

UNIVERSIDADE FEDERAL DO PARANÁ

RANGEL DAYKES DOMINGOS DOS SANTOS

DB ACESSO FÁCIL

CURITIBA
2012

RANGEL DAYKES DOMINGOS DOS SANTOS

DB ACESSO FÁCIL

Trabalho de Conclusão de Curso, apresentado como requisito parcial para a obtenção do título de especialista no curso de Especialização em Engenharia de Software, Setor de Educação Profissional e Tecnológica, Universidade Federal do Paraná.

Orientação professor: MsC. Jaime Wojciechowski

CURITIBA
2012

TERMO DE APROVAÇÃO

RANGEL DAYKES DOMINGOS DOS SANTOS

DB ACESSO FÁCIL

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

Orientador professor: MsC. Jaime Wojciechowski

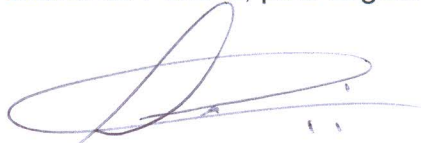
Curitiba, 20 de dezembro de 2012

TERMO DE APROVAÇÃO

RANGEL DAYKES DOMINGOS DOS SANTOS

O DESENVOLVIMENTO DE UM APLICATIVO WEB PARA ACESSO A BANCO DE DADOS UTILIZANDO O RATIONAL UNIFIED PROCESS COMO PROCESSO DE ENGENHARIA DE SOFTWARE

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, pela seguinte banca examinadora:



Orientador: Professor Jaime Wojciechowski

Curitiba, 19 de dezembro de 2012

RESUMO

Este sistema tem por objetivo auxiliar profissionais da área de informática que tem em sua rotina a tarefa de manipulação de bancos de dados. O sistema permitirá um acesso prático, rápido e acessível à de bancos de dados locais através da internet. O sistema foi desenvolvido seguindo o modelo de processos Rational Unified Process (RUP), para auxiliar na transformação de requisitos em software e na documentação. Para mensurar prazos, custos, gerar um cronograma e tornar viável o desenvolvimento do sistema foi utilizado conceitos de gerência de projetos. Para o desenvolvimento do software foi utilizado a linguagem JAVA.

Palavras Chave: Banco de dados, manipulação de dados e acesso a base de dados.

ABSTRACT

This system aims to help IT professionals who have in their routine task of manipulating databases. The system will allow for convenient access, fast and affordable to the local databases through the internet. The system was developed following the model of processes RUP-Rational Unified Process, to assist in the transformation of requirements into software and documentation. To measure time, cost, generate a schedule and make feasible the development of the system was used concepts of project management. To develop the software was used to linguem JAVA.

Keywords: Database, data manipulation and access the database.

SUMARIO

1 INTRODUÇÃO	8
1.2 OBJETIVOS.....	9
1.2.1 Objetivos específicos.....	9
2 REVISÃO DA LITERATURA	10
2.1 BANCO DE DADOS.....	10
2.2 SGDB.....	10
2.3 PROGRAMAÇÃO ORIENTADA A OBJETOS – POO.....	11
2.4 UNIFIED MODELLING LANGUAGE – UML.....	11
2.5 RATIONAL UNIFIED PROCESS - RUP.....	11
2.6 LINGUAGEM JAVA.....	12
2.6.1 Servlets.....	12
2.6.2 Applets.....	12
2.6.3 JavaServer Pages – JSP.....	13
2.7 LINGUAGEM HTML	13
2.8 GERENCIAMENTO DE PROJETOS.....	13
2.9 PADRÕES DE PROJETOS – DESIGN PATTERNS.....	14
3 METODOLOGIA	15
3.1 PROCESSO DE ENGENHARIA DE SOFTWARE.....	15
3.2 GERENCIAMENTO DE PROJETOS.....	15
3.2.1 PLANO DE ATIVIDADE.....	15
4 SOFTWARE	17
4.1 TELA DE LOGIN.....	17
4.2 TELA DE CADASTRO DE USUARIO.....	17
4.3 TELA DE BANCO DE DADOS	17
4.4 TELA REGISTRAR BANCO DE DADOS.....	18
4.5 TELA EXECUTAR CONSULTA.....	18
4.6 TELA EXIBIR DADOS.....	19
4.7 TELA SALVAR DADOS DA CONSULTA.....	20
5 CONCLUSÃO	21
REFERENCIA BIBLIOGRAFICA	22
APENDICE	23

LISTA DE FIGURAS

Figura 01: WBS DB Acesso Fácil.....	15
Figura 02: Gráfico de Gantt.....	16
Figura 03: Tela de Login	17
Figura 04: Tela de Cadastro de Usuário	17
Figura 05: Tela de Banco de Dados	18
Figura 06: Tela Registrar Banco de Dados	18
Figura 07: Tela Executar Consuta	19
Figura 08: Tela Exibir Dados	19
Figura 09: Tela Salvar Dados da Consulta	20

1 INTRODUÇÃO

Para trabalhar com massas de dados hoje os sistemas informatizados tem o apoio de outros sistemas que gerenciam de forma rápida e confiável esses dados. Este sistema de gerencia de dados ou Sistema Gerenciador de Banco de Dados (SGBD) traz com ele um conjunto de ferramentas que possibilitam a interação de um usuário com os dados armazenados. Mas essas ferramentas geralmente não oferecem um bom suporte visual ao usuário. Assim foi necessária a construção de outros sistemas que fizessem o papel de interação com os dados de um SGBD, mas que oferecesse uma boa interface visual e recursos para facilitar a rotina dos profissionais que trabalham com sistemas informatizados de acesso a dados.

