

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

ESCOLA POLITÉCNICA

DEPARTAMENTO DE ELETRÔNICA

**FAGP/ SISTEMA WEB PARA  
ACOMPANHAMENTO E GESTÃO DE PROJETOS**

Autor:

---

Rodrigo Cardoso Pereira

Orientador :

---

Aloysio de Castro P. Pedroza

Examinador :

---

Antônio Cláudio Gómez de Sousa

Examinador :

---

Marcelo Luiz Drumond Lanza

DEL

Agosto de 2004

## **Agradecimentos**

À minha família pela dedicação e ajuda.

Ao Professor Aloysio de Castro P. Pedroza pela paciência e compreensão durante a orientação deste trabalho.

Ao Prof. Antônio Cláudio Gómez de Sousa pelo auxílio com relação à modelagem do sistema.

Ao Prof. Sérgio Palma da Justa Medeiros pelo auxílio com relação a modelagem dos dados.

Aos demais amigos que contribuíram para a realização deste trabalho.

## **Resumo**

A gerência e o acompanhamento de projetos em empresas de médio e grande porte, em que os funcionários nem sempre situam-se no mesmo espaço físico, torna-se cada vez mais complicado sem o auxílio de uma ferramenta que auxilie nas tarefas exercidas pelos gerentes ou líderes de projeto.

Com o auxílio da Intranet, uma máquina servidora, linguagens ativas nos servidores, um SGBD (sistema gerenciador de banco de dados, que pode ser desde um Ingres até um SQL Server ou Oracle), podemos fazer ferramentas que podem ser acessadas por funcionários em qualquer lugar desde que possua acesso à rede e um browser.

O sistema desenvolvido nesse trabalho consiste de uma ferramenta para auxiliar gerentes a administrarem de uma forma automatizada e por meio da rede o andamento de um projeto (riscos, pontos de atenção, solicitações de mudanças, lições aprendidas, relatório de status). Neste projeto são apresentadas todas as etapas do desenvolvimento do sistema: desde a modelagem dos dados e do sistema, passando pelo desenvolvimento do software e chegando na confecção de um manual para o usuário.

# Índice

<b>AGRADECIMENTOS.....</b>	<b>II</b>
<b>RESUMO.....</b>	<b>III</b>
<b>1. INTRODUÇÃO.....</b>	<b>1</b>
<b>2. PLANO DE PROJETO PARA O FAGP.....</b>	<b>3</b>
2.1 Introdução.....	3
2.1.1 Finalidade do Plano.....	3
2.1.2 Escopo e Objetivo do Projeto.....	3
2.2 Estimativas.....	6
2.2.1 Dados Históricos Usados nas Estimativas.....	6
2.2.2 Técnicas de Estimativa.....	6
2.2.3 Estimativa do Esforço, Custo e Duração.....	6
2.3 Estratégia para o Gerenciamento dos Riscos.....	8
2.3.1 Tabela de Riscos.....	8
2.3.2 Discussão dos Riscos a Serem Gerenciados.....	8
2.3.3 Plano de RMMM para cada Risco.....	9
2.4 Cronograma.....	10
2.4.1 Estrutura do Projeto.....	10
2.4.2 Rede de Tarefas.....	10
2.4.3 Cartas de Tempo (Gantt).....	11
2.4.4 Tabela de Recursos.....	12
2.5 Recursos do Projeto.....	12
2.5.1 Pessoal.....	13
2.5.2 Software e Hardware.....	13
2.5.3 Recursos Especiais.....	14
2.6 Organização da Equipe.....	14
2.7 Mecanismos de Controle e Acompanhamento.....	14
2.7.1 Controle e Garantia da Qualidade.....	14
2.7.2 Controle e Gerenciamento das Mudanças.....	15
<b>3. ESPECIFICAÇÃO DE REQUISITOS DO FAGP.....</b>	<b>18</b>
3.1 Introdução.....	18
3.1.1 Finalidade.....	18
3.1.2 Escopo.....	18
3.1.3 Definições, Acrônimos e Abreviaturas.....	19
3.1.4 Referências.....	19
3.1.5 Resumo.....	20
3.2 Descrição Geral.....	20
3.2.1 Perspectiva do Produto.....	20
3.2.2 Funções do Produto.....	21
3.2.3 Características do Usuário.....	22
3.2.4 Restrições.....	22
3.2.5 Pressupostos e Dependências.....	23

