retornaForOvo();

```

```

CodCorOvo=cont.retornaCorOvo();
CodCel=cont.retornaCelula();
AquisOvo=cont.retornaPropriedade();
%>
<body>
<form id="MainForm" name="MainForm" method="post" action="EntidadeOvo">
<table width="100%" border="0" cellspacing="5" cellpadding="0">
<tr>
<td colspan="3" class="titulo">Cadastrar Nova Entidade: Ovo </td>
</tr>
<tr>
<td colspan="3">&nbsp;&nbsp;&nbsp;</td>
</tr>
<tr>
<td width="20%" align="left" valign="top"><strong>C&oacute;digo de Barras:</strong></td>
<td width="80%" colspan="2" align="left" valign="top">
<input name="CodBarras" type="text" id="CodBarras" size="14" maxlength="14" />
<br />
<span class="descricao"><span class="atencao">NOTA:</span> O c&oacute;digo deste
campo &eacute; gerado automaticamente se for deixado em branco.</span> </td>
</tr>
<tr>
<td align="left" valign="top"><strong>C&eacute;lula de Origem: </strong></td>
<td colspan="2" align="left" valign="top"><select name="CodCel" id="CodCel">
<%
String aux = "";
int index = 0;
for(int i = 0 ; i < CodCel.size() ; i++)
{
aux = (String) CodCel.elementAt(i);
index = aux.indexOf('-');
%>
<option value="<%=aux.substring(0,index-1)%"><%=aux.substring(index+1)%"></option>
<}%%>
</select></td>
</tr>
<tr>
<td height="22" align="left" valign="top"><strong>Cor do Ovo:</strong></td>
<td colspan="2" align="left" valign="top"><select name="CodCorOvo" id="CodCorOvo">
<%
aux = "";
index = 0;
for(int i = 0 ; i < CodCorOvo.size() ; i++)
{
aux = (String) CodCorOvo.elementAt(i);
index = aux.indexOf('-');
%>
<option value="<%=aux.substring(0,index-1)%"><%=aux.substring(index+1)%"></option>
<}%%>
</select></td>
</tr>
<tr>
<td height="22" align="left" valign="top"><strong>Formato do Ovo:</strong></td>
<td colspan="2" align="left" valign="top">
<select name="CodForOvo" id="CodForOvo">
<%
aux = "";
index = 0;
for(int i = 0 ; i < CodForOvo.size() ; i++)
{
aux = (String) CodForOvo.elementAt(i);
index = aux.indexOf('-');
%>
<option value="<%=aux.substring(0,index-1)%"><%=aux.substring(index+1)%"></option>
<}%%>
</select> </td>
</tr>
<tr>
<td height="22" align="left" valign="top"><strong>Tipo de
Desinfec&ccedil;&atilde;o:</strong></td>
<td colspan="2" align="left" valign="top"><select name="CodDesOvo" id="CodDesOvo">
<%
aux = "";
index = 0;
for(int i = 0 ; i < CodDesOvo.size() ; i++)
{

```

```

                aux = (String) CodDesOvo.elementAt(i);
                index = aux.indexOf('-');
            %>
            <option value="<%=aux.substring(0,index-1)%>"><%=aux.substring(index+1)%></option>
            <%}%>
        </select></td>
    </tr>

    <tr>
        <td align="left" valign="top"><strong>Local de Aquisi&ccedil;&atilde;o: </strong></td>
        <td colspan="2" align="left" valign="top"><select name="AquisOvo" id="AquisOvo">
            <%
                aux = "";
                index = 0;
                for(int i = 0 ; i < AquisOvo.size() ; i++)
                {
                    aux = (String) AquisOvo.elementAt(i);
                    index = aux.indexOf('-');
                }
            %>
            <option value="<%=aux.substring(0,index-1)%>"><%=aux.substring(index+1)%></option>
            <%}%>
        </select>
        <br />
        <span class="descricao"><span class="atencao">ATEN&Ccedil;&Atilde;O:</span> Se o ovo
n&atilde;o foi comprado, selecione sua pr&ocute;pria Propriedade.</span> </td>
    </tr>
    <tr>
        <td align="left" valign="top"><strong>Data da Coleta: </strong>
        <td colspan="2" align="left" valign="top"><input name="DataColOvo" type="text"
id="DataColOvo" size="10" maxlength="10" /></td>
    </tr>
    <tr>
        <td align="left" valign="top"><strong>Aspecto:</strong>
        <td colspan="2" align="left" valign="top"><span class="descricao">
            <input name="AspOvo" type="text" id="AspOvo" size="50" maxlength="50" />
        </span></td>
    </tr>
    <tr>
        <td align="left" valign="top"><strong>Peso:</strong>
        <td colspan="2" align="left" valign="top"><input name="PesoOvo" type="text" id="PesoOvo"
size="6" maxlength="6" />
        <span class="descricao">gr. </span></td>
    </tr>
    <tr>
        <td align="left" valign="top"><strong>Porosidade:</strong>
        <td colspan="2" align="left" valign="top"><input name="PorOvo" type="text" id="PorOvo"
size="10" maxlength="10" />
        <span class="descricao">un.</span></td>
    </tr>
    <tr>
        <td align="left" valign="top"><strong>Incub&aacute;vel:</strong></td>
        <td colspan="2" align="left" valign="top">
            <input name="Incubavel" type="text" id="Incubavel" size="2" maxlength="1" />
            <span class="atencao">(S/N) </span> </td>
    </tr>
    <tr>
        <td align="left" valign="top"><strong>Descri&ccedil;&atilde;o:</strong>
        <td colspan="2" align="left" valign="top"><label>
            <textarea name="DescEnt" cols="38" rows="2" id="DescEnt"></textarea>
        </label></td>
    </tr>
    <tr>
        <td align="left" valign="top">&nbsp;</td>
        <td colspan="2" align="left" valign="top">&nbsp;</td>
    </tr>
    <tr>
        <td colspan="3" align="left" valign="top">
            <hr align="center" width="100%" size="1" noshade="noshade" /> </td>
    </tr>
</table>
<table width="100%" border="0" cellspacing="0" cellpadding="0">
    <tr>
        <td>
            <input name="Acao" type="submit" class="rodape" id="Acao" onclick="return v.exec()"
value="Gravar" />
        </td>
    </tr>

```

```

        <td>
            <input name="Limpar" type="reset" class="rodape" id="Limpar" value="Limpar" />
        </td>
        <td>
            <input name="IrPara" type="button" id="IrPara" onclick="IrPara()"
            value="Alterar/Excluir" />
        </td>
    </tr>
</table>
</form>
<jsp:include page="../include/cadastro.jsp" flush="true"/>
</body>
</html>

```

12.4.1.2 gerenciamento

12.4.1.2.1 usualt

```

<%@ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<meta name="robots" content="noindex, nofollow" />
<meta name="description" content="Página de alteração/exclusão de usuário." />
<meta name="author" content="Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano
Padilha" />
<title>SISINC - Alterar/Excluir Usuário</title>
<link href="estilos/principal.css" rel="stylesheet" type="text/css" />
<!-- validador do formulário -->
<script language="JavaScript" src="scripts/validator.js"></script>
<!-- limita entrada do formulário -->
<script language="JavaScript" src="scripts/form.js"></script>
<!-- posiciona na primeira linha -->
<script language="JavaScript" src="scripts/focus.js"></script>
<!-- validador da senha -->
<script language="JavaScript" src="scripts/valpass.js"></script>
</head>
<%
controladora.ControladoraConsulta cont = new controladora.ControladoraConsulta();
classepersistente.Usuario auxUsuario =
(classepersistente.Usuario)request.getAttribute("dados");

String MatrUsu=auxUsuario.getMatricula();
String NomeUsu=auxUsuario.getNome();
String CodGru=auxUsuario.getGrupo();
String SenCrt=auxUsuario.getSenha();
String TelRes=auxUsuario.getTelres();
String TelCom=auxUsuario.getTelcom();
String TelCel=auxUsuario.getTelcel();
String Email=auxUsuario.getEmail();
String CodStatusUsu=auxUsuario.getStatus();

// acerto de nulos
if (TelRes==null){TelRes="";}
if (TelCom==null){TelCom="";}
if (TelCel==null){TelCel="";}

java.util.Vector vCodGru,vCodStatusUsu;
vCodGru=cont.retornaGrupo();
vCodStatusUsu=cont.retornaStatusUsu();
%>
<body OnLoad="placeFocus(); disableFirst()">
<form id="MainForm" name="MainForm" method="post" action="UsuarioAlterar" onSubmit="return
v.exec(); return validatePwd('SenCrt');">

```

```

<table width="100%" border="0" cellspacing="5" cellpadding="0">
  <tr>
    <td colspan="3" class="titulo">Alterar/Excluir Usuário </td>
  </tr>
  <tr>
    <td colspan="3">&nbsp;&nbsp;&nbsp;</td>
  </tr>
  <tr>
    <td width="20%" align="left" valign="top"><strong>Matrícula:</strong></td>
    <td width="80%" colspan="2" align="left" valign="top">
      <input name="MatrUsul" type="text" class="atform" id="MatrUsul" size="8" maxlength="8"
onKeyUp="doCheckInteger(MainForm.MatrUsul)" value="<%=MatrUsul%>" />
      <input name="MatrUsu" type="hidden" id="MatrUsu" size="8" value="<%=MatrUsul%>" />
      <span class="descricao"> (ex.: 20060101)</span>
    </td>
  </tr>
  <tr>
    <td align="left" valign="top"><strong>Nome Completo:</strong></td>
    <td colspan="2" align="left" valign="top">
      <input name="NomeUsul" type="text" id="NomeUsul" size="50" maxlength="100"
onKeyUp="doCheckAlpha(MainForm.NomeUsul)" value="<%=NomeUsul%>" />
    </td>
  </tr>
  <tr>
    <td height="22" align="left" valign="top"><strong>Grupo de Acesso:</strong></td>
    <td colspan="2" align="left" valign="top">
      <select name="CodGru" id="CodGru">
        <%
String aux = "";
int index = 0;
for(int i = 0 ; i < vCodGru.size() ; i++)
{
  aux = (String) vCodGru.elementAt(i);
  index = aux.indexOf('-');
}
<%
<if(aux.substring(0,index-1).compareTo(CodGru)==0) { %>
  <option value="<%=aux.substring(0,index-1)%>"
selected><%=aux.substring(index+1)%></option>
<else{%>
  <option value="<%=aux.substring(0,index-1)%>"><%=aux.substring(index+1)%></option>
<}%>
<}%>
</select>
</td>
  </tr>
  <tr>
    <td height="22" align="left" valign="top"><strong>Senha:</strong></td>
    <td colspan="2" align="left" valign="top">
      <input name="SenCrt" type="password" id="SenCrt" size="8" />
      <br />
      <span class="descricao"><span class="atencao">ATENÇÃO:</span> Senha deve
ter entre 6 e 8 caracteres.</span>
    </td>
  </tr>
  <tr>
    <td align="left" valign="top"><strong>Confirmar Senha: </strong></td>
    <td colspan="2" align="left" valign="top">
      <input name="SenCrt1" type="password" id="SenCrt1" size="8" />
      <br />
      <span class="descricao"><span class="atencao">ATENÇÃO:</span> Favor
digitar novamente a Senha.</span>
    </td>
  </tr>
  <tr>
    <td align="left" valign="top">Telefone Residencial: </td>
    <td colspan="2" align="left" valign="top">
      <input name="TelRes" type="text" id="TelRes" size="10" maxlength="10"
onKeyUp="doCheckInteger(MainForm.TelRes)" value="<%=TelRes%>" />
      <span class="descricao"> (ex.: XX12345678)</span>
    </td>
  </tr>
  <tr>
    <td align="left" valign="top">Telefone Comercial:</td>
    <td colspan="2" align="left" valign="top">
      <input name="TelCom" type="text" id="TelCom" size="10" maxlength="10"
onKeyUp="doCheckInteger(MainForm.TelCom)" value="<%=TelCom%>" />
      <span class="descricao"> (ex.: XX12345678)</span>
    </td>
  </tr>

```

```

        </td>
    </tr>
    <tr>
        <td align="left" valign="top">Telefone Celular:</td>
        <td colspan="2" align="left" valign="top">
            <input name="TelCel" type="text" id="TelCel" size="10" maxlength="10"
onKeyUp="doCheckInteger(MainForm.TelCel)" value="<%=TelCel%>" />
            <span class="descricao"> (ex.: XX12345678)</span>
        </td>
    </tr>
    <tr>
        <td align="left" valign="top"><strong>E-mail:</strong></td>
        <td colspan="2" align="left" valign="top">
            <input name="Email" type="text" id="Email" size="50" maxlength="50"
value="<%=Email%>" />
        </td>
    </tr>
    <tr>
        <td align="left" valign="top"><em>STATUS</em></td>
        <td colspan="2" align="left" valign="top">
            <select name="CodStatusUsu" id="CodStatusUsu">
                <%
                aux = "";
                index = 0;
                for(int i = 0 ; i < vCodStatusUsu.size() ; i++)
                {
                    aux = (String) vCodStatusUsu.elementAt(i);
                    index = aux.indexOf('-');
                }
                <%
                if(aux.substring(0,index-1).compareTo(CodStatusUsu)==0) { %>
                <option value="<%=aux.substring(0,index-1)%>"
selected<%=aux.substring(index+1)%></option>
                <%
                }else{%>
                <option value="<%=aux.substring(0,index-1)%>"><%=aux.substring(index+1)%></option>
                <%}%>
                <%}%>
            </select>
        </td>
    </tr>
    <tr>
        <td colspan="3" align="left" valign="top"><hr align="center" width="100%" size="1"
noshade="noshade" /></td>
    </tr>
</table>
<table width="100%" border="0" cellspacing="0" cellpadding="0">
    <tr>
        <td>
            <input name="Acao" type="submit" class="rodape" id="Acao" value="Alterar" />
        </td>
        <td>
            <input name="Acao" type="button" class="rodape" id="Acao" value="Excluir"
onClick="Delete()" />
        </td>
        <td>
            <input name="Voltar" type="reset" class="rodape" id="Voltar" value="Voltar"
onClick="history.back(1)" />
        </td>
        <td>
            <input name="IrPara" type="button" id="IrPara" value="Cadastrar Novo"
OnClick="window.open('Usuario','_self')" />
        </td>
    </tr>
</table>
</form>
<!-- chamada da página de rodapé -->
<jsp:include page="../include/gerenciamento.jsp" flush="true"/>
<!-- validação do formulário -->
<script language="JavaScript">
// descrição da estrutura de campos de formulário
var a_fields = {
    'MatrUsu': {'l': 'Matrícula', 'r': true, 'f': 'integer', 't': 'MatrUsu', 'mn': 6},
    'NomeUsu': {'l': 'Nome Completo', 'r': true, 'f': 'alpha', 't': 'NomeUsu'},
    'CodGru': {'l': 'Grupo de Acesso', 'r': true, 'f': 'integer', 't': 'CodGru'},
    'SenCrt': {'l': 'Senha', 'r': true, 'f': 'alphanumeric', 't': 'SenCrt', 'mn': 6, 'mx': 14},

```



```

</tr>
<tr>
  <td height="22" align="left" valign="top"><strong>Grupo de Acesso:</strong></td>
  <td colspan="2" align="left" valign="top">
    <select name="CodGru" id="CodGru">
      <%
        String aux = "";
        int index = 0;
        for(int i = 0 ; i < CodGru.size() ; i++)
        {
          aux = (String) CodGru.elementAt(i);
          index = aux.indexOf('-');
        }
      <%
        <option value="<%=aux.substring(0,index-1)%"><%=aux.substring(index+1)%></option>
      <%>
    </select>
  </td>
</tr>
<tr>
  <td height="22" align="left" valign="top"><strong>Senha:</strong></td>
  <td colspan="2" align="left" valign="top">
    <input name="SenCrt" type="password" id="SenCrt" size="8" />
    <br />
    <span class="descricao"><span class="atencao">ATEN&Ccedil;&Atilde;O:</span> Senha deve
ter entre 6 e 8 caracteres.</span> </td>
</tr>
<tr>
  <td align="left" valign="top"><strong>Confirmar Senha: </strong></td>
  <td colspan="2" align="left" valign="top">
    <input name="SenCrt1" type="password" id="SenCrt1" size="8" />
    <br />
    <span class="descricao"><span class="atencao">ATEN&Ccedil;&Atilde;O:</span> Favor
digitar novamente a Senha.</span>
  </td>
</tr>
<tr>
  <td align="left" valign="top">Telefone Residencial: </td>
  <td colspan="2" align="left" valign="top">
    <input name="TelRes" type="text" id="TelRes" size="10" maxlength="10"
onKeyUp="doCheckInteger(MainForm.TelRes)" />
    <span class="descricao"> (ex.: XX12345678)</span>
  </td>
</tr>
<tr>
  <td align="left" valign="top">Telefone Comercial: </td>
  <td colspan="2" align="left" valign="top">
    <input name="TelCom" type="text" id="TelCom" size="10" maxlength="10"
onKeyUp="doCheckInteger(MainForm.TelCom)" />
    <span class="descricao"> (ex.: XX12345678)</span>
  </td>
</tr>
<tr>
  <td align="left" valign="top">Telefone Celular: </td>
  <td colspan="2" align="left" valign="top">
    <input name="TelCel" type="text" id="TelCel" size="10" maxlength="10"
onKeyUp="doCheckInteger(MainForm.TelCel)" />
    <span class="descricao"> (ex.: XX12345678)</span>
  </td>
</tr>
<tr>
  <td align="left" valign="top"><strong>E-mail:</strong></td>
  <td colspan="2" align="left" valign="top">
    <input name="Email" type="text" id="Email" size="50" maxlength="50" />
  </td>
</tr>
<tr>
  <td align="left" valign="top">&nbsp;</td>
  <td colspan="2" align="left" valign="top">&nbsp;</td>
</tr>
<tr>
  <td colspan="3" align="left" valign="top">
    <hr align="center" width="100%" size="1" noshade="noshade" />
  </td>
</tr>
</table>
<table width="100%" border="0" cellspacing="0" cellpadding="0">

```

```

<tr>
  <td>
    <input name="Acao" type="submit" class="rodape" id="Acao" value="Gravar" />
  </td>
  <td>
    <input name="Limpar" type="reset" class="rodape" id="Limpar" value="Limpar" />
  </td>
  <td>
    <input name="IrPara" type="button" id="IrPara" value="Alterar/Excluir"
onClick="window.open('UsuarioAlterar','_self')" />
  </td>
</tr>
</table>
</form>
<!-- chamada da página de rodapé -->
<jsp:include page="../../include/gerenciamento.jsp" flush="true"/>
<!-- validação do formulário -->
<script language="JavaScript">
// descrição da estrutura de campos para o validador
var a_fields = {
  'MatrUsu':{'l':'Matrícula','r':true,'f':'integer','t':'MatrUsu','mn':6},
  'NomeUsu':{'l':'Nome Completo','r':true,'f':'alpha','t':'NomeUsu'},
  'CodGru':{'l':'Grupo de Acesso','r':true,'f':'integer','t':'CodGru'},
  'SenCrt':{'l':'Senha','r':true,'t':'SenCrt','mn':6,'mx':14},
  'SenCrt1':{'l':'Confirmação da
Senha','r':true,'t':'SenCrt1','m':'SenCrt','mn':6,'mx':14},
  'TelRes':{'l':'Telefone Residencial','r':false,'f':'integer','t':'TelRes','mn':7},
  'TelCom':{'l':'Telefone Comercial','r':false,'f':'integer','t':'TelCom','mn':7},
  'TelCel':{'l':'Telefone Celular','r':false,'f':'integer','t':'TelCel','mn':7},
  'Email':{'l':'E-mail','r':true,'f':'email','t':'Email','mn':6}
},
o_config = {
  'to_disable':['Gravar','Limpar'],
  'alert':1
}
// chamada do construtor do validador
var v = new validator('MainForm', a_fields, o_config);
</script>
</body>
</html>

```

12.4.1.2.3 usucon