Muitas vezes com a implantação de um sistema também é feita a implantação de um SGBD para o cliente que irá usar o sistema. E isso pode ser feito em no ambiente do cliente em seu servidor por exemplo. Assim sempre que necessário uma manutenção no sistema que envolva manutenção no SGBD, o profissional de informática terá que ter sempre em mãos um sistema para acesso aos dados com interface para uma manutenção e manipulação mais ágil e eficiente.

Para que não seja necessário ter sempre em mãos esse sistema e não precisar se incomodar com transporte ou instalação de um sistema com interface para acesso aos dados o profissional de informática poderá fazer o uso do Sistema de acesso a banco de dados pela internet.

1.2 OBJETIVOS

Desenvolver o projeto de um sistema para acesso de banco de dados pela internet.

1.2.1 Objetivos específicos

- Permitir acesso a banco de dados pela Internet
- Tornar acessível o acesso a um sistema para manipulação de banco de dados
- Executar instruções no banco de dados acessado.

2 REVISÃO DA LITERATURA

2.1 BANCO DE DADOS

Banco de dados geralmente é utilizado para armazenamento dos dados de um sistema de informação. Dados estes que podem ser gerados por pessoas imputando dados manualmente, através de processamento de dados gerando outros dados. O banco de dados deve prover um arquivamento duradouro dos dados nele inserido e possibilitar o consumo destes dados para gerar novos dados ou apenas apresenta-los para tirar informações.

Pode-se dizer que banco de dados é um conjunto de dados estruturados e organizados entre si para armazenamento persistente e com a possibilidade de recuperação de dados para formar uma informação.

Um sistema de uma organização utiliza ferramentas oferecidas pelo banco de dados para manipular os dados a fim de gerar um contexto com o resultado para tomada de decisões. Assim o banco de dados precisa garantir a consistência dos dados armazenados.

Podemos conceituar banco de dados (ou abreviadamente, BD) como sendo um conjunto de dados com certa organização característica, com o objetivo de armazenamento persistente dos dados e dotados de mecanismos de manipulação para obtenção de informações e recuperação posterior, dentro de um sistema de informação. (MEDEIROS, 2007, p.14)

2.2 SGBD

Para poder acessar os dados um sistema tem que possuir ferramentas que possibilitem a comunicação com os dados. Para que não seja necessário que cada linguagem de programação tenha suas próprias bibliotecas ou ferramentas de acesso a cada tipo de banco de dados foi criado um conjunto de sistemas para tal tarefa. Um SGBD que significa Sistemas Gerenciadores de Banco de Dados é responsável pelo gerenciamento de uma base de dados. De acordo com (MEDEIROS, 2007, p.26) “Um SGBD é o responsável por todas as tarefas pertinentes ao armazenamento, à recuperação, à segurança e ao gerenciamento dos dados”.

Hoje existem vários SGBD's de diversos fabricantes cada um com suas características, concorrentes entre si ou destinados para tarefas diferentes, mas em

suas maioria segue padrões para facilitar seu uso como por exemplo a Linguagem SQL a sigla para Standard Query Language. Este é um padrão de linguagem para consulta para banco de dados relacionais que possibilita através de comandos a manipulação dos dados de um banco de dados de forma clara e é aceita na maioria dos SGBDs.

2.3 PROGRAMAÇÃO ORIENTADA A OBJETOS – POO

Técnica de desenvolvimento de software que visa aproximar o mundo real ao mundo virtual de programação de sistemas. Por exemplo, uma cadeira ou uma pessoa para a POO seria um objeto cadeira ou um objeto pessoa. Uma classe define os atributos, comportamentos e relacionamento que um objeto possui. Os objetos de um software conversão entre sim através de mensagens. O programador define quais mensagens cada objeto poderá receber como ele vai receber e qual comportamento ele terá com determinadas mensagens.

2.4 UNIFIED MODELLING LANGUAGE – UML

A UML é uma linguagem visual para modelagem de sistemas orientados a Objetos. Podemos dizer que o objetivo da UML é facilitar a interpretação de modelos criados por um analista por toda a equipe do projeto. Pode ser utilizada independente de linguagem de programação que o sistema será desenvolvido, basta que a linguagem de apoio à orientação a objetos que a modelagem feita na UML será aplicada.

