

UNIVERSIDADE FEDERAL DO PARANÁ

CECILIA MACHADO DE FELIPPE

SISTEMA DE APROVAÇÃO DE CONTRATOS DE CÂMBIO

CURITIBA

2015

CECÍLIA MACHADO DE FELPPE

SISTEMA DE APROVAÇÃO DE CONTRATOS DE CÂMBIO

Trabalho de Conclusão de Curso do curso de
Especialização em Engenharia de Software da
Universidade Federal do Paraná.

Orientador Prof. Jaime Wojciechowski

CURITIBA

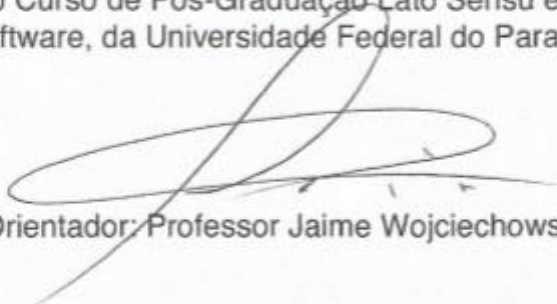
2015

TERMO DE APROVAÇÃO

CECILIA MACHADO DE FELIPPE

**SISTEMA DE APROVAÇÃO DE CONTRATOS DE
CÂMBIO**

Monografia apresentada como requisito parcial para a obtenção da titulação de especialista, pelo Curso de Pós-Graduação Lato Sensu em Engenharia de Software, da Universidade Federal do Paraná.



Orientador: Professor Jaime Wojciechowski

Curitiba, 31 de Janeiro de 2015

RESUMO

O desenvolvimento de projetos de software dentro do prazo e custo esperados é um dos grandes desafios para os gestores de sistemas. Para isso, a Engenharia de Software provê metodologias para auxiliar as equipes a desenvolver sistemas de uma forma planejada e com qualidade. Considerando a complexidade e dinamismo dos sistemas surgiu o *Rational Unified Process*, que adota práticas para minimizar os problemas encontrados dentro do cenário atual de desenvolvimento de software.

Este trabalho apresenta um estudo sobre o *Rational Unified Process* e o desenvolvimento de um sistema baseado nesse processo, evidenciando todas as etapas e artefatos gerados durante o andamento do projeto.

Palavras-chave: Engenharia de Software, metodologia, *RUP*.

ABSTRACT

The software projects development on time and cost expected is a major challenge for systems managers. For this, Software Engineering provides methodologies to help teams to develop planned manner systems with quality. Considering the systems complexity and dynamism the Rational Unified Process has come, which adopts practices to minimize the problems found within the current software development scenario.

This work gives a study on the *Rational Unified Process* and the development of a system based in this process, showing all the steps and artifacts generated during the project progress.

Keywords: Software Engineering, methodology, *RUP*.

SUMÁRIO

RESUMO	4
ABSTRACT	5
1 INTRODUÇÃO	8
1.1 INTRODUÇÃO	8
1.2 OBJETIVOS DO PROJETO.....	8
2 FUNDAMENTAÇÃO TEÓRICA	9
2.1 METODOLOGIAS DE DESENVOLVIMENTO DE SOFTWARE	9
2.2 RATIONAL UNIFIED PROCESS.....	10
2.2.1 Melhores Práticas	11
2.2.2 Fases do RUP	13
2.2.3 Fluxos do RUP	15
3 METODOLOGIA	18
3.1 METODOLOGIA	18
3.2 MODELO DE PROCESSO DE ENGENHARIA DE SOFTWARE.....	18
3.3 PLANO DE ATIVIDADE	18
3.3.1 WBS	19
3.4 PLANO DE RISCOS	19
3.5 RESPONSABILIDADES.....	20
3.6 MATERIAIS.....	20
3.7 DESENVOLVIMENTO DO PROJETO	20
4 APRESENTAÇÃO DO SOFTWARE	21
4.1 INICIAÇÃO.....	22
4.1.1 Fluxo de Modelo de Negócio	22
4.2 ELABORAÇÃO	22
4.2.1 Fluxo de Requisitos	22
4.2.2 Fluxo de Análise e Design	22
4.2.3 Fluxo de Testes	23
4.3 CONSTRUÇÃO E TRANSIÇÃO	23
4.3.1 Workflow de Implementação	23
5 CONSIDERAÇÕES FINAIS	25
5.1 QUANTO AOS OBJETIVOS DO PROJETO	25
REFERÊNCIAS	26
APÊNDICE 1	27
APÊNDICE 2	30
APÊNDICE 3	32
APÊNDICE 4.....	33
APÊNDICE 5.....	34
APÊNDICE 6.....	38
APÊNDICE 7	58
APÊNDICE 8.....	59
APÊNDICE 9.....	60
APÊNDICE 10.....	61
APÊNDICE 11.....	62
APÊNDICE 12.....	69
APÊNDICE 13.....	70
APÊNDICE 14.....	72

APÊNDICE 15.....	73
APÊNDICE 16.....	77
APÊNDICE 17.....	84

1 INTRODUÇÃO

1.1 INTRODUÇÃO

A preocupação com a qualidade de software deixou de ser um diferencial competitivo e passou a ser um pré-requisito básico para a participação no mercado. Para alcançar a qualidade desejada, a Engenharia de Software dedica esforços na tentativa de melhorar os processos de desenvolvimento de software.

Sommerville (2004) define “processo de software” como um conjunto de atividades cujo objetivo é o desenvolvimento ou a evolução de um software. Estas atividades precisam estar coordenadas e todas as pessoas envolvidas precisam estar cientes de seus papéis no processo para garantir o seu sucesso. Logo, existem metodologias que definem uma série de passos para que o processo de desenvolvimento siga um padrão e assim atinja bons níveis de qualidade.

Um grande problema nos projetos atuais é o grande dinamismo e complexidade dos negócios nos dias de hoje. Cada vez mais os sistemas são complexos e precisam estar prontos em menos tempo. As necessidades mudam ao longo do tempo e a especificação de um sistema provavelmente será alterada durante seu desenvolvimento (BROERING, 2015). O *Rational Unified Process* adota algumas premissas básicas visando atacar estes problemas.

1.2 OBJETIVOS DO PROJETO

O objetivo geral é estudar e aplicar o *Rational Unified Process* através da construção de um software fictício. Como objetivos específicos deste trabalho podem ser citados:

- Realizar o estudo do processo de modelagem do *Rational Unified Process* por meio da bibliografia existente;
- Avaliação da aplicabilidade do *RUP* através do desenvolvimento de um software simples.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta a revisão bibliográfica na qual este trabalho está fundamentado. Será apresentada uma breve descrição sobre metodologias de desenvolvimento de software e posteriormente o processo de desenvolvimento *Rational Unified Process*. Para confecção do capítulo 2.1 foi utilizado como base o Trabalho de Conclusão de Curso submetido à Universidade Federal de Santa Catarina do autor deste trabalho (FELIPPE, 2007).

2.1 METODOLOGIAS DE DESENVOLVIMENTO DE SOFTWARE

Na década de 70 a atividade de “desenvolvimento de software” era executada de forma desorganizada, desestruturada e sem planejamento, gerando um produto final de má qualidade, que não correspondia às reais necessidades do cliente (PRESSMAN, 2001). A partir deste cenário surgiu a necessidade de tornar o desenvolvimento de software como um processo estruturado, planejado e padronizado (NETO, 2004).

Segundo Pressman (2001), processo de software é um conjunto de tarefas necessárias para a produção de um software com alta qualidade. O resultado do processo é um produto (software) que reflete a forma como o processo foi realizado. Embora existam vários métodos de desenvolvimento de software existem atividades genéricas comuns a todos eles (SOMMERVILLE, 2004):

- a) Especificação do Software: definição do que o sistema deve fazer (requisitos e restrições do software);
- b) Desenvolvimento do Software: projeto e desenvolvimento do software (o sistema é desenvolvido conforme sua especificação);
- c) Validação do Software: o software é validado para garantir que as funcionalidades desenvolvidas estejam de acordo com o especificado.
- d) Evolução do Software: evolução do software conforme a necessidade dos clientes.

Visando a qualidade final do produto gerado são propostos métodos que buscam padronizar as atividades e o ciclo de vida de um processo de software.

Esta padronização implica em criação de documentos, artefatos e marcos capazes de representar o contexto do software (NETO, 2004).

Portanto, uma metodologia de desenvolvimento de software é a representação simplificada do processo que combina descrição do trabalho, procedimentos e convenções utilizadas por seus membros. Ou ainda, uma metodologia de desenvolvimento de software é um conjunto de práticas recomendadas para o desenvolvimento de software, sendo que essas práticas geralmente passam por fases ou passos, que são subdivisões do processo, para ordená-lo e melhor gerenciá-lo (SOMMERVILLE, 2004).

O desafio mais comum dentro do desenvolvimento de software é entregar um produto que atenda as reais necessidades do cliente, dentro do prazo e orçamento previstos (FAGUNDES, 2005). Como suporte a esta atividade, a Engenharia de Software oferece metodologias que auxiliam os desenvolvedores durante o processo de desenvolvimento, entre elas o processo de desenvolvimento de software *Rational Unified Process*.

2.2 RATIONAL UNIFIED PROCESS

O *Rational Unified Process (RUP)* é uma metodologia completa para viabilizar que grandes projetos de software sejam bem sucedidos (BROERING, 2015). É um produto de processo desenvolvido e mantido pela *Rational Software* e integrado com seu conjunto de ferramentas de desenvolvimento de software (KRUCHTEN, 2003).

O *RUP* usa a abordagem da orientação a objetos em sua concepção e é projetado e documentado utilizando a notação *UML (Unified Modeling Language)* para ilustrar os processos em ação. Utiliza técnicas e práticas aprovadas comercialmente (WIKIPEDIA, 2015). Ele fornece uma abordagem disciplinada para assumir tarefas e responsabilidades dentro de uma organização de desenvolvimento (KRUCHTEN, 2003).

O *Rational Unified Process* define perfeitamente quem é responsável pelo que, como as coisas deverão ser feitas e quando devem ser realizadas, descrevendo todas as metas de desenvolvimento especificamente para que sejam alcançadas (MARTINEZ, 2014). Seu objetivo é assegurar a produção de

software de alta qualidade que satisfaça as necessidades de seus usuários finais dentro de prazo e orçamento previsíveis (KRUCHTEN, 2003).

O *RUP* é também uma estrutura de processo que pode ser adaptada e estendida para compor as necessidades de uma organização que o esteja adotando (KRUCHTEN, 2003). Um processo descreve quem está fazendo o quê, como e quando. O *Rational Unified Process* é representado usando quatro elementos primários de modelagem (KRUCHTEN, 2003):

- a) Trabalhadores (quem): define o comportamento e as responsabilidades de um indivíduo ou grupo de indivíduos que trabalham juntos com uma equipe. Trabalhadores não são os indivíduos, eles descrevem como os indivíduos devem se comportar no negócio e as responsabilidades de cada um;
- b) Atividades (como): unidade do trabalho que um indivíduo naquele papel pode ser pedido a desempenhar e isso produz um resultado significativo no contexto do projeto. Tem um propósito claro, normalmente expresso em termos de criar ou atualizar artefatos como modelos, classes ou um plano;
- c) Artefatos (o quê): pedaço de informação que é produzida, modificada ou usada por um processo. Coisas que o projeto produz ou usa enquanto trabalha para o produto final;
- d) Fluxos (quando): seqüência de atividades que produz um resultado de valor observável. É a descrição das seqüências significantes de atividades que produzem algum resultado valioso e mostrar interações entre trabalhadores.

2.2.1 Melhores Práticas

Um grande problema nos projetos atuais é o grande dinamismo e complexidade dos negócios nos dias de hoje. Cada vez mais os sistemas são complexos e precisam estar prontos em menos tempo. As necessidades mudam ao longo do tempo e a especificação de um sistema provavelmente será alterada durante seu desenvolvimento (BROERING, 2015).

O *Rational Unified Process* reúne muitas das melhores práticas do desenvolvimento moderno de software de forma satisfatória para uma grande faixa de projetos e organizações. Em particular, o *RUP* cobre as seguintes

práticas comerciais que atacam a raiz dos problemas de desenvolvimento de software (KRUCHTEN, 2003):

- a) Desenvolver software iterativamente: abordagem iterativa extremamente controlada (iterações planejadas em número, duração e objetivos), tarefas e responsabilidades dos participantes bem definidas, medidas objetivas de progresso adotadas;
- b) Gerenciar requisitos: uma abordagem sistemática para concluir, organizar, comunicar e gerenciar os requisitos mutáveis de um sistema de software, visando melhor controle de projetos complexos, qualidade aperfeiçoada de software e satisfação do cliente, além de custos de projeto reduzidos;
- c) Usar arquiteturas baseadas em componente: produzir e validar uma arquitetura de software que no ciclo de desenvolvimento inicial leva a forma de um protótipo arquitetônico executável que gradualmente evolui para se tornar o sistema final em iterações posteriores;
- d) Modelar visualmente o software: desenvolver e manter modelos do sistema de acordo com o desenvolvimento (um modelo é uma simplificação da realidade que ajuda a compreender um sistema grande e complexo que não pode ser compreendido em sua totalidade);
- e) Verificar continuamente a qualidade do software: a qualidade é de responsabilidade de todos os membros da organização de desenvolvimento, focada em duas áreas: qualidade de produto (qualidade do produto principal que é produzido e todos os elementos que ele abrange) e qualidade do processo (o grau para qual um processo aceitável foi implementado e aderiu durante a fabricação do produto);
- f) Controlar mudanças do software: permitir flexibilidade no planejamento e execução do desenvolvimento e evolução dos requisitos, enfatizando as questões vitais de localizar mudanças e assegurar que tudo e todos estejam em sincronia.

Além das práticas supracitadas, de acordo com Krutchen (2003) três características importantes do *Rational Unified Process* valem ser mencionadas:

- a) Desenvolvimento orientado por caso de uso: os casos de uso são a fundação para o resto do processo de desenvolvimento, exercem um papel fundamental

em vários fluxos de processo, especialmente requisitos, projeto, teste e gerenciamento;

- b) Configuração de processo: seu uso como uma estrutura de processo que pode ser adaptada e estendida por uma organização que o esteja adotando, podendo ser modificados, personalizados, acrescentados ou suprimidos conforme necessidade;
- c) Suporte a ferramentas: necessidade de ferramentas adequadas de desenvolvimento de software para dar suporte ao processo (o *RUP* é suportado por uma vasta palheta de ferramentas que automatizam as etapas em muitas atividades).

O *RUP*, porém, contém diversas outras práticas em todos os aspectos de desenvolvimento de software.

2.2.2 Fases do RUP

O *RUP* organiza o desenvolvimento de software em quatro fases, onde são tratadas questões sobre planejamento, levantamento de requisitos, análise, implementação, teste e implantação do software. Cada fase tem um papel fundamental para que o objetivo seja cumprido (MARTINEZ, 2014).

As fases são compostas de iterações. As iterações são janelas de tempo e possuem prazo definido enquanto as fases são objetivas. Todas as fases geram artefatos que serão utilizados nas próximas fases e documentam o projeto, além de permitir melhor acompanhamento (WIKIPEDIA, 2015).

A Figura 1 mostra o ciclo de vida do RUP. Segundo Kruchten (2003), o processo tem duas dimensões:

- O eixo horizontal representa o tempo e mostra os aspectos do ciclo de vida do processo como se desdobra (aspecto dinâmico expresso em termos de ciclos, fases, iterações, marcos);
- O eixo vertical representa os fluxos essenciais do processo, que agrupa logicamente as atividades por natureza (aspecto estático descrito em termos de componentes de processo, atividades, fluxos, artefatos, trabalhadores).

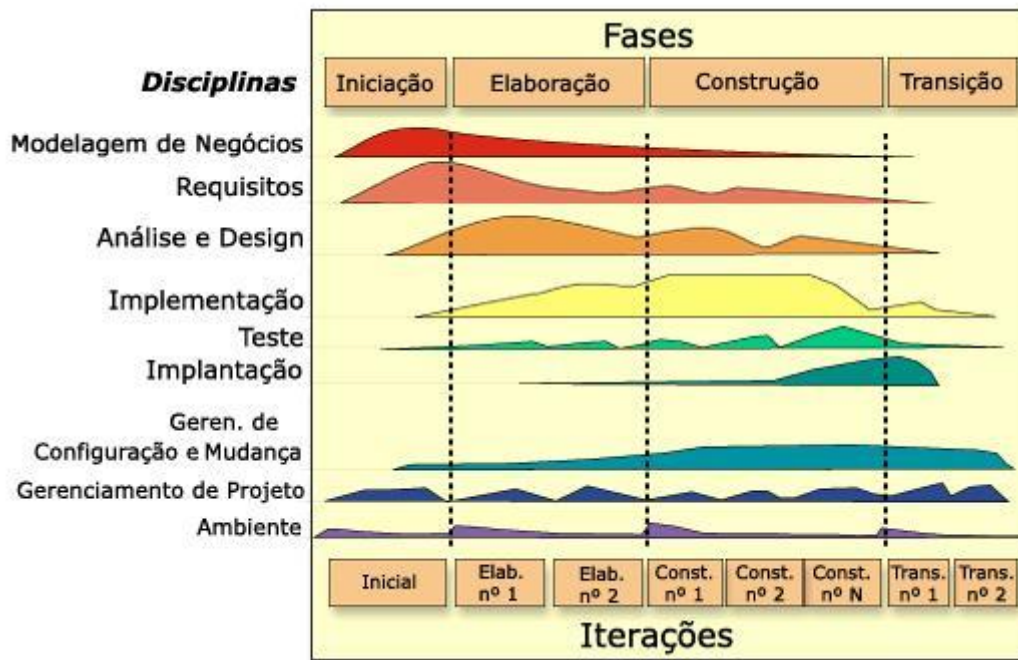


Figura 1 - Ciclo de vida do RUP (WIKIPEDIA, 2015)

Apesar de parecer um modelo em cascata, na verdade cada fase é composta de uma ou mais iterações, o que se assemelha a um modelo em espiral. Estas iterações são em geral curtas e abordam algumas poucas funções do sistema, reduzindo o impacto de mudanças, pois quanto menor o tempo, menor a probabilidade de haver uma mudança neste período para as funções em questão (BROERING, 2015).