```

<%@ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<meta name="robots" content="noindex, nofollow" />
<meta name="description" content="Página de consulta de usuário." />
<meta name="author" content="Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano
Padilha" />
<title>SISINC - Consultar Usuário</title>
<link href="estilos/principal.css" rel="stylesheet" type="text/css" />
<!-- validador do formulário -->
<script language="JavaScript" src="scripts/validator.js"></script>
<!-- limita entrada do formulário -->
<script language="JavaScript" src="scripts/form.js"></script>
<!-- posiciona na primeira linha -->
<script language="JavaScript" src="scripts/focus.js"></script>
</head>
<%@ page import="controladora.ControladoraUsuario" %>
<%@ page import="servlet.UsuarioAlterar" %>
<%
String matricula = (String) request.getAttribute("matricula");
String nome = (String) request.getAttribute("nome");
String status = (String) request.getAttribute("status");

controladora.ControladoraConsulta cont = new controladora.ControladoraConsulta();
// monta a lista Status

```

```

java.util.Vector vCodStatusUsu;
vCodStatusUsu=cont.retornaStatusUsu();
%>
<body OnLoad="placeFocus()">
<form id="MainForm" name="MainForm" method="post" action="UsuarioAlterar" onSubmit="return
v.exec()">
<table width="600" border="0" cellspacing="5" cellpadding="0">
<tr>
<td colspan="2">
<span class="titulo">Alterar/Excluir Usuário </span> </td>
</tr>
<tr>
<td colspan="2">&nbsp;</td>
</tr>
<tr>
<td width="15%">Matrícula:</td>
<td width="465%">
<input name="MatrUsu" type="text" id="MatrUsu" size="8" maxlength="8"
onKeyUp="doCheckInteger(MainForm.MatrUsu)" <%if(matricula!=null){%> value="<%=matricula%"
<%}%>/>
<span class="descricao"> (ex.: 20060101)</span>
</td>
</tr>
<tr>
<td>Nome:</td>
<td>
<input name="NomeUsu" type="text" id="NomeUsu" size="50" maxlength="100"
onKeyUp="doCheckAlpha(MainForm.NomeUsu)" <%if(nome!=null){%> value="<%=nome%" <%}%>/>
</td>
</tr>
<tr>
<td><em>STATUS</em></td>
<td>
<select name="CodStatusUsu" id="CodStatusUsu">
<%
String aux = "";
int index = 0;
for(int i = 0 ; i < vCodStatusUsu.size() ; i++)
{
aux = (String) vCodStatusUsu.elementAt(i);
index = aux.indexOf('-');
%>
<%if((status!=null) && (aux.substring(0,index-1).compareTo(status)==0)) { %>
<option value="<%=aux.substring(0,index-1)%">
selected><%=aux.substring(index+1)%></option>
<%}else{%>
<option value="<%=aux.substring(0,index-1)%"><%=aux.substring(index+1)%></option>
<%}%>
<%}%>
</select>
</td>
</tr>
<tr>
<td colspan="2">
<hr align="center" width="100%" size="1" noshade="noshade" />
</td>
</tr>
<tr>
<td colspan="2"><table width="100%" border="0" cellspacing="0" cellpadding="0">
<tr>
<td>
<input name="Acao" type="submit" class="rodape" id="Acao" value="Consultar"
/>
</td>
<td>
<input name="Limpar" type="reset" class="rodape" id="Limpar" value="Limpar"
/>
</td>
<td align="right">
<input name="IrPara" type="button" id="IrPara" value="Cadastrar Novo"
OnClick="window.open('Usuario','_self')" />
</td>
</tr>
<tr>
<td colspan="3">&nbsp;</td>
</tr>

```

```

        </table></td>
    </tr>
</table>
</form>
<table width="600" border="1" cellpadding="0" cellspacing="0" bordercolor="#000000">
    <tr>
        <td width="15%"><div align="center"><strong>Matrícula</strong></div></td>
        <td width="85%"><div align="center"><strong>Nome</strong></div></td>
    </tr>
    <%
java.util.Vector tabela = new java.util.Vector();
tabela=(java.util.Vector)request.getAttribute("tabela");
%>
<%if((tabela!=null) && (tabela.size()!=0)) {%>
    <%
        aux = "";
        index = 0;
        int index2 = 0;
        for(int i = 0; i < tabela.size(); i++)
        {
            aux = (String) tabela.elementAt(i);
            index = aux.indexOf('-');
            index2 = aux.indexOf(':');
        }
    <%>
    <tr>
        <td>
            <a href="UsuarioAltera?codigo=<%=aux.substring(0,index-1)%>&Acao=Carregar"><%=aux.substring(index+2,index2-1)%></a>
        </td>
        <td>
            <a href="UsuarioAltera?codigo=<%=aux.substring(0,index-1)%>&Acao=Carregar"><%=aux.substring(index2+1)%></a>
        </td>
    </tr>
    <%=}%>
    <%=}else{%>
<tr>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
    </tr>
    <%=}%>
</table>
<!-- chamada da página de rodapé -->
<jsp:include page="../include/gerenciamento.jsp" flush="true"/>
<!-- validação do formulário -->
<script language="JavaScript">
// descrição da estrutura de campos de formulário
var a_fields = {
    'MatrUsu':{'l':'Matrícula','r':false,'f':'integer','t':'MatrUsu'},
    'NomeUsu':{'l':'Nome','r':false,'f':'alpha','t':'NomeUsu'},
},
o_config = {
    'to_disable':['Consultar','Limpar'],
    'alert':1
}
// chamada do construtor do validator
var v = new validator('MainForm', a_fields, o_config);
</script>
</body>
</html>

```

12.4.1.3 incubacao

12.4.1.3.1 incban

```

<%@ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>

```

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>SISINC - Cadastrar Bandeja em Incubadora</title>
<link href="estilos/principal.css" rel="stylesheet" type="text/css" />
<!-- verifica se suporta javascript
<script>function lastmod() {}</script>
<!-- validador do formulário
<script language="javascript" type="text/javascript" src="scripts\validator.js"></script>
<!-- caminho botão adicional -->

<script language="javascript" type="text/javascript">
function IrPara() {parent.location='jsp/incubacao/inccad.jsp';}
</script>
<script>
function mudaEstadoF(){
    document.MainForm.incubadora.disabled=false;
    document.MainForm.nascedouro.disabled=true;
    document.MainForm.posnas.disabled=true;
    document.MainForm.posinc.disabled=false;
}
function mudaEstadoT(){
    document.MainForm.incubadora.disabled=true;
    document.MainForm.nascedouro.disabled=true;
    document.MainForm.posnas.disabled=true;
    document.MainForm.posinc.disabled=true;
}
function mudaEstado2F(){
    document.MainForm.incubadora.disabled=true;
    document.MainForm.nascedouro.disabled=false;
    document.MainForm.posnas.disabled=false;
    document.MainForm.posinc.disabled=true;
}
function mudaEstado2T(){
    document.MainForm.incubadora.disabled=true;
    document.MainForm.nascedouro.disabled=true;
    document.MainForm.posnas.disabled=true;
    document.MainForm.posinc.disabled=true;
}
</script>
</head>
<%@ page import="java.lang.*" %>
<%
controladora.ControladoraConsulta cont=new controladora.ControladoraConsulta();

java.util.Vector nas,inc,vetposN,vetposI,CodBan;
int x,var;
String marcaI="",marcaN="";
inc=new java.util.Vector();
nas=new java.util.Vector();
vetposI=new java.util.Vector();
vetposN=new java.util.Vector();

String ban = request.getParameter("Bandes");
String incubadora=request.getParameter("incubadora");
String nascedouro=request.getParameter("nascedouro");

CodBan=cont.retornaBandejaDesalocada();
inc=cont.retornaIncubadora();
nas=cont.retornaNascedouro();

if(incubadora!=null){
    int posinc = cont.retornaPosinc(incubadora);
    for(x=0;x<posinc;x++){
        vetposI.addElement(new Integer(x));
    }
    marcaN="disabled";
}else{
    if(nascedouro!=null){

```



```

        aux1 = (String) inc.elementAt(i);
        index1 = aux1.indexOf('-');
        %>
        <%if(incubadora!=null){%>
        <%if(aux1.substring(0,index1-1).compareTo(incubadora)==0){ %>
        <option value="<%=aux1.substring(0,index1-1)%>"
selected="selected"><%=aux1.substring(index1+1)%></option>
        <%}else{%>
        <option value="<%=aux1.substring(0,index1-
1)%>"><%=aux1.substring(index1+1)%></option>
        <%}%>
        <%}else{ %>
        <option value="<%=aux1.substring(0,index1-
1)%>"><%=aux1.substring(index1+1)%></option>
        <%}%>
        <%}%>
        </select>
        <!-- acaba o select -->
    </td>
    <td width="59%" align="left" valign="top"><input type="checkbox" name="checkbox"
value="checkbox" onclick="if(this.checked){mudaEstado2F();}else{mudaEstado2T();}" />
    <select name="nascvoudouro" id="nascvoudouro" <%=marcaN%>
onchange="document.MainForm.submit();">
    <%
    aux1 = "";
    index1 = 0;
    for(int i = 0 ; i < nas.size() ; i++)
    {
        aux1 = (String) nas.elementAt(i);
        index1 = aux1.indexOf('-');
        %>
        <%if(nascvoudouro!=null){%>
        <%if(aux1.substring(0,index1-1).compareTo(nascvoudouro)==0){ %>
        <option value="<%=aux1.substring(0,index1-1)%>"
selected="selected"><%=aux1.substring(index1+1)%></option>
        <%}else{%>
        <option value="<%=aux1.substring(0,index1-
1)%>"><%=aux1.substring(index1+1)%></option>
        <%}%>
        <%}else{ %>
        <option value="<%=aux1.substring(0,index1-
1)%>"><%=aux1.substring(index1+1)%></option>
        <%}%>
        <%}%>
        </select> </td>
    </tr>
    <tr>
    <td align="left" valign="top"><strong>Posi&ccedil;&atilde;o: </strong></td>
    <td align="left" valign="top">
    <!-- trecho de código que carrega o campo pai-->
    <select name="posinc" id="posinc" <%=marcaI%> >
    <%

    for(int i = 0 ; i < vetposI.size() ; i++)
    {

        Integer varI=(Integer)vetposI.elementAt(i);
        var=varI.intValue();
        var=var+1;

    %>

    <option value="<%=var%>"><%=var%></option>
    <%}%>
    </select>
    <!-- acaba o select -->
    </td>
    <td align="left" valign="top"> <select name="posnas" id="posnas" <%=marcaN%> >
    <%

    for(int i = 0 ; i < vetposN.size() ; i++)
    {

        Integer varI=(Integer)vetposN.elementAt(i);
        var=varI.intValue();
        var=var+1;
    %>

```

```

        <option value="<%=var%>"><%=var%></option>
    <%}%>
</select></td>
</tr>
<tr>
    <td colspan="3" align="left" valign="top">&nbsp;</td>
</tr>
<tr>
    <td colspan="3" align="left" valign="top"> <hr align="center" width="100%" size="1"
noshade="noshade" />
    </td>
</tr>
</table>
<table width="100%" border="0" cellspacing="0" cellpadding="0">
    <tr>
        <td>
            <input name="AcaoA" type="submit" class="rodape" id="AcaoA" onclick="return v.exec()"
value="Alocar" /> </td>
            <td><input name="Limpar" type="reset" class="rodape" id="Limpar" value="Limpar" /></td>
        <td>&nbsp;</td>
        <td>
            <input name="IrPara" type="button" id="IrPara" onclick="IrPara()" value="Voltar para
Incuba&ccedil;&atilde;o" /> </td>
    </tr>
</table>
</form>
<jsp:include page="../include/incubacao.jsp" flush="true"/>
<script>
// descrição da estrutura de campos de formulário
var a_fields = {
    'MatrUsu':{'l':'Matrícula','r':true,'f':'integer','t':'MatrUsu','mn':8},
    'NomeUsu':{'l':'Nome','r':true,'f':'alphaesp','t':'NomeUsu'},
    'CodGru':{'l':'Grupo','r':true,'f':'integer','t':'CodGru'},
    'SenCrt':{'l':'Senha','r':true,'f':'alphanum','t':'SenCrt','mn':6},
    'SenCrt1':{'l':'Confirmação da
Senha','r':true,'f':'alphanum','t':'SenCrt1','m':'SenCrt','mn':6},
    'TelRes':{'l':'Telefone Residencial','r':false,'f':'integer','t':'TelRes','mn':7},
    'TelCom':{'l':'Telefone Comercial','r':false,'f':'integer','t':'TelCom','mn':7},
    'TelCel':{'l':'Telefone Celular','r':false,'f':'integer','t':'TelCel','mn':7},
    'Email':{'l':'E-mail','r':true,'f':'email','t':'Email','mn':11},
},

o_config = {
    'to_disable' : ['Acao', 'Limpar'],
    'alert' : 1
}

// chamada do construtor do validator
var v = new validator('MainForm', a_fields, o_config);

</script>
</body>
</html>

```

12.4.1.3.2 incbandes

```

<%@ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>SISINC - Cadastrar Bandeja em Incubadora</title>
<link href="estilos/principal.css" rel="stylesheet" type="text/css" />
<!-- verifica se suporta javascript
<script>function lastmod() {}</script>
<!-- validador do formulário
<script language="javascript" type="text/javascript" src="scripts/validator.js"></script>
<!-- caminho botão adicional -->
<link rel="shortcut icon" href="frango.jpg">
<script language="javascript" type="text/javascript">

```

```

function IrPara() {parent.location='jsp/incubacao/inccad.jsp';}
</script>
<script>
function mudaEstadoIF() {
    document.MainForm.CodBan.disabled=false;
    document.MainForm.Bannas.disabled=true;
    document.MainForm.Acao1.disabled=false;
    document.MainForm.Acao2.disabled=true;
}
function mudaEstadoIT() {
    document.MainForm.CodBan.disabled=true;
    document.MainForm.Bannas.disabled=true;
    document.MainForm.Acao1.disabled=true;
    document.MainForm.Acao2.disabled=true;
}
function mudaEstadoNF() {
    document.MainForm.CodBan.disabled=true;
    document.MainForm.Bannas.disabled=false;
    document.MainForm.Acao1.disabled=true;
    document.MainForm.Acao2.disabled=false;
}
function mudaEstadoNT() {
    document.MainForm.CodBan.disabled=true;
    document.MainForm.Bannas.disabled=true;
    document.MainForm.Acao1.disabled=true;
    document.MainForm.Acao2.disabled=true;
}
</script>
</head>
<%
controladora.ControladoraConsulta cont=new controladora.ControladoraConsulta();

java.util.Vector CodBan,CodBanN,bannas,filho,pai,status;
CodBan=cont.retornaBandejaAlocadaI();
bannas=cont.retornaBandejaAlocadaN();

String filhoX = request.getParameter("filho");

if (filhoX==null){
    filho=cont.retornaIncubadora();
    status=cont.retornaStatus();
    pai=cont.retornaPosEnc();
}else{
    filho=cont.retornaIncubadora();
    status=cont.retornaStatus();
    pai=(java.util.Vector) request.getAttribute("encs");
}
%>
<body>
<form id="MainForm" name="MainForm" method="post" action="IncubadoraBandejaDes">
    <table width="100%" border="0" cellspacing="5" cellpadding="0">
        <tr>
            <td colspan="3" class="titulo">Desalocar Bandeja em Incubadora/Nascedouro
            </td>
        </tr>
        <tr>
            <td colspan="2">&nbsp;&nbsp;&nbsp;</td>
            <td bgcolor="#EEEEFF"><a href="IncubadoraBandeja">ir para alocar IncubadoraBandeja</a>
        </td>
        </tr>
        <tr>
            <td colspan="3" align="left" valign="top"><hr align="center" width="100%" size="1"
noshade="noshade" /></td>
        </tr>
        <tr>
            <td colspan="3" align="left" valign="top"><em><strong>DESALOCAR</strong>
            </em></td>
        </tr>
        <tr>
            <td align="left" valign="top">&nbsp;&nbsp;&nbsp;</td>
            <td align="left" valign="top"><input type="checkbox" name="checkbox" value="checkbox"
onclick="if(this.checked){mudaEstadoIF();}else{mudaEstadoIT();}"/>
            Na Incubadora</td>
    </table>

```

```

        <td align="left" valign="top"><input type="checkbox" name="checkbox2" value="checkbox"
onclick="if(this.checked){mudaEstadoNF();}else{mudaEstadoNT();}" />
        No Nascidouro</td>
    </tr>
    <tr>
        <td width="20%" align="left" valign="top"><em><strong>Selecione
Bandeja:</strong></em></td>
        <td width="25%" align="left" valign="top"><select name="CodBan" id="CodBan"
disabled="disabled" >
            <%
                String aux = "";
                int index = 0;
                for(int i = 0 ; i < CodBan.size() ; i++)
                {
                    aux = (String) CodBan.elementAt(i);
                    index = aux.indexOf('-');
                }
            %>
            <option value="<%=aux.substring(0,index-1)%"><%=aux.substring(index+1)%></option>
            <%}%>
        </select> </td>
        <td width="55%" align="left" valign="top"> <select name="Bannas" id="Bannas"
disabled="true">
            <%
                aux = "";
                index = 0;
                for(int i = 0 ; i < bannas.size() ; i++)
                {
                    aux = (String) bannas.elementAt(i);
                    index = aux.indexOf('-');
                }
            %>
            <option value="<%=aux.substring(0,index-1)%"><%=aux.substring(index+1)%></option>
            <%}%>
        </select> </td>
    </tr>
    <tr>
        <td colspan="3" align="left" valign="top"> <hr align="center" width="100%" size="1"
noshade="noshade" />
    </td>
    </tr>
</table>
<table width="100%" border="0" cellspacing="0" cellpadding="0">
    <tr>
        <td><input name="Acao" type="submit" class="rodape" id="Acao" onclick="return v.exec()"
value="Desalocar" />
        </td> <td>&nbsp;</td>
        <td><input name="Limpar" type="reset" class="rodape" id="Limpar" value="Limpar" /></td>
        <td>
            <input name="IrPara" type="button" id="IrPara" onclick="IrPara()" value="Voltar para
Incuba&ccedil;&atilde;o" /> </td>
    </tr>
</table>
</form>
<jsp:include page="../../include/incubacao.jsp" flush="true"/>
<script>
// descrição da estrutura de campos de formulário
var a_fields = {
    'MatrUsu':{'l':'Matrícula','r':true,'f':'integer','t':'MatrUsu','mn':8},
    'NomeUsu':{'l':'Nome','r':true,'f':'alphaesp','t':'NomeUsu'},
    'CodGru':{'l':'Grupo','r':true,'f':'integer','t':'CodGru'},
    'SenCrt':{'l':'Senha','r':true,'f':'alphanum','t':'SenCrt','mn':6},
    'SenCrt1':{'l':'Confirmação da
Senha','r':true,'f':'alphanum','t':'SenCrt1','m':'SenCrt','mn':6},
    'TelRes':{'l':'Telefone Residencial','r':false,'f':'integer','t':'TelRes','mn':7},
    'TelCom':{'l':'Telefone Comercial','r':false,'f':'integer','t':'TelCom','mn':7},
    'TelCel':{'l':'Telefone Celular','r':false,'f':'integer','t':'TelCel','mn':7},
    'Email':{'l':'E-mail','r':true,'f':'email','t':'Email','mn':11},
},
o_config = {
    'to_disable' : ['Acao', 'Limpar'],
    'alert' : 1
}

// chamada do construtor do validator
var v = new validator('MainForm', a_fields, o_config);