A UML é uma linguagem visual para modelar sistemas orientados a objetos. Isso quer dizer que a UML, é uma linguagem constituída de elementos gráficos (visuais) utilizados na modelagem que permitem representar os conceitos do paradigma da orientação a objetos. Através dos elementos gráficos definidos nesta linguagem pode-se construir diagramas que representam diversas perspectivas de um sistema (BEZERRA, 2002, p.14).

2.5 RATIONAL UNIFIED PROCESS - RUP

O RUP é um processo de Engenharia de software criado pela Rational Software Corporation, oferece um conjunto de técnicas para conduzir o processo de produção de um software. Pela grande quantidade de artefatos produzidos o RUP inicialmente foi dirigido a softwares e equipes de grande porte. Mas pela sua

característica de customização com algumas alterações é possível ser aplicado ao desenvolvimento de projetos menores. É um processo que visa o desenvolvimento em incrementos dividindo todo o trabalho em processos menores ou iterações.

2.6 LINGUAGEM JAVA

Java é uma linguagem de programação de sintaxe amigável com muitos recursos, entre eles à orientação a objetos, ela foi proposta como uma linguagem em que os programas gerados por ela seriam portáteis, ou seja, o programador escreve um programa e tem a liberdade de executar este programa em diversas plataformas. Com isso ficou famosa a frase “Escreva uma vez, execute em qualquer lugar”. Com o tempo por características de robustez, segurança e apoio a redes, o Java passou a ser utilizado por sistemas em geral mais fortemente em sistemas de grande porte. A compilação de um código fonte em Java não gera um arquivo executável pelo hardware diretamente, mas sim um arquivo chamado bytecode que é um intermediário entre a linguagem de alto nível e a linguagem de máquina. De acordo com (SIERRA, 2005, p.11) “O Java vem com centenas de classes predefinidas. Você não terá que reinventar a roda se souber como encontrar o que precisa na biblioteca Java, normalmente conhecida como API Java”.

2.6.1 Servlets

São objetos Java que funcionam como um pequeno servidor recebendo e respondendo requisições de clientes. Mais utilizado para gerar conteúdo dinâmico para paginas html. O servlet retorna, por exemplo, um conteúdo html ou uma imagem JPEG. Tem a capacidade de acessar banco de dados já que fica instalado no lado servidor e pode manter informações de sessão por cliente entre outras características.

2.6.2 Applets

Applets são componentes Java que rodam no Cliente ou seja no browser. Podem executar tarefas complexas e dependem de uma Jre instalada no lado do cliente. Por serem pequenos programas escritos em Java eles podem rodar independentes de plataforma. No momento da requisição de uma página html que contenha um Applet o pequeno programa é transferido para o cliente e roda em cima

da maquina virtual Java instalada, por esta característica o Applet pode se beneficiar de alguns recursos da maquina cliente.

2.6.3 JavaServer Pages – JSP

Utilizado para gerar conteúdo dinâmico em uma pagina web JSP é um arquivo com conteúdo html junto com código Java. É uma extensão da tecnologia servlet sendo assim, seu conteúdo Java é convertido em um servlet assim que carregado pela primeira vez. De acordo com Deitel (2010, p.951) “As JavaServer Pages permitem aos programadores de aplicativo Web criar conteúdo dinâmico reutilizando componentes predefinidos e interagindo com componentes que utilizam script do lado do servidor.”

2.7 LINGUAGEM HTML

A linguagem HTML (Hiper Text Markup Language) é utilizada para formatar textos que serão exibidos na internet, nela é possível fazer referências a outras páginas na web, inserir imagens e animações. O arquivo com o fonte Html é puramente um arquivo de texto produzido com qualquer editor do mais simples ao mais sofisticado, o arquivo é identificado com a extensão “.html” e é executado pelo navegador ou browser que interpreta os comandos contidos no arquivo e exibe o conteúdo.

Quando o browser lê seu HTML, ele interpreta todas as tags que envolvem seu texto. As tags são apenas palavras ou caracteres entre os sinais de menor e maior, como <head>, <p>, <h1> e assim por diante. As tags informam ao browser a estrutura e o significado de seu texto (FREEMAN, 2008, p.5).

2.8 GERENCIAMENTO DE PROJETOS

O processo de gerenciamento de projeto de software deve levantar as atividades para produção de um software assim como, estabelecer essas atividades, atribuí-las aos envolvidos, coordenar a equipe monitorando as tarefas, estipular prazos criar cronogramas dentro de um orçamento estabelecido. Segundo (BERKUN, 2008, p.28) “O gerente de projeto deve ser capaz de executar a façanha de pensar, liderar e criar estratégias que causem impacto positivo na equipe de maneira tal que poucos o conseguiriam”.

2.9 PADRÕES DE PROJETOS – DESIGN PATTERNS

Padrões de projeto ajudam desenvolvedores a solucionar problemas que ocorrem com frequências nos projetos desenvolvidos. Assim essa solução é documentada para ser aplicada quando o cenário problema ocorrer novamente em qualquer outro projeto. Esses padrões são compartilhados entre desenvolvedores do mesmo projeto e com a comunidade fazendo assim padrões que já são bem conhecidos e aplicados. Segundo (LARMAN, 2007, p.32) “Fórmulas do tipo problema-solução, devidamente nomeadas, que codificam princípios exemplares de projeto”.