As quatro fases do *RUP* são (KRUCHTEN, 2003), (MARTINEZ, 2014):

- Concepção: abrange as tarefas de comunicação com o cliente e planejamento. O objetivo desta fase é entender os requisitos globais e determinar a extensão do esforço de desenvolvimento, alcançando o consentimento de todos os interessados nos objetivos do ciclo de vida do projeto. Assim, haverá uma anuência das partes interessadas na definição do escopo do projeto, onde são examinados os objetivos para se decidir sobre a continuidade do desenvolvimento;
- Elaboração: abrange a modelagem do modelo genérico do processo. O objetivo desta fase é analisar o domínio do problema, estabelecer uma fundação arquitetônica executável que servirá como linha base para a próxima fase, desenvolver o plano de projeto e eliminar os elementos de alto

risco do projeto. Indagações como "O plano do projeto é confiável?", "Os custos são admissíveis?" são esclarecidas nesta etapa.

- c) Construção: desenvolve ou adquire os componentes de software. O objetivo desta fase é a construção do projeto e do sistema através do desenvolvimento dos componentes restantes e todas as características da aplicação são completamente testadas. É na fase de Construção que a maior parte de codificação ocorre;
- d) Transição: Abrange a entrega do software ao usuário e a fase de testes. O objetivo desta fase é levar o produto de software à comunidade de usuário, assegurando que o sistema tem o nível certo de qualidade para satisfazer seus objetivos. As atividades desta fase incluem o treinamento dos usuários finais e também a realização de testes da versão beta do sistema visando garantir que o mesmo possua o nível adequado de qualidade.

2.2.3 Fluxos do RUP

Além das fases e iterações, existem os fluxos. Cada fluxo é na verdade uma seqüência de tarefas encadeadas e relacionadas a um aspecto importante do projeto (BROERING, 2015).

Há nove fluxos no *Rational Unified Process* e eles representam uma partição de todos os trabalhadores e atividades em áreas de interesse ou disciplinas (Figura 1). Kruchten (2003) os divide em seis fluxos centrados na engenharia e três de suporte.

Cada um dos fluxos cobre determinado aspecto do sistema. Para decompô-los, o *RUP* usa detalhes de fluxo para expressar um grupo específico de atividades próximas relacionadas. Detalhes do fluxo também exibem o fluxo da informação (os artefatos que estão entrando e saindo das atividades), mostrando como as atividades interagem pelos vários artefatos (KRUCHTEN, 2003).

Os fluxos de engenharia são (KRUCHTEN, 2003):

- a) Modelagem de Negócio: descreve como desenvolver uma visão da nova organização alvo e, baseando-se nesta visão, definir os processos, papéis e responsabilidades daquela organização em um modelo de negócio;

- b) Requisitos: descreve como definir uma visão do sistema e traduzi-la num modelo de caso de uso que, com especificações adicionais, define os requisitos do software detalhados do sistema. Além disso, descreve como usar atributos de requisitos para auxiliar na administração da extensão e nas mudanças de requisitos do sistema;
- c) Análise e Projeto: traduzir os requisitos em uma especificação que descreva como implementar o sistema. O propósito da análise é transformar os requisitos do sistema para um conjunto de classes e subsistemas. O propósito do projeto é adaptar os resultados da análise às restrições impostas pelos requisitos não funcionais, o ambiente de implementação, os requisitos de desempenho, entre outros;
- d) Implementação: definir a organização do código em termos de subsistemas de implementação organizados em camadas, implementar classes e objetos em termos de componentes, testar os componentes desenvolvidos como unidades e integrar em um sistema executável os resultados produzidos por desenvolvedores individuais ou equipes;
- e) Teste: avaliar a qualidade do produto, o que não só envolve o produto final como inicia logo no projeto com a avaliação da arquitetura e continua pela avaliação do produto final entregue aos clientes;
- f) Implantação: entregar o produto pronto aos seus usuários através do teste do software em seu ambiente operacional final, empacotamento do software para entrega, distribuição do software, instalação do software, treinamento dos usuários finais, entre outros.

Os fluxos centrados no suporte são (KRUCHTEN, 2003):

- a) Gerenciamento de Projeto: equilibrar os objetivos concorrentes, gerenciar o risco e superar restrições para entregar um produto que satisfaça as necessidades dos clientes e dos usuários finais. Este fluxo no *RUP* não cobre questões como administração de pessoal, orçamento e contratos com os provedores e clientes;
- b) Configuração e Gerenciamento de Mudança: localizar e manter a integridade dos recursos de evolução do projeto, fornecendo a informação do estado dos

artefatos fundamentais do projeto e reunir as medidas relacionadas às suas mudanças;

- c) Ambiente: suportar a organização do desenvolvimento através de seleção e aquisição de ferramentas de configuração, configuração das ferramentas para harmonizar a organização, configuração de processo, melhoria do processo e prestação de serviços técnicos para suportar o processo.

Embora os nomes dos fluxos centrados na engenharia possam evocar as fases seqüenciais em um processo linear tradicional, as fases de um processo iterativo são diferentes. Estes fluxos são revisitados novamente e novamente ao longo do ciclo de vida. O fluxo atual completo de um projeto intercala estes nove fluxos e os repete com várias ênfases e níveis de intensidade a cada iteração (KRUCHTEN, 2003).

3 METODOLOGIA

3.1 METODOLOGIA

Este trabalho é caracterizado por uma pesquisa experimental, visto que todo o conhecimento adquirido será utilizado para um objetivo específico, o estudo da metodologia *Rational Unified Process* através do desenvolvimento de um software fictício.

Como metodologia de pesquisa para reunir informações foi utilizada a pesquisa bibliográfica, praticamente ao longo de todo o trabalho, a fim de obter conhecimento sobre o assunto abordado.

3.2 MODELO DE PROCESSO DE ENGENHARIA DE SOFTWARE

A metodologia de desenvolvimento de software utilizada para a construção do sistema foi a metodologia *RUP*, cuja revisão bibliográfica foi apresentada no capítulo 2.2 deste documento. Todos os detalhes relativos ao desenvolvimento do software e documentação do sistema estão descritos no capítulo 4 deste trabalho.

3.3 PLANO DE ATIVIDADE

O plano de atividades do projeto será demonstrado no WBS e no gráfico de Gantt apresentados a seguir. O trabalho foi dividido em iterações na fase de Elaboração. Como o software desenvolvido é fictício, as fases de Construção e Transição foram planejadas em conjunto.

3.3.1 WBS

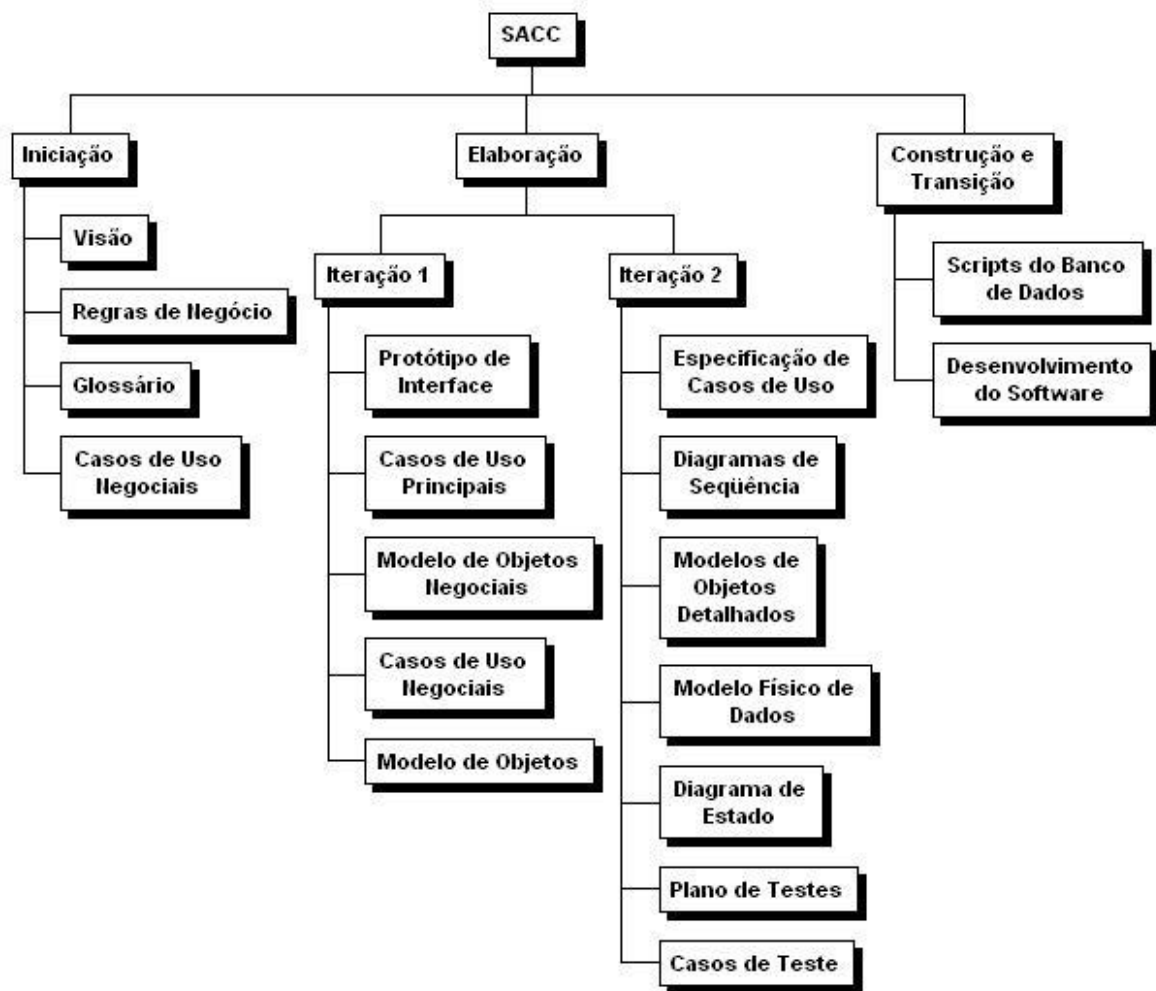


Figura 2 - Work Breakdown Structure

3.4 PLANO DE RISCOS

Abaixo é apresentado o plano de riscos elaborado para o projeto:

N	Condição	Limite	Conseqüência	Ação	Monitoramento	Probabilidade	Impacto	Classificação
1	Não adequação do trabalho aos objetivos da disciplina	20/08/13	Alteração do projeto	Negociar os prazos com o orientador	Orientador	Baixa	Alto	5
2	Deficiência na capacitação técnica	30/08/13	Atraso nas atividades de construção	Considerar contingência de tempo das atividades	Desenvolvedor	Baixa	Alto	5
3	Indisponibilidade do recurso humano	Sem Data	Inviabilidade da conclusão do projeto até a data limite	Negociar os prazos com o orientador	Todos	Muito Alta	Muito Alto	6

Tabela 1 - Plano de Riscos

3.5 RESPONSABILIDADES

Para a construção deste projeto necessita-se de um profissional que realize as atividades de confecção da monografia, gerenciamento de projetos, análise e arquitetura do sistema, bem como o seu desenvolvimento e testes. Portanto, o autor do trabalho possui responsabilidade de todas as atividades.

3.6 MATERIAIS

Como o projeto é fictício e será desenvolvido somente por uma pessoa, será utilizado 1 computador pessoal com processador de 2.1 GHz de 64 bits, 3 GB de RAM, 320 GB de HD e sistema operacional Windows Vista Home Premium. Não serão necessários equipamentos adicionais, servidores de desenvolvimento e de gerenciamento de configurações.

3.7 DESENVOLVIMENTO DO PROJETO

O desenvolvimento do projeto ocorreu de forma esperada, entretanto não no prazo esperado. O risco 3 (indisponibilidade do recurso humano) foi o risco mais evidente durante a realização projeto. A indisponibilidade do recurso ocorreu devido a atividades fora do escopo deste trabalho que tiveram prioridade à realização do mesmo. Durante o desenvolvimento os atrasos foram reportados ao orientador e foram tentados novos prazos, mesmo assim o trabalho corre o risco de não ser aceito.

4 APRESENTAÇÃO DO SOFTWARE

O produto deste projeto trata-se de um sistema que deve permitir a aprovação de contratos de câmbio pelas entidades corretora, cliente e banco através de alçadas, uma vez que o contrato seja inserido no sistema. O sistema deve controlar o fluxo de aprovação do contrato enquanto fica pendente com cada uma das entidades. Suas principais funcionalidades são:

- Possibilitar o cadastro das entidades corretora e cliente do contrato;
- Permitir o cadastro das pessoas autorizadas pelas entidades corretora e cliente do contrato;
- Permitir o cadastro do contrato de câmbio no sistema;
- Possibilitar que as pessoas que representem as entidades corretora, cliente e banco possam autorizar ou rejeitar os contratos da entidade que estão representando;
- Armazenar os dados de autorização e rejeição do contrato e a pessoa que realizou a ação para a consulta pelo usuário nos detalhes do contrato.

Uma vez que o projeto é fictício, não haverá necessidade de suporte e nem de gerência de configuração. Pelo mesmo motivo, o projeto não apresentará um plano de implantação e instalação.

Para o desenvolvimento do software foram utilizadas as seguintes ferramentas e tecnologias:

- Linguagem de programação: Java EE 7;
- IDE de Programação: NetBeans IDE 8.0.1;
- Servidor Web: Apache Tomcat 8.0.9;
- Banco de Dados: PostgreSQL 9.3;
- Ferramenta para gerência de banco de dados: pgAdmin III;
- Ferramenta para construção de artefatos UML: IBM Rational Software Architect 8.5;
- Ferramenta para modelagem de banco de dados: brModelo.

A seguir serão demonstrados os artefatos gerados para realização deste projeto, contextualizados nas fases e fluxos do *RUP*, conforme Plano de Atividade apresentado no capítulo 3.3.

4.1 INICIAÇÃO

4.1.1 Fluxo de Modelo de Negócio

Nesta etapa foram criados os artefatos:

- Documento de Visão: este documento apresenta a visão geral do processo de negócio e pode ser encontrado no Apêndice 1;
- Regras de Negócio: este documento contém as regras de definições de negócio e pode ser encontrado no Apêndice 2;
- Glossário: neste documento contém as definições do termo de negócio e pode ser encontrado no Apêndice 3;
- Casos de Uso Negociais: documento que descreve as funcionalidades macro do sistema e pode ser encontrado no Apêndice 4.

4.2 ELABORAÇÃO

4.2.1 Fluxo de Requisitos

Nesta etapa foram desenvolvidos os artefatos:

- Protótipo de Interfaces: apresentação do esboço das telas do sistema e pode ser encontrado no Apêndice 5;
- Casos de Uso Principais: diagrama dos casos de uso principais do sistema e pode ser encontrado no Apêndice 6;
- Modelo de Objetos Negociais: diagrama de classes de negócio do sistema (sem atributos) e pode ser encontrado no Apêndice 7.

4.2.2 Fluxo de Análise e Design

Nesta etapa foram desenvolvidos os artefatos:

- Casos de Uso Negociais: diagrama e especificação detalhada de casos de uso e pode ser encontrado no Apêndice 8;

- Modelo de Objetos: diagrama de classes com todas as classes e atributos e pode ser encontrado no Apêndice 9;
- Especificação de Casos de Uso: diagrama e especificação detalhada de todos os casos de uso e pode ser encontrado no Apêndice 10;
- Diagramas de Seqüência: diagramas de seqüência de todos os casos de uso e podem ser encontrados no Apêndice 11;
- Modelos de Objetos Detalhados: diagrama de classes com atributos e métodos e pode ser encontrado no Apêndice 12;
- Modelo Físico de Dados: tabelas do banco de dados e script para criação do banco de dados e podem ser encontrados no Apêndice 13;
- Diagrama de Estados: diagrama suplementar necessário para entendimento do sistema e pode ser encontrado no Apêndice 14;

4.2.3 Fluxo de Testes

Nesta etapa foram criados os artefatos:

- Plano de Testes: planejamento de como ocorrerão os testes do sistema e pode ser encontrado no Apêndice 15;
- Casos de Teste: detalhamento dos testes que serão realizados no sistema e podem ser encontrados no Apêndice 16;

4.3 CONSTRUÇÃO E TRANSIÇÃO

4.3.1 Workflow de Implementação

Nesta etapa foram gerados os artefatos:

- Scripts do Banco de Dados: scripts para popular as bases de Entidade, Pessoa Física e Funcionário e podem ser encontrados no Apêndice 17;
- Código Fonte: classes Java, arquivos XML e páginas JSP. Para a geração do código fonte as classes foram estruturadas em camadas, sendo as classes relativas à persistência no pacote *DAO*, a classe que trata as requisições web no pacote *SERVLET*, as classes que representam as entidades do banco de dados no pacote *BEANS* e as classes de apoio no pacote *UTIL*. A estrutura de diretórios do sistema é apresentada na Figura 4, a seguir.



Figura 3 – Estrutura de Diretórios

5 CONSIDERAÇÕES FINAIS

De uma forma geral objetivo do projeto foi alcançado. O estudo e aplicação do *Rational Unified Process* foram realizados e apresentados durante o desenvolvimento do trabalho.

5.1 QUANTO AOS OBJETIVOS DO PROJETO

Quanto ao objetivo específico “*realizar o estudo do processo de modelagem do Rational Unified Process por meio da bibliografia existente*”, no capítulo x.x deste trabalho foi apresentado o resultado da pesquisa bibliográfica realizada sobre o *RUP*.