```

```

</script>
</body>
</html>

```

12.4.1.3.3 inccad

```

<%@ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>SISINC - Cadastrar Incubação</title>
<script>
function lastmod() {}
</script>
<script src="scripts\cancelar.js"></script>

<link href="estilos/principal.css" rel="stylesheet" type="text/css" />
</head>

<%
controladora.ControladoraConsulta auxCons = new controladora.ControladoraConsulta();
javaUtil.TransferData transf=new javaUtil.TransferData();
//retorna as listas de projetos e bandejas //
java.util.Vector projeto,bandeja;
projeto = auxCons.retornaProjeto();
bandeja = auxCons.retornaBandeja();
String cod=request.getParameter("CodBan");
String incb="",a="",incubacao="",entmin="",inc="",codpos="",pos="",esp="";
String dataPrev="",date="";

//retorna a posição da bandeja no momento não esta na pagina//
if(cod!=null){
int ind1,ind2;
incb=auxCons.retornaPosBan(cod);
System.out.println("incb:"+incb+"cod:"+cod);
ind1=incb.indexOf("-");
ind2=incb.indexOf(":");
inc=incb.substring(0,ind1-1);
codpos=incb.substring(ind1+2,ind2-1);
pos=incb.substring(ind2+2);
inc=inc.replace( ' ', '_');
}
//*****

//pega o valor de CodProj direto*****//
//do servlet se ele nao for nulo carrega a pagina como estava//
String proj=request.getParameter("CodProj");
incubacao=(String)request.getAttribute("incubacao");

java.util.Vector entidade=new Vector();

if(proj!=null){
entmin=(String)auxCons.retornaEntMin(proj);
System.out.println(entmin);
entmin=entmin.substring(4);
entidade=auxCons.retornaEntidadeProj(proj);
dataPrev=auxCons.retornaDataEspecie(proj)+"";
esp=auxCons.retornaEspProjeto(proj);
}

}

else{

a="1 - ";
entidade.addElement(a);
}

%>

```

```

<%@page import="java.util.*"%>
<%!
Calendar tData;
int dia;
int mes;
int ano;
String mensagem;
%>
<%

//funcao que joga data no campo data de inicio da incubacao//
tData=Calendar.getInstance();
dia=tData.get(Calendar.DATE);
mes=tData.get(Calendar.MONTH)+1;
ano=tData.get(Calendar.YEAR);
mensagem="";
mensagem=mensagem+dia+"/";
mensagem=mensagem+mes+"/";
mensagem=mensagem+ano;

%>
<%
int col=0;
int lin=0;

String tam=(String)request.getAttribute("tam");
String aux3 = "";
int marcax=0,marcay=0;

if(tam!=null){

int index3 = 0;

        aux3 = tam;
        index3 = aux3.indexOf('-');
        System.out.println("tam:"+tam);
        lin=Integer.parseInt(aux3.substring(0,index3));
        col=Integer.parseInt(aux3.substring(index3+1));

}
int[][] posmark=new int[lin][col];
if(tam==null){
System.out.println("tam é null");
}else{
//se tam não nulo
//bc é o vetor com as posições ocupadas
Vector bandejaCheia=(Vector)request.getAttribute("bc");
System.out.println("bandeja cheia"+bandejaCheia+"tam vale:"+tam);

for(int x=0;x<lin;x++){
for(int y=0;y<col;y++){
posmark[x][y]=0;
}
}

//pega todos os elementos do vetor
for(int i=0;i<bandejaCheia.size();i++){
aux3=(String)bandejaCheia.elementAt(i);

int index1=aux3.indexOf("-");
int index2=aux3.indexOf(":");
marcax=Integer.parseInt(aux3.substring(index1+1,index2));
marcay=Integer.parseInt(aux3.substring(index2+1));
posmark[marcax][marcay]=1;

}

}

System.out.println("cod"+cod+"entmin"+entmin);
%>
<body>

<form name="MainForm" method="post" action="Incubacao">

```



```

        <td colspan="2" align="left" valign="top"><input name="CodTipoEnt" type="text"
disabled="disabled" size="35" maxlength="50"<%if(proj!=null)%> value=<%=entmin%> /></td>
    </tr>
    <tr>
        <td align="left" valign="top"><strong>Incubac&atilde;o:</strong></td>
        <td colspan="2" align="left" valign="top"><strong>
            <input name="CodInc" type="text" size="35" maxlength="50" <%if(incubacao!=null)%>
value="<%=incubacao%>" />
            </strong></td>
    </tr>
    <tr>
        <td colspan="3" align="left" valign="top"><hr align="center" width="100%" size="1"
noshade="noshade" /></td>
    </tr>
    <tr>
        <td align="left" valign="top"><strong>Data de In&iacute;cio:</strong></td>
        <td colspan="2" align="left" valign="top"><input name="DataIniInc" type="text"
id="DataIniInc" tabindex="3" disabled="true" size="10" maxlength="10" value="<%=mensagem
%>" /></td>
    </tr>
    <tr>
        <td align="left" valign="top">Previs&atilde;o de T&eacute;rmino: </td>
        <td colspan="2" align="left" valign="top"><input name="DataPrevInc" type="text"
disabled="true" size="10" maxlength="10"
<%if((String)request.getAttribute("DataPrevInc")!=null) {
date=transf.DataPApresentar((String)request.getAttribute("DataPrevInc"));}%> value="<%=date%>"
/></td>
    </tr>
    <tr>
        <td colspan="3" align="left" valign="top"><hr align="center" width="100%" size="1"
noshade="noshade" /></td>
    </tr>
    <tr>
        <td align="left" valign="top" bgcolor="#EEEEFF"><a href="IncubadoraBandeja">ir para
alocar bandeja</a></td>
        <td colspan="2" align="left" valign="top">&nbsp;</td>
    </tr>
    <tr>
        <td align="left" valign="top"><em><strong>Selecione Entidade:</strong></em></td>
        <td colspan="2" align="left" valign="top"><select name="CodEnt" >
            <%
                String aux2 = "";
                int index2 = 0;

                for(int i = 0 ; i< entidade.size() ; i++)
                {
                    aux2 = (String) entidade.elementAt(i);
                    index2 = aux2.indexOf('-');
                    if(proj!=null){
                        String aux4=aux2.substring(0,index2-1);
                        String auxEnt=request.getParameter("CodEnt");

                        if(auxEnt==null){
                            auxEnt="0";
                        }

                        if(aux4.compareTo(auxEnt)==0) { %>
                            <option value=<%=aux2.substring(0,index2-1) %> selected><%=aux2.substring(index2+2)
%></option>
                            <%}else{ %>
                                <option value=<%=aux2.substring(0,index2-1) %>><%=aux2.substring(index2+2)
%></option>
                            <%}
                                }else{
                                    %>
                                    <option value=<%=aux2.substring(0,index2-1) %>><%=aux2.substring(index2 +2)
%></option>
                                    <%
                                        } %>
                                <%} %>
                            </select> </td>
    </tr>
    <tr>
        <td align="left" valign="top"><em><strong>Selecione Bandeja:</strong></em></td>
        <td colspan="2" align="left" valign="top"><select name="CodBan"
onchange="document.MainForm.submit();" >

```

```

    <%
String aux1 = "";
int index1 = 0;

for(int i = 0 ; i < bandeja.size() ; i++)
{
    aux1 = (String) bandeja.elementAt(i);
    index1 = aux1.indexOf('-');
    if(proj!=null){

        String b=request.getParameter("CodBan");
        if(b==null){b=""};a=aux1.substring(0,index1-1);
        if(a.compareTo(b)==0){ %>
    <option value=<%=aux1.substring(0,index1-1) %> selected><%=aux1.substring(index1+2)
%></option>
    <%}else{
        %>
    <option value=<%=aux1.substring(0,index1-1) %>><%=aux1.substring(index1+2)
%></option>
    <%} %>
    <%}else{%>
    <option value=<%=aux1.substring(0,index1-1) %>><%=aux1.substring(index1+2) %>
</option>
    <%} %>
    <%} %>
    </select> <input type="submit" class="rodape" name="acao" value="Buscar Bandeja"/>
</td>
</tr>
<tr>
<td height="24" align="left" valign="top"><em><strong>Na Incubadora:</strong></em></td>
<td colspan="2" align="left" valign="top"><input name="CodEnc" type="text"
disabled="true" size="50" maxlength="100" value="<%=inc%>" /></td>
</tr>
<tr>
<td colspan="3" align="left" valign="top">&nbsp;</td>
</tr>
<tr>
<td colspan="3" align="left" valign="top"><hr /></td>
</tr>
<tr>
<td colspan="4" align="left" valign="top"></td>
</tr>
<tr>
<td height="24" colspan="4" align="left" valign="top"> <table width="100%" border="0"
cellspacing="0" cellpadding="0">
<tr>
<td><input name="acao" type="submit" class="rodape" value="Incubar" /></td>
<td><input name="Limpar" type="reset" class="rodape" value="Limpar" /></td>
</tr>
</table></td>
</tr>
</table>
</form>
</body>
</html>

```

12.4.1.4 nascimento

12.4.1.4.1 nascad

```

<%@ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>

```

```

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>SISINC - Cadastrar Nascimento</title>
<link href="estilos/principal.css" rel="stylesheet" type="text/css" />
<script>
    function lastmod() {}
</script>
<script>
    function transferir(nome) {

        if(document.MainForm.QtdeEntidade.value > 0){

            document.MainForm.QtdeEntidade.value=document.MainForm.QtdeEntidade.value-1;

            document.MainForm.QtdeTotal.value=document.MainForm.QtdeTotal.value-1;

            document.MainForm.HQtdeEntidade.value=document.MainForm.HQtdeEntidade.value-1;

        }
        else{
            alert('Você já transferiu todos os ovos desta entidade!');
            document.MainForm.nome.checked=false;
            document.MainForm.nome.checked=false;
        }

    }

    function aumenta(){

        x=0,y=0;
        x=parseInt(document.MainForm.QtdeTotal.value);
        y=parseInt(document.MainForm.QtdeEntidade.value);
        document.MainForm.QtdeTotal.value=x+1;
        document.MainForm.QtdeEntidade.value=y+1;
        document.MainForm.HQtdeEntidade.value=y+1;

    }
</script>
<script src="scripts\cancelar.js"></script>
</head>
<!--*****
*
*JSPs
*
*****-->
<%
controladora.ControladoraConsulta auxCons = new controladora.ControladoraConsulta();
javaUtil.TransferData transf=new javaUtil.TransferData();
//retorna as listas de projetos e bandejas //
java.util.Vector projeto,bandeja;
projeto = auxCons.retornaProjeto();
bandeja = auxCons.retornaBandejaNas();
String cod=request.getParameter("CodBan");
String incb="",a="",incubacao,entmin="",inc="",pos="",codpos="",esp="";
String dataPrev="",date="",qTotal="0",qEntidade="0",HQtdeEntidade="",codnasc="",hcodnasc="";
int index1,index2;

//retorna a posição da bandeja *****//
if(cod!=null){
    int ind1,ind2;
    incb=auxCons.retornaPosBanNas(cod);
    System.out.println("incb:"+incb+"cod:"+cod);
    ind1=incb.indexOf("-");
    ind2=incb.indexOf(":");
    inc=incb.substring(0,ind1-1);
    codpos=incb.substring(ind1+2,ind2-1);
    pos=incb.substring(ind2+2);
    inc=inc.replace(' ','_');
}

//pega o valor de CodProj direto*****//
//do servlet se ele nao for nulo carrega a pagina como estava//
String proj=request.getParameter("CodProj");

```

```

incubacao=(String) request.getAttribute("incubacao");
String ent=(String) request.getAttribute("CodEnt");

java.util.Vector entidade=new Vector();
java.util.Vector listaInc=new Vector();
codnasc=(String) request.getAttribute("codnasc");
hcodnasc=codnasc;
if(proj!=null){
    entmin=(String) auxCons.retornaEntMin(proj);
    entmin=entmin.substring(4);
    entidade=auxCons.retornaEntidadeProj(proj);
    dataPrev=auxCons.retornaDataEspecieNas(proj)+"";
    esp=auxCons.retornaEspProjeto(proj);
    listaInc=auxCons.retornaIncubacao(proj);

}else{

    a=" - ";
    entidade.addElement(a);
    listaInc.addElement(a);

}

}
//*****//
%>
    <%@page import="java.util.*"%>
    <%!
    Calendar tData;
    int dia;
    int mes;
    int ano;
    String mensagem;
    %>
    <%

    //funcao que joga data no campo data de inicio da incubacao//
    tData=Calendar.getInstance();
    dia=tData.get(Calendar.DATE);
    mes=tData.get(Calendar.MONTH)+1;
    ano=tData.get(Calendar.YEAR);
    mensagem="";
    mensagem=mensagem+dia+"/";
    mensagem=mensagem+mes+"/";
    mensagem=mensagem+ano;
    String msg=(String) request.getAttribute("msg");

%>
<%
//*****//
    int col=0;
    int lin=0;

    String tam=(String) request.getAttribute("tam");
    String aux3 = "";
    int marcax=0,marcay=0;

    if(tam!=null){

        int index3 = 0;

            aux3 = tam;
            index3 = aux3.indexOf('-');
            System.out.println("tam:"+tam);
            lin=Integer.parseInt(aux3.substring(0,index3));
            col=Integer.parseInt(aux3.substring(index3+1));

        }
        int[][] posmark=new int[lin][col];
        if(tam==null){
            System.out.println("tam é null");
        }
        }else{
            //se tam não nulo
            //bc é o vetor com as posições ocupadas
            Vector bandejaCheiaEnt=(Vector) request.getAttribute("bc");
            System.out.println("bandeja cheia"+bandejaCheiaEnt+"tam vale:"+tam);

```



```

        if(request.getParameter("CodBan")!=null && lin>0 )out.println("<b>C&ocirc; digo da
Bandeja: </b>"+request.getParameter("CodBan"));
        int x,y;
        String nome="";

        for(x=0;x<lin;x++){
        %>
        <tr>
        <%for (y=0;y<col;y++){ nome=x+"0"+y;%>
        <td>
        <%if (posmark[x][y]==1){%>
        <input type="checkbox" disabled="disable" checked name=<%=nome%> >
        <%}else{%>
        <input type="checkbox" name=<%=nome%> onclick="if (this.checked
) {if (document.MainForm.QtEntidade.value > 0) {

        document.MainForm.QtEntidade.value=document.MainForm.QtEntidade.value-1;

        document.MainForm.QtTotal.value=document.MainForm.QtTotal.value-1;

        document.MainForm.HQtEntidade.value=document.MainForm.HQtEntidade.value-1;
        }
        else{
        alert('Voc&eacute; j&agrave; transferiu todos os ovos desta entidade!');
        this.checked=false;
        }
        } else {almenta();}"/>
        <%}%>
        </td>
        <%}%>
        <td bgcolor="#cccccc"><%=x+1 %>&nbsp;&nbsp;&nbsp;</td>
        </tr>
        <%}%>
        <tr>
        <% for (x=0;x<col;x++){%>
        <td bgcolor="#cccccc"><%=x+1 %></td>
        <%} %>
        </tr>
        </table>
        <!--*****
        *
        *acaba montagem da tabela da bandeja
        *
        *****--
>
        </td>
        </tr>
        <tr >
        <td align="left" valign="top" bgcolor="#EEEEEDD"><strong>Total:</strong></td>
        <td colspan="2" align="left" valign="top"><input name="QtTotal" type="text"
disabled="disabled" size="10" maxlength="20"<%if (incubacao!=null)%> value="<%=qTotal%>"
/></td>
        </tr>
        <tr >
        <td align="left" valign="top" bgcolor="#EEEEEDD"><strong>Esta entidade:</strong></td>
        <td colspan="2" align="left" valign="top"><input name="QtEntidade" type="text"
disabled="disabled" size="10" maxlength="50"<%if (incubacao!=null) %> value="<%=qEntidade%>"
/></td>
        </tr>
        <tr >
        <td colspan="3" align="left" valign="top">|-----
        -----|</td>
        </tr>
        <!--*****
        *2° parte
        *Os dados s&agrave;o coletados
        *
        *****--
>
        <tr >
        <td colspan="3" align="left" valign="top">&nbsp;&nbsp;&nbsp;</td>
        </tr>
        <tr >
        <td width="269" align="left" valign="top"><strong>Projeto:</strong></td>

```

```

        <td colspan="2" align="left" valign="top"><select name="CodProj"
onchange="document.MainForm.submit();" >
    <%
String aux = "";
int index = 0;
for(int i = 0 ; i < projeto.size() ; i++)
{
    aux = (String) projeto.elementAt(i);
    index = aux.indexOf('-');
    if(proj==null){ %>
<option value=<%=aux.substring(0,index-1) %>><%=aux.substring(index+2) %></option>
<%}else{
    a=aux.substring(0,index-1);
    %>
<%if(a.compareTo(proj)==0){ %>
<option value= <%=aux.substring(0,index-1) %> selected><%=aux.substring(index+2)
%></option>
<%
    }else{%>
<option value= <%=aux.substring(0,index-1) %>><%=aux.substring(index+2) %></option>
<%} %>
<% }%>
<%}%>
</select> <strong>Especie:</strong> <input name="CodEspecie" type="text"
disabled="disabled" size="12" maxlength="20" value=<%=esp%>>
</td>
</tr>
<tr>
<td align="left" valign="top"><strong>Incubac&atilde;o:</strong></td>
<td colspan="2" align="left" valign="top"><strong>
<select name="CodInc" >
    <%
String aux1 = "";
index1 = 0;

for(int i = 0 ; i < listaInc.size() ; i++)
{
    aux1 = (String) listaInc.elementAt(i);
    index1=aux1.indexOf(" ");

    if(incubacao!=null){
String b=incubacao;
if(b==null){b="";};a=aux1.substring(0,index1);
if(a.compareTo(b)==0){ %>
<option value=<%=aux1.substring(0,index1)%> selected><%=aux1 %></option>
<%}else{ %>
<option value=<%=aux1.substring(0,index1)%>><%=aux1 %></option>
<%} %>
<%}else{%>
<option value=<%=aux1.substring(0,index1)%>><%=aux1%> </option>
<%} %>
<%} %>
</select>
</strong></td>
<td width="31" colspan="2" align="left" valign="top">&nbsp;</td>
</tr>
<tr>
<td align="left" valign="top"><strong>Nascimento:</strong>&nbsp;</td>
<td colspan="2" align="left" valign="top"><input name="CodNascimento" type="text"
disabled="disabled" size="12" maxlength="20" <%if(codnasc!=null)%> value="<%=codnasc%>"
/></td>
<td colspan="2" align="left" valign="top">&nbsp;</td>
</tr>
<tr>
<td colspan="3" align="left" valign="top"><hr align="center" width="100%" size="1"
noshade="noshade" />
<strong> </strong></td>
<td colspan="2" align="left" valign="top">&nbsp;</td>
</tr>
<tr>
<td align="left" valign="top"><strong>Data de In&iacute;cio:</strong></td>
<td colspan="2" align="left" valign="top"><input name="DataIniInc" type="text"
id="DataIniProj" tabIndex="3" disabled="true" size="10" maxlength="10" value=<%=mensagem
%>></td>
</tr>
<tr>