3 METODOLOGIA

3.1 PROCESSO DE ENGENHARIA DE SOFTWARE

Para fazer o projeto foi utilizado o conjunto de técnicas do RUP. Todos os documentos necessários para produção do projeto estão em anexo. Como o RUP utiliza a UML como linguagem de diagramação esta linguagem também foi utilizada para gerar os diagramas do projeto, todos os diagramas utilizados também estão em anexo

3.2 GERENCIAMENTO DE PROJETOS

3.2.1 PLANO DE ATIVIDADE

Para mostrar o plano de atividade a seguir encontramos a figura 01 o Work Breakdown Structure (WBS) ou estrutura analítica do projeto. Ela faz com que o trabalho seja subdividido para melhor gerenciamento.

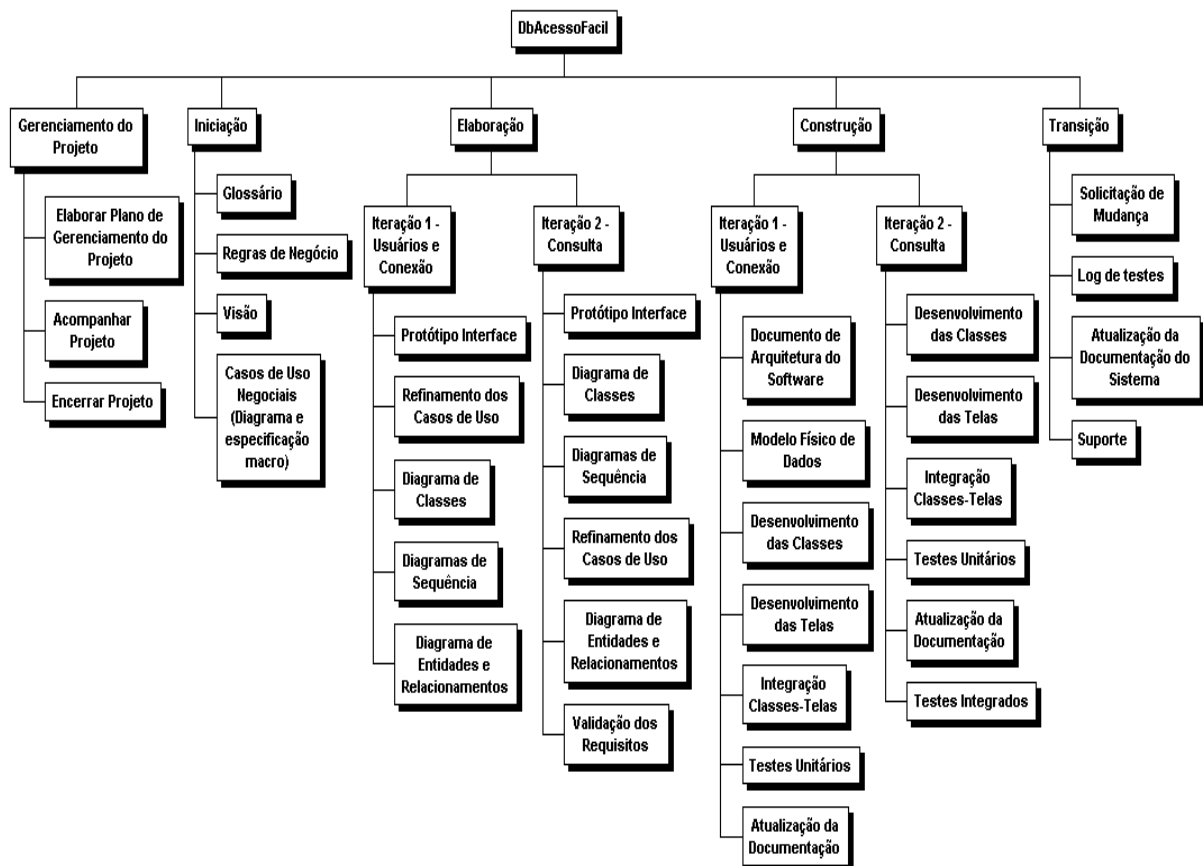


Figura 1: WBS DB Acesso Fácil

Fonte: O autor (2012)

Continuando o plano de atividades a seguir encontramos a figura 02 o Gráfico de Gantt que mostra a seqüência de atividades e quando cada uma está programada para ocorrer.