Quanto ao objetivo específico “*avaliação da aplicabilidade do RUP através do desenvolvimento de um software simples*”, no capítulo 4 deste trabalho foram apresentados os artefatos do *RUP* desenvolvidos para o projeto. Sob o ponto de vista de especificação, a partir dos artefatos gerados o sistema comportou-se conforme o esperado. Entretanto, a quantidade de artefatos gerada foi desnecessária para esse tipo de sistema, visto que o software que foi desenvolvido é simples e a equipe foi composta por uma pessoa. Portanto, mesmo com o software construído de acordo com a especificação, percebeu-se que o *RUP* é indicado para projetos com times maiores e mais complexos que o desenvolvido, pois fornece várias formas de alinhar a comunicação, gestão e requisitos de softwares robustos.

REFERÊNCIAS

_____. brModelo. **Ferramenta de Modelagem Conceitual de Banco de Dados**. Disponível em: <<http://sis4.com/brModelo/monografia/monografia.htm>>. Acessado em: 25 nov 2014.

BROERING, Evandro. RUP – Rational Unified Process. Disponível em: <<http://www.devmedia.com.br/rup-rational-unified-process/4574>>. Acessado em: 10 jan 2015.

FAGUNDES, Priscila Bastos. **Framework para Comparação e Análise de Métodos Ágeis**. Dissertação de Mestrado. Universidade Federal de Santa Catarina. Florianópolis: 2005.

FELIPPE, Cecília Machado de. **Aplicação de Metodologia Ágil no Desenvolvimento de um Componente de Software para Prefeituras**. Trabalho de Conclusão de Curso. Universidade Federal de Santa Catarina. Florianópolis: 2007.

HEUSER, Carlos Alberto. **Projeto de Banco de Dados**. 6. ed. Volume 4 da Série Livros Didáticos Informática UFRGS. Porto Alegre: Bookman, 2009.

KRUTCHTEN, Philippe. **Introdução ao RUP - Rational Unified Process**. Rio de Janeiro: Editora Ciência Moderna Ltda., 2003.

LARMAN, Craig. **Applying UML and Patterns**. 2nd Edition, 2002.

MARTINEZ, Marina. **RUP - Rational Unified Process - Desenvolvimento de Softwares**. Disponível em: <<http://www.infoescola.com/engenharia-de-software/rup/>>. Acessado em: 20 dez 2014.

NETO, Oscar Nogueira de Souza. **Análise Comparativa das Metodologias de Desenvolvimento de Softwares Tradicionais e Ágeis**. Trabalho de Conclusão de Curso. Universidade da Amazônia. Belém: 2004.

_____. PostgreSQL. **PostgreSQL 9.1.4 Documentation**. Disponível em: <<http://www.postgresql.org/docs/9.1/static/index.html>>. Acessado em: 15 jun 2014.

PRESSMAN, Roger. **Software Engineering - A Practitioner's Approach**. McGraw-Hill, 5th Edition, 2001.

SOMMERVILLE, Ian. **Software Engineering**. 7th Edition, 2004.

_____. Wikipedia. IBM Rational Unified Process. Disponível em: <http://pt.wikipedia.org/wiki/IBM_Rational_Unified_Process>. Acessado em: 15 jan 2015.

APÊNDICE 1

Visão

Introdução

A finalidade deste documento é coletar, analisar e definir as necessidades e características de nível superior do SACC (Sistema de Aprovação de Contratos de Câmbio). Ele enfoca os recursos de que os envolvidos e usuários precisam para automatizar o processo de aprovação de contratos de câmbio. Os detalhes de como o SACC atende a essas necessidades estão descritos nas especificações suplementares e de caso de uso.

Posicionamento

Descrição do Problema

Um banco possui um processo de compra e venda de moeda estrangeira manual para as empresas que são seus clientes. Quando um cliente deseja comprar ou vender uma quantia em moeda estrangeira deve ser gerado um contrato de câmbio que precisa ter a aprovação das pessoas responsáveis pelas 3 entidades do contrato: a corretora de câmbio que irá realizar a transação, a empresa cliente que deseja realizar a transação e do banco. O aceite das entidades corretora de câmbio e cliente do contrato é realizado por pessoas autorizadas através de um cadastro de alçadas. Funcionários não necessitam de alçada cadastrada, pois podem realizar aprovações em qualquer contrato pela entidade banco.

O problema	Processo manual de aprovação de contrato de câmbio
afeta	Corretoras de Câmbio, Clientes Pessoa Jurídica e Funcionários do banco.
cujo impacto é	Quando um contrato de câmbio é gerado as entidades do contrato devem ir até a instituição bancária realizar a aprovação do mesmo manualmente. Também, não se tem o controle de quantos contratos precisam ser aprovados por uma determinada entidade.
uma boa solução seria	Um sistema de informação que realize a gestão de aprovação de contratos de câmbio para seus clientes pessoa jurídica. Deseja-se que o funcionário do banco realize o cadastro do contrato e que o sistema controle a ordem e as permissões das aprovações dos usuários (corretora, cliente e banco), permitindo assim que o fluxo de aprovação seja automático.

	Uma vez cadastrada a alçada para um usuário, este terá permissão para realizar a aprovação do contrato ou para rejeitar o mesmo. Logo, o sistema também deverá armazenar os detalhes das ações das pessoas de cada entidade do contrato
--	---

Descrições dos Envolvidos e Usuários

O sistema está direcionado a todas as pessoas envolvidas no processo de aprovação de contratos de câmbio.

Resumo dos Envolvidos

Nome	Descrição	Responsabilidades
Funcionário	Funcionário do banco.	Cadastrar outras entidades dos contratos, cadastrar pessoas autorizadas
Corretora	Corretora de Câmbio	Entidade participante do contrato.
Cliente	Cliente Pessoa Jurídica	Entidade participante do contrato.

Resumo dos Usuários

Nome	Descrição	Envolvido
Funcionário	Cadastrar entidades corretora e cliente, cadastrar contratos, cadastrar alçadas, visualizar, aprovar e rejeitar todos os contratos.	Auto-representado
Corretora	Visualizar, aprovar e rejeitar contratos de suas entidades.	Auto-representado
Cliente	Visualizar, aprovar e rejeitar contratos de suas entidades.	Auto-representado

Visão Geral do Produto

O presente projeto consiste no desenvolvimento de um sistema de informação para uma entidade bancária que deseja automatizar o processo de aprovação de contratos de câmbio de seus clientes

pessoa jurídica. O projeto deve ser modelado de acordo com a metodologia RUP, desenvolvido em Java e banco de dados PostgreSQL.

O projeto deve:

- Permitir o cadastro de alçadas para as entidades corretora e cliente;
- Permitir aos usuários que realizem a consulta, aprovação ou rejeição de contratos das entidades que representam;
- Agilizar o processo de aprovação de contratos de câmbio

APÊNDICE 2

Regras de Negócio

Regras de Negócios

1. Introdução

Este documento apresenta as definições de negócio pertinentes ao sistema a ser desenvolvido e deve ser utilizado como guia para geração de todos os artefatos do sistema.

1.1 Finalidade

A finalidade deste documento é definir as regras de negócio essenciais para o Sistema de Aprovação de Contratos de Câmbio. Todos os artefatos do projeto irão contemplar as regras descritas neste documento.

2. Definições

2.1 Regra de Negócio 1 – Menu de Acesso

O menu de acesso ao sistema deve ser apresentado conforme perfil do usuário logado.

2.2 Regra de Negócio 2 – CNPJ da Entidade

O CNPJ cadastrado para uma entidade deve ser único no sistema. Um mesmo CNPJ não pode ser associado como corretora e cliente ao mesmo tempo.

2.3 Regra de Negócio 3 – Alçadas

Devem ser cadastradas alçadas únicas no sistema. Uma pessoa não pode estar vinculada a mais de uma entidade no sistema.

2.4 Regra de Negócio 4 – Entidades válidas no contrato

As entidades corretora e cliente de um contrato devem ser válidas no sistema. As entidades para pertencerem a um contrato devem ter sido previamente cadastradas no sistema pelo funcionário do banco.

2.5 Regra de Negócio 5 – Consulta de contratos por entidade

O usuário logado somente poderá visualizar os contratos da entidade na qual ele representa. O funcionário do banco por sua vez poderá visualizar todos os contratos do sistema.

2.6 Regra de Negócio 7 – Ações de acordo com o status do contrato

As ações de aprovação ou rejeição de somente devem ser realizadas por usuários que representem a entidade na qual o contrato está pendente. O contrato só poderá ser aprovado ou rejeitado quando o status estiver pendente com a entidade em que o usuário logado esteja representando.

2.7 Regra de Negócio 7 – Workflow de status de contrato

Uma vez cadastrado no sistema, o status do contrato de câmbio deve ser “Pendente Corretora”. Quando o representante da corretora de câmbio do contrato realizar a aprovação, o status do contrato deve ser “Pendente Cliente”. Quando o mesmo for aprovado pelo representante da entidade cliente do contrato, o status passa a ser “Pendente Banco”. Quando o contrato for aprovado pelo funcionário do banco, o contrato passa a ter o status “Aprovado”. A qualquer momento pode haver a rejeição do contrato pelo representante da entidade na qual o contrato está pendente e neste caso o status do contrato será “Rejeitado”.

APÊNDICE 3

Glossário

Glossário de Negócios

1. Introdução

Este documento pode ser usado como um dicionário de dados informal, capturando definições de dados para que as descrições de casos de uso e outros documentos do projeto possam se concentrar no que o sistema deve fazer com as informações.

1.1 Finalidade

A finalidade deste documento é definir a terminologia específica para o desenvolvimento do Sistema de Aprovação de Contratos de Câmbio.

2. Definições

2.1 Entidade

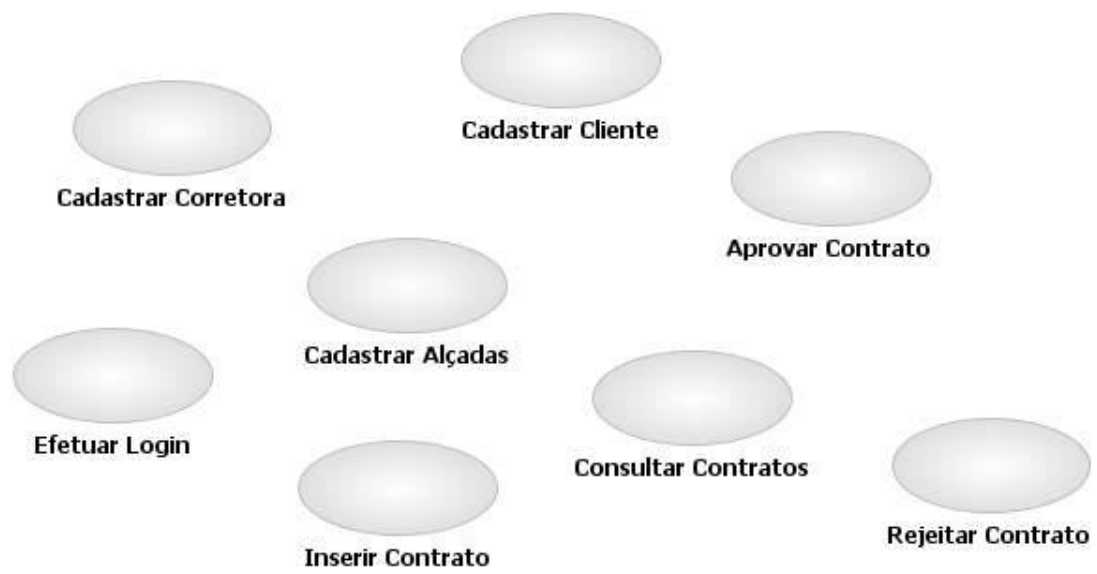
Parte representante do contrato de câmbio. No escopo do sistema, as entidades que fazem parte do contrato de câmbio são: entidade corretora, entidade cliente e entidade banco. As entidades corretora e cliente são representadas por pessoas nas quais precisam possuir alçada cadastrada para representar uma corretora ou um cliente específico. A entidade banco é representada pelo funcionário do banco.

2.2 Alçada

Permissão concedida a uma pessoa para representar uma corretora ou um cliente. No sistema, a permissão para uma pessoa representar uma corretora ou um cliente específico é concedida pelo funcionário do banco.

APÊNDICE 4

Diagrama de Caso de Uso Negocial



APÊNDICE 5

Protótipo de Interfaces

Login

Login:

Senha:

Cadastro de Corretora

CNPJ:

Razão Social:

Cadastro de Cliente

CNPJ:

Razão Social:

Cadastro de Alçada

CNPJ Entidade:

CPF Pessoa:

Cadastrar Contrato

CNPJ Corretora:

Razão Social:

CNPJ Cliente:

Razão Social:

Tipo: Moeda:

Valor Moeda Nacional: Valor Moeda Estrangeira:

Contratos de Câmbio

Número	Corretora	Cliente	Tipo	Valor	Moeda	Status
--------	-----------	---------	------	-------	-------	--------

Detalhes Contrato

CNPJ Corretora:

Razão Social:

CNPJ Cliente:

Razão Social:

Tipo:

Moeda:

Valor Moeda Nacional: Valor Moeda Estrangeira:

Status:

Histórico de Aprovações

Ação	Entidade	CPF	Nome	Data
------	----------	-----	------	------

[Aprovar](#) [Rejeitar](#)

Aprovar Contrato

CNPJ Corretora:

Razão Social:

CNPJ Cliente:

Razão Social:

Tipo:

Moeda:

Valor Moeda Nacional: Valor Moeda Estrangeira:

Status:

Rejeitar Contrato

CNPJ Corretora:

Razão Social:

CNPJ Cliente:

Razão Social:

Tipo:

Moeda:

Valor Moeda Nacional: Valor Moeda Estrangeira:

Status:

APÊNDICE 6

Especificação de Caso de Uso

UC001 – Efetuar Login

Controle do Documento

Versão	Autor	Data	Descrição
1.0	Cecília	11/12/2012	Elaboração do documento.

Descrição

Este caso de uso serve para validar e autenticar o usuário no sistema.

Data Views

DV1 - Tela de Login



The image shows a screenshot of a web-based login form. The form is titled "Login" in a bold, dark blue font. Below the title, there are two input fields: the first is labeled "Login:" and the second is labeled "Senha:". Below these fields are two buttons: "Login" and "Limpar". The entire form is enclosed in a dark red border.

Pré-condições

Este use case pode iniciar somente se:

1. O sistema estiver disponível.

Pós-condições

Após o fim normal deste use case o sistema deve:

1. Ter validado e autenticado usuário;
2. Apresentar o menu do sistema de acordo com o perfil do usuário.

Ator Primário

Funcionário

Fluxo de Eventos Principal

1. O sistema apresenta a tela **DV1 – Tela de Login**; (DV1)
2. O usuário preenche os campos da tela e clica no botão **Entrar**; (A1) (R1) (R2) (E1) (E2)
3. O sistema autentica o usuário e apresenta o menu do sistema. (R3)
4. O use case é encerrado.

Fluxos Alternativos

A1: Limpar campos.

1. O usuário clica no botão **Limpar**;
2. O sistema limpa os campos da tela;
3. O use case retorna ao passo 2 do Fluxo Principal.

Fluxos de Exceção

E1. Campos obrigatórios não preenchidos (R1)

1. O sistema apresenta a mensagem “Campo(s) obrigatório(s) não preenchido(s).”.
2. O sistema apresenta uma lista com os campos obrigatórios que não foram preenchidos.
3. O use case retorna ao passo 2 do Fluxo Principal.

E2. Usuário/Senha inválido(s) (R2)

1. O sistema apresenta a mensagem “Usuário e/ou Senha inválido(s).”.

2. O use case retorna ao passo 2 do Fluxo Principal.

Regras de Negócio

R1. Os campos Usuário e Senha são obrigatórios.

R2. Os campos Usuário e Senha devem ser preenchidos com valores válidos.

R3. O sistema deve apresentar o menu de acesso conforme perfil do usuário logado.

UC002 – Cadastrar Corretora

Controle do Documento

Versão	Autor	Data	Descrição
1.0	Cecília	11/12/2012	Elaboração do documento.

Descrição

Este caso de uso serve para cadastrar Corretoras de Câmbio no sistema.

Data Views

DV1 - Tela de Cadastro de Corretora



A captura de tela mostra uma interface web com o título "Cadastro de Corretora" em azul. Abaixo do título, há dois campos de entrada de texto: "CNPJ:" e "Razão Social:". Abaixo dos campos, há um botão cinza com o texto "Cadastrar".

Pré-condições

Este use case pode iniciar somente se:

1. O usuário estiver logado e autenticado no perfil Banco.

Pós-condições

Após o fim normal deste use case o sistema deve:

1. Ter cadastrado os dados da corretora.

Ator Primário

Funcionário

Fluxo de Eventos Principal

1. O sistema apresenta a tela **DV1 – Tela de Cadastro de Corretora**; (DV1)
2. O usuário preenche os campos da tela e clica no botão **Salvar**; (R1) (R2) (R3) (E1) (E2) (E3)
3. O sistema consiste os dados e apresenta a mensagem “Corretora cadastrada com sucesso.”.
4. O use case é encerrado.

Fluxos Alternativos

N/A

Fluxos de Exceção

E1. Campos obrigatórios não preenchidos (R1)

1. O sistema apresenta a mensagem “Campo(s) obrigatório(s) não preenchido(s).”.
2. O sistema apresenta uma lista com os campos obrigatórios que não foram preenchidos.
3. O use case retorna ao passo 2 do Fluxo Principal.