```

```

        <td align="left" valign="top">Previs&atilde;o de T&eacute;rmino: </td>
        <td colspan="2" align="left" valign="top"><input name="DataPrevInc" type="text"
disabled="true" size="10" maxlength="10"
<%if((String)request.getAttribute("DataPrevInc")!=null) {
date=transf.DataPApresentar((String)request.getAttribute("DataPrevInc"));}%> value=<%=date%>
></td>
    </tr>
    <tr>
        <td colspan="3" align="left" valign="top"><hr align="center" width="100%" size="1"
noshade="noshade" /></td>
    </tr>
    <tr>
        <td align="left" valign="top" bgcolor="#EEEEFF"><a href="IncubadoraBandeja">ir para
alocar bandeja</a></td>
        <td colspan="2" align="left" valign="top">&nbsp;&nbsp;&nbsp;</td>
    </tr>
    <tr>
        <td align="left" valign="top"><em><strong>Selecione Entidade:</strong></em></td>
        <td colspan="2" align="left" valign="top"><select name="CodEnt" >
            <%
String aux2 = "";
index2 = 0;

for(int i = 0 ; i < entidade.size() ; i++)
{
    aux2 = (String) entidade.elementAt(i);
    index2 = aux2.indexOf('-');
    if(proj!=null){
        String aux4=aux2.substring(0,index2-1);
        String auxEnt=request.getParameter("CodEnt");
        if(auxEnt==null){
            auxEnt=" - ";
        }
        if(aux4.compareTo(auxEnt)==0) { %>
            <option value= <%=aux2.substring(0,index2-1) %> selected><%=aux2.substring(index2+2)
%></option>
            <%}else{ %>
            <option value=<%=aux2.substring(0,index2-1) %>><%=aux2.substring(index2+2)
%></option>
            <%}
        }else{
            %>
            <option value=<%=aux2.substring(0,index2-1) %>><%=aux2.substring(index2 +2)
%></option>
            <%
                } %>
            <%} %>
        </select> </td>
    </tr>
    <tr>
        <td align="left" valign="top"><em><strong>Selecione Bandeja:</strong></em></td>
        <td colspan="2" align="left" valign="top"><select name="CodBan"
onchange="document.MainForm.submit();" >
            <%
aux1 = "";
index1 = 0;

for(int i = 0 ; i < bandeja.size() ; i++)
{
    aux1 = (String) bandeja.elementAt(i);
    index1 = aux1.indexOf('-');
    if(proj!=null){

        String b=request.getParameter("CodBan");
        if(b==null){b="";};a=aux1.substring(0,index1-1);
        if(a.compareTo(b)==0) { %>
            <option value=<%=aux1.substring(0,index1-1) %> selected><%=aux1.substring(index1+2)
%></option>
            <%}else{
                %>
            <option value=<%=aux1.substring(0,index1-1) %>><%=aux1.substring(index1+2)
%></option>
            <%} %>
            <%}else{%>
            <option value=<%=aux1.substring(0,index1-1) %>><%=aux1.substring(index1+2) %>

```



```

</tr>
<% java.util.Vector tabela= new java.util.Vector();

tabela = (java.util.Vector)request.getAttribute("tabela");

%>

<%if(tabela.size() != 0){%>

<%
String aux = "";
int index = 0;
int index2 = 0;
int index3 = 0;
for(int i = 0 ; i < tabela.size() ; i++){
    aux = (String) tabela.elementAt(i);
    index = aux.indexOf('-');
    index2 = aux.indexOf(':');
    index3 = aux.indexOf('|');
String tudo=aux.substring(index+2,index2-1);
String parte1=tudo.substring(0,4);
String parte2=tudo.substring(5,7);
String parte3=tudo.substring(8);
String dataini=parte3+"/"+parte2+"/"+parte1;
tudo="";
tudo=aux.substring(index2+2, index3-1);
parte1=tudo.substring(0,4);
parte2=tudo.substring(5,7);
parte3=tudo.substring(8,10);
String datafim=parte3+"/"+parte2+"/"+parte1;
String nome=aux.substring(0,index-1);
String codigo=aux.substring(index3+2);
%>

<tr>
<td height="26">
<a href= "ProjetoAltera?codigo=<%=codigo %>&acao=carregar" > <%=nome %></a>
</td>
<td>
<font color=red><b><%=dataini%></b></font>
</td>
<td>
<font color=red><b><%=datafim%></b></font>
</td>

</tr>
<%} %>
<% } else {%>
<tr>
<td>
</td>
<td>
</td>
</tr>
</table>
</body>
</html>

```

12.4.1.5.2 projcad

```

<%@ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
<%@page import="java.util.*"%>
<%@ include file="../../scripts/data.jsp"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<meta name="robots" content="noindex, nofollow" />
<meta name="description" content="Página de cadastro de projeto." />

```



```

        index = aux.indexOf('-');
    %>
    <option value="<%=aux.substring(0,index-1)%>"><%=aux.substring(index+1)%></option>
    <%}%>
    </select>
    <br />
    <span class="atencao">CUIDADO:</span> <span class="descricao">Define a esp&eacute;cie
para tratamentos, repeti&ccedil;&otilde;es, incuba&ccedil;&otilde;es e nascimentos de
<strong>TODO</strong> o projeto.</span>
    </td>
</tr>
<tr>
    <td align="left" valign="top"><strong>Data de In&iacute;cio: </strong></td>
    <td colspan="2" align="left" valign="top">
        <input name="DataIniProj" type="text" id="DataIniProj" size="10" maxlength="10"
value="<%=mensagem%>" onKeyDown="return dFilter(event.keyCode,this,'##/##/####') " /></td>
    </tr>
<tr>
    <td align="left" valign="top">Pevis&atilde;o de T&eacute;rmino: </td>
    <td colspan="2" align="left" valign="top">
        <input name="DataPrevProj" type="text" id="DataPrevProj" size="10" maxlength="10"
onKeyDown="return dFilter(event.keyCode,this,'##/##/####') " /></td>
    </tr>
<tr>
    <td align="left" valign="top">Descri&ccedil;&atilde;o:</td>
    <td colspan="2" align="left" valign="top">
        <textarea name="DescProj" cols="50" rows="2" id="DescProj"></textarea>
    </td>
</tr>
<tr>
    <td colspan="3" align="left" valign="top">
        <hr align="center" width="100%" size="1" noshade="noshade" />
    </td>
</tr>
</table>
<table width="100%" border="0" cellspacing="0" cellpadding="0">
    <tr>
    <td>
        <input name="Acao" type="submit" class="rodape" id="Acao" value="Criar" />
    </td>
    <td>
        <input name="Acao" type="reset" class="rodape" id="Acao" value="Limpar" />
    </td>
    <td>
        <input name="IrPara" type="button" id="IrPara" value="Alterar/Excluir"
onClick="window.open('ProjetoAlterar','_self') " />
    </td>
</tr>
</table>
</form>
<!-- chamada da p&agrave;gina de rodap&eacute; -->
<jsp:include page="../include/projeto.jsp" flush="true"/>
<script language="JavaScript">
// descri&ccedil;&atilde;o da estrutura de campos para o validador
var a_fields = {
    'SiglaProj':{ 'l':'Sigla do Projeto', 'r':true, 'f':'alphanum', 't':'SiglaProj', 'mn':8},
    'NomeProj':{ 'l':'Nome Completo', 'r':true, 'f':'alpha', 't':'NomeProj'},
    'DataIniProj':{ 'l':'Data de In&iacute;cio', 'r':true, 'f':'date', 't':'DataIniProj'},
    'DataPrevProj':{ 'l':'Previs&atilde;o de T&eacute;rmino', 'r':false, 'f':'date', 't':'DataPrevProj'},
    'DescProj':{ 'l':'Descri&ccedil;&atilde;o', 'r':false, 'f':'alphanum', 't':'DescProj'},
},
o_config = {
    'to_disable':['Criar','Limpar'],
    'alert':1
}
// chamada do construtor do validador
var v = new validator('MainForm', a_fields, o_config);
</script>
</body>
</html>

```

12.4.1.5.3 projcon

```

<%@ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<meta name="robots" content="noindex, nofollow" />
<meta name="description" content="Página de consulta de projeto." />
<meta name="author" content="Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano
Padilha" />
<title>SISINC - Consultar Projeto</title>
<link href="estilos/principal.css" rel="stylesheet" type="text/css" />
<!-- validador do formulário -->
<script language="JavaScript" src="scripts/validator.js"></script>
<!-- limita entrada do formulário -->
<script language="JavaScript" src="scripts/form.js"></script>
<!-- máscara de entrada -->
<script language="JavaScript" src="scripts/dFilter.js"></script>
<!-- posiciona na primeira linha -->
<script language="JavaScript" src="scripts/focus.js"></script>
<!-- desabilita campo em detrimento de outro -->
<script language="JavaScript" src="scripts/disabled.js"></script>
</head>
<%@ page import="controladora.ControladoraProjeto" %>
<%@ page import="servlet.ProjetoAlterar" %>
<%
String sigla = (String) request.getAttribute("sigla");
String nome = (String) request.getAttribute("nome");
String dataInicio = (String) request.getAttribute("dataInicio");
String status = (String) request.getAttribute("status");

javaUtil.TransferData td = new javaUtil.TransferData();
controladora.ControladoraConsulta cont = new controladora.ControladoraConsulta();
// monta a lista Status
java.util.Vector vCodStatusProj;
vCodStatusProj=cont.retornaStatusProj();
%>
<body OnLoad="placeFocus()">
<form id="MainForm" name="MainForm" method="post" action="ProjetoAlterar" onSubmit="return
v.exec();">
<table width="600" border="0" cellspacing="5" cellpadding="0">
<tr>
<td colspan="2">
<span class="titulo">Alterar/Excluir Projeto </span> </td>
</tr>
<tr>
<td colspan="2">&nbsp;</td>
</tr>
<tr>
<td width="15%">Sigla:</td>
<td><input name="SiglaProj" type="text" id="SiglaProj" size="14" maxlength="14"
onBlur="doDisabled(SiglaProj, NomeProj)" <%if (sigla!=null) {%> value="<%=sigla%>" <%}%>/>
<span class="descricao"> (ex.: 200601-A)</span></td>
</tr>
<tr>
<td>Nome:</td>
<td><input name="NomeProj" type="text" id="NomeProj" size="50" maxlength="100"
onKeyUp="doCheckAlpha(MainForm.NomeProj)" onBlur="doDisabled(NomeProj, SiglaProj)"
<%if (nome!=null) {%> value="<%=nome%>" <%}%>/></td>
</tr>
<tr>
<td>Data de In&iacute;cio: </td>
<td><input name="DataIniProj" type="text" id="DataIniProj" size="10" maxlength="10"
onKeyDown="return dFilter(event.keyCode,this,'###/###/####')" <%if (dataInicio!=null) {%>
value="<%=dataInicio%>" <%}%>/></td>
</tr>
<tr>
<td><em>STATUS</em></td>
<td>
<select name="CodStatusProj" id="CodStatusProj">
<%
String aux = "";
int index = 0;
for(int i = 0 ; i < vCodStatusProj.size() ; i++)
{

```

```

        aux = (String) vCodStatusProj.elementAt(i);
        index = aux.indexOf('-');
    %>
    <%if((status!=null) && (aux.substring(0,index-1).compareTo(status)==0)) { %>
        <option value="<%=aux.substring(0,index-1)%>"
selected><%=aux.substring(index+1)%></option>
        <%}else{%>
            <option value="<%=aux.substring(0,index-1)%>"><%=aux.substring(index+1)%></option>
            <%}%>
        <%}%>
    </select>
    </td>
</tr>
<tr>
    <td colspan="2">
        <hr align="center" width="100%" size="1" noshade="noshade" />
    </td>
</tr>
<tr>
    <td colspan="2"><table width="100%" border="0" cellspacing="0" cellpadding="0">
        <tr>
            <td>
                <input name="Acao" type="submit" class="rodape" id="Acao" value="Consultar"
/>
            </td>
            <td>
                <input name="Acao" type="button" class="rodape" id="Acao" value="Limpar"
onClick="doEnbled(SiglaProj,NomeProj)" />
            </td>
            <td align="right">
                <input name="IrPara" type="button" id="IrPara" value="Cadastrar Novo"
OnClick="window.open('Projeto','_self');" />
            </td>
        </tr>
        <tr>
            <td colspan="3">&nbsp;&nbsp;&nbsp;</td>
        </tr>
    </table></td>
</tr>
</table>
</form>
<table width="600" border="1" cellpadding="0" cellspacing="0" bordercolor="#000000">
    <tr>
        <td width="15%"><div align="center"><strong>Sigla</strong></div></td>
        <td width="70%"><div align="center"><strong>Nome</strong></div></td>
        <td width="15%"><div align="center"><strong>Data de Início</strong></div></td>
    </tr>
    <%
    java.util.Vector tabela = new java.util.Vector();
    tabela=(java.util.Vector)request.getAttribute("tabela");
    %>
    <%if((tabela!=null) && (tabela.size() !=0)) {%>
    <%
        aux = "";
        index = 0;
        int index2 = 0;
        int index3 = 0;
        int index4 = 0;
        for(int i = 0; i < tabela.size(); i++)
        {
            aux = (String) tabela.elementAt(i);
            index = aux.indexOf('*');
            index2 = aux.indexOf(':');
            index3 = aux.indexOf(',');
            index4 = aux.indexOf('|');
        }
    %>
    <tr>
        <td>
            <a href="UsuarioAltera?codigo=<%=aux.substring(0,index-1)%>&Acao=Carregar"><%=aux.substring(index3+2,index-1)%></a>
            </td>
        <td>
            <a href="UsuarioAltera?codigo=<%=aux.substring(0,index-1)%>&Acao=Carregar"><%=aux.substring(index+2,index2-1)%></a>
            </td>
        <td>
            <a href="UsuarioAltera?codigo=<%=aux.substring(0,index-1)%>&Acao=Carregar"><%=aux.substring(index2+2,index4-1)%></a>
            </td>
    </tr>

```

```

        <%}%>
    <%}else{%>
<tr>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
</tr>
<%}%>
</table>
<!-- chamada da página de rodapé -->
<jsp:include page="../include/projeto.jsp" flush="true"/>
<!-- validação do formulário -->
<script language="JavaScript">
// descrição da estrutura de campos de formulário
var a_fields = {
    'SiglaProj':{'l':'Sigla do Projeto','r':false,'f':'alphanum','t':'SiglaProj'},
    'NomeProj':{'l':'Nome Completo','r':false,'f':'alpha','t':'NomeProj'},
    'DataIniProj':{'l':'Data de Início','r':false,'f':'date','t':'DataIniProj'},
},
o_config = {
    'to_disable':['Consultar','Limpar'],
    'alert':1
}
// chamada do construtor do validator
var v = new validator('MainForm', a_fields, o_config);
</script>
</body>
</html>

```

12.4.1.6 saque

12.4.1.6.1 embcad

```

<%@ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>SISINC - Cadastrar Incubação</title>
<link href="estilos/principal.css" rel="stylesheet" type="text/css" />
<script>
    function lastmod() {}
</script>
<script>
    function transferir(nome) {

        if(document.MainForm.QtdEntidade.value > 0){

            document.MainForm.QtdEntidade.value=document.MainForm.QtdEntidade.value-1;

            document.MainForm.QtdTotal.value=document.MainForm.QtdTotal.value-1;

            document.MainForm.HQtdEntidade.value=document.MainForm.HQtdEntidade.value-1;

        }
        else{
            alert('Você já transferiu todos os ovos desta entidade!');
            document.MainForm.nome.checked=false;
            document.MainForm.nome.checked=false;
        }
    }
}

```

```

function aumenta(){
    x=0,y=0;
    x=parseInt(document.MainForm.QtdTotal.value);
    y=parseInt(document.MainForm.QtdEntidade.value);
    document.MainForm.QtdTotal.value=x+1;
    document.MainForm.QtdEntidade.value=y+1;
    document.MainForm.HQtdEntidade.value=y+1;
}
</script>
<script src="scripts\cancelar.js"></script>
</head>
<!--*****
*
*JSPs
*
*****-->
<%
controladora.ControladoraConsulta auxCons = new controladora.ControladoraConsulta();
javaUtil.TransferData transf=new javaUtil.TransferData();
//retorna as listas de projetos e bandejas //
java.util.Vector projeto;
projeto = auxCons.retornaProjeto();
String ban=request.getParameter("CodBan");
String incb="",a="",nascimento,entmin="",inc="",pos="",codpos="",esp="",codent="";
String dataPrev="",date="",codInc="",hcodnasc="",DataFimInc="",datainiinc="",tam="";
int index1,index2;
int col=0;
int lin=0;
int[][] posmark=new int[17][26];

//pega o valor de CodProj direto*****//
//do servlet se ele nao for nulo carrega a pagina como estava//
String proj=request.getParameter("CodProj");
nascimento=request.getParameter("CodNas");
String ent=request.getParameter("CodEnt");

java.util.Vector entidade=new Vector();
java.util.Vector listaNas=new Vector();

hcodnasc=nascimento;
if(proj!=null){
    entmin=(String)auxCons.retornaEntMin(proj);
    entmin=entmin.substring(4);

    dataPrev=auxCons.retornaDataEspecieNas(proj)+"";
    esp=auxCons.retornaEspProjeto(proj);
    listaNas=auxCons.retornaNascimento(proj);

    if ((int)listaNas.size()==1){
        String auxnas=(String)listaNas.elementAt(0);
        nascimento=auxnas.substring(0,auxnas.indexOf(" "));
    }else{
        System.out.println("antes do erro"+nascimento+"--");
        if(nascimento==null){
            nascimento="";
        }
        if(nascimento.equals("") && listaNas.size() >= 1){
            String auxnas=(String)listaNas.elementAt(0);
            nascimento=auxnas.substring(0,auxnas.indexOf(" "));
        }else{
            if(listaNas.size()< 1){
                nascimento="";
            }
        }
    }
}

}else{
    a=" - ";
    entidade.addElement(a);
}