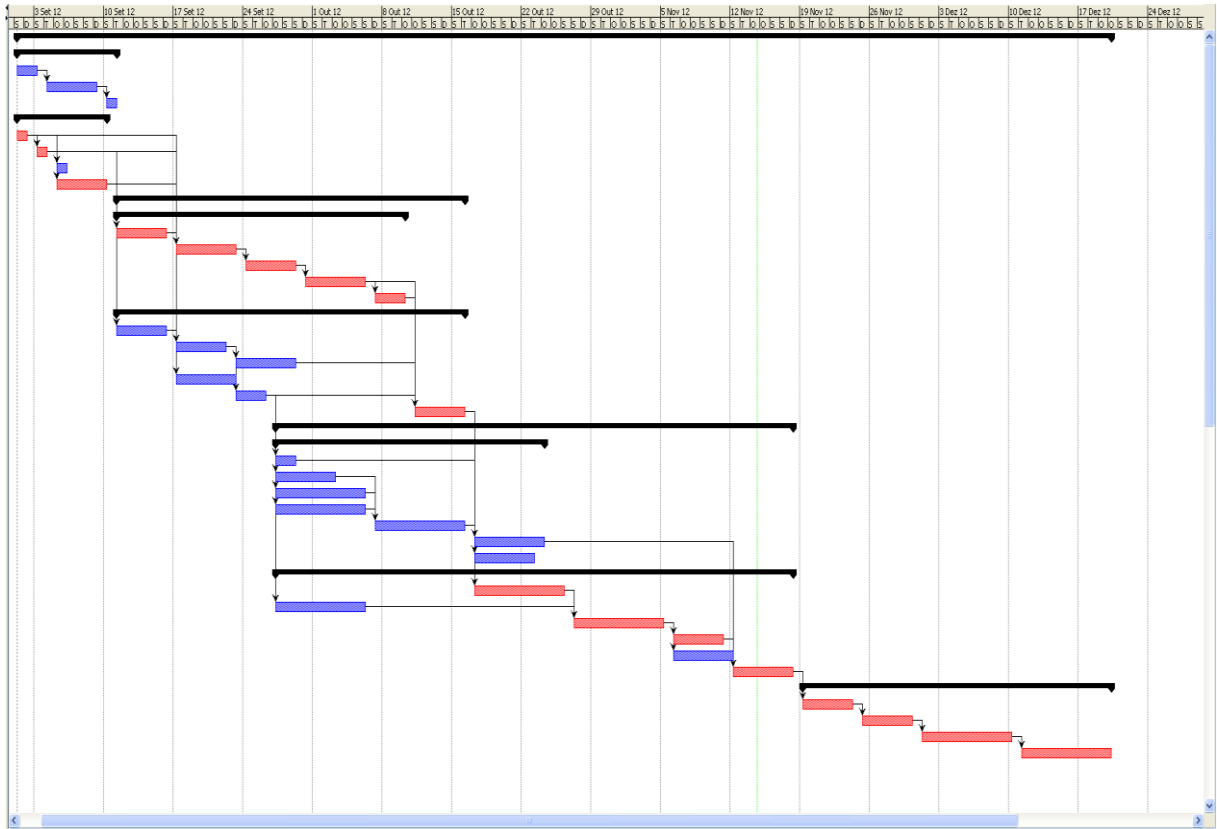


Figura 2: Gráfico de Gantt
Fonte: O autor (2012)

4 SOFTWARE

4.1 TELA DE LOGIN

A figura 03 apresenta a tela responsável pelo acesso ao sistema que somente deve estar disponível a usuários cadastrados.



A tela de login é apresentada dentro de um retângulo tracejado azul. Ela contém os seguintes elementos:

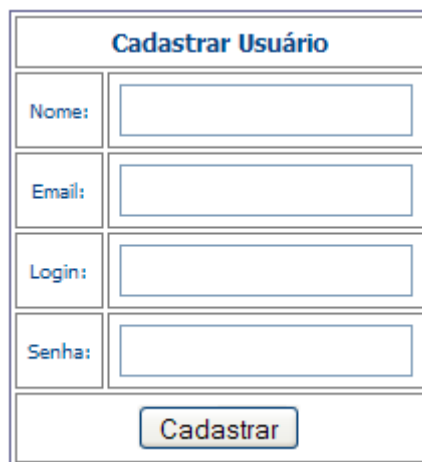
- Um rótulo "Login:" seguido de um campo de entrada de texto.
- Um rótulo "Senha:" seguido de um campo de entrada de texto.
- Um link "Novo Usuário" em azul e sublinhado, localizado na parte inferior esquerda.
- Um botão "Entrar" com um fundo cinza e borda azul, localizado na parte inferior direita.

Figura 3: Tela de Login

Fonte: O autor (2012)

4.2 TELA DE CADASTRO DE USUÁRIO

A figura 04 apresenta a tela responsável pelo cadastro dos usuários que pode ser feito através da tela de login.



A tela de cadastro de usuário é apresentada dentro de um retângulo com uma borda dupla cinza. Ela contém os seguintes elementos:

- Um cabeçalho "Cadastrar Usuário" em azul.
- Quatro campos de entrada de texto, cada um precedido por um rótulo: "Nome:", "Email:", "Login:" e "Senha:".
- Um botão "Cadastrar" com um fundo cinza e borda azul, localizado na parte inferior central.