E2. CNPJ Inválido (R2)

1. O sistema apresenta a mensagem “CNPJ inválido.”.
2. O use case retorna ao passo 2 do Fluxo Principal.

E3. CNPJ existente (R3)

1. O sistema apresenta a mensagem “CNPJ já cadastrado no sistema.”.
2. O use case retorna ao passo 2 do Fluxo Principal.

Regras de Negócio

R1. Os campos CNPJ e Razão Social são obrigatórios.

R2. O campo CNPJ deve ser um CNPJ válido.

R3. O CNPJ informado deve ser único no sistema.

UC003 – Cadastrar Cliente

Controle do Documento

Versão	Autor	Data	Descrição
1.0	Cecília	11/12/2012	Elaboração do documento.

Descrição

Este caso de uso serve para cadastrar Clientes Pessoa Jurídica no sistema.

Data Views

DV1 - Tela de Cadastro de Cliente



Cadastro de Cliente

CNPJ:

Razão Social:

Pré-condições

Este use case pode iniciar somente se:

1. O usuário estiver logado e autenticado no perfil Banco.

Pós-condições

Após o fim normal deste use case o sistema deve:

1. Ter cadastrado os dados do cliente.

Ator Primário

Funcionário

Fluxo de Eventos Principal

1. O sistema apresenta a tela **DV1 – Tela de Cadastro de Cliente**; (DV1)
2. O usuário preenche os campos da tela e clica no botão **Salvar**; (R1) (R2) (R3) (E1) (E2) (E3)
3. O sistema consiste os dados e apresenta a mensagem “Cliente cadastrado com sucesso.”.
4. O use case é encerrado.

Fluxos Alternativos

N/A

Fluxos de Exceção

E1. Campos obrigatórios não preenchidos (R1)

1. O sistema apresenta a mensagem “Campo(s) obrigatório(s) não preenchido(s).”.
2. O sistema apresenta uma lista com os campos obrigatórios que não foram preenchidos.
3. O use case retorna ao passo 2 do Fluxo Principal.

E2. CNPJ Inválido (R2)

1. O sistema apresenta a mensagem “CNPJ inválido.”.
2. O use case retorna ao passo 2 do Fluxo Principal.

E3. CNPJ existente (R3)

1. O sistema apresenta a mensagem “CNPJ já cadastrado no sistema.”.
2. O use case retorna ao passo 2 do Fluxo Principal.

Regras de Negócio

R1. Os campos CNPJ e Razão Social são obrigatórios.

R2. O campo CNPJ deve ser um CNPJ válido.

R3. O CNPJ informado deve ser único no sistema.

UC004 – Cadastrar Alçadas

Controle do Documento

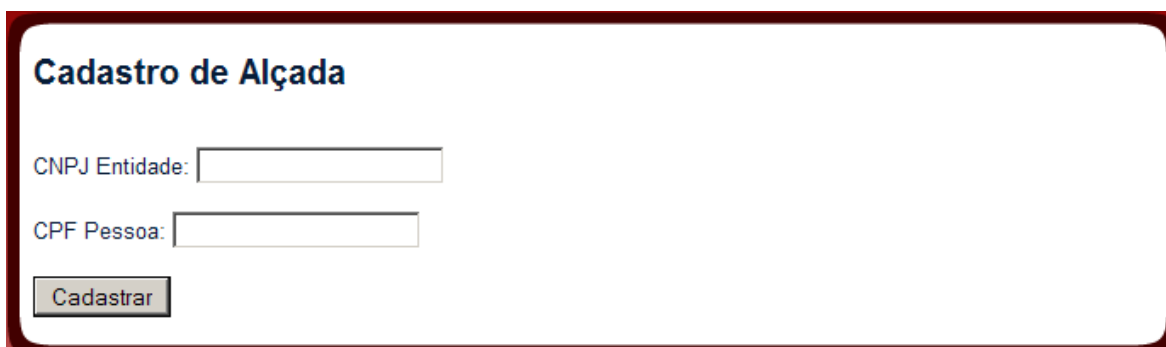
Versão	Autor	Data	Descrição
1.0	Cecília	11/12/2012	Elaboração do documento.

Descrição

Este caso de uso serve para cadastrar os usuários que representam as entidades Corretora e Cliente.

Data Views

DV1 - Tela de Cadastro de Alçadas



Cadastro de Alçada

CNPJ Entidade:

CPF Pessoa:

Pré-condições

Este use case pode iniciar somente se:

1. O usuário estiver logado e autenticado no perfil Banco.

Pós-condições

Após o fim normal deste use case o sistema deve:

1. Ter associado o CPF do usuário ao CNPJ da entidade.

Ator Primário

Funcionário

Fluxo de Eventos Principal

1. O sistema apresenta a tela **DV1 – Tela de Cadastro de Alçadas**; (DV1)
2. O usuário preenche os campos da tela e clica no botão **Salvar**; (R1) (R2) (R3) (R4) (R5) (E1) (E2) (E3) (E4) (E5)
3. O sistema apresenta a mensagem “Alçada cadastrada com sucesso”.
4. O use case é encerrado.

Fluxos Alternativos

N/A

Fluxos de Exceção

E1. Campos obrigatórios não preenchidos (R1)

1. O sistema apresenta a mensagem “Campo(s) obrigatório(s) não preenchido(s).”.
2. O sistema apresenta uma lista com os campos obrigatórios que não foram preenchidos.
3. O use case retorna ao passo 2 do Fluxo Principal.

E2. CNPJ inexistente (R2)

1. O sistema apresenta a mensagem “CNPJ não cadastrado no sistema.”.
2. O use case retorna ao passo 2 do Fluxo Principal.

E3. CPF inexistente (R3)

1. O sistema apresenta a mensagem “CPF não cadastrado no sistema.”.
2. O use case retorna ao passo 2 do Fluxo Principal.

E4. Data Final de Vigência inválida (R4)

1. O sistema apresenta a mensagem “A Data Final de Vigência deve ser maior que a data atual.”.
2. O use case retorna ao passo 2 do Fluxo Principal.

E5. Alçada existente (R5)

1. O sistema apresenta a mensagem “Alçada já cadastrada no sistema”.

2. O use case retorna ao passo 2 do Fluxo Principal.

Regras de Negócio

R1. Os campos CNPJ, CPF e Data Final de Vigência são obrigatórios.

R2. O campo CNPJ deve ser um CNPJ existente no sistema.

R3. O campo CPF deve ser um CPF existente no sistema.

R4. O campo Data Final de Vigência deve ser maior que a data atual.

R5. A alçada informada deve ser única no sistema.

UC005 – Inserir Contrato

Controle do Documento

Versão	Autor	Data	Descrição
1.0	Cecília	11/12/2012	Elaboração do documento.

Descrição

Este caso de uso serve para cadastrar um contrato de câmbio no sistema.

Data Views

DV1 - Tela de Cadastro de Contrato

Cadastrar Contrato

CNPJ Corretora:

Razão Social:

CNPJ Cliente:

Razão Social:

Tipo: Moeda:

Valor Moeda Nacional: Valor Moeda Estrangeira:

Pré-condições

Este use case pode iniciar somente se:

1. O usuário estiver logado e autenticado no perfil Banco.

Pós-condições

Após o fim normal deste use case o sistema deve:

1. Ter cadastrado os dados do contrato de câmbio.

Ator Primário

Funcionário

Fluxo de Eventos Principal

1. O sistema apresenta a tela **DV1 – Tela de Cadastro de Contrato**; (DV1)
2. O usuário preenche os campos da tela e clica no botão **Inserir**; (R1) (R2) (R3) (E1) (E2) (E3)
3. O sistema consiste os dados e apresenta a mensagem “Contrato de Câmbio inserido com sucesso.”. (R5)
4. O use case é encerrado.

Fluxos Alternativos

N/A

Fluxos de Exceção

E1. Campos obrigatórios não preenchidos (R1)

1. O sistema apresenta a mensagem “Campo(s) obrigatório(s) não preenchido(s).”.
2. O sistema apresenta uma lista com os campos obrigatórios que não foram preenchidos.
3. O use case retorna ao passo 2 do Fluxo Principal.

E2. CNPJ Corretora inexistente (R2)

1. O sistema apresenta a mensagem “CNPJ Corretora não cadastrado no sistema.”.
2. O use case retorna ao passo 2 do Fluxo Principal.

E3. CNPJ Cliente inexistente (R3)

1. O sistema apresenta a mensagem “CNPJ Cliente não cadastrado no sistema.”.
2. O use case retorna ao passo 2 do Fluxo Principal.

E4. Data de Encerramento inválida (R4)

1. O sistema apresenta a mensagem “A Data de Encerramento deve ser maior que a data atual.”.
2. O use case retorna ao passo 2 do Fluxo Principal.

Regras de Negócio

- R1.** Os campos CNPJ Corretora, CNPJ Cliente, são obrigatórios.
- R2.** O campo CNPJ Corretora deve ser um CNPJ existente no sistema.
- R3.** O campo CNPJ Cliente deve ser um CNPJ existente no sistema.
- R4.** O campo Data Final de Vigência deve ser maior que a data atual.
- R5.** Ao inserir o contrato o sistema deve gerar um número de contrato seqüencial.

UC006 – Consultar Contratos

Controle do Documento

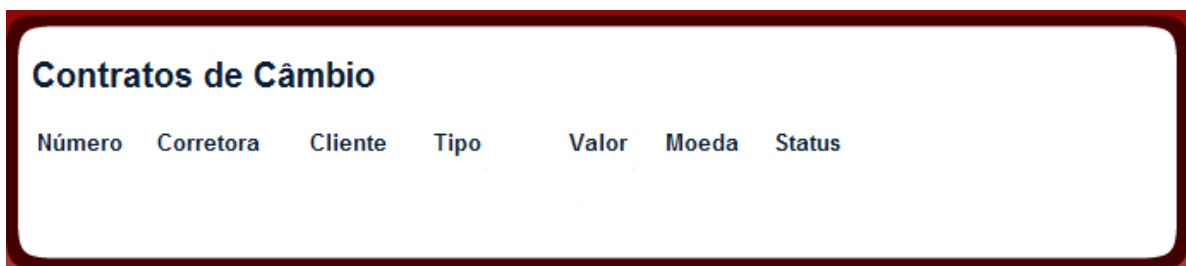
Versão	Autor	Data	Descrição
1.0	Cecília	11/12/2012	Elaboração do documento.

Descrição

Este caso de uso serve para consultar contratos de câmbio no sistema.

Data Views

DV1 - Tela de Consulta de Contratos



Contratos de Câmbio

Número	Corretora	Cliente	Tipo	Valor	Moeda	Status
--------	-----------	---------	------	-------	-------	--------

DV2 - Tela de Detalhes do Contrato

Detalhes Contrato

CNPJ Corretora:

Razão Social:

CNPJ Cliente:

Razão Social:

Tipo:

Moeda:

Valor Moeda Nacional: Valor Moeda Estrangeira:

Status:

Histórico de Aprovações

Ação	Entidade	CPF	Nome	Data
------	----------	-----	------	------

[Aprovar](#) [Rejeitar](#)

Pré-condições

Este use case pode iniciar somente se:

1. O usuário estiver logado e autenticado nos perfis Corretora, Cliente ou Banco.

Pós-condições

Após o fim normal deste use case o sistema deve:

1. Ter visualizado dados de contratos de câmbio.

Ator Primário

Funcionário / Corretora / Cliente

Fluxo de Eventos Principal

1. O sistema apresenta a tela **DV1 – Tela de Consulta de Contratos**; (DV1) (R1) (R2) (E1)
2. O usuário clica no link do número do contrato; (A1) (A2)
3. O sistema apresenta a tela **DV2 – Tela de Detalhes do Contrato**;
4. O use case é encerrado.

Fluxos Alternativos

A1: Aprovar contrato.

1. O usuário clica no link **Aprovar**;
2. << Extends – SACC_UC007 – Aprovar Contrato >>

A2: Rejeitar Contrato.

1. O usuário clica no link **Rejeitar**;
2. << Extends – SACC_UC008 – Rejeitar Contrato >>

Fluxos de Exceção

E1. Nenhum registro encontrado (R1)

1. O sistema apresenta a mensagem “Não existem contratos.”.
2. O use case é encerrado.

Regras de Negócio

R1. O sistema deve retornar todos os contratos da entidade na qual o usuário logado possui alçada cadastrada, ordenados pela data de criação do contrato.

R2. O sistema deve habilitar os links Aprovar e Rejeitar se o contrato estiver pendente no status da entidade na qual o usuário está representando.

UC007 – Aprovar Contrato

Controle do Documento

Versão	Autor	Data	Descrição
1.0	Cecília	11/12/2012	Elaboração do documento.

Descrição

Este caso de uso serve para realizar a aprovação por uma entidade das entidades do contrato.

Data Views

DV1 - Tela de Aprovação

Aprovar Contrato

CNPJ Corretora:

Razão Social:

CNPJ Cliente:

Razão Social:

Tipo:

Moeda:

Valor Moeda Nacional: Valor Moeda Estrangeira:

Status:

Pré-condições

Este use case pode iniciar somente se:

1. O usuário estiver logado e autenticado nos perfis Corretora, Cliente ou Banco;

2. Ter consultado contratos e ter clicado no link Aprovar.

Pós-condições

Após o fim normal deste use case o sistema deve:

1. Ter realizado a aprovação do contrato por uma das entidades.

Ator Primário

Funcionário / Corretora / Cliente

Fluxo de Eventos Principal

1. O sistema apresenta a tela **DV1 – Tela de Aprovação**; (DV1)
2. O usuário visualiza os campos da tela e clica no botão **Aprovar Contrato**;
O sistema consiste a aprovação e apresenta a mensagem “Aprovação cadastrada.”. (R1) (R2)
3. O use case é encerrado.

Fluxos Alternativos

N/A

Fluxos de Exceção

N/A

Regras de Negócio

R1. O sistema deve alterar o status do contrato como segue:

- se PENDENTE CORRETORA – atualizar para PENDENTE CLIENTE.
- se PENDENTE CLIENTE, atualizar para PENDENTE BANCO;
- se PENDENTE BANCO, atualizar para APROVADO.

R2. O sistema deve armazenar os dados da aprovação do contrato.

UC008 – Rejeitar Contrato

Controle do Documento

Versão	Autor	Data	Descrição
1.0	Cecília	11/12/2012	Elaboração do documento.

Descrição

Este caso de uso serve para realizar a rejeição por uma entidade das entidades do contrato.

Data Views

DV1 - Tela de Rejeição

Rejeitar Contrato

CNPJ Corretora:

Razão Social:

CNPJ Cliente:

Razão Social:

Tipo:

Moeda:

Valor Moeda Nacional: Valor Moeda Estrangeira:

Status:

Pré-condições

Este use case pode iniciar somente se:

1. O usuário estiver logado e autenticado nos perfis Corretora, Cliente ou Banco;

2. Ter consultado contratos e ter clicado no link Rejeitar.

Pós-condições

Após o fim normal deste use case o sistema deve:

1. Ter realizado a rejeição do contrato por uma das entidades.

Ator Primário

Funcionário / Corretora / Cliente

Fluxo de Eventos Principal

1. O sistema apresenta a tela **DV1 – Tela de Rejeição**; (DV1)
2. O usuário visualiza os campos da tela e clica no botão **Rejeitar Contrato**;
O sistema consiste a aprovação e apresenta a mensagem “Rejeição cadastrada.”. (R1) (R2)
3. O use case é encerrado.

Fluxos Alternativos

N/A

Fluxos de Exceção

N/A

Regras de Negócio

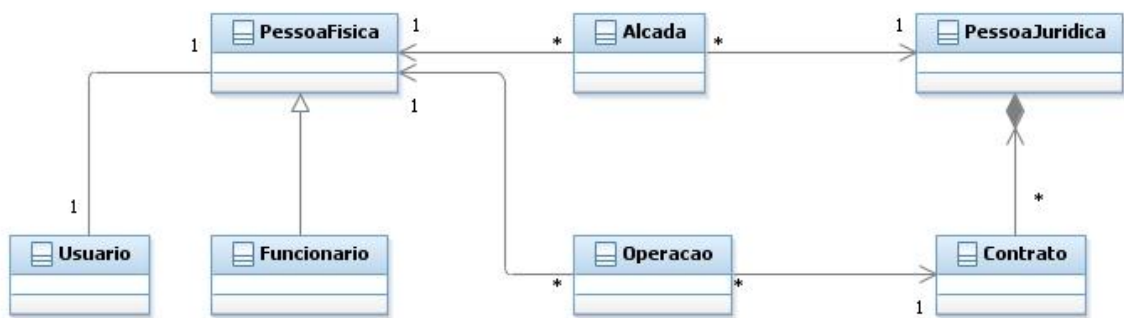
R1. O sistema deve alterar o status para REJEITADO.

R2. O sistema deve armazenar os dados da rejeição do contrato.