```

```

        listaNas.addElement(a);
    }
    //*****//
    %>
        <%@page import="java.util.*"%>
        <%!
        Calendar tData;
        int dia;
        int mes;
        int ano;
        String mensagem;
        %>
        <%

        //funcao que joga data no campo data de inicio da incubacao//
        tData=Calendar.getInstance();
        dia=tData.get(Calendar.DATE);
        mes=tData.get(Calendar.MONTH)+1;
        ano=tData.get(Calendar.YEAR);
        mensagem="";
        mensagem=mensagem+dia+"/";
        mensagem=mensagem+mes+"/";
        mensagem=mensagem+ano;
        String msg=(String)request.getAttribute("msg");
    %>
    <%
    //*****//

    if (nascimento!=null && nascimento!=""){
        String dados=auxCons.retornaDadosNas(nascimento);
        entidade=auxCons.retornaDadosNas2(nascimento);
        int index=dados.indexOf("-");
        codInc=dados.substring(0,index);
        DataFimInc=dados.substring(index+1);
        datainiinc=auxCons.retornaDataIniInc(codInc);
        //if (codent != ""){}
        //index=dados2.indexOf("-");
        //index2=dados2.indexOf("|");
        //codigo para quando só retorna um registro no vetor
        if ((int)entidade.size()==1){

            String aux=(String)entidade.elementAt(0);
            ent=aux.substring(aux.indexOf(":")+1,aux.lastIndexOf("-"));
        }else{

            if( ent==null){
                String aux=(String)entidade.elementAt(0);
                ent=aux.substring(aux.indexOf(":")+1,aux.lastIndexOf("-"));
            }
        }
    }
    if (ent!=null && ent!=""){

        int i,int j;
        controladora.ControladoraNascimento auxContNas= new
controladora.ControladoraNascimento();
        for(i=0;i<entidade.size();i++){
            String aux=(String)entidade.elementAt(i);

            String aux1=aux.substring(aux.indexOf(":")+1,aux.lastIndexOf("-"));
            System.out.println("ent:"+ent+"aux1"+aux1);
            if(aux1.equals(ent)){

                ban=aux.substring(0,aux.indexOf("-"));

                //retorna a posição da bandeja *****//
                if(ban!=null){
                    int ind1,ind2;
                    incb=auxCons.retornaPosBanNas(ban);
                    System.out.println("incb:"+incb+"cod:"+ban);
                    ind1=incb.indexOf("-");
                    ind2=incb.indexOf(":");
                    inc=incb.substring(0,ind1-1);

```



```

<form name="MainForm" method="post" action="Saque">

<input type="hidden" name="cordenada" value="<%=tam%>"/>
<input type="hidden" name="CodEncBan" value="<%=codpos%>"/>
<input type="hidden" name="Data" <%if(proj!=null) %> value="<%=dataPrev%>"/>
<input type="hidden" name="HCodNascimento" <%if(hcodnasc!=null) %> value="<%=hcodnasc%>" />

        <!--*****
        *1° parte
        *informação dos totais
        *por entidade e por incubação
        *****-->

<table width="100%" border="0" cellpadding="0" cellspacing="5">
<tr>
<td colspan="6" class="titulo" bgcolor="#DFEAFE"> Novo Embriodiagn&ocute;stico</td>
</tr>
<tr>
<td height="21" colspan="6">&nbsp;</td>
</tr>
<!--*****
        *2° parte
        *Os dados são coletados
        *
        *****-->

<tr >
<td width="117" height="24" align="left" valign="top"><strong>Projeto:</strong></td>
<td width="85" align="left" valign="top"><select name="CodProj"
onchange="document.MainForm.submit();" >
<%
String aux = "";
int index = 0;
for(int i = 0 ; i < projeto.size() ; i++)
{
    aux = (String) projeto.elementAt(i);
    index = aux.indexOf('-');
    if(proj==null){ %>
<option value=<%=aux.substring(0,index-1) %>><%=aux.substring(index+2) %></option>
<%}else{
    a=aux.substring(0,index-1);
    %>
<%if(a.compareTo(proj)==0){ %>
<option value= <%=aux.substring(0,index-1) %> selected><%=aux.substring(index+2)
%></option>
<%
                                }else{%>
<option value= <%=aux.substring(0,index-1) %>><%=aux.substring(index+2) %></option>
<%} %>
<% }%>
<%}%>
</select> <strong></strong> </td>
<td width="238" align="left"
valign="top"><strong>Especie:&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</strong>
<input name="CodEspecie" type="text" disabled="disabled" size="12" maxlength="20"
value="<%=esp%>" />
</td>
<td width="228" align="left" valign="top"><strong>Data de In&iacute;cio
da incuba&ccedil;&atilde;o: </strong></td>
<td width="322" colspan="2" align="left" valign="top"><strong>
<input name="DataIniInc" type="text" disabled="true" size="10" maxlength="10"
<%if(datainiinc!="") %> value="<%=datainiinc%>"<%} %>/>
</strong></td>
</tr>
<tr>
<td height="24" align="left" valign="top" ><strong>Nascimento:</strong></td>
<td align="left" valign="top"><strong>
<select name="CodNas" onchange="document.MainForm.submit();" >
<%
String aux1 = "";
index1 = 0;

for(int i = 0 ; i < listaNas.size() ; i++)
{
    aux1 = (String) listaNas.elementAt(i);

```

```

        index1=aux1.indexOf(" ");

        if(nascimento!=null){
            String b=nascimento;
            if(b==null){b="";};
            a=aux1.substring(0,index1);
            if(a.compareTo(b)==0){ %>
                <option value=<%=aux1.substring(0,index1)%> selected><%=aux1 %></option>
            <%>else{ %>
                <option value=<%=aux1.substring(0,index1)%>><%=aux1 %></option>
            <%>} %>
            <%>else{<%>
                <option value=<%=aux1.substring(0,index1)%>><%=aux1%> </option>
            <%>} %>
        } %>
    </select>
</strong></td>
<td align="left" valign="top"><strong>Incubac&atilde;o:
    <input name="CodIncubacao" type="text" disabled="disabled" size="12" maxlength="20"
<%if(codInc!=null)%> value="<%=codInc%>" />
    </strong>&nbsp;<strong> </strong></td>
<td align="left" valign="top"><strong>Data de T&eacuterm;mino do nascimento</strong></td>
<td colspan="4" align="left" valign="top"><strong>
    <input name="DataFimNas" type="text" disabled="true" size="10" maxlength="10"
<%if(DataFimInc !="")%> value="<%=DataFimInc%>"<%> %> />
</strong></td>
</tr>
<tr>
<td height="24" align="left" valign="top" ><em><strong>Entidade:</strong></em></td>
<td align="left" valign="top"><strong>
    <select name="CodEnt" onchange="document.MainForm.submit();" >
        <%
            String aux2 = "";
            index2 = 0;

            for(int i = 0 ; i < entidade.size() ; i++)
            {
                aux1 = (String) entidade.elementAt(i);
                aux2=aux1.substring(aux1.indexOf(":")+1);
                index2 = aux2.indexOf('-');
                if(proj!=null){
                    String aux4=aux2.substring(0,index2);
                    System.out.println("valor de ent para compara&ccedil;o:"+aux4);
                    if(ent==null){
                        ent=" - ";
                    }
                    if(aux4.compareTo(ent)==0){ %>
                        <option value= <%=aux2.substring(0,index2) %> selected><%=aux2.substring(index2+1)
%></option>
                    <%>else{ %>
                        <option value=<%=aux2.substring(0,index2) %>><%=aux2.substring(index2+1) %></option>
                    <%>}
                }else{
                    %>
                        <option value=<%=aux2.substring(0,index2-1) %>><%=aux2.substring(index2 +2)
%></option>
                    <%
                        } %>
                <%>} %>
            </select>
        </strong></td>
<td align="left" valign="top">&nbsp;</td>
<td align="left" valign="top">&nbsp;</td>
<td colspan="4" align="left" valign="top">&nbsp;</td>
</tr>
<tr>
<td height="24" colspan="2" align="left" valign="top"
><em><strong>Bandeja:</strong></em></td>
<td align="left" valign="top"><input name="CodBan" type="text" disabled="true"
size="30" maxlength="100" value="<%=ban%>" /></td>
<td align="left" valign="top">&nbsp;</td>
<td colspan="4" align="left" valign="top">&nbsp;</td>
</tr>
<tr>
<td height="24" colspan="2" align="left" valign="top" ><em><strong>No
Nascedouro:</strong></em><strong>

```

```

        </strong></td>
        <td align="left" valign="top"><strong>
          <input name="CodEnc" type="text" disabled="true" size="30" maxlength="100"
value="<%=inc%>" />
        </strong></td>
        <td align="left" valign="top">&nbsp;</td>
        <td colspan="4" align="left" valign="top">&nbsp;</td>
      </tr>
      <tr>
        <td height="24" colspan="2" align="left" valign="top" ><strong>Motivo do
descarte :</strong></td>
        <td align="left" valign="top"><strong> </strong><strong><em><strong>
          <select name="nasceu" >
            <option value="S" selected="true">SIM</option>
            <option value="N" >NAO</option>
          </select>
        </strong></em> </strong></td>
        <td align="left" valign="top"><strong>:
          <input name="acao" type="submit" class="rodape" value="Confirmar" />
        </strong></td>
        <td colspan="4" align="left" valign="top"><strong> </strong></td>
      </tr>
      <tr>
        <td colspan="8" align="left" valign="top"><strong> </strong> <hr align="center"
width="100%" size="1" noshade="noshade" />
        </td>
      </tr>
    </table>
  </form>
</body>
</html>

```

12.4.1.6.2 saqcad

```

<%@ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>SISINC - Cadastrar Incubação</title>
<link href="estilos/principal.css" rel="stylesheet" type="text/css" />
<script>
  function lastmod() {}
</script>
<script>
  function transferir(nome) {

    if(document.MainForm.QtdeEntidade.value > 0){

      document.MainForm.QtdeEntidade.value=document.MainForm.QtdeEntidade.value-1;
      document.MainForm.QtdeTotal.value=document.MainForm.QtdeTotal.value-1;
      document.MainForm.HQtdeEntidade.value=document.MainForm.HQtdeEntidade.value-1;

    }
    else{
      alert('Você já transferiu todos os ovos desta entidade!');
      document.MainForm.nome.checked=false;
      document.MainForm.nome.checked=false;
    }
  }

  function aumenta(){

    x=0,y=0;
    x=parseInt(document.MainForm.QtdeTotal.value);

```

```

        y=parseInt(document.MainForm.QtdeEntidade.value);
        document.MainForm.QtdeTotal.value=x+1;
        document.MainForm.QtdeEntidade.value=y+1;
        document.MainForm.HQtdeEntidade.value=y+1;
    }
</script>
<script src="scripts\cancelar.js"></script>
</head>
<!--*****
*
*JSPs
*
*****-->
<%
controladora.ControladoraConsulta auxCons = new controladora.ControladoraConsulta();
javaUtil.TransferData transf=new javaUtil.TransferData();
//retorna as listas de projetos e bandejas //
java.util.Vector projeto;
projeto = auxCons.retornaProjeto();
String ban=request.getParameter("CodBan");
String incb="",a="",nascimento,entmin="",inc="",pos="",codpos="",esp="",codent="";
String dataPrev="",date="",codInc="",hcodnasc="",DataFimInc="",datainiinc="",tam="";
int index1,index2;
int col=0;
int lin=0;
int[][] posmark=new int[17][26];

//pega o valor de CodProj direto*****//
//do servlet se ele nao for nulo carrega a pagina como estava//
String proj=request.getParameter("CodProj");
nascimento=request.getParameter("CodNas");
String ent=request.getParameter("CodEnt");

java.util.Vector entidade=new Vector();
java.util.Vector listaNas=new Vector();

hcodnasc=nascimento;
if(proj!=null){
    entmin=(String)auxCons.retornaEntMin(proj);
    entmin=entmin.substring(4);

    dataPrev=auxCons.retornaDataEspecieNas(proj)+"";
    esp=auxCons.retornaEspProjeto(proj);
    listaNas=auxCons.retornaNascimento(proj);

    if ((int)listaNas.size()==1){
        String auxnas=(String)listaNas.elementAt(0);
        nascimento=auxnas.substring(0,auxnas.indexOf(" "));
    }else{
        System.out.println("antes do erro"+nascimento+"--");
        if(nascimento==null){
            nascimento="";
        }
        if(nascimento.equals("") && listaNas.size() >= 1){
            String auxnas=(String)listaNas.elementAt(0);
            nascimento=auxnas.substring(0,auxnas.indexOf(" "));
        }else{
            if(listaNas.size()< 1){
                nascimento="";
            }
        }
    }
}

}else{

    a=" - ";
    entidade.addElement(a);
    listaNas.addElement(a);
}
//*****//

```

```

%>
<%@page import="java.util.*"%>
<%!
Calendar tData;
int dia;
int mes;
int ano;
String mensagem;
%>
<%

//funcao que joga data no campo data de inicio da incubacao//
tData=Calendar.getInstance();
dia=tData.get(Calendar.DATE);
mes=tData.get(Calendar.MONTH)+1;
ano=tData.get(Calendar.YEAR);
mensagem="";
mensagem=mensagem+dia+"/";
mensagem=mensagem+mes+"/";
mensagem=mensagem+ano;
String msg=(String)request.getAttribute("msg");
%>
<%
//*****//

if (nascimento!=null && nascimento!=""){
    String dados=auxCons.retornaDadosNas(nascimento);
    entidade=auxCons.retornaDadosNas2(nascimento);
    int index=dados.indexOf("-");
    codInc=dados.substring(0,index);
    DataFimInc=dados.substring(index+1);
    datainiinc=auxCons.retornaDataIniInc(codInc);
    //if (codent != ""){}
    //index=dados2.indexOf("-");
    //index2=dados2.indexOf("|");
    //codigo para quando só retorna um registro no vetor
    if ((int)entidade.size()==1){

        String aux=(String)entidade.elementAt(0);
        ent=aux.substring(aux.indexOf(":")+1,aux.lastIndexOf("-"));
    }else{

        if( ent==null){
            String aux=(String)entidade.elementAt(0);
            ent=aux.substring(aux.indexOf(":")+1,aux.lastIndexOf("-"));
        }
    }
}
if (ent!=null && ent!=""){

    int i:int j;
    controladora.ControladoraNascimento auxContNas= new
controladora.ControladoraNascimento();
    for(i=0;i<entidade.size();i++){
        String aux=(String)entidade.elementAt(i);

        String aux1=aux.substring(aux.indexOf(":")+1,aux.lastIndexOf("-"));
        System.out.println("ent:"+ent+"aux1"+aux1);
        if(aux1.equals(ent)){

            ban=aux.substring(0,aux.indexOf("-"));

            //retorna a posição da bandeja *****//
            if(ban!=null){
                int ind1,ind2;
                incb=auxCons.retornaPosBanNas(ban);
                System.out.println("incb:"+incb+"cod:"+ban);
                ind1=incb.indexOf("-");
                ind2=incb.indexOf(":");
                inc=incb.substring(0,ind1-1);
                codpos=incb.substring(ind1+2,ind2-1);
                pos=incb.substring(ind2+2);
                inc=inc.replace( ' ', '_');
                tam=auxCons.retornaTamBan(ban);
            }
        }
    }
}
}

```



```

        if (b==null) {b=""};
        a=aux1.substring(0,index1);
        if (a.compareTo(b)==0) { %>
<option value=<%=aux1.substring(0,index1)%> selected<%=aux1 %></option>
<%>else{ %>
<option value=<%=aux1.substring(0,index1)%><%=aux1 %></option>
<%> %>
<%>else{%>
<option value=<%=aux1.substring(0,index1)%><%=aux1%> </option>
<%> %>
<%> %>
</select>
</strong></td>
<td align="left" valign="top"><strong> Incubac&atilde;o:
<input name="CodIncubacao" type="text" disabled="disabled" size="12" maxlength="20"
<%if(codInc!=null)%> value="<%=codInc%>" />
</strong>&nbsp;<strong> </strong><strong> </strong></td>
<td align="left" valign="top"><strong>Data de T&eacuterm;mino do nascimento:
</strong></td>
<td colspan="4" align="left" valign="top"><strong>
<input name="DataFimNas" type="text" disabled="true" size="10" maxlength="10"
<%if(DataFimInc != ""){%> value="<%=DataFimInc%>"<%> %> />
</strong></td>
</tr>
<tr>
<td colspan="8" align="left" valign="top"><strong> </strong> <hr align="center"
width="100%" size="1" noshade="noshade" />
</td>
</tr>
<tr>
<td colspan="3" rowspan="5" align="left" valign="top" bgcolor="#EEEEEE"><strong>
<!--*****
*
*montagem da tabela din&acirc;mica da bandeja
*
*****-->
>
</strong> <table border="1" cellspacing="0" cellpadding="0" >
<%

        if (request.getParameter("CodBan")!=null && lin>0 )out.println("<b>C&odigo da Bandeja:
</b>"+request.getParameter("CodBan"));
        int x,y;
        String nome="";

        for (x=0;x<lin;x++){
        %>
        <tr>
        <%for (y=0;y<col;y++){ nome=x+"0"+y;%>
        <td>
        <%if (posmark[x][y]==1){%>
        <input type="checkbox" checked="checked" name="<%=nome%>" />
        <%>else{%>
        <input type="checkbox" disabled="disable" name="<%=nome%>" />
        <%>%>
        </td>
        <%>%>
        <td bgcolor="#cccccc"><%=x+1 %>&nbsp;</td>
        </tr>
        <%>%>
        <tr>
        <% for (x=0;x<col;x++){%>
        <td bgcolor="#cccccc"><%=x+1 %></td>
        <%> %>
        </tr>
        </table>
        <!--*****
        *
        *acaba montagem da tabela da bandeja
        *
        *****-->
        >
        </td>
        <td height="24" align="left" valign="top"><em><strong>Selecione
        Entidade:</strong></em></td>

```



```

<%@ include file="../../../scripts/damenu.jsp"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<meta name="robots" content="noindex, nofollow" />
<meta name="description" content="Menu." />
<meta name="author" content="Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano
Padilha" />
<title>SISINC - Menu</title>
<link href="../../../estilos/menu.css" rel="stylesheet" type="text/css" />
<!-- retorna submenu para menu -->
<script language="JavaScript" src="../../../scripts/menu.js"></script>
</head>

<body>
<br />
<ul id="primary-nav">
  <li class="menuparent"><a href="gerenciamento.jsp"target="principal" >Gerenciamento</a>
    <ul>
    </ul>
  </li>
  <li class="menuparent"><a href="cadastro.jsp"target="principal" >Cadastro</a>
    <ul>
    </ul>
  </li>
  <li class="menuparent"><a href="projeto.jsp"target="principal" >Projeto</a>
    <ul>
    </ul>
  </li>
  <li class="menuparent"><a href="incubacao.jsp"target="principal"
>Incuba&ccedil;&atilde;o</a>
    <ul>
    </ul>
  </li>
  <li class="menuparent"><a href="nascimento.jsp"target="principal" >Nascimento</a>
    <ul>
    </ul>
  </li>
  <li class="menuparent"><a href="relatorio.jsp"target="principal" >Relat&oacute;rio</a>
    <ul>
    </ul>
  </li>
  <li class="menuparent"><a href="../../../help.jsp"target="principal" >Ajuda</a>
    <ul>
    </ul>
  </li>

  <!-- <li class="menuparent"><a href="../../../sair/sair.jsp"target="_parent" >Sair</a> -->
    <li class="menuparent"><a href="../../../sair/sair.jsp"target="principal" >Sair</a>
      <ul>
      </ul>
    </li>
  </ul>
<font color="#ffffff"size="1">
<br />
&nbsp;&nbsp;&nbsp;In&iacute;cio do Acesso
<br />
&nbsp;&nbsp;&nbsp;<%= mensagem1%>
<br />
&nbsp;&nbsp;&nbsp;<%= mensagem %>
</font>
</body>
</html>

```

12.4.2.1.5 nascimento

```

<%@ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