Figura 4: Tela de Cadastro de Usuário

Fonte: O autor (2012)

4.3 TELA DE BANCO DE DADOS

A figura 05 apresenta a tela que exibe todos os bancos de dados cadastrados pelo usuário ele pode através dela abrir uma conexão, editar e excluir um banco cadastrado além cadastrar novos bancos.

Banco de Dados				
Descrição	Banco de dados			
Produção	Firebird	Abrir	Excluir	Editar
Desenvolvimento	Oracle	Abrir	Excluir	Editar
Novo				

Figura 5: Tela de Banco de dados

Fonte: O autor (2012)

4.4 TELA REGISTRAR BANCO DE DADOS

A figura 06 apresenta a tela que permite o cadastro de novos bancos de dados pelo usuário é possível testar a conexão com o banco nesta tela.

Registrar Banco de Dados	
Descrição:	<input type="text"/>
Host:	<input type="text"/>
Caminho:	<input type="text"/>
Login:	<input type="text"/>
Senha:	<input type="text"/>
<input type="button" value="Testar Conexão"/> <input type="button" value="Salvar"/> <input type="button" value="Cancelar"/>	

Figura 6: Tela Registrar Banco de dados

Fonte: O autor (2012)

4.5 TELA EXECUTAR CONSULTA

A figura 07 apresenta a tela que permite o usuário executar consulta no banco de dados. O usuário escreve o comando e clica em executar.

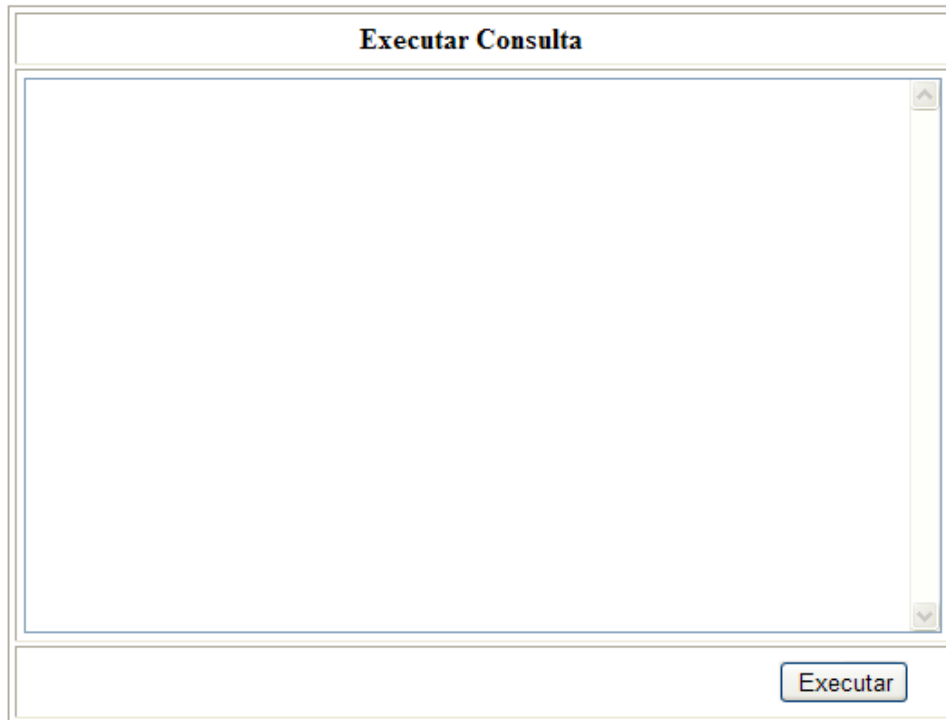


Figura 7: Tela Executar Consulta
 Fonte: O autor (2012)

4.6 TELA EXIBIR DADOS

A figura 08 apresenta a tela que mostra o resultado da consulta executada pelo usuário, mostra as colunas com os dados referentes. Caso o usuário queira salvar os dados em arquivo ele pode clicar em salvar dados.

Exibir Dados			
COIGO	NOME	ENDERECO	CPF
1	João Paulo	Rua cidade de laguana	05117212605
2	Maria Rita	Travessa da lapa	04612513521
3	Pedro Enrique	Av. jão paiva	04962112503

Figura 8: Tela Exibir dados
 Fonte: O autor (2012)

4.7 TELA SALVAR DADOS DA CONSULTA

A figura 09 apresenta a tela que permite o usuário salvar os dados da consulta realizada no banco de dados para arquivo do tipo “csv”. O usuário clica em caminho e escolhe o local onde será salvo o arquivo, depois clica em salvar.

Salvar dados da Consulta	
Salvar dados para arquivo "csv"	
<input type="button" value="Caminho"/> :	<input type="text"/>
	<input type="button" value="Salvar"/>
	<input type="button" value="Sair"/>

Figura 9: Tela Salvar Dados da Consulta

Fonte: O autor (2012)

5 CONCLUSÃO

O Processo RUP contribuiu muito na documentação de cada iteração do projeto apesar de ser um pouco trabalhosa pela quantidade de documentos que o processo propõe. Foi possível verificar que o RUP pode ser aplicado em projetos pequenos como no caso deste trabalho, pois permite adaptação não sendo necessário gerar todos os documentos apresentados por ele.