APÊNDICE 7

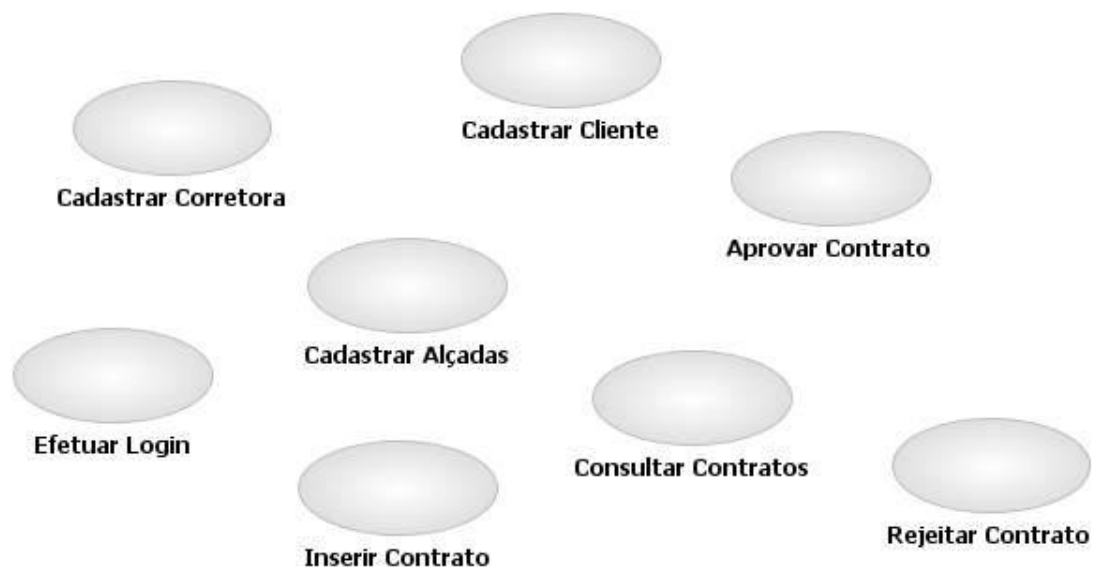
Modelo de Objetos Negociais

SACC Analysis Elements



APÊNDICE 8

Casos de Uso Negociais

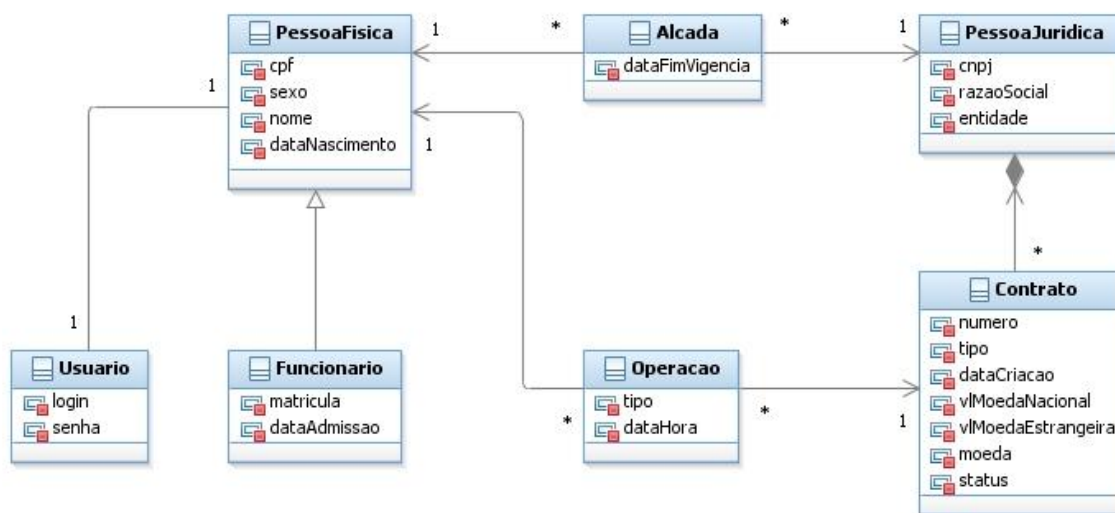


- Efetuar login: login no sistema;
- Cadastrar Corretora: cadastro da entidade Corretora de Câmbio pelo funcionário do banco;
- Cadastrar Cliente: cadastro da entidade Cliente pelo funcionário do banco;
- Cadastrar Alçadas: cadastro das pessoas que poderão aprovar ou rejeitar os contratos de câmbio das entidades corretora ou cliente das quais estarão representando, também pelo funcionário do banco;
- Inserir Contrato: geração do contrato de câmbio pelo funcionário do banco no sistema;
- Consultar Contrato: nesta funcionalidade o usuário poderá visualizar os contratos de câmbio no sistema que estão para a entidade. Também poderá visualizar os detalhes de aprovação e rejeição dos contratos;
- Aprovar Contrato: aprovação do contrato de câmbio pela entidade responsável a partir da consulta do contrato;
- Rejeitar Contrato: rejeição do contrato de câmbio pela entidade responsável a partir da consulta do contrato.

APÊNDICE 9

Modelo de Objetos

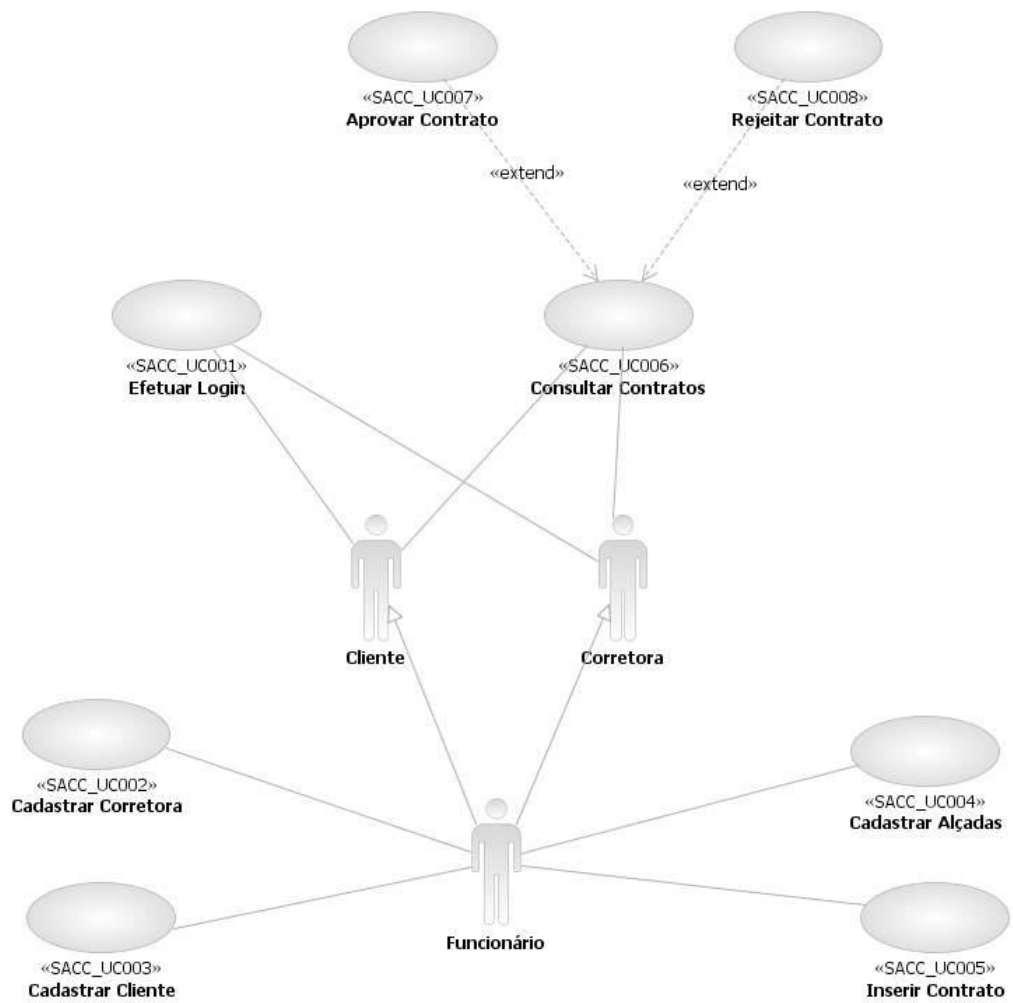
SACC Analysis Elements



APÊNDICE 10

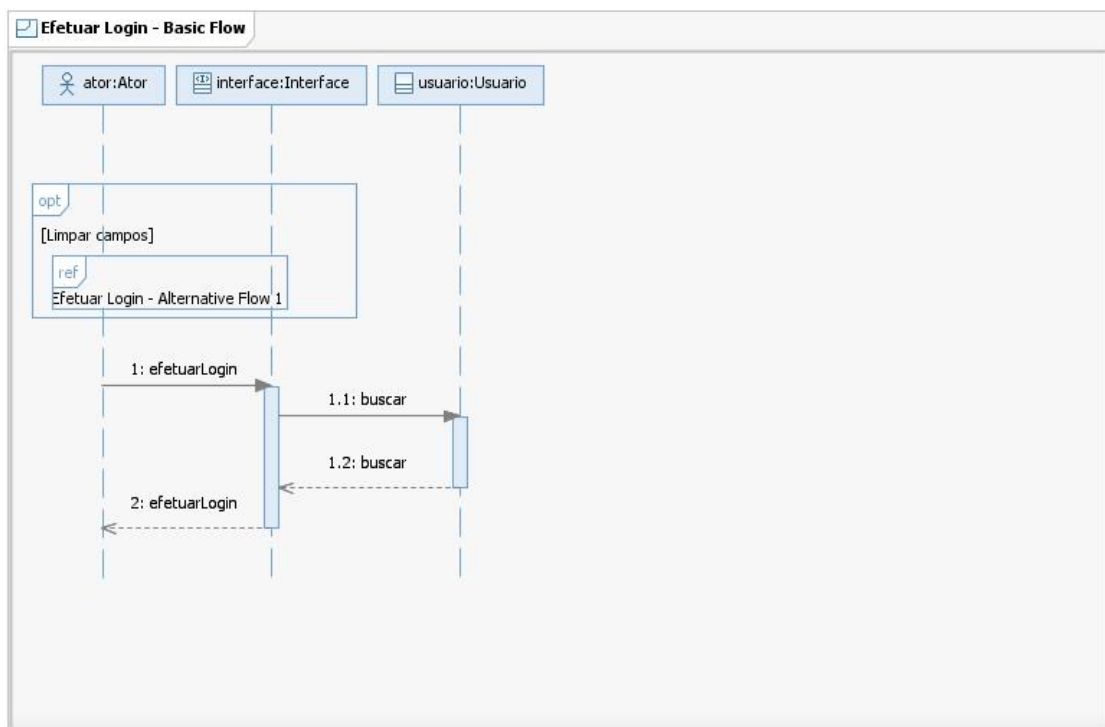
Casos de Uso

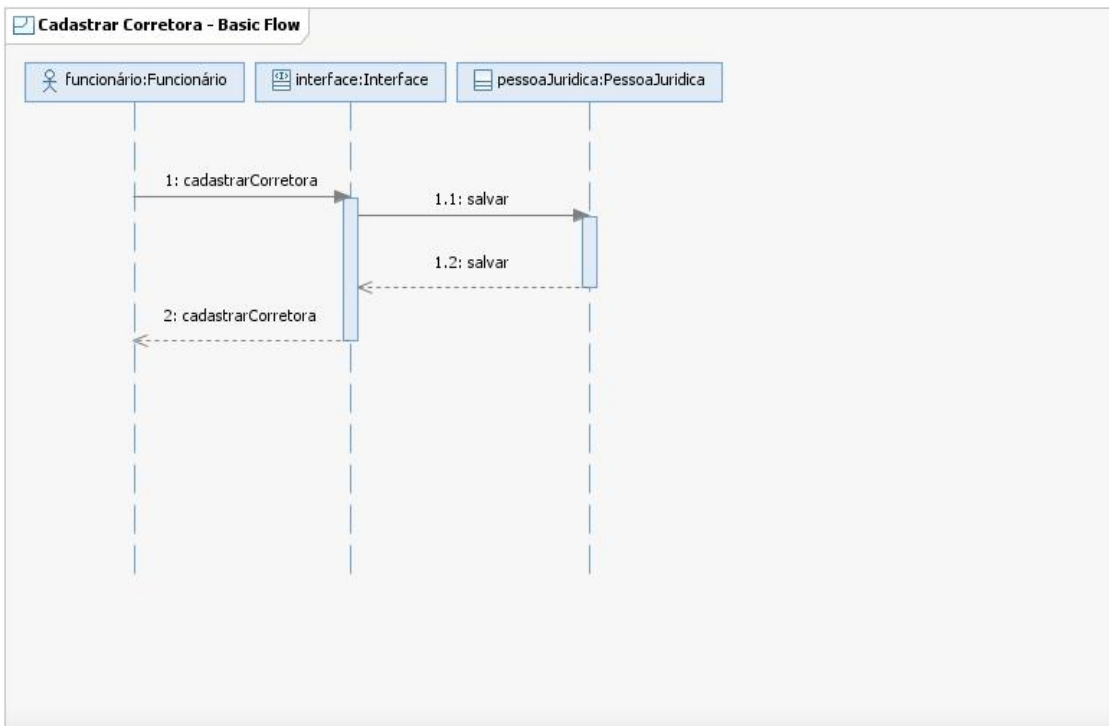
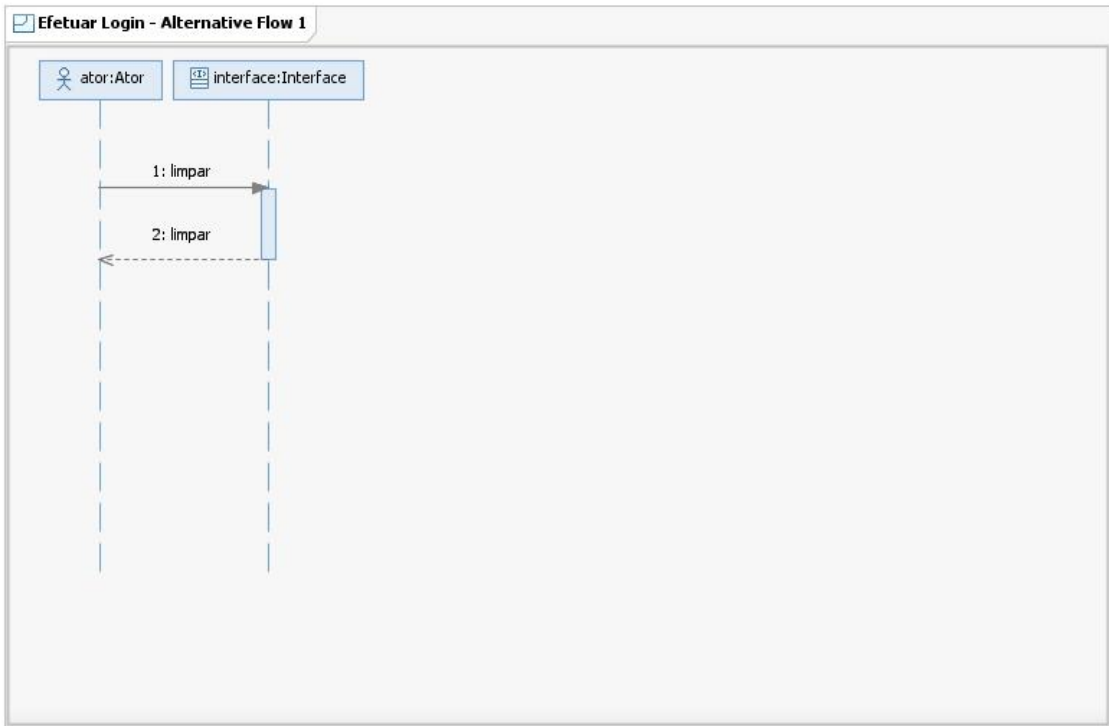
SACC Use Cases

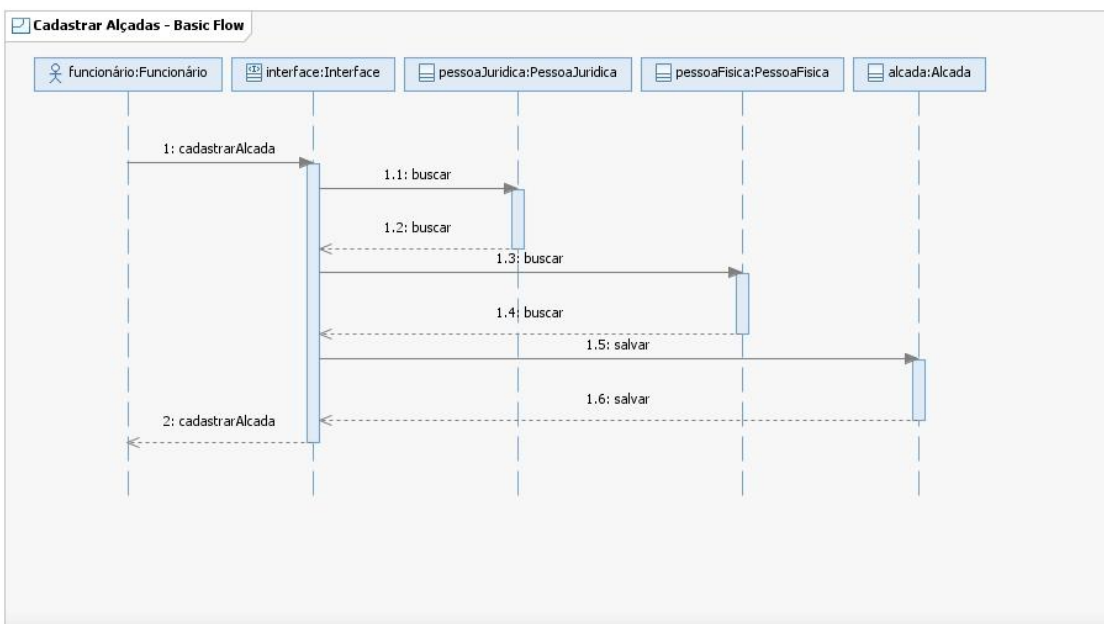
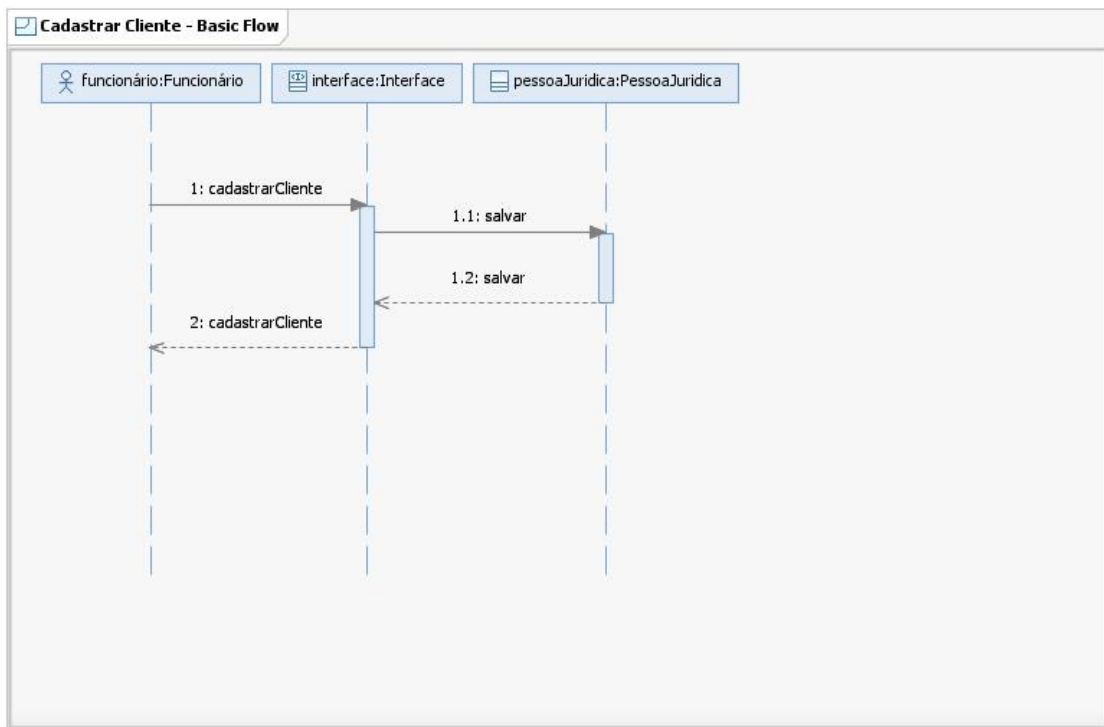