```



```

        ele retorna a incubação a data de início da
incubação
        e a data de término do nascimento e as entidades que estão
em nascimento.</p>
<p>Para uma entidade selecionada, retornada bandeja e nascedouro
onde se encontra, a bandeja retorna com as posições marcadas
onde existem ovos assim que os ovos retirados deve-se desmarcar
na bandeja e depois confirmar se eles nasceram ou não.</p>
<p>&nbsp;</p></td>
<td></td></tr>
<tr>
<td colspan="2" rowspan="8" align="left" valign="top" bgcolor="#EEEEFF">&nbsp;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td colspan="2" align="left" valign="top">&nbsp;</td>
</tr>
<tr>
<td colspan="2" align="left" valign="top"><hr align="center" width="100%" size="1"
noshade="noshade" />
</td>
</tr>
</table>
<table width="100%" border="0" cellspacing="0" cellpadding="0">
<tr>
<td>&nbsp;</td>
</tr>
</table>
</form>
<jsp:include page="../../include/incubacao.jsp" flush="true"/>
<script>
// descrição da estrutura de campos de formulário
var a_fields = {
'MatrUsu':{'l':'Matrícula','r':true,'f':'integer','t':'MatrUsu','mn':8},
'NomeUsu':{'l':'Nome','r':true,'f':'alphaesp','t':'NomeUsu'},
'CodGru':{'l':'Grupo','r':true,'f':'integer','t':'CodGru'},
'SenCrt':{'l':'Senha','r':true,'f':'alphanum','t':'SenCrt','mn':6},
'SenCrt1':{'l':'Confirmação da
Senha','r':true,'f':'alphanum','t':'SenCrt1','m':'SenCrt','mn':6},
'TelRes':{'l':'Telefone Residencial','r':false,'f':'integer','t':'TelRes','mn':7},
'TelCom':{'l':'Telefone Comercial','r':false,'f':'integer','t':'TelCom','mn':7},
'TelCel':{'l':'Telefone Celular','r':false,'f':'integer','t':'TelCel','mn':7},
'Email':{'l':'E-mail','r':true,'f':'email','t':'Email','mn':11},
},
o_config = {
'to_disable' : ['Acao', 'Limpar'],
>alert' : 1
}

// chamada do construtor do validator
var v = new validator('MainForm', a_fields, o_config);

</script>
</body>
</html>

```

12.4.3.2 include

12.4.3.2.1 cadastro

```
<table width="100%" border="0" cellpadding="0" cellspacing="0">
  <tr>
    <td colspan="3">&nbsp;</td>
  </tr>
  <tr>
    <td colspan="3">
      <strong>Ir para:</strong>   <a href="jsp/principal/principal.jsp">Principal</a>
</strong></strong>   <a href="jsp/menu/administrador/cadastro.jsp">Cadastro</a>   </td>
    </tr>
</table>
```

12.4.3.2.2 gerenciamento

```
<table width="100%" border="0" cellpadding="0" cellspacing="0">
  <tr>
    <td colspan="3">&nbsp;</td>
  </tr>
  <tr>
    <td colspan="3">
      <strong>Ir para:</strong>   <a href="jsp/principal/principal.jsp">Principal</a>
</strong></strong>   <a href="jsp/menu/administrador/gerenciamento.jsp">Gerenciamento</a>
    </td>
  </tr>
</table>
```

12.4.3.2.3 incubacao

```
<table width="100%" border="0" cellpadding="0" cellspacing="0">
  <tr>
    <td colspan="3">&nbsp;</td>
  </tr>
  <tr>
    <td colspan="3">
      <strong>Ir para:</strong>   <a href="jsp/principal/principal.jsp">Principal</a>
</strong></strong>   <a href="jsp/menu/administrador/incubacao.jsp">Incuba&ccedil;&atilde;o</a>   </td>
    </tr>
</table>
```

12.4.3.2.4 nascimento

```
<table width="100%" border="0" cellpadding="0" cellspacing="0">
  <tr>
    <td colspan="3">&nbsp;</td>
  </tr>
  <tr>
    <td colspan="3">
      <strong>Ir para:</strong>   <a href="jsp/principal/principal.jsp">Principal</a>
</strong></strong>   <a href="jsp/menu/administrador/nascimento.jsp">Nascimento</a>   </td>
    </tr>
</table>
```

12.4.3.2.5 projeto

```
<table width="100%" border="0" cellpadding="0" cellspacing="0">
  <tr>
    <td colspan="3">&nbsp;</td>
  </tr>
```

```
|  |  |  |
| --- | --- | --- |
| Ir para: Principal | | |
| \ Projeto | | |

```

12.4.3.2.6 rodape

```

<br />
<div align="center" class="rodape">
  <hr align="center" width="100%" size="1" noshade />
  <table width="100%" border="0" align="center" cellpadding="0" cellspacing="0">
    <tr>
      <td width="300"><div align="center">Curso de Zootecnia</div></td>
      <td width="300"><div align="center">Universidade Federal do
Paraná</div></td>
      <td width="300"><div align="center">Tecnologia em Inform&acute;tica</div></td>
    </tr>
    <tr>
      <td><div align="center"><a href="http://www.ccz.agrarias.ufpr.br"></a></div></td>
      <td><div align="center"><a href="http://ww.ufpr.br"></a> <br />
2005</div></td>
      <td><div align="center"><a href="http://et.ufpr.br"></a></div> </td>
    </tr>
  </table>
</div>

```

12.4.3.3 login

12.4.3.3.1 cabeçalho

```

<%@ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<meta name="robots" content="noindex, nofollow" />
<meta name="description" content="Frame de cabe alho da p gina de Login." />
<title>SISINC - Cabe alho</title>
<link href="../../estilos/cabe alho.css" rel="stylesheet" type="text/css" />
</head>

<body>
<div id="cabe alho">
  <div align="center"></div>
</div>
</body>
</html>

```



```

<tr>
  <td height="620" align="left" valign="top"><b>Internet Explorer 6.X <br />
  </b>
  <ol>
    <li>Selecione <em>Op&ccedil;&otilde;es da Internet (Internet Options)</em>
do menu <em>Ferramentas (Tools)</em>. </li>
    <li>Na caixa de di&acutelo;logo <em>Op&ccedil;&otilde;es da Internet</em>
<em>Internet Options)</em> selecione a barra <em>Seguran&ccedil;a</em>
<em>Security)</em>. </li>
    <li>Clique no bot&atilde;o <em>N&iacute;vel Personalizado (Custom
level)</em>. A caixa de di&acutelo;logo com configura&ccedil;&otilde;es de seguran&ccedil;a
<em>Security Settings)</em> aparece. </li>
    <li>Localize, nesta caixa, a categoria <em>Scripts</em>
<em>Scripting)</em>. </li>
    <li>Habilite as op&ccedil;&otilde;es <em>ativar scripts</em> (<em>active
scripting)</em>, <em>permitir opera&ccedil;&otilde;es de colagem atrav&eacute;s de script
(allow paste operations via script)</em> e <em>scripts de miniaplicativos Java (scripting
of java applets)</em>. </li>
    <li>Clique OK na caixa de di&acutelo;logo de <em>configura&ccedil;&otilde;es
de seguran&ccedil;a</em> (<em>Security Settings)</em> e clique OK na caixa de
di&acutelo;logo <em>Op&ccedil;&otilde;es da Internet (Internet Options)</em>. </li>
  </ol>
<p><b>Internet Explorer 5.X </b></p>
<ol>
  <li>Selecione <em>Op&ccedil;&otilde;es da Internet (Internet
Options) </em>do menu <em>Ferramentas</em> (<em>Tools)</em>. </li>
  <li>Na caixa de di&acutelo;logo <em>Op&ccedil;&otilde;es da Internet</em>
<em>Internet Options)</em> selecione a barra <em>Seguran&ccedil;a</em>
<em>Security)</em>. </li>
  <li>Clique no bot&atilde;o <em>N&iacute;vel Personalizado (Custom
level)</em>. A caixa de di&acutelo;logo <em>Configura&ccedil;&otilde;es de Seguran&ccedil;a
(Security</em> settings) aparece. </li>
  <li>Localize, nesta caixa, a categoria <em>Scripts</em>
<em>Scripting)</em>. </li>
  <li>Habilite as op&ccedil;&otilde;es <em>ativar scripts</em> (<em>active
scripting)</em>, <em>permitir opera&ccedil;&otilde;es de colagem atrav&eacute;s de script
(allow paste operations via script)</em> e <em>scripts de miniaplicativos Java (scripting
of java applets)</em>. </li>
  <li>Clique OK na caixa de di&acutelo;logo de <em>configura&ccedil;&otilde;es
de seguran&ccedil;a</em> (<em>Security Settings)</em> e clique OK na caixa de
di&acutelo;logo <em>Op&ccedil;&otilde;es da Internet (Internet Options)</em>. </li>
</ol>
<p><b>Internet Explorer 4.X </b></p>
<ol>
  <li>Selecione <em>Op&ccedil;&otilde;es da Internet (Internet Options)
</em>do menu <em>Visualizar</em> (<em>View)</em>. </li>
  <li>Clique na barra <em>Seguran&ccedil;a</em> (<em>Security)</em>. </li>
  <li>Clique <em>Personalizar</em> (<em>Custom)</em>. </li>
  <li>Clique <em>Configura&ccedil;&otilde;es</em> (<em>Settings)</em>. </li>
  <li>Localize a categoria <em>Scripts</em> (<em>Scripting)</em>. </li>
  <li>Clique <em>Habilitar</em> para <em>Ativar Scripts (Active
Scripting)</em>. </li>
  <li>Clique <em>OK</em>. </li>
  <li>Clique no bot&atilde;o <em>Recarregar</em> (<em>Reload)</em>. </li>
</ol>
<p><b>Internet Explorer 3.X </b></p>
<ol>
  <li>Selecione <em>Op&ccedil;&otilde;es </em>do menu <em>Visualizar
(View)</em>. </li>
  <li>Clique na barra <em>Seguran&ccedil;a (Security)</em>. </li>
  <li>Clique <em>Habilitar Programas Java (Enable Java Programs)</em>. </li>
  <li>Clique <em>OK</em>. </li>
  <li>Clique no bot&atilde;o <em>Recarregar (Reload)</em>. </li>
</ol>
<p><b>Netscape Navigator 5.X e 4.X </b></p>
<ol>
  <li>Clique <em>Editar (Edit)</em>. </li>
  <li>Clique <em>Prefer&ecirc;ncias</em> (<em>Preferences)</em>. </li>
  <li>Clique <em>Avan&ccedil;ado</em> (<em>Advanced)</em>. </li>
  <li>Habilite as op&ccedil;&otilde;es <em>Habilitar Java (Enable Java),
Habilitar JavaScript (Enable JavaScript) </em>e <em>Habilitar JavaScript para mensagens
e not&iacute;cias (Enable JavaScript for Mail and News)</em>. </li>
  <li>Clique OK. </li>
</ol>
<p><b>Netscape Navigator 3.X </b></p>
<ol>

```

```

                <li> Seleccione <em>Op&ccedil;&otilde;es</em> (<em>Options</em>) </em>do menu
<em>Editar (Edit)</em>.</li>
                <li> Clique na op&ccedil;&atilde;o<em> Prefer&ecirc;ncias de Rede</em>
(<em>Network Preferences</em>).</li>
                <li> Clique <em>Linguagens</em> (<em>Languages</em>).</li>
                <li> Habilite ambas as op&ccedil;&otilde;es <em>Habilitar Java (Enable
Java)</em> e <em>Habilitar JavaScript (Enable JavaScript)</em>.</li>
                <li> Clique <em>OK</em>.</li>
            </ol>
        </td>
    </tr>
    <tr>
        <td height="29" align="center" valign="middle">
            <form id="MainForm" name="MainForm" method="post" action=""
target="_self" onSubmit="javascript:history.go(0)">
                <label>
                    <input name="Recarregar" type="submit" class="tag" id="Recarregar"
value="Recarregar" />
                </label>
            </form>
        </td>
    </tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
</noscript>
</body>
</html>

```

12.4.3.3.4 menu

```

<%@ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<meta name="robots" content="noindex, nofollow" />
<meta name="description" content="Frame de menu da página de Login." />
<title>SISINC - Menu</title>
<link href="../../../estilos/menu.css" rel="stylesheet" type="text/css" />
</head>

<body>
</body>
</html>

```

12.4.3.3.5 principal

```

<%@ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<script language="JavaScript">
<!--
if(aux==null)window.open("../Login","_main");
aux="";
//-->
</script>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<meta name="robots" content="noindex, nofollow" />

```

```

<meta name="description" content="Página de Login do SISINC." />
<title>SISINC - Login</title>
<link href="../../estilos/principal.css" rel="stylesheet" type="text/css" />
<!-- limita entrada do formulário -->
<script language="JavaScript" src="../../scripts/form.js"></script>
<!-- posiciona na primeira linha -->
<script language="JavaScript" src="../../scripts/focus.js"></script>

</head>
<%@ page import="controladora.ControladoraLogin" %>
<%@ page import="servlet.Login" %>
<%
String login = (String) request.getAttribute("login");
String retorno = (String) request.getAttribute("retorno");
%>
<%if(retorno!=null){%>
<script language="JavaScript">
var retorno = '<%=retorno%>';
alert(retorno);
</script>
<%}%>
<body OnLoad="placeFocus()">
<table width="100%" border="0" align="center" cellpadding="0" cellspacing="0">
<tr class="cabecalho">
<td height="20" align="center" valign="top">
<div align="center" class="cabecalho">Acesso ao Sistema</div>
</td>
</tr>
<tr>
<td align="center" valign="top">&nbsp;</td>
</tr>
<tr>
<td height="19">
<form id="MainForm" name="MainForm" method="post" action="../../Login" target="_parent"
>
<table width="600" height="100%" border="1" align="center" cellpadding="0"
cellspacing="0" bordercolor="#666666">
<tr>
<td width="92">
<table width="67%" border="0" cellspacing="8" cellpadding="8">
<tr>
<td align="left" valign="top" class="style10">
<span class="titulo">Login</span>
</td>
</tr>
<tr>
<td height="138" align="left" valign="top">
<span class="tag">Matr&iacute;cula: </span>
<label>
<input name="MatrUsu" type="text" id="MatrUsu" size="8" maxlength="8"
onKeyUp="doCheckInteger(MainForm.MatrUsu)" <%if(login!=null){%> value="<%=login%>" <%}%> />
</label>
<p>
<span class="tag">Senha: </span>
<br />
<input name="SenCrt" type="password" id="SenCrt" size="8" />
</p>
<label>
<input name="Acao" type="submit" class="tag" id="Acao" value="Entrar" />
</label>
</td>
</tr>
</table>
</td>
<td width="502"><table width="100%" border="0" cellspacing="8" cellpadding="8">
<tr>
<td align="left" valign="top" class="titulo"><div align="center"
class="titulo">
<div align="left" class="titulo">Instru&ccedil;&otilde;es</div>
</div>
</td>
</tr>
<tr>
<td height="138" align="left" valign="top">
<div align="justify">

```

```

        <span class="body">Para entrar no SISINC voc&ecirc; deve ter
        sido cadastrado, se ainda n&atilde;o possui sua matr&iacute;cula cadastrada favor entrar em
        contato com um dos Administradores do Sistema.
        <br />
        <br />
        Clique aqui para obter mais ajuda.<br />
        <br />
        Clique aqui para entrar em contato com o Administrador.</span>
        <br />
    </div>
</td>
</tr>
</table>
</td>
</tr>
</table>
</form>
</td>
</tr>
</table>
<!-- chamada da p&agrave;gina de rodap&eacute; -->
<jsp:include page="../include/rodape.jsp" flush="true"/>
</body>
</html>

```

12.4.3.4 principal

12.4.3.4.1 cabeçalho

```

<%@ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<meta name="robots" content="noindex, nofollow" />
<meta name="description" content="Frame de cabeçalho da p&agrave;gina de Login." />
<meta name="author" content="Arthur Maçaneiro Magyar, F&abreve;bio Glauco dos Santos, Luciano
Padilha" />
<title>SISINC - Cabe&ccedil;alho</title>
<link href="../../estilos/cabecalho.css" rel="stylesheet" type="text/css" />
</head>

<body>
<div id="cabecalho">
    <div align="center"><a href="principal.jsp" target="principal"></a></div>
</div>
</body>
</html>

```

12.4.3.4.2 index

```

<%@ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
<%@page import="java.util.*"%>
<%
classepersistente.Usuario auxUsuario = new classepersistente.Usuario();
javaUtil.TransferData td = new javaUtil.TransferData();
%>

```

```

<%
HttpSession s = request.getSession(false);
if(s != null) { auxUsuario = (classepersistente.Usuario) s.getAttribute("login"); }
%>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<meta name="robots" content="noindex, nofollow" />
<meta name="description" content="Frameset de entrada do SISINC." />
<meta name="author" content="Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano
Padilha" />
<title>SISINC - Sistema de Controle de Incuba&ccedil;&atilde;o Experimental</title>
</head>

<frameset rows="80,*" cols="*" frameborder="no" border="0" framespacing="0">
<frame src="jsp/principal/cabecalho.jsp" name="cabecalho" scrolling="No" noresize="noresize"
id="cabecalho" title="cabecalho" />

<frameset cols="160,*" frameborder="no" border="0" framespacing="0">
<%if(auxUsuario.getGrupo().equals("1")){%>
<frame src="jsp/menu/administrador/menu.jsp" name="menu" scrolling="No"
noresize="noresize" id="menu" title="menu" />
<%}else{%>
<%if(auxUsuario.getGrupo().equals("2")){%>
<frame src="jsp/menu/operador/menu.jsp" name="menu" scrolling="No" noresize="noresize"
id="menu" title="menu" />
<%}else{%>
<%if(auxUsuario.getGrupo().equals("3")){%>
<frame src="jsp/menu/consultor/menu.jsp" name="menu" scrolling="No" noresize="noresize"
id="menu" title="menu" />
<%}else{%>
<frame src="jsp/login/menu.jsp" name="menu" scrolling="No" noresize="noresize" id="menu"
title="menu" />
<%}}}%>

<frame src="jsp/principal/principal.jsp" name="principal" scrolling="Yes" id="principal"
title="principal" />
</frameset>
</frameset>
<noframes>
</noframes>
<body>
<noframeset>
</noframeset>
</body>
</html>

```

12.4.3.4.3 principal

```

<%@ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
<%@page import="java.util.*"%>
<%
classepersistente.Usuario auxUsuario = new classepersistente.Usuario();
controladora.ControladoraConsulta auxCons = new controladora.ControladoraConsulta();
javaUtil.TransferData td = new javaUtil.TransferData();
%>
<%@ include file="../../../scripts/damssg.jsp"%>
<%@ include file="../../../scripts/rdados.jsp"%>
<%
HttpSession s = request.getSession(false);
if(s != null) { auxUsuario = (classepersistente.Usuario) s.getAttribute("login"); }
%>
%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>