A UML permitiu a visualização mais ampla do sistema sendo possível analisar cada particularidade do sistema apenas no diagrama o que garantiu menos trabalho nas mudanças de requisitos ou alteração no software, pois se já tivesse implementação de código o trabalho seria muito maior. Dentro da gestão de projetos pode-se observar que os prazos apresentados pelo planejamento foi bem próximo do real visto que o projeto foi todo desenvolvido por uma pessoa. O modelo de dimensionamento de sistema Pontos por Caso de Uso mostrou-se um modelo de dimensionamento eficiente para ajudar a aproximar os prazos planejados da realidade.

O sistema atendeu as expectativas e executou as tarefas propostas, vários novos recursos devem ser adicionados como a execução de outros comandos SQL, salvar dados em outros tipos de arquivos

REFERÊNCIAS

- BERKUN, S. **A arte do gerenciamento de projetos**. São Paulo: Artmed, 2008.
- BEZERRA, E. **Princípios de análise e projeto de sistemas com UML**. Rio de Janeiro: CAMPUS, 2002.
- DEITEL, P; DEITEL, H. **Java: como programar**. 8. ed. São Paulo: PEARSON, 2010.
- FREEMAN, E; FREEMAN, E. **Use a cabeça: HTML com CSS e XHTML**. 2. Ed. Rio de Janeiro: ALTA BOOKS, 2008.
- LARMAN, C. **Utilizando UML e padrões**. 3. ed. São Paulo: Artmed, 2007.
- MEDEIROS, L. F. de. **Banco de dados: princípios e práticas**. Curitiba: IBPEX, 2007.
- SIERRA, K.; BATES, B. **Use a cabeça - Java**. 2. ed. Rio de Janeiro: ALTA BOOKS, 2005.

APÊNDICE

Gestão de Projetos

Estimativa de esforço de desenvolvimento para o sistema DB Acesso Fácil.

1. Peso dos Atores	
Ator	Classificação
Usuário	Simple

Tipo	Quantidade	Peso	Total
Simple	1	1	1
Médio	0	2	0
Complexo	0	3	0
		TPNAA	1

2. Peso dos Casos de Uso	
Caso de Uso	Classificação
Manter Cadastro de Usuários	Médio
Login	Simple
Registrar Banco de Dados	Médio
Download de Driver	Complexo
Abrir Banco	Médio
Executar Consulta	Complexo
Salvar Dados da Consulta	Médio
Salvar csv	Médio

Tipo	Quantidade	Peso	Total
Simple	1	1	1
Médio	5	2	10
Complexo	2	3	6
		TPNAUC	17

3. Pontos por Caso de Uso não ajustados	
PCUNA:	18

4. Fator de Complexidade Técnica			
Descrição	Peso	Fator	Fator * Peso
Sistemas Distribuídos	2	0	0
Desempenho da Aplicação	1	2	2
Eficiência do usuário final	1	3	3
Processamento interno complexo	1	1	1
Reusabilidade do código	1	2	2
Facilidade de Instalação	0,5	5	2,5
Usabilidade	0,5	4	2
Portabilidade	2	4	8
Manutenibilidade	1	3	3
Concorrência	1	0	0
Características especiais de segurança	1	3	3
Acesso direto para terceiros	1	3	3
Facilidades especiais de treinamento	1	1	1
FCT = 0,6 + (0,01 * Somatório) =			0,905

5. Fator de Complexidade Ambiental				
Descrição		Peso	Fator	Peso * Fator
Familiaridade com o processo de desenvolvimento de software	F1	1,5	3	4,5
Experiência na aplicação	F2	0,5	3	1,5
Experiência com OO, na linguagem e na técnica de desenvolvimento	F3	1	2	2
Capacidade do líder de análise	F4	0,5	3	1,5
Motivação	F5	1	5	5
Requisitos estáveis	F6	2	4	8
Trabalhadores com dedicação parcial	F7	-1	5	-5
Dificuldade na linguagem de programação	F8	-1	4	-4
FCA = 1,4 + (-0,03 * Somatório)=				0,995

6. Calcular PCUs Ajustados

PCUA = PCUNA * FCT * FCA

PCUA = 16

Estimativa de Horas:

12,3977 * 20= 324 hh

12,3977 * 28= 454 hh

Gráfico de Gantt (Cronograma)