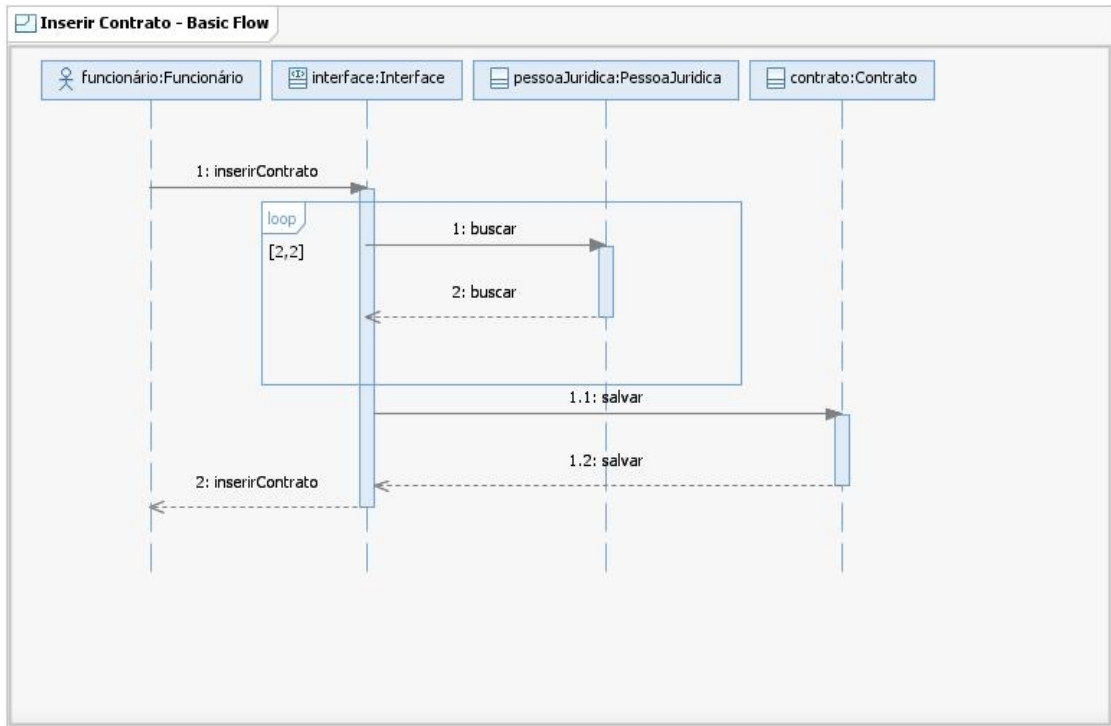
APÊNDICE 11

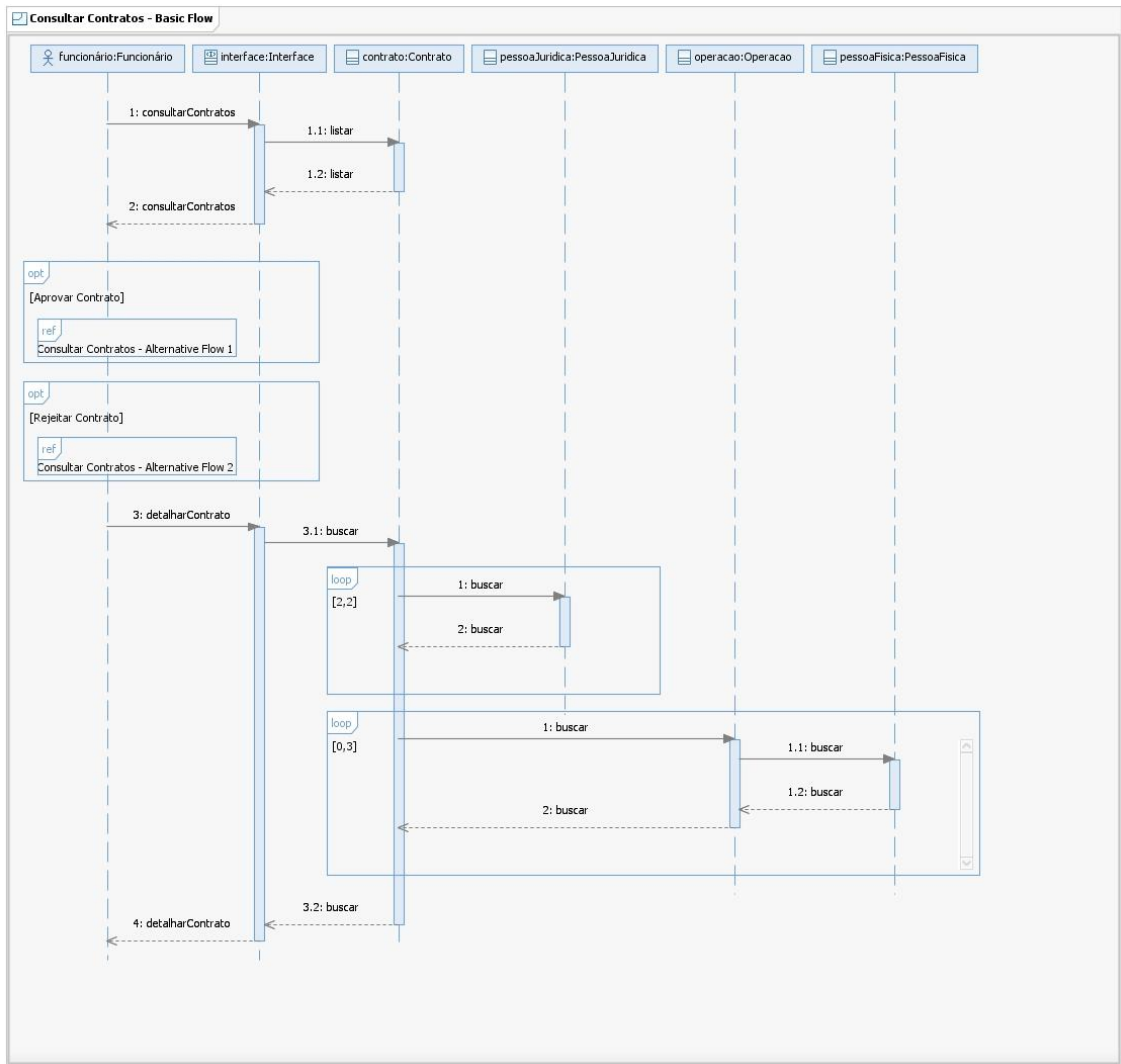
Diagramas de seqüência

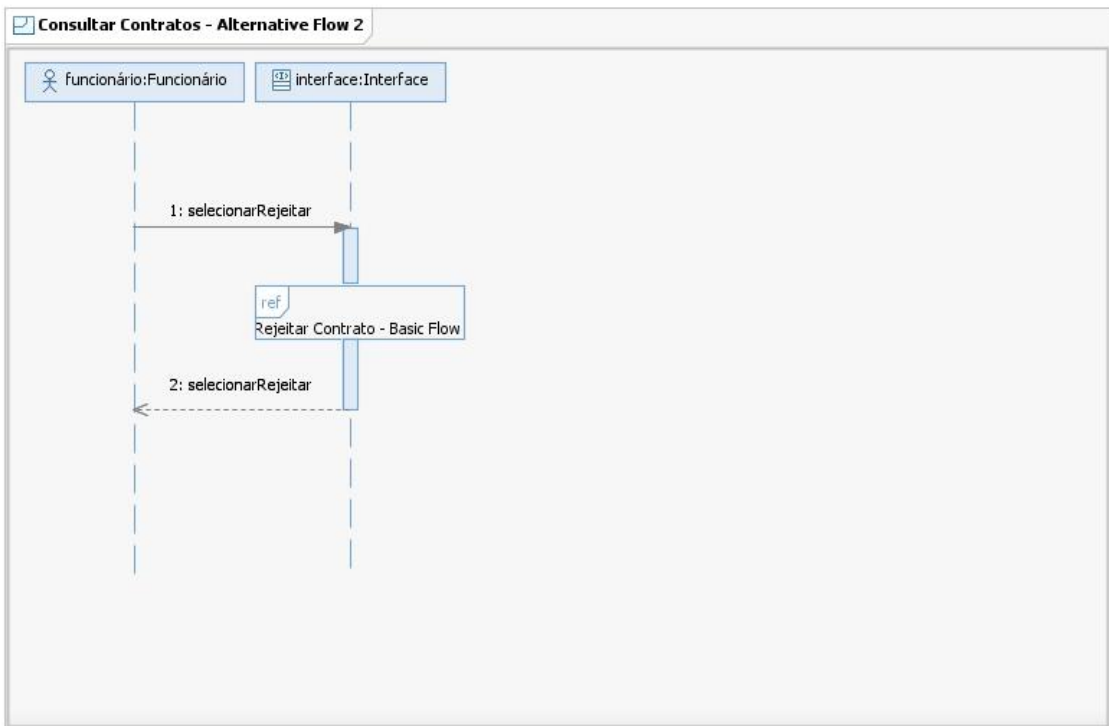
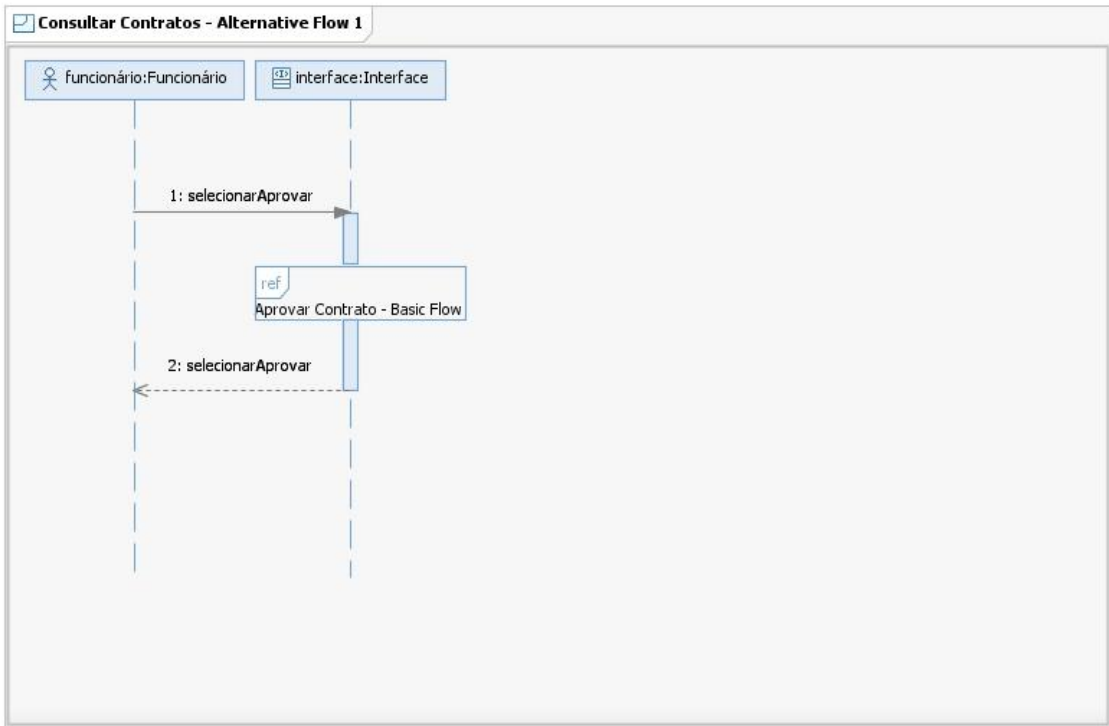


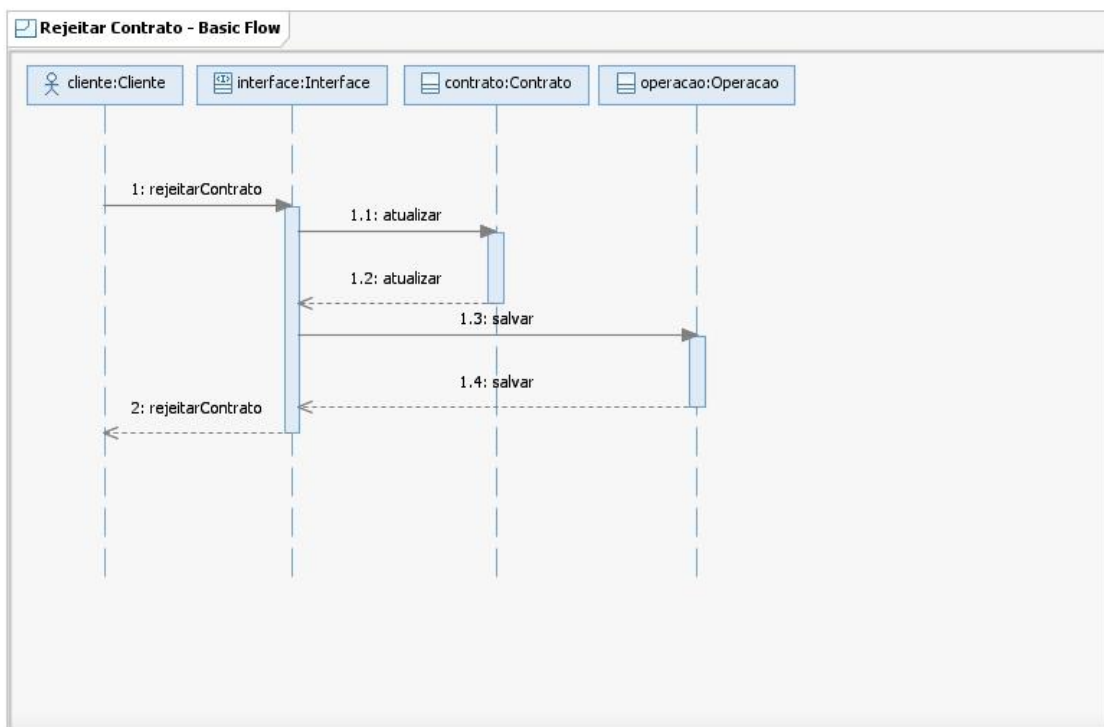
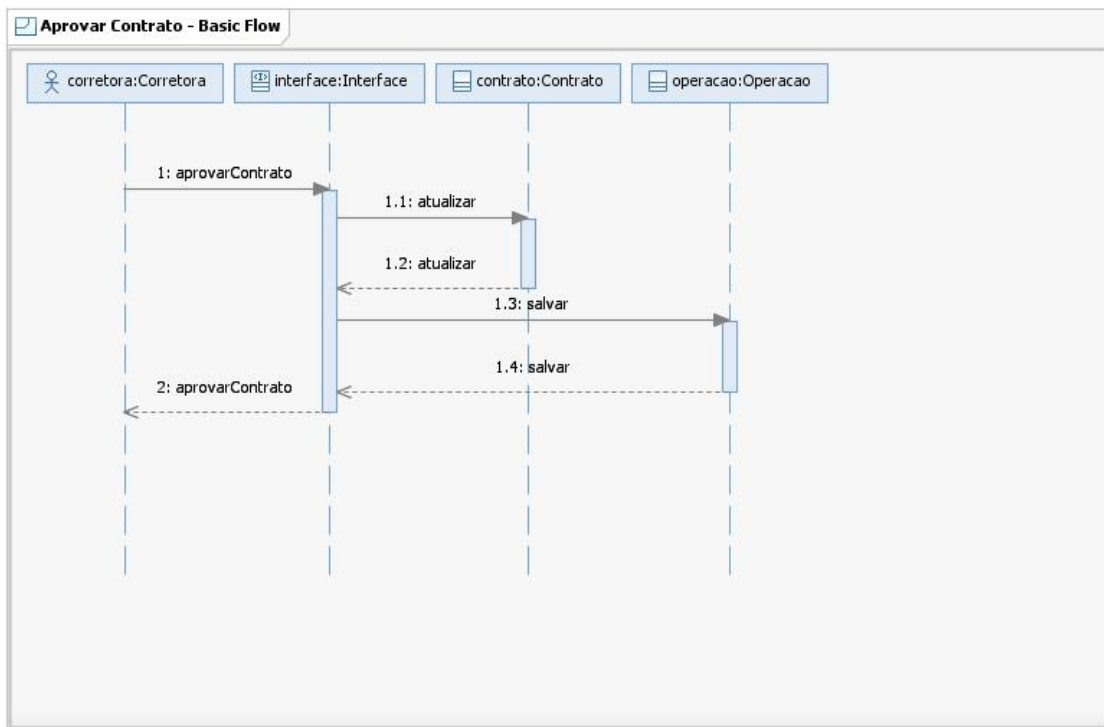