```

```

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<meta name="robots" content="noindex, nofollow" />
<meta name="description" content="Página de entrada do SISINC." />
<meta name="author" content="Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano Padilha" />
<title>SISINC - Entrada</title>
<link href="../../estilos/principal.css" rel="stylesheet" type="text/css" />
</head>

<body>
<table width="100%" border="0" align="center" cellpadding="0" cellspacing="0">
  <tr class="cabecalho">
    <td height="20" align="center" valign="top">
      <div align="center" class="cabecalho">
        <%=mensagem%>, Seja Bem-Vindo <%=auxUsuario.getNome()%> </div></td>
    </tr>
  <tr>
    <td align="center" valign="top">&nbsp;</td>
  </tr>
  <tr>
    <td height="19"><form id="login" name="login" method="post" action="principal.jsp">
      <table width="600" height="100%" border="1" align="center" cellpadding="0" cellspacing="0" bordercolor="#666666">
        <tr>
          <td><table width="100%" border="0" align="center" cellpadding="5" cellspacing="5">
            <tr>
              <td align="center" valign="middle"><div align="center">
                <p>&Uacute;ltimo Acesso como <strong><%=auxUsuario.getDesGru()%></strong> em <strong><%=td.DataPApresentar(auxUsuario.getUltLogon())%></strong><br />
                N&uacute;mero de Projetos: <a href="../../Projeto" target="principal"><%=aux.substring(0,index1)%></a><br />
              </p>
            </div></td>
          </tr>
        </table></td>
        <tr>
          <td align="center" valign="middle"><table width="100%" border="0" cellspacing="5" cellpadding="5">
            <tr>
              <td><div align="center"><span class="titulo">Quadro de Avisos</span></div></td>
            </tr>
            <tr>
              <td align="center" valign="middle"><b>EM ANDAMENTO<br />
                </b><a href="../../Projeto" target="principal"><%=aux.substring(index1+1,index2)%></a> Projetos<br />
                <a href="../../Incubacao"><%=aux.substring(index2+1,index3)%></a> Incuba&ccedil;&otilde;es<br />
                <a href="../../Nascimento"><%=aux.substring(index3+1)%></a> Nascimentos</td>
            </tr>
          </table></td>
        </tr>
      </table>
    </form></td>
  </tr>
</table>
<!-- chamada da página de rodapé -->
<jsp:include page="../../include/rodape.jsp" flush="true"/>
</body>
</html>

```

12.4.3.5 sair

12.4.3.5.1 sair

```

<%@ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<meta name="robots" content="noindex, nofollow" />
<meta name="description" content="Página de saída do SISINC." />
<meta name="author" content="Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano
Padilha" />
<title>SISINC - Sair</title>
<link href="../../estilos/principal.css" rel="stylesheet" type="text/css" />
</head>
<%! classepersistente.Usuario auxUsuario; %>
<%
HttpSession s = request.getSession(false);
if(s != null){ auxUsuario = (classepersistente.Usuario) s.getAttribute("login"); }
%>
<body>
<table width="100%" border="0" align="center" cellpadding="0" cellspacing="0">
<tr class="cabecalho">
<td height="20" align="center" valign="top">
<div align="center" class="cabecalho">Sair do SISINC </div>
</td>
</tr>
<tr>
<td align="center" valign="top">&nbsp;  </td>
</tr>
<tr>
<td align="center" valign="top">
<form id="MainForm" name="MainForm" method="post" action="../../Sair"
target="_parent" >
<table width="600" border="1" align="center" cellpadding="0" cellspacing="0"
bordercolor="#666666">
<tr>
<td height="41"><table width="100%" border="0" align="center" cellpadding="5"
cellspacing="5">
<tr>
<td><div align="center"><span class="titulo"><%=auxUsuario.getNome()%>, deseja
realmente sair? </span></div></td>
</tr>
<tr bordercolor="#666666">
<td align="center" valign="middle">
<table width="100%" border="0" cellspacing="0" cellpadding="0">
<tr>
<td align="center"><input name="Acao" type="submit" class="rodape"
id="Acao" value="Sim" /></td>
<td align="center"><input name="Acao" type="submit" class="rodape"
id="Acao" value="N&atilde;o" /> </td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
</form>
</td>
</tr>
</table>
</body>
</html>

```

12.5 PÁGINAS CSS

12.5.1 cabecalho

```

/* estilo para cabeçalho da página de login e principal */
body {
    background-image: url(../imagens/cabecalho_fundo.gif);
}

#cabecalho {
    position: absolute;
    width: 100%;
    height: 80px;
    z-index: 1;
    left: 0px;
    top: 0px;
}

```

12.5.2 menu

```

/* estilo para menu da página de login e principal */
body {
    font: normal 65.5% verdana;
    background-image: url(../imagens/menu.gif);
}

ul#primary-nav,
ul#primary-nav ul {
    margin: 0;
    padding: 0;
    width: 142px; /* Width of Menu Items */
    border-bottom: 1px solid #ccc;
    background: #6699CC; /* IE6 Bug */
    font-size: 100%;
}

ul#primary-nav li {
    position: relative;
    list-style: none;
}

ul#primary-nav li a {
    display: block;
    text-decoration: none;
    color: #ffffff;
    padding: 5px;
    border: 1px solid #ccc;
    border-bottom: 0;
}

/* Fix IE. Hide from IE Mac */
* html ul#primary-nav li { float: left; height: 1%; }
* html ul#primary-nav li a { height: 1%; }
/* End */

ul#primary-nav ul {
    position: absolute;
    display: none;
    left: 150px; /* Set 1px less than menu width */
    top: 0;
}

ul#primary-nav li ul li a { padding: 2px 5px; } /* Sub Menu Styles */

ul#primary-nav li: hover ul ul,
ul#primary-nav li: hover ul ul ul,
ul#primary-nav li. over ul ul,
ul#primary-nav li. over ul ul ul { display: none; } /* Hide sub-menus initially */

ul#primary-nav li: hover ul,
ul#primary-nav li li: hover ul,
ul#primary-nav li li li: hover ul,
ul#primary-nav li. over ul,
ul#primary-nav li li. over ul,
ul#primary-nav li li li. over ul { display: block } /* The magic */

```

```

ul#primary-nav li.menuparent:hover,
ul#primary-nav li.over { background-color: #CECBCE; }

ul#primary-nav li a:hover { color: #666666; }

```

12.5.3 principal

```

/* estilo padrão */
body {
    font-size: 12;
    background-image: url(../imagens/fundo.gif);
}
form.textfield {
    font-family: Verdana, Arial, Helvetica, sans-serif;
    font-size: 12px;
}

.atencao {
    font-size: 9px;
    color: #FF0000;
}
.atform {
    color: #FF0000;
}
.cabecalho {
    font-size: 36px;
    font-family: Georgia, "Times New Roman", Times, serif;
    color: #666666;
}
.descricao {
    font-size: 9px;
    color: #000000;
}
.rodape {
    font-family: Verdana, Arial, Helvetica, sans-serif;
    font-weight: bold;
}
.tag {
    font-family: Arial, Helvetica, sans-serif;
    font-size: 14px;
}
.titulo {
    font-family: Verdana, Arial, Helvetica, sans-serif;
    font-size: 24px;
}

```

12.5.4 submenu

```

/* estilo para páginas de submenu */
body {
    font: normal 65.5% verdana;
    background-image: url(../imagens/fundo.gif);
}

ul#primary-nav,
ul#primary-nav ul {
    margin: 0;
    padding: 0;
    width: 142px; /* Width of Menu Items */
    border-bottom: 1px solid #ccc;
    background: #6699CC; /* IE6 Bug */
    font-size: 100%;
}

ul#primary-nav li {
    position: relative;
    list-style: none;
}

```

```

ul#primary-nav li a {
    display: block;
    text-decoration: none;
    color: #ffffff;
    padding: 5px;
    border: 1px solid #ccc;
    border-bottom: 0;
}

/* Fix IE. Hide from IE Mac */
* html ul#primary-nav li { float: left; height: 1%; }
* html ul#primary-nav li a { height: 1%; }
/* End */

ul#primary-nav ul {
    position: absolute;
    display: none;
    left: 141px; /* Set 1px less than menu width */
    top: 0;
}

ul#primary-nav li ul li a { padding: 2px 5px; } /* Sub Menu Styles */

ul#primary-nav li:hover ul ul,
ul#primary-nav li:hover ul ul ul,
ul#primary-nav li.over ul ul,
ul#primary-nav li.over ul ul ul { display: none; } /* Hide sub-menus initially */

ul#primary-nav li:hover ul,
ul#primary-nav li li:hover ul,
ul#primary-nav li li li:hover ul,
ul#primary-nav li.over ul,
ul#primary-nav li li.over ul,
ul#primary-nav li li li.over ul { display: block; } /* The magic */

ul#primary-nav li.menuparent:hover,
ul#primary-nav li.over { background-color: #CECBCE; }

ul#primary-nav li a:hover { color: #666666; }

.titulo {
    font-family: Verdana, Arial, Helvetica, sans-serif; font-size: 12px;
    color: #666666;
    font-weight: bold;
}

```

12.6 SCRIPTS

12.6.1 Páginas JavaScript

12.6.1.1 cripto

```

// Equipe: Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano Padilha
// Criado por: Arthur Maçaneiro Magyar
// Aplicação de um cript do tipo virgenere para criptografar senhas.

var TID = 0;
var longer = 0;
var thirdwrite = "";

```

```

var alphabet = new Array("0@", "1#", "2$", "3%", "4^", "5&", "6*", "7 = ", "8-
", "9+", "Aa", "Bb", "Cc", "Dd", "Ee", "Ff", "Gg", "Hh", "Ii", "Jj", "Kk", "Ll", "Mm", "Nn", "Oo", "Pp", "Qq", "R
r", "Ss", "Tt", "Uu", "Vv", "Ww", "Xx", "Yy", "Zz");
function preTime(coco, uinput, question) {
    longer = coco.length - 1;
    placement = 0;
    ref = 0 ;
    secondWrite = "";
    firstWrite = "";
    thirdWrite = "";
    code1(coco, uinput, longer, question);
}
function findshift(letter) {
    for (i = 0; i < 36; i++) {
        if ((alphabet[i].substring(0,1) == letter) || (alphabet[i].substring(1,2) == letter))
        {
            if (alphabet[i].substring(0,1) == letter) {
                x = 0;
            }
            if (alphabet[i].substring(1,2) == letter) {
                x = 1;
            }
            return i;
        }
    }
}
function check(letter) {
    bill = "no";
    for (i = 0; i < 36; i++) {
        if ((alphabet[i].substring(0,1) == letter) || (alphabet[i].substring(1,2) == letter))
        {
            bill = "yes";
        }
    }
    if (bill == "yes") {
        return true;
    }
    else {
        return false;
    }
}
function fixoffset(takin) {
    if (takin > 35) {
        return (takin - 36);
    }
    else {
        return takin;
    }
}
function code1(inval1, inval2, inval3, sage) {
    if (check(inval2.charAt(placement))) {
        if (check(inval1.charAt(ref)) && ((sage==0) || (sage==1))) {
            shift = findshift(inval1.charAt(ref));
        }
        else {
            shift = 0;
        }
        orig = findshift(inval2.charAt(placement));
        firstWrite = firstWrite+alphabet[fixoffset(orig+shift)].charAt(x);
    }
    else {
        firstWrite = firstWrite+inval2.charAt(placement);
    }
    secondWrite = inval2.substring(placement+1, inval2.length+1);
    thirdWrite = firstWrite+secondWrite;
    if (sage == 0) document.MainForm.SenCrt.value = thirdWrite;
    if (sage == 1) document.MainForm.SenCrt1.value = thirdWrite;
    placement++;
    if (ref == inval3) {
        ref = 0
    }
    else {
        ref++;
    }
}
if (placement == (inval2.length)) {
    window.clearTimeout(TID);
}

```

```

        return false;
    }
    one = inval1;
    two = inval2;
    three = inval3;
    goer = sage;
    TID = window.setTimeout("code1(one, two, three, goer)", 0);
}

```

12.6.1.2 data

```

// Equipe: Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano Padilha
// Criado por: Arthur Maçaneiro Magyar
// Script que verifica se data 1 é maior ou igual a data 2 e retorna mensagem.

// compara as datas
function dataMI(dt1,dt2) {
    aler("XX");
    var hoje = new Date();
    var ano = hoje.getYear();
    if(ano >= 50 && ano <= 99)
        ano = 1900 + ano
    else
        ano = 2000 + ano;
    var pos1 = dt1.indexOf("/",0)
    var dd = dt1.substring(0,pos1)
    pos2 = dt1.indexOf("/", pos1 + 1)
    var mm = dt1.substring(pos1 + 1,pos2)
    var aa = dt1.substring(pos2 + 1,10)
    if(aa.length < 4)
        if(ano > 1999)
            aa = (2000 + parseInt(aa,10))
        else
            aa = (1900 + parseInt(aa,10));
    var data1 = new Date(parseInt(aa,10),parseInt(mm,10) - 1, parseInt(dd,10));
    var pos1 = dt2.indexOf("/",0)
    var dd = dt2.substring(0,pos1)
    pos2 = dt2.indexOf("/", pos1 + 1)
    var mm = dt2.substring(pos1 + 1,pos2)
    var aa = dt2.substring(pos2 + 1,10)
    if(aa.length < 4)
        if(ano > 80 && ano <= 99)
            aa = (1900 + parseInt(aa,10))
        else
            aa = (2000 + parseInt(aa,10));
    var data2 = new Date(parseInt(aa,10),parseInt(mm,10) - 1,parseInt(dd,10));
    alert("1");
    if(data1 < data2)
        alert("2");
        return false;
        alert("3");
        if(data1 == null || data2 == null)
            alert("4");
        return false
    else
        alert("5");
        return true;
        alert("6");
}

```

12.6.1.3 dFilter

```

// Equipe: Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano Padilha
// Traduzido por: Arthur Maçaneiro Magyar
// Script que cria modelo de máscara para entrada de dados em formulários.

// [dFilter] - A Numerical Input Mask for JavaScript
// Written By Dwayne Forehand - March 27th, 2003

```

```

// Please reuse & redistribute while keeping this notice.

var dFilterStep
function dFilterStrip (dFilterTemp, dFilterMask) {
    dFilterMask = replace(dFilterMask,'#','');
    for (dFilterStep = 0; dFilterStep < dFilterMask.length++; dFilterStep++) {
        dFilterTemp =
replace(dFilterTemp,dFilterMask.substring(dFilterStep,dFilterStep+1),'');
    }
    return dFilterTemp;
}

function dFilterMax (dFilterMask) {
    dFilterTemp = dFilterMask;
    for (dFilterStep = 0; dFilterStep < (dFilterMask.length+1); dFilterStep++) {
        if (dFilterMask.charAt(dFilterStep)!='#') {
            dFilterTemp = replace(dFilterTemp,dFilterMask.charAt(dFilterStep),'');
        }
    }
    return dFilterTemp.length;
}

function dFilter (key, textbox, dFilterMask) {
    dFilterNum = dFilterStrip(textbox.value, dFilterMask);
    if (key==9) {
        return true;
    }
    else if (key==8&&dFilterNum.length!=0) {
        dFilterNum = dFilterNum.substring(0,dFilterNum.length-1);
    }
    else if (((key>47&&key<58)|| (key>95&&key<106)) &&
dFilterNum.length<dFilterMax(dFilterMask)) {
        dFilterNum=dFilterNum+String.fromCharCode(key);
    }
    var dFilterFinal='';
    for (dFilterStep = 0; dFilterStep < dFilterMask.length; dFilterStep++) {
        if (dFilterMask.charAt(dFilterStep)=='#') {
            if (dFilterNum.length!=0) {
                dFilterFinal = dFilterFinal + dFilterNum.charAt(0);
                dFilterNum = dFilterNum.substring(1,dFilterNum.length);
            }
            else {
                dFilterFinal = dFilterFinal + " ";
            }
        }
        else if (dFilterMask.charAt(dFilterStep)!='#') {
            dFilterFinal = dFilterFinal + dFilterMask.charAt(dFilterStep);
        }
    }
    // dFilterTemp =
replace(dFilterTemp,dFilterMask.substring(dFilterStep,dFilterStep+1),'');
    }
    textbox.value = dFilterFinal;
    return false;
}

function replace(fullString,text,by) {
    // substitui trechos de texto que estão numa string
    var strLength = fullString.length, txtLength = text.length;
    if ((strLength == 0) || (txtLength == 0)) return fullString;
    var i = fullString.indexOf(text);
    if ((!i) && (text != fullString.substring(0,txtLength))) return fullString;
    if (i == -1) return fullString;
    var newstr = fullString.substring(0,i) + by;
    if (i+txtLength < strLength)
        newstr += replace(fullString.substring(i+txtLength,strLength),text,by);
    return newstr;
}

```

12.6.1.4 disabled

// Equipe: Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano Padilha

```

// Criado por: Arthur Maçaneiro Magyar
// Script que desabilita campos exclusivos e os habilita.

// função que desabilita o segundo campo quando o primeiro está preenchido.
function doDisabled(t1,t2){
    if ((t1.value == null) || (t1.value == ""))
        t2.disabled=false
    else
        t2.disabled=true
}

// função que habilita campos
function doEnabled(t1,t2){
    document.forms[0].reset();
    t1.disabled=false;
    t2.disabled=false;
    document.forms[0].reset();
}

```

12.6.1.5 focus

```

// Equipe: Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano Padilha
// Criado por: Arthur Maçaneiro Magyar
// Script relativo ao posicionamento e seleção no formulário.

// função que desabilita o primeiro campo do formulário
function disableFirst(){
    document.forms[0].elements[0].disabled=true
}

// função que posiciona o foco no primeiro campo editável do formulário.
function placeFocus(){
    if (document.forms.length > 0) {
        var field = document.forms[0];
        for (i = 0; i < field.length; i++) {
            if ((field.elements[i].type == "text") || (field.elements[i].type == "textarea") ||
(field.elements[i].type.toString().charAt(0) == "s")) {
                document.forms[0].elements[i].focus();
                break;
            }
        }
    }
}

// função que confirma exclusão
function Delete(){
    del=confirm('Tem certeza que deseja excluir este registro?');
    if (del) document.forms[0].submit();
}

```

12.6.1.6 form

```

// Equipe: Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano Padilha
// Criado por: Arthur Maçaneiro Magyar
// Script que limita a entrada nos campos de formulário, apagando as erradas.

// Original: Mikhail Esteves (miks80@yahoo.com)
// Web Site: http://www.freebox.com/jackol

// limita entrada de campo para valores caracteres
function doCheckAlpha(val) {
    var mikExp = /^[a-zA-Z\ \.\-]*$/;
    var strPass = val.value;
    var strLength = strPass.length;
    var lchar = val.value.charAt((strLength)-1);
    if(lchar.search(mikExp) == -1) {
        var tst = val.value.substring(0, (strLength)-1);
        val.value = tst;
    }
}

```

```

    }
}

// limita entrada de campo para valores numéricos inteiros
function doCheckInteger(val) {
    var mikExp = /^[+\-]?\d*$/;
    var strPass = val.value;
    var strLength = strPass.length;
    var lchar = val.value.charAt((strLength)-1);
    if(lchar.search(mikExp) == -1) {
        var tst = val.value.substring(0, (strLength)-1);
        val.value = tst;
    }
}

// limita entrada de campo para valores numéricos flutuantes
function doCheckReal(val) {
    var mikExp = /^[+\-]?\d*\.\d*$/;
    var strPass = val.value;
    var strLength = strPass.length;
    var lchar = val.value.charAt((strLength)-1);
    if(lchar.search(mikExp) == -1) {
        var tst = val.value.substring(0, (strLength)-1);
        val.value = tst;
    }
}

// função acessória para remover espaços
function trim(s) {
    while ((s.substring(0,1) == ' ') || (s.substring(0,1) == '\n') || (s.substring(0,1) == '\r')){
        s = s.substring(1,s.length);
    }
    return s;
}