	Ⓜ	Nome	Trabalho	Predecessoras	Duração	Início	Término
1		<input type="checkbox"/> DbAcessoFacil	456 horas		112 dias?	01/09/12 08:00	20/12/12 10:00
2		<input type="checkbox"/> Gerenciamento do Projeto	22 horas		11 dias?	01/09/12 08:00	11/09/12 10:00
3		Elaborar Plano de Gerenciamento do Projeto	6 horas		3 dias?	01/09/12 08:00	03/09/12 10:00
4		Acompanhar Projeto	12 horas	3	6 dias?	04/09/12 08:00	09/09/12 10:00
5		Encerrar Projeto	4 horas	4	2 dias?	10/09/12 08:00	11/09/12 10:00
6		<input type="checkbox"/> Iniciação	24 horas		10 dias?	01/09/12 08:00	10/09/12 10:00
7		Glossário	4 horas		2 dias?	01/09/12 08:00	02/09/12 10:00
8		Regras de Negócio	4 horas	7	2 dias?	03/09/12 08:00	04/09/12 10:00
9		Visão	4 horas	8	2 dias?	05/09/12 08:00	06/09/12 10:00
10		Casos de Uso Negociais (Diagrama e especificação)	12 horas	8;7	6 dias?	05/09/12 08:00	10/09/12 10:00
11		<input type="checkbox"/> Elaboração	132 horas		36 dias?	11/09/12 08:00	16/10/12 10:00
12		<input type="checkbox"/> Iteração 1 - Usuários e Conexão	60 horas		30 dias?	11/09/12 08:00	10/10/12 10:00
13		Protótipo Interface	12 horas	10;8	6 dias?	11/09/12 08:00	16/09/12 10:00
14		Refinamento dos Casos de Uso	14 horas	13;10	7 dias?	17/09/12 08:00	23/09/12 10:00
15		Diagrama de Classes	12 horas	14	6 dias?	24/09/12 08:00	29/09/12 10:00
16		Diagramas de Sequência	14 horas	15	7 dias?	30/09/12 08:00	06/10/12 10:00
17		Diagrama de Entidades e Relacionamentos	8 horas	16	4 dias?	07/10/12 08:00	10/10/12 10:00
18		<input type="checkbox"/> Iteração 2 - Consulta	72 horas		36 dias?	11/09/12 08:00	16/10/12 10:00
19		Protótipo Interface	12 horas	10;8	6 dias?	11/09/12 08:00	16/09/12 10:00
20		Diagrama de Classes	12 horas	19;8;7	6 dias?	17/09/12 08:00	22/09/12 10:00
21		Diagramas de Sequência	14 horas	20	7 dias?	23/09/12 08:00	29/09/12 10:00
22		Refinamento dos Casos de Uso	14 horas	19;10	7 dias?	17/09/12 08:00	23/09/12 10:00
23		Diagrama de Entidades e Relacionamentos	8 horas	20	4 dias?	23/09/12 08:00	26/09/12 10:00
24		Validação dos Requisitos	12 horas	23;21;17;16	6 dias?	11/10/12 08:00	16/10/12 10:00
25		<input type="checkbox"/> Construção	214 horas		54 dias?	27/09/12 08:00	18/11/12 10:00
26		<input type="checkbox"/> Iteração 1 - Usuários e Conexão	110 horas		28 dias?	27/09/12 08:00	24/10/12 10:00
27		Documento de Arquitetura do Software	6 horas	23	3 dias?	27/09/12 08:00	29/09/12 10:00
28		Modelo Físico de Dados	14 horas	23	7 dias?	27/09/12 08:00	03/10/12 10:00
29		Desenvolvimento das Classes	20 horas	23	10 dias?	27/09/12 08:00	06/10/12 10:00
30		Desenvolvimento das Telas	20 horas	23	10 dias?	27/09/12 08:00	06/10/12 10:00
31		Integração Classes-Telas	20 horas	30;29;28	10 dias?	07/10/12 08:00	16/10/12 10:00
32		Testes Unitários	16 horas	31	8 dias?	17/10/12 08:00	24/10/12 10:00
33		Atualização da Documentação	14 horas	31	7 dias?	17/10/12 08:00	23/10/12 10:00
34		<input type="checkbox"/> Iteração 2 - Consulta	104 horas		54 dias?	27/09/12 08:00	18/11/12 10:00
35		Desenvolvimento das Classes	20 horas	27;24	10 dias?	17/10/12 08:00	26/10/12 10:00
36		Desenvolvimento das Telas	20 horas	23	10 dias?	27/09/12 08:00	06/10/12 10:00
37		Integração Classes-Telas	20 horas	36;35	10 dias?	27/10/12 08:00	05/11/12 10:00
38		Testes Unitários	14 horas	37	7 dias?	06/11/12 08:00	11/11/12 10:00
39		Atualização da Documentação	16 horas	37	8 dias?	06/11/12 08:00	12/11/12 10:00
40		Testes Integrados	14 horas	38;32	7 dias?	12/11/12 08:00	18/11/12 10:00
41		<input type="checkbox"/> Transição	64 horas		32 dias?	19/11/12 08:00	20/12/12 10:00
42		Solicitação de Mudança	12 horas	40	6 dias?	19/11/12 08:00	24/11/12 10:00
43		Log de testes	12 horas	42	6 dias?	25/11/12 08:00	30/11/12 10:00
44		Atualização da Documentação do Sistema	20 horas	43	10 dias?	01/12/12 08:00	10/12/12 10:00
45		Suporte	20 horas	44	10 dias?	11/12/12 08:00	20/12/12 10:00