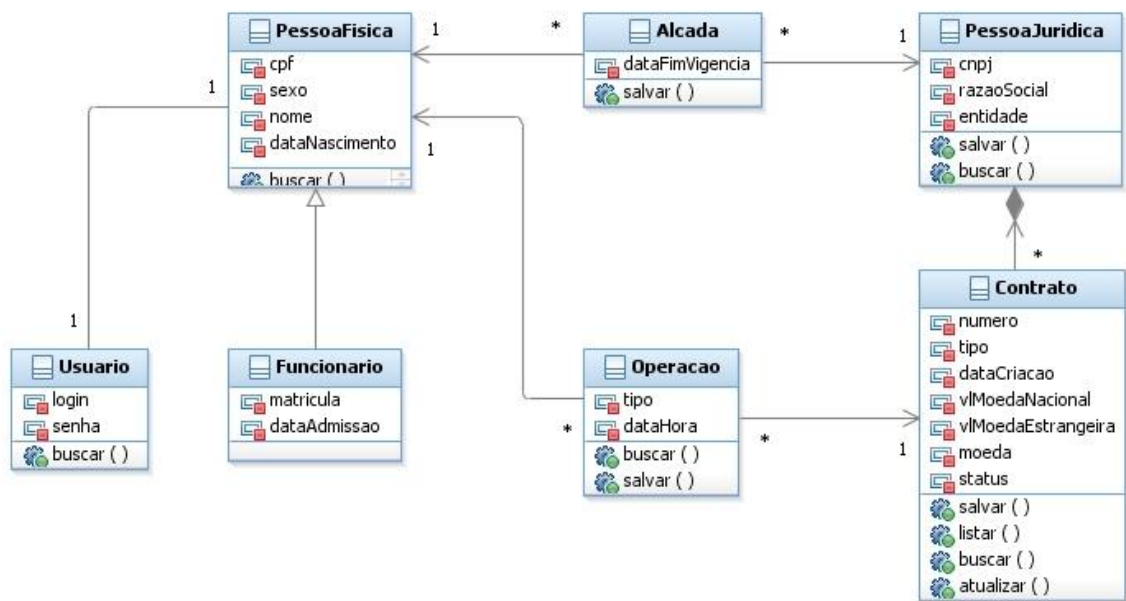




APÊNDICE 12

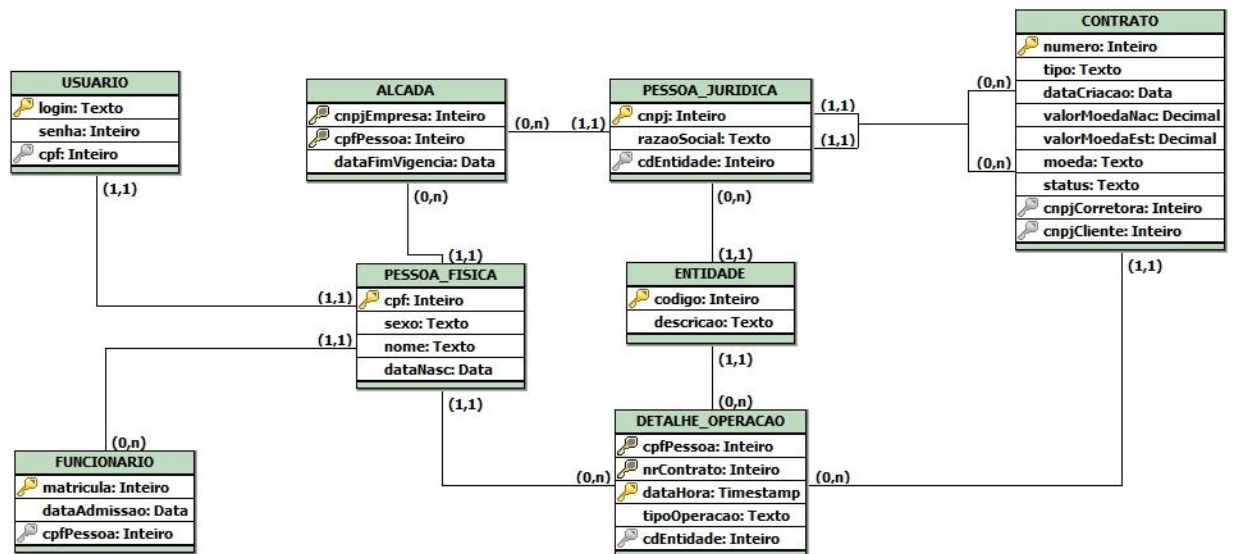
Modelo de Objetos

SACC Analysis Elements



APÊNDICE 13

Modelo Físico de Dados



```
CREATE TABLE ENTIDADE (
codigo serial PRIMARY KEY,
descricao character varying(10) NOT NULL );
```

```
CREATE TABLE PESSOA_JURIDICA (
cnpj bigint PRIMARY KEY,
razaoSocial character varying(50) NOT NULL,
cdEntidade smallint NOT NULL,
FOREIGN KEY(cdEntidade) REFERENCES ENTIDADE (codigo) );
```

```
CREATE TABLE PESSOA_FISICA (
cpf bigint PRIMARY KEY,
sexo character varying(10) NOT NULL,
nome character varying(50) NOT NULL,
```

```
dataNasc date NOT NULL );
```

```
CREATE TABLE USUARIO (  
login character varying(12) PRIMARY KEY,  
senha character varying(10) NOT NULL,  
cpf bigint NOT NULL,  
FOREIGN KEY(cpf) REFERENCES PESSOA_FISICA (cpf) );
```

```
CREATE TABLE FUNCIONARIO (  
matricula smallint PRIMARY KEY,  
dataAdmissao date NOT NULL,  
cpfPessoa bigint NOT NULL,  
FOREIGN KEY(cpfPessoa) REFERENCES PESSOA_FISICA (cpf) );
```

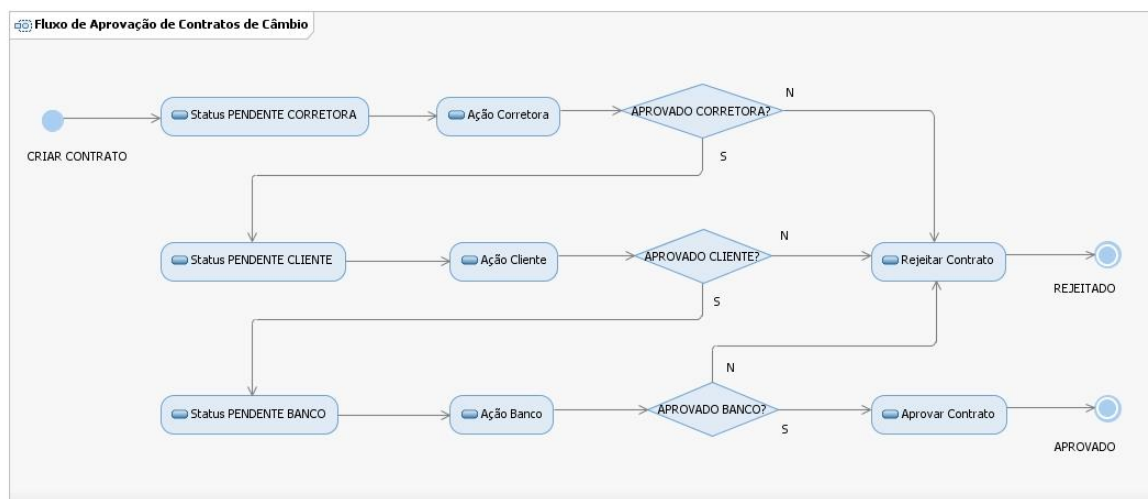
```
CREATE TABLE ALCADA (  
cnpjEmpresa bigint,  
cpfPessoa bigint,  
dataFimVigencia date NOT NULL,  
PRIMARY KEY(cnpjEmpresa,cpfPessoa),  
FOREIGN KEY(cnpjEmpresa) REFERENCES PESSOA_JURIDICA (cnpj),  
FOREIGN KEY(cpfPessoa) REFERENCES PESSOA_FISICA (cpf) );
```

```
CREATE TABLE CONTRATO (  
numero serial PRIMARY KEY,  
tipo character varying(10) NOT NULL,  
dataCriacao date NOT NULL,  
valorMoedaNac real NOT NULL,  
valorMoedaEst real NOT NULL,  
moeda character varying(20) NOT NULL,  
status character varying(20) NOT NULL,  
cnpjCorretora bigint NOT NULL,  
cnpjCliente bigint NOT NULL,  
FOREIGN KEY(cnpjCorretora) REFERENCES PESSOA_JURIDICA (cnpj),  
FOREIGN KEY(cnpjCliente) REFERENCES PESSOA_JURIDICA (cnpj) );
```

```
CREATE TABLE DETALHE_OPERACAO (  
cpfPessoa bigint,  
nrContrato smallint,  
dataHora timestamp,  
tipoOperacao character varying(10) NOT NULL,  
cdEntidade smallint NOT NULL,  
PRIMARY KEY(cpfPessoa,nrContrato,dataHora),  
FOREIGN KEY(cpfPessoa) REFERENCES PESSOA_FISICA (cpf),  
FOREIGN KEY(nrContrato) REFERENCES CONTRATO (numero),  
FOREIGN KEY(cdEntidade) REFERENCES ENTIDADE (codigo) );
```

APÊNDICE 14

Diagramas Suplementares



APÊNDICE 15

Plano de Testes

PLANEJAMENTO DO TESTE

Os testes serão feitos em 5 baterias na ordem correta para que os contratos passem pelo fluxo de aprovação e rejeição em todos os status do contrato.

Bateria 0 – Produzir todas as exceções previstas nos casos de uso 001, 002, 003, 004, 005 e 006.

Bateria 1 – Testar login, cadastrar entidades CORRETORA, CLIENTE e cadastrar de alçadas para CORRETORA e CLIENTE.

Bateria 2 – Criação e aprovação de contrato.

Bateria 3 – Criação e rejeição de contrato no status PENDENTE CORRETORA.

Bateria 4 – Criação e rejeição de contrato no status PENDENTE CLIENTE.

Bateria 5 – Criação e rejeição de contrato no status PENDENTE BANCO.

Detalhamento das Baterias:

BATERIA 0

Caso de Teste 1.1 - Não preencher campos obrigatórios.

Caso de Teste 1.3 - Preencher dados inválidos.

Caso de Teste 2.1 - Não preencher campos obrigatórios.

Caso de Teste 2.2 - Preencher CNPJ inválido.

Caso de Teste 2.3 - Preencher CNPJ existente.

Caso de Teste 3.1 - Não preencher campos obrigatórios.

Caso de Teste 3.2 - Preencher CNPJ inválido.

Caso de Teste 3.3 - Preencher CNPJ existente.

Caso de Teste 4.1 - Não preencher campos obrigatórios.

Caso de Teste 4.2 - Preencher CNPJ inválido.

Caso de Teste 4.3 - Preencher CNPJ inexistente.

Caso de Teste 4.4 - Preencher CPF inválido.

Caso de Teste 4.5 - Preencher CPF inexistente.

Caso de Teste 4.6 - Preencher data inválida.

Caso de Teste 4.7 - Preencher data com valor inválido.

Caso de Teste 4.8 - Preencher alçada existente.

Caso de Teste 5.1 - Não preencher campos obrigatórios.

Caso de Teste 5.2 - Preencher CNPJ Corretora inválido.

Caso de Teste 5.3 - Preencher CNPJ Corretora inexistente.

- Caso de Teste 5.4** - Preencher CNPJ Cliente inválido.
- Caso de Teste 5.5** - Preencher CNPJ Cliente inexistente.
- Caso de Teste 5.6** - Preencher data inválida.
- Caso de Teste 5.7** - Preencher data com valor inválido.
- Caso de Teste 6.1** - Não possuir contratos.

BATERIA 1

- Caso de Teste 1.2** - Limpar dados.
- Caso de Teste 1.4** - Efetuar login.
- Caso de Teste 2.4** - Cadastrar Corretora.
- Caso de Teste 3.4** - Cadastrar Cliente.
- Caso de Teste 4.9** - Cadastrar alçada para Corretora do CT 2.4 e para Cliente do CT 3.4.

BATERIA 2

- Caso de Teste 1.4** - Efetuar login com usuário perfil BANCO.
- Caso de Teste 5.8** - Inserir contrato para Corretora do CT 2.4 e Cliente do CT 3.4.
- Caso de Teste 1.4** - Efetuar login com usuário perfil CORRETORA para Corretora do CT 2.4.
- Caso de Teste 6.2** - Consultar Contratos. O sistema apresenta o contrato com status PENDENTE CORRETORA.
- Caso de Teste 6.3** - Detalhar Contrato.
- Caso de Teste 6.4** - Abrir Aprovar Contrato.
- Caso de Teste 7.1** - Aprovar Contrato – perfil Corretora.
- Caso de Teste 1.4** - Efetuar login com usuário perfil CLIENTE para Cliente do CT 3.4.
- Caso de Teste 6.2** - Consultar Contratos. O sistema apresenta o contrato com status PENDENTE CLIENTE.
- Caso de Teste 6.3** - Detalhar Contrato.
- Caso de Teste 6.4** - Abrir Aprovar Contrato.
- Caso de Teste 7.2** - Aprovar Contrato – perfil Cliente.
- Caso de Teste 1.4** - Efetuar login com usuário perfil BANCO.
- Caso de Teste 6.2** - Consultar Contratos. O sistema apresenta o contrato com status PENDENTE BANCO.
- Caso de Teste 6.3** - Detalhar Contrato.
- Caso de Teste 6.4** - Abrir Aprovar Contrato.
- Caso de Teste 7.3** - Aprovar Contrato – perfil Banco.
- Caso de Teste 6.3** - Detalhar Contrato. O sistema apresenta o contrato com o status APROVADO, com os detalhes de cada aprovação.

BATERIA 3

- Caso de Teste 1.4** - Efetuar login com usuário perfil BANCO.
- Caso de Teste 5.8** - Inserir contrato para Corretora do CT 2.4 e Cliente do CT 3.4.

Caso de Teste 1.4 - Efetuar login com usuário perfil CORRETORA para Corretora do CT 2.4.

Caso de Teste 6.2 - Consultar Contratos. O sistema apresenta o contrato com status PENDENTE CORRETORA.

Caso de Teste 6.3 - Detalhar Contrato.

Caso de Teste 6.5 - Abrir Rejeitar Contrato.

Caso de Teste 8.1 - Rejeitar Contrato – perfil Corretora.

Caso de Teste 1.4 - Efetuar login com usuário perfil BANCO.

Caso de Teste 6.2 - Consultar Contratos.

Caso de Teste 6.3 - Detalhar Contrato. O sistema apresenta o contrato com o status REJEITADO, com os detalhes da rejeição.

BATERIA 4

Caso de Teste 1.4 - Efetuar login com usuário perfil BANCO.

Caso de Teste 5.8 - Inserir contrato para Corretora do CT 2.4 e Cliente do CT 3.4.

Caso de Teste 1.4 - Efetuar login com usuário perfil CORRETORA para Corretora do CT 2.4.

Caso de Teste 6.2 - Consultar Contratos. O sistema apresenta o contrato com status PENDENTE CORRETORA.

Caso de Teste 6.3 - Detalhar Contrato.

Caso de Teste 6.4 - Abrir Aprovar Contrato.

Caso de Teste 7.1 - Aprovar Contrato – perfil Corretora.

Caso de Teste 1.4 - Efetuar login com usuário perfil CLIENTE para Cliente do CT 3.4.

Caso de Teste 6.2 - Consultar Contratos. O sistema apresenta o contrato com status PENDENTE CLIENTE.

Caso de Teste 6.3 - Detalhar Contrato.

Caso de Teste 6.5 - Abrir Rejeitar Contrato.

Caso de Teste 8.2 - Rejeitar Contrato – perfil Cliente.

Caso de Teste 1.4 - Efetuar login com usuário perfil BANCO.

Caso de Teste 6.2 - Consultar Contratos.

Caso de Teste 6.3 - Detalhar Contrato. O sistema apresenta o contrato com o status REJEITADO, com os detalhes da aprovação e rejeição.

BATERIA 5

Caso de Teste 1.4 - Efetuar login com usuário perfil BANCO.

Caso de Teste 5.8 - Inserir contrato para Corretora do CT 2.4 e Cliente do CT 3.4.

Caso de Teste 1.4 - Efetuar login com usuário perfil CORRETORA para Corretora do CT 2.4.

Caso de Teste 6.2 - Consultar Contratos. O sistema apresenta o contrato com status PENDENTE CORRETORA.

Caso de Teste 6.3 - Detalhar Contrato.

Caso de Teste 6.4 - Abrir Aprovar Contrato.

Caso de Teste 7.1 - Aprovar Contrato – perfil Corretora.

Caso de Teste 1.4 - Efetuar login com usuário perfil CLIENTE para Cliente do CT 3.4.

Caso de Teste 6.2 - Consultar Contratos. O sistema apresenta o contrato com status PENDENTE CLIENTE.

Caso de Teste 6.3 - Detalhar Contrato.

Caso de Teste 6.4 - Abrir Aprovar Contrato.

Caso de Teste 7.2 - Aprovar Contrato – perfil Cliente.

Caso de Teste 1.4 - Efetuar login com usuário perfil BANCO.

Caso de Teste 6.2 - Consultar Contratos. O sistema apresenta o contrato com status PENDENTE BANCO.