```

12.6.1.7 menu

```

// Equipe: Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano Padilha
// Criado por: Arthur Maçaneiro Magyar
// Função que carrega janelas de submenu nas páginas de submenu.

function IEHoverPseudo() {
    var navItems = document.getElementById("primary-nav").getElementsByTagName("li");
    for (var i=0; i<navItems.length; i++) {
        if(navItems[i].className == "menuparent") {
            navItems[i].onmouseover=function() { this.className += " over"; }
            navItems[i].onmouseout=function() { this.className = "menuparent"; }
        }
    }
}
window.onload = IEHoverPseudo;

```

12.6.1.8 validator

```

// Equipe: Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano Padilha
// Traduzido por: Arthur Maçaneiro Magyar
// Script que valida campos de formulário por obrigatório, formato e tamanho.

// Title: Tigra Form Validator
// URL: http://www.softcomplex.com/products/tigra_form_validator/
// Version: 1.3
// Date: 08/25/2005 (mm/dd/yyyy)
// Notes: Registration needed to use this script legally. Visit official site for details.

// expressões regulares ou funções para validar o formato
var re_dt = /^(\\d{1,2})\\/(\\d{1,2})\\/(\\d{4})$/;
re_tm = /^(\\d{1,2})\\:(\\d{1,2})\\:(\\d{1,2})$/;

```

```

a_formats = {
  'alpha' : /^[a-zA-Z\ \.-]*$/,
  'alphanum': /^[\\w\\ \\st]+$/,
  'unsigned': /^[d+$/,
  'integer': /^[+\\-]?\\d*$/,
  'real' : /^[+\\-]?\\d*\\.?\\d*$/,
  'email' : /^[\\w-\\.]+@[\\w\\.]+\\. [a-z]{2,4}$/,
  'phone' : /^[d\\.\\s\\-]+$/,
  'date' : function (s_date) {
    // checa o fomato
    if (!re_dt.test(s_date))
      return false;
    // checa valores permitidos
    if (RegExp.$1 > 31 || RegExp.$2 > 12)
      return false;
    // checa número de dias do mês
    var dt_test = new Date(RegExp.$3, Number(RegExp.$2-1), RegExp.$1);
    if (dt_test.getMonth() != Number(RegExp.$2-1))
      return false;
    return true;
  },
  'time' : function (s_time) {
    // checa formato
    if (!re_tm.test(s_time))
      return false;
    // checa valores permitidos
    if (RegExp.$1 > 23 || RegExp.$2 > 59 || RegExp.$3 > 59)
      return false;
    return true;
  }
},
a_messages = [
  'Não há nome de formulário passado para a rotina construtora do validator.',
  'Não há campos de formulário de "%form%" passados para a rotina construtora do
validator.',
  'Formulário "%form%" não foi encontrado neste documento.',
  'Incompleto em "%n%" a entrada descritiva de campo de formulário. "l" atributo está
faltando.',
  'Não encontrou o campo "%n%" no formulário "%form%"',
  'Não encontrou a label tag (id="%t%")',
  'Não há como verificar se há equivalência. O campo "%m%" não foi encontrado.',
  '"%l%" é um campo obrigatório.',
  'Valor para "%l%" precisa ter %mn% caracteres ou mais.',
  'Valor para "%l%" precisa ter %mx% caracteres no máximo.',
  '"%v%" não há um valor válido para "%l%".',
  '"%l%" precisa ser igual a "%ml%".'
]

// rotina construtora do validator
function validator(s_form, a_fields, o_cfg) {
  this.f_error = validator_error;
  this.f_alert = o_cfg && o_cfg.alert
    ? function(s_msg) { alert(s_msg); return false }
    : function() { return false };

  // checa parâmetros requeridos
  if (!s_form)
    return this.f_alert(this.f_error(0));
  this.s_form = s_form;

  if (!a_fields || typeof(a_fields) != 'object')
    return this.f_alert(this.f_error(1));
  this.a_fields = a_fields;

  this.a_2disable = o_cfg && o_cfg['to_disable'] && typeof(o_cfg['to_disable']) ==
'object'
    ? o_cfg['to_disable']
    : [];

  this.exec = validator_exec;
}

// método de execução do validator
function validator_exec() {
  var o_form = document.forms[this.s_form];
  if (!o_form)

```

```

        return this.f_alert(this.f_error(2));

b_dom = document.body && document.body.innerHTML;

// checa a integridade da entrada descritiva dos campos de formulário
for (var n_key in this.a_fields) {
    // checa a descrição da entrada de dados
    this.a_fields[n_key]['n'] = n_key;
    if (!this.a_fields[n_key]['l'])
        return this.f_alert(this.f_error(3, this.a_fields[n_key]));
    o_input = o_form.elements[n_key];
    if (!o_input)
        return this.f_alert(this.f_error(4, this.a_fields[n_key]));
    this.a_fields[n_key].o_input = o_input;
}

// reinicia highlight dos títulos
if (b_dom)
    for (var n_key in this.a_fields)
        if (this.a_fields[n_key]['t']) {
            var s_labeltag = this.a_fields[n_key]['t'], e_labeltag =
get_element(s_labeltag);
            if (!e_labeltag)
                return this.f_alert(this.f_error(5,
this.a_fields[n_key]));
            this.a_fields[n_key].o_tag = e_labeltag;

            // parâmetros de estado normais definidos aqui
            e_labeltag.className = 'tfvNormal';
        }

// coleta valores dependendo do tipo de entrada
for (var n_key in this.a_fields) {
    var s_value = '';
    o_input = this.a_fields[n_key].o_input;
    if (o_input.type == 'checkbox') // checkbox
        s_value = o_input.checked ? o_input.value : '';
    else if (o_input.value) // text, password, hidden
        s_value = o_input.value;
    else if (o_input.options) // select
        s_value = o_input.selectedIndex > -1
            ? o_input.options[o_input.selectedIndex].value
            : null;
    else if (o_input.length > 0) // radiobuton
        for (var n_index = 0; n_index < o_input.length; n_index++)
            if (o_input[n_index].checked) {
                s_value = o_input[n_index].value;
                break;
            }
    this.a_fields[n_key]['v'] = s_value.replace(/(^s+)|(\s+$)/g, '');
}

// checa por erros
var n_errors_count = 0,
    n_another, o_format_check;
for (var n_key in this.a_fields) {
    o_format_check = this.a_fields[n_key]['f'] &&
a_formats[this.a_fields[n_key]['f']]
        ? a_formats[this.a_fields[n_key]['f']]
        : null;

    // reinicia erros anteriores se houver
    this.a_fields[n_key].n_error = null;

    // checa campos requeridos
    if (this.a_fields[n_key]['r'] && !this.a_fields[n_key]['v']) {
        this.a_fields[n_key].n_error = 1;
        n_errors_count++;
    }
    // checa tamanho
    else if (this.a_fields[n_key]['mn'] && this.a_fields[n_key]['v'] != '' &&
String(this.a_fields[n_key]['v']).length < this.a_fields[n_key]['mn']) {
        this.a_fields[n_key].n_error = 2;
        n_errors_count++;
    }
}

```

```

        else if (this.a_fields[n_key]['mx'] && String(this.a_fields[n_key]['v']).length
> this.a_fields[n_key]['mx']) {
            this.a_fields[n_key].n_error = 3;
            n_errors_count++;
        }
        // checa formato
        else if (this.a_fields[n_key]['v'] && this.a_fields[n_key]['f'] && (
            (typeof(o_format_check) == 'function'
            && !o_format_check(this.a_fields[n_key]['v']))
            || (typeof(o_format_check) != 'function'
            && !o_format_check.test(this.a_fields[n_key]['v'])))
        ) {
            this.a_fields[n_key].n_error = 4;
            n_errors_count++;
        }
        // checa equivalência
        else if (this.a_fields[n_key]['m']) {
            for (var n_key2 in this.a_fields)
                if (n_key2 == this.a_fields[n_key]['m']) {
                    n_another = n_key2;
                    break;
                }
            if (n_another == null)
                return this.f_alert(this.f_error(6, this.a_fields[n_key]));
            if (this.a_fields[n_another]['v'] != this.a_fields[n_key]['v']) {
                this.a_fields[n_key]['ml'] = this.a_fields[n_another]['l'];
                this.a_fields[n_key].n_error = 5;
                n_errors_count++;
            }
        }
    }

    // coleta mensagens de erro e títulos highlight para os campos errados
    var s_alert_message = '',
        e_first_error;

    if (n_errors_count) {
        for (var n_key in this.a_fields) {
            var n_error_type = this.a_fields[n_key].n_error,
                s_message = '';

            if (n_error_type)
                s_message = this.f_error(n_error_type + 6,
this.a_fields[n_key]);

            if (s_message) {
                if (!e_first_error)
                    e_first_error = o_form.elements[n_key];
                s_alert_message += s_message + "\n";
                // parâmetros com highlight definidos aqui
                if (b_dom && this.a_fields[n_key].o_tag)
                    this.a_fields[n_key].o_tag.className = 'tfvHighlight';
            }
        }
        alert(s_alert_message);
        // foca no primeiro campo errado
        if (e_first_error.focus && e_first_error.type != 'hidden' &&
!e_first_error.disabled)
            eval("e_first_error.focus()");
        // cancela o submit do formulário se erros forem encontrados
        return false;
    }

    for (n_key in this.a_2disable)
        if (o_form.elements[this.a_2disable[n_key]])
            o_form.elements[this.a_2disable[n_key]].disabled = true;

    return true;
}

function validator_error(n_index) {
    var s_ = a_messages[n_index], n_i = 1, s_key;
    for (; n_i < arguments.length; n_i++)
        for (s_key in arguments[n_i])
            s_ = s_.replace('%' + s_key + '%', arguments[n_i][s_key]);
}

```

```

        s_ = s_.replace('%form%', this.s_form);
        return s_
    }

function get_element (s_id) {
    return (document.all ? document.all[s_id] : (document.getElementById ?
document.getElementById(s_id) : null));
}

```

12.6.1.9 valpass

```

// Equipe: Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano Padilha
// Traduzido por: Arthur Maçaneiro Magyar
// Script que valida senha.

function validatePwd(fieldname) {
//+++++
//Copyright April 2004 Sani, A. I. (MCSE, MCSA, CCNA)
//sanijean@yahoo.co.uk
//+++++

// inicializa variáveis
var errorMsg = "";
var space = " ";
fieldname = document.forms[0].SenCrt;
fieldvalue = fieldname.value;
fieldlength = fieldvalue.length;

// precisa não conter espaço
if (fieldvalue.indexOf(space) > -1) {
    errorMsg += "\nSenha não deve conter espaço.\n";
}
// precisa conter ao menos um número
if (!(fieldvalue.match(/\d/))) {
    errorMsg += "\nSenha deve incluir ao menos 1 número.\n";
}
// precisa iniciar com ao menos uma letra
if (!(fieldvalue.match(/^[a-zA-Z]/))) {
    errorMsg += "\nSenha deve incluir ao menos uma letra.\n";
}
/*
// precisa conter ao menos um caractere maiúsculo
if (!(fieldvalue.match(/[A-Z]/))) {
    errorMsg += "\nSenha deve incluir ao menos uma letra em maiúsculo.\n";
}
// precisa conter ao menos um caractere minúsculo
if (!(fieldvalue.match(/[a-z]/))) {
    errorMsg += "\nSenha deve incluir ao menos uma letra em minúsculo.\n";
}
// precisa ter ao menos 7 de comprimento
if (!(fieldlength >= 7)) {
    errorMsg += "\nSenha precisa ter ao menos 7 de comprimento.\n";
}
*/
// precisa conter ao menos um caractere especial
if (!(fieldvalue.match(/\W+/))) {
    errorMsg += "\nSenha deve incluir ao menos um caractere especial - #,@,%!\n";
}

//If there is aproblem with the form then display an error
if (errorMsg != ""){
    msg = "_____ \n\n";
    msg += "Por favor corrija o problema com sua senha e tente novamente.\n";
    msg += "_____ \n";
    errorMsg += alert(msg + errorMsg + "\n\n");
    fieldname.focus();
    return false;
}

return true;
}

```

12.6.2 Páginas JSP

12.6.2.1 damenu

```
<%--
Equipe: Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano Padilha
Criado por: Arthur Maçaneiro Magyar
Script que retorna data atual exibida em menagem separada para data e hora.
--%>
<%!
// declaração de variáveis
Calendar tData;
int dia;
int mes;
int ano;
String mensagem;
int hora;
int minuto;
int segundo;
String mensagem1;
%>
<%
// retorna data
tData=Calendar.getInstance();
dia=tData.get(Calendar.DATE);
mes=tData.get(Calendar.MONTH);
ano=tData.get(Calendar.YEAR);
mensagem="Data:";
mensagem+=(dia<10?"0":"")+dia+"/";
mensagem+=(mes<10?"0":"")+mes+"/";
mensagem+=(ano<10?"0":"")+ano;

// retorna hora
tData=Calendar.getInstance();
hora=tData.get(Calendar.HOUR_OF_DAY);
minuto=tData.get(Calendar.MINUTE);
segundo=tData.get(Calendar.SECOND);
mensagem1="Hor&aacute;rio:";
mensagem1+=(hora<10?"0":"")+hora+":";
mensagem1+=(minuto<10?"0":"")+minuto+":";
mensagem1+=(segundo<10?"0":"")+segundo;
%>
```

12.6.2.2 damssg

```
<%--
Equipe: Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano Padilha
Criado por: Arthur Maçaneiro Magyar
Script que retorna data atual exibida em menagem separada para data e hora.
--%>
<%
// declaração de variáveis
Calendar tData;
int hora;
String mensagem;

// instancia a classe
tData=Calendar.getInstance();
hora=tData.get(Calendar.HOUR_OF_DAY);

// exibe mensagem de boas-vindas
if ((hora >= 0) && (hora < 6)){
    mensagem="Ol&aacute;";
} else if (hora < 12){
    mensagem="Bom Dia";
} else if (hora < 18){
```

```

    mensagem="Boa Tarde";
} else {
    mensagem="Boa Noite";
}
%>

```

12.6.2.3 data

```

<%--
Equipe: Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano Padilha
Criado por: Arthur Maçaneiro Magyar
Script que retorna data atual exibida em menagem separada para data e hora.
--%>
<%!
// declaração de variáveis
Calendar tData;
int dia;
int mes;
int ano;
String mensagem;
%>
<%
// retorna data
tData=Calendar.getInstance();
dia=tData.get(Calendar.DATE);
mes=tData.get(Calendar.MONTH);
ano=tData.get(Calendar.YEAR);
mensagem=((dia<10)?"0":"")+dia+"/";
mensagem+=(mes<10)?"0":""+(mes+1)+"/";
mensagem+=(ano<10)?"0":""+ano;
System.out.println(mensagem);
%>

```

12.6.2.4 falha

```

<%@ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
<%--
Equipe: Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano Padilha
Criado por: Arthur Maçaneiro Magyar
Script que retorna mensagem em caso de falha ao gravar no banco de dados.
--%>
<%@ page import="controladora.*" %>
<%@ page import="servlet.*" %>
<% String retorno = (String) request.getAttribute("retorno"); %>
<script language="JavaScript">
var retorno = '<%=retorno%>';
alert("Falha ao Gravar: "+retorno);
history.go(-1);
</script>

```

12.6.2.5 manut

```

<script language="JavaScript">
// Equipe: Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano Padilha
// Criado por: Arthur Maçaneiro Magyar
// Script que retorna mensagem de manutenção para páginas incompletas.
    alert("Página em Construção/Manutenção.");
    javascript:history.go(-1);
</script>

```

12.6.2.6 perm

```
<%@ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
<%--
Equipe: Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano Padilha
Criado por: Arthur Maçaneiro Magyar
Script que retorna mensagem em caso de falha de permissão de usuário.
--%>
<%@ page import="controladora.*" %>
<%@ page import="servlet.*" %>
<% String retorno = (String) request.getAttribute("retorno"); %>
<script language="JavaScript">
var retorno = '<%=retorno%>';
alert("Acesso Não Autorizado. Necessário ao menos Nível "+retorno+".");
history.go(-1);
</script>
```

12.6.2.7 rdados

```
<%
int index1,index2,index3;
String aux;
aux=auxCons.retornaDadosGerais();
System.out.println(aux);
index1=aux.indexOf("-");
index2=aux.indexOf(":");
index3=aux.lastIndexOf("-");
%>
```

12.6.2.8 sessao

```
<%@ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
<%--
Equipe: Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano Padilha
Criado por: Arthur Maçaneiro Magyar
Script que retorna mensagem em caso de sessão expirada.
--%>
<script language="JavaScript">
alert("Sessão Expirada. Favor efetue novo Login.");
window.open("Login","_parent");
</script>
```

12.6.2.9 status

```
<%@ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
<%--
Equipe: Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano Padilha
Criado por: Arthur Maçaneiro Magyar
Script que retorna mensagem em caso de falha de status de usuário.
--%>
<script language="JavaScript">
alert("Acesso Não Autorizado. Status não está Ativo.");
history.go(-1);
</script>
```

12.6.2.10 sucesso

```
<%@ page contentType="text/html; charset=iso-8859-1" language="java" import="java.sql.*"
errorPage="" %>
<%--
Equipe: Arthur Maçaneiro Magyar, Fábio Glauco dos Santos, Luciano Padilha
Criado por: Arthur Maçaneiro Magyar
Script que retorna página nova caso sucesso ao gravar no banco de dados.
--%>
<%@ page import="controladora.*" %>
<%@ page import="servlet.*" %>

<% String retorno = (String) request.getAttribute("retorno"); %>
<script language="JavaScript">

<%if(retorno.equals("Entidade")) {%>
    alert("Entidade Gravada com Sucesso!");
    window.open("Entidade", "_self");
<%}else{%>
<%if(retorno.equals("EntidadeAlterar")) {%>
    alert("Entidade Alterada com Sucesso!");
    window.open("EntidadeAlterar", "_self");
<%}else{%>
<%if(retorno.equals("EntidadeExcluir")) {%>
    alert("Entidade Excluída com Sucesso!");
    window.open("EntidadeAlterar", "_self");
<%}else{%>
<%if(retorno.equals("Projeto")) {%>
    alert("Projeto Gravado com Sucesso!");
    window.open("Projeto", "_self");
<%}else{%>
<%if(retorno.equals("ProjetoAlterar")) {%>
    alert("Projeto Alterado com Sucesso!");
    window.open("ProjetoAlterar", "_self");
<%}else{%>
<%if(retorno.equals("ProjetoExcluir")) {%>
    alert("Projeto Excluído com Sucesso!");
    window.open("ProjetoAlterar", "_self");
<%}else{%>
<%if(retorno.equals("Usuario")) {%>
    alert("Usuário Gravado com Sucesso!");
    window.open("Usuario", "_self");
<%}else{%>
<%if(retorno.equals("UsuarioAlterar")) {%>
    alert("Usuário Alterado com Sucesso!");
    window.open("UsuarioAlterar", "_self");
<%}else{%>
<%if(retorno.equals("UsuarioExcluir")) {%>
    alert("Usuário Excluído com Sucesso!");
    window.open("UsuarioAlterar", "_self");
<%}%>
<%}}}}}}}};%>
    window.focus();
</script>
```