Caso de Teste 6.3 - Detalhar Contrato.

Caso de Teste 6.5 - Abrir Rejeitar Contrato.

Caso de Teste 8.3 - Rejeitar Contrato – perfil Banco.

Caso de Teste 6.3 - Detalhar Contrato. O sistema apresenta o contrato com o status REJEITADO, com os detalhes de cada aprovação e rejeição.

APÊNDICE 16

Casos de Teste

Caso de Uso	SACC_UC001 – Efetuar Login			
Pré-condições	1. A tela correspondente ao <i>Caso de Uso</i> deve estar aberta.			
Elaborador	Cecília Felipe	Data Elaboração de	14/12/2012	
Executor	Cecília Felipe	Data Execução de	22/01/2015	
N.º	Pré-condições	Entrada	Ação	Resultado Esperado
01		Não preencher nenhum campo	Clicar no botão Entrar.	O sistema apresenta a mensagem “Campo(s) obrigatório(s) não preenchido(s).”. O sistema apresenta uma lista com os campos obrigatórios que não foram preenchidos.
02	Dados digitados nos campos Usuário e Senha.		Clicar no botão Limpar.	O sistema limpa os valores dos campos Usuário e Senha.
03	Usuário cadastrado na tabela USUARIO.	Preencher o(s) campo(s) Usuário/Senha com dados inválidos.	Clicar no botão Entrar.	O sistema apresenta a mensagem “Usuário e/ou Senha inválido(s).”.
04	Usuário cadastrado na tabela USUARIO.	Preencher o campo Usuário e Senha corretamente.	Clicar no botão Entrar.	O sistema autentica o usuário e apresenta o menu do sistema conforme perfil logado.
Caso de Uso	SACC_UC002 – Cadastrar Corretora			
Pré-condições	1. O usuário deve estar logado e autenticado no perfil Banco; 2. A tela correspondente ao <i>Caso de Uso</i> deve estar aberta.			
Elaborador	Cecília Felipe	Data Elaboração de	22/01/2015	
Executor	Cecília Felipe	Data Execução de	22/01/2015	
N.º		Entrada	Ação	Resultado Esperado

01		Não preencher nenhum campo	Clicar no botão Salvar.	O sistema apresenta a mensagem "Campo(s) obrigatório(s) não preenchido(s)". O sistema apresenta uma lista com os campos obrigatórios que não foram preenchidos.
02		Preencher o campo CNPJ com um CNPJ inválido.	Clicar no botão Salvar.	O sistema apresenta a mensagem "CNPJ inválido".
03	CNPJ cadastrado na tabela PESSOA_JURIDICA.	Preencher o campo CNPJ com um CNPJ existente.	Clicar no botão Salvar.	O sistema apresenta a mensagem "CNPJ já cadastrado no sistema".
04		Preencher os campos corretamente.	Clicar no botão Salvar.	O sistema inclui um registro na tabela PESSOA_JURIDICA da entidade Corretora; O sistema apresenta a mensagem "Corretora cadastrada com sucesso".
Caso de Uso		SACC_UC003 – Cadastrar Cliente		
Pré-condições		1. O usuário deve estar logado e autenticado no perfil Banco; 2. A tela correspondente ao <i>Caso de Uso</i> deve estar aberta.		
Elaborador		Cecília Felipe	Data de Elaboração	14/12/2012
Executor		Cecília Felipe	Data de Execução	22/01/2015
N.º	Pré-condições	Entrada	Ação	Resultado Esperado
01		Não preencher nenhum campo	Clicar no botão Salvar.	O sistema apresenta a mensagem "Campo(s) obrigatório(s) não preenchido(s)". O sistema apresenta uma lista com os campos obrigatórios que não foram preenchidos.
02		Preencher o campo CNPJ com um CNPJ inválido.	Clicar no botão Salvar.	O sistema apresenta a mensagem "CNPJ inválido".
03	CNPJ cadastrado na tabela PESSOA_JURIDICA.	Preencher o campo CNPJ com um CNPJ existente.	Clicar no botão Salvar.	O sistema apresenta a mensagem "CNPJ já cadastrado no sistema".
04		Preencher os campos corretamente.	Clicar no botão Salvar.	O sistema inclui um registro na tabela PESSOA_JURIDICA da entidade Cliente; O sistema apresenta a mensagem "Cliente cadastrado com sucesso".
Caso de Uso		SACC_UC004 – Cadastrar Alçadas		
Pré-condições		1. O usuário deve estar logado e autenticado no perfil Banco; 2. O A tela correspondente ao <i>Caso de Uso</i> deve estar aberta.		

Elaborador	Cecília Felipe	Data Elaboração	de	14/12/2012
Executor	Cecília Felipe	Data Execução	de	23/01/2015
N.º	Pré-condições	Entrada	Ação	Resultado Esperado
01		Não preencher nenhum campo	Clicar no botão Salvar.	O sistema apresenta a mensagem "Campo(s) obrigatório(s) não preenchido(s)". O sistema apresenta uma lista com os campos obrigatórios que não foram preenchidos.
02		Preencher o campo CNPJ com um CNPJ inválido.	Clicar no botão Salvar.	O sistema apresenta a mensagem "CNPJ inválido".
03	CNPJ não cadastrado na tabela PESSOA_JURIDICA;	Preencher o campo CNPJ com um CNPJ inexistente.	Clicar no botão Salvar.	O sistema apresenta a mensagem "CNPJ não cadastrado no sistema".
04		Preencher o campo CPF com um CPF inválido.	Clicar no botão Salvar.	O sistema apresenta a mensagem "CPF inválido".
05	CPF não cadastrado na tabela PESSOA FISICA;	Preencher o campo CPF com um CPF inexistente.	Clicar no botão Salvar.	O sistema apresenta a mensagem "CPF não cadastrado no sistema".
06		Preencher o campo Data Final de Vigência com uma data inválida.	Clicar no botão Salvar.	O sistema apresenta a mensagem "Data inválida".
07		Preencher o campo Data Final de Vigência com data atual.	Clicar no botão Salvar.	O sistema apresenta a mensagem "A Data Final de Vigência deve ser maior que a data atual".
08	CNPJ cadastrado na tabela PESSOA_JURIDICA; CPF cadastrado na tabela PESSOA FISICA; Relacionamento do CNPJ com CPF cadastrado na tabela ALCADA.	Preencher os campos CNPJ e CPF com valores existentes.	Clicar no botão Salvar.	O sistema apresenta a mensagem "Alçada já cadastrada no sistema".
09	CNPJ cadastrado na tabela PESSOA_JURIDICA. CPF cadastrado na tabela PESSOA FISICA;	Preencher os campos corretamente.	Clicar no botão Salvar.	O sistema inclui um registro na tabela ALCADA, relacionando o CNPJ da entidade com o CPF da pessoa; O sistema apresenta a mensagem "Corretora cadastrada com sucesso".
Caso de Uso	SACC_UC005 – Inserir Contrato			

Pré-condições	1. O usuário deve estar logado e autenticado no perfil Banco; 2. A tela correspondente ao <i>Caso de Uso</i> deve estar aberta.			
Elaborador	Cecília Felipe	Data de Elaboração	14/12/2012	
Executor	Cecília Felipe	Data de Execução	23/01/2015	
N.º	Pré-condições	Entrada	Ação	Resultado Esperado
01		Não preencher nenhum campo	Clicar no botão Inserir.	O sistema apresenta a mensagem "Campo(s) obrigatório(s) não preenchido(s)". O sistema apresenta uma lista com os campos obrigatórios que não foram preenchidos.
02		Preencher o campo CNPJ Corretora com um CNPJ inválido.	Clicar no botão Inserir	O sistema apresenta a mensagem "CNPJ Corretora inválido."
03	CNPJ não cadastrado na tabela PESSOA_JURIDICA como entidade Corretora;	Preencher o campo CNPJ Corretora com um CNPJ inexistente.	Clicar no botão Inserir	O sistema apresenta a mensagem "CNPJ Corretora não cadastrado no sistema."
04		Preencher o campo CNPJ Cliente com um CNPJ inválido.	Clicar no botão Inserir	O sistema apresenta a mensagem "CNPJ Cliente inválido."
05	CNPJ não cadastrado na tabela PESSOA_JURIDICA como entidade Cliente;	Preencher o campo CNPJ Cliente com um CNPJ inexistente.	Clicar no botão Inserir	O sistema apresenta a mensagem "CNPJ Cliente não cadastrado no sistema."
06		Preencher o campo Data de Encerramento com uma data inválida.	Clicar no botão Inserir	O sistema apresenta a mensagem "Data inválida."
07		Preencher o campo Data de Encerramento com uma data anterior à data atual.	Clicar no botão Inserir	O sistema apresenta a mensagem "A Data de Encerramento deve ser maior que a data atual."
08	CNPJ Corretora cadastrado na tabela PESSOA_JURIDICA; CNPJ Cliente cadastrado na tabela PESSOA_JURIDICA.	Preencher os campos corretamente.	Clicar no botão Inserir	O sistema inclui um registro na tabela CONTRATO com status PENDENTE CORRETORA; O sistema apresenta a mensagem "Contrato de Câmbio inserido com sucesso."
Caso de Uso	SACC_UC006 – Consultar Contrato			
Pré-condições	1. O usuário deve estar logado e autenticado em qualquer perfil; 2. A tela correspondente ao <i>Caso de Uso</i> deve estar aberta.			

Elaborador	Cecília Felipe	Data Elaboração	de 14/12/2012	
Executor	Cecília Felipe	Data Execução	de 24/01/2015	
N.º	Pré-condições	Entrada	Ação	Resultado Esperado
01	Não existir contratos para o usuário logado.	Nenhum registro encontrado.		O sistema apresenta a mensagem "Não existem contratos."
02	Contrato cadastrado na tabela CONTRATO; Relacionamento cadastrado na tabela ALCADA do usuário logado com a entidade do contrato; Registros na tabela DELALHE_OPERACAO.			O sistema apresenta todos os contratos da entidade na qual o usuário logado possui alçada cadastrada.
03	O Caso de Teste 02 foi executado.		Clicar no link Número do Contrato.	O sistema apresenta as informações do contrato e os detalhes das aprovações.
04	O Caso de Teste 03 foi executado.		Clicar no link Aprovar.	O sistema executa caso de uso SACC_UC007 – Aprovar Contrato.
05	O Caso de Teste 03 foi executado.		Clicar no link Rejeitar.	O sistema executa caso de uso SACC_UC007 – Rejeitar Contrato.
Caso de Uso	SACC_UC007 – Aprovar Contrato			
Pré-condições	<ol style="list-style-type: none"> 1. O sistema deverá ter executado o caso de uso UC006 – Consultar Contrato; 2. O usuário deve estar logado e autenticado em qualquer perfil; 3. A tela correspondente ao <i>Caso de Uso</i> deve estar aberta. 			
Elaborador	Cecília Felipe	Data Elaboração	de 14/12/2012	
Executor	Cecília Felipe	Data Execução	de 25/01/2015	
N.º	Pré-condições	Entrada	Ação	Resultado Esperado
01	Usuário logado com alçada cadastrada para a Corretora participante do contrato; Contrato com status PENDENTE CORRETORA.		Clicar no botão Aprovar Contrato.	O sistema insere um registro na tabela DETALHE_OPERAÇÃO da entidade do usuário logado de aprovação; O sistema atualiza o campo status do contrato da tabela CONTRATO como segue para PENDENTE CLIENTE; O sistema apresenta a mensagem: "Aprovação cadastrada."

02	Usuário logado com alçada cadastrada para a Corretora participante do contrato; Contrato com status PENDENTE CLIENTE.		Clicar no botão Aprovar Contrato.	O sistema insere um registro na tabela DETALHE_OPERAÇÃO da entidade do usuário logado de aprovação; O sistema atualiza o campo status do contrato da tabela CONTRATO como segue para PENDENTE BANCO; O sistema apresenta a mensagem: "Aprovação cadastrada."
03	Usuário logado ser funcionário; Contrato com status PENDENTE BANCO.		Clicar no botão Aprovar Contrato.	O sistema insere um registro na tabela DETALHE_OPERAÇÃO da entidade do usuário logado de aprovação; O sistema atualiza o campo status do contrato da tabela CONTRATO como para APROVADO. O sistema apresenta a mensagem: "Aprovação cadastrada.."
Caso de Uso	SACC_UC008 – Rejeitar Contrato			
Pré-condições	<ol style="list-style-type: none"> 1. O sistema deverá ter executado o caso de uso UC006 – Consultar Contrato; 2. O usuário deve estar logado e autenticado em qualquer perfil; 3. A tela correspondente ao <i>Caso de Uso</i> deve estar aberta. 			
Elaborador	Cecília Felipe	Data de Elaboração	14/12/2012	
Executor	Cecília Felipe	Data de Execução	25/01/2015	
N.º	Pré-condições	Entrada	Ação	Resultado Esperado
01	Usuário logado com alçada cadastrada para a Corretora participante do contrato; Contrato com status PENDENTE CORRETORA.		Clicar no botão Rejeitar Contrato.	O sistema insere um registro na tabela DETALHE_OPERAÇÃO da entidade do usuário logado de rejeição; O sistema atualiza o campo <i>status</i> do contrato da tabela CONTRATO como segue para REJEITADO; O sistema apresenta a mensagem: "Contrato rejeitado."
02	Usuário logado com alçada cadastrada para a Corretora participante do contrato; Contrato com status PENDENTE CLIENTE.		Clicar no botão Rejeitar Contrato.	O sistema insere um registro na tabela DETALHE_OPERAÇÃO da entidade do usuário logado de rejeição; O sistema atualiza o campo <i>status</i> do contrato da tabela CONTRATO como segue para REJEITADO; O sistema apresenta a mensagem: "Contrato rejeitado."

03	<p>Usuário logado ser funcionário;</p> <p>Contrato com status PENDENTE BANCO.</p>		<p>Clicar no botão Rejeitar Contrato.</p>	<p>O sistema insere um registro na tabela DETALHE_OPERAÇÃO da entidade do usuário logado de rejeição;</p> <p>O sistema atualiza o campo <i>status</i> do contrato da tabela CONTRATO como para APROVADO.</p> <p>O sistema apresenta a mensagem: "Contrato rejeitado."</p>
----	---	--	---	---

APÊNDICE 17

Scripts do Banco de Dados

POPULAR ENTIDADES:

```
INSERT INTO ENTIDADE VALUES (default, 'CORRETORA');  
INSERT INTO ENTIDADE VALUES (default, 'CLIENTE');  
INSERT INTO ENTIDADE VALUES (default, 'BANCO');
```

POPULAR PESSOAS:

```
INSERT INTO PESSOA_FISICA VALUES (55046621681, 'M', 'João da Silva', '1975-12-08');  
INSERT INTO PESSOA_FISICA VALUES (12360619853, 'M', 'Lucas de Souza', '1969-07-23');  
INSERT INTO PESSOA_FISICA VALUES (68535583912, 'M', 'Cláudio de Oliveira', '1975-01-28');  
INSERT INTO PESSOA_FISICA VALUES (26061531389, 'M', 'Francisco Araújo', '1979-02-05');  
INSERT INTO PESSOA_FISICA VALUES (53887875753, 'F', 'Maria Fonseca', '1976-11-15');  
INSERT INTO PESSOA_FISICA VALUES (11493269216, 'F', 'Carmem Machado', '1981-06-22');  
INSERT INTO PESSOA_FISICA VALUES (32489884018, 'F', 'Lúcia de Paula', '1971-03-10');  
INSERT INTO PESSOA_FISICA VALUES (46537877946, 'F', 'Carla Amarante', '1984-03-13');  
INSERT INTO PESSOA_FISICA VALUES (32638994731, 'M', 'Luiz Fonseca', '1972-06-04');  
INSERT INTO PESSOA_FISICA VALUES (26262126791, 'M', 'Flávio Machado', '1965-04-11');  
INSERT INTO PESSOA_FISICA VALUES (16257591937, 'M', 'Ademir Amarante', '1980-01-05');  
INSERT INTO PESSOA_FISICA VALUES (22734549700, 'F', 'Michele da Silva', '1966-05-07');  
INSERT INTO PESSOA_FISICA VALUES (14390273310, 'F', 'Camila de Souza', '1971-03-09');  
INSERT INTO PESSOA_FISICA VALUES (03288825180, 'F', 'Maria da Graça Araújo', '1974-06-26');
```

POPULAR USUÁRIOS:

```
INSERT INTO USUARIO VALUES ('jsilva', '12345', 55046621681);  
INSERT INTO USUARIO VALUES ('lsouza', '12345', 12360619853);  
INSERT INTO USUARIO VALUES ('coliveira', '12345', 68535583912);  
INSERT INTO USUARIO VALUES ('faraujo', '12345', 26061531389);  
INSERT INTO USUARIO VALUES ('mfonseca', '12345', 53887875753);  
INSERT INTO USUARIO VALUES ('cmachado', '12345', 11493269216);  
INSERT INTO USUARIO VALUES ('lpaula', '12345', 32489884018);  
INSERT INTO USUARIO VALUES ('camarante', '12345', 46537877946);  
INSERT INTO USUARIO VALUES ('lfonseca', '12345', 32638994731);  
INSERT INTO USUARIO VALUES ('fmachado', '12345', 26262126791);  
INSERT INTO USUARIO VALUES ('aamarante', '12345', 16257591937);  
INSERT INTO USUARIO VALUES ('msilva', '12345', 22734549700);  
INSERT INTO USUARIO VALUES ('csouza', '12345', 14390273310);  
INSERT INTO USUARIO VALUES ('maraujo', '12345', 03288825180);
```

POPULAR FUNCIONÁRIOS:

```
INSERT INTO FUNCIONARIO VALUES(12345, '2000-11-01', 32489884018);  
INSERT INTO FUNCIONARIO VALUES(13456, '2005-01-10', 68535583912);
```