3.3 Requisitos Específicos.....	23
3.3.1Requisitos Funcionais.....	23
3.3.2Interfaces Externas.....	30
3.3.3Requisitos de Desempenho.....	30
3.3.4Restrições de Projeto.....	30
3.3.5Atributos.....	31
<b>4. DESCRIÇÃO DE PROJETO DO FAGP.....</b>	<b>33</b>
4.1 Introdução.....	33
4.1.1Finalidade.....	33
4.1.2Escopo.....	34
4.1.3Definições e Acrônimos.....	34
4.2 Referências.....	34
4.3 Decomposição.....	34
4.3.1Decomposição em módulos.....	34
4.3.2Decomposição em processos concorrentes.....	38
4.3.3Decomposição de dados.....	38
4.4 Descrição das Dependências.....	56
4.4.1Dependência entre módulos.....	56
4.4.2Dependência entre processos.....	56
4.4.3Dependência entre dados.....	56
4.5 Descrição das Interfaces.....	58
4.5.1Interfaces dos Módulos.....	58
4.5.2Interfaces entre Processos.....	59
4.6 Projeto Detalhado.....	59
4.6.1Projeto Detalhado dos Módulos.....	59
4.6.2Projeto Detalhado das Entidades de Dados.....	69
<b>5. PROCEDIMENTO DE TESTES.....</b>	<b>81</b>
<b>6. AUTENTICAÇÃO.....</b>	<b>84</b>
<b>7. DISCUSSÃO.....</b>	<b>91</b>
7.1 Escolha das Ferramentas e Linguagens.....	91
<b>8. SUGESTÕES PARA AS PRÓXIMAS VERSÕES.....</b>	<b>95</b>
<b>9. CONCLUSÃO.....</b>	<b>97</b>
<b>10. REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>98</b>
<b>APÊNDICES.....</b>	<b>99</b>

## 1. Introdução

Aplicativos de acompanhamento e gerência de projetos são fundamentais na tarefa de administrar de maneira organizada e integrada diversos projetos ocorrendo em paralelo. Hoje em dia, nas empresas, existe cada vez mais a preocupação de existir uma metodologia de gerenciamento em que todos os funcionários participantes do projeto possam estar dando sua contribuição (seja essa contribuição através de alguma lição aprendida no decorrer do projeto ou, mais na parte gerencial, através da identificação de um ponto de atenção que poderia ter um grande impacto na fase de implantação do projeto caso não tivesse sido detectado).

Muitas empresas possuem equipes envolvidas com projetos que possuem a única finalidade de auxiliar no andamento de outros projetos. Essas equipes auxiliam o trabalho dos gerentes de cada um desses projetos, cobrando *feedbacks* (termo utilizado para poder se ter conhecimento do grau de satisfação de um funcionário a partir de seu próprio discurso) dos membros de equipe sobre o andamento de suas partes nos projeto, lembrando ao gerente da necessidade de realizar relatórios de status do projeto em prazos inferiores a um mês (geralmente quinze dias) para não perder o andamento do projeto.

O sistema desenvolvido nesse trabalho descentraliza algumas funções administrativas (na medida que permite mais a participação de cada um dos integrantes) e centraliza as informações do projeto (permitindo aos membros participantes acompanharem de maneira automatizada o seu andamento).

O software será desenvolvido para ambiente WEB em uma máquina servidora contendo um SGBD (Sistema Gerenciador de Banco de Dados) de maneira que os usuários não precisarão ter pré-compiladores ou interpretadores instalados em suas máquinas. Para a utilização do sistema bastará ao usuário final possuir um browser instalado em seu computador que interprete HTML (que pode ser quaisquer uns desses mais utilizados: Konqueror, Netscape, Internet Explorer).

No capítulo 2 é apresentado o plano de projeto do FAGP descrevendo as estimativas, as estratégias para gerenciamento dos riscos, cronograma, recursos utilizados, mecanismos para

controle e acompanhamento de qualidade e mudanças durante o projeto. O capítulo 3 tem a especificação de requisitos do FAGP apresentando uma descrição geral do produto e requisitos específicos. No capítulo 4 é apresentada a descrição de projeto do FAGP: decomposição em módulos, processos e dados, descrição das dependências entre estes, descrição das interfaces e o projeto detalhado dos módulos e entidades de dados. Em seguida será apresentado o plano de testes realizado (capítulo 5) seguido pelo capítulo que discutirá o plano de segurança utilizado no sistema (capítulo 6). O capítulo 7 tem uma discussão sobre linguagens e ferramentas utilizadas. No capítulo 8 tem a conclusão do projeto com suas considerações finais. No capítulo 9 encontra-se toda a referência bibliográfica utilizada no trabalho. No apêndice encontra-se o passo a passo de instalação e o manual do sistema.

## **2. Plano de Projeto para o FAGP**

### **2.1 Introdução**

#### **2.1.1 Finalidade do Plano**

Este capítulo tem como objetivo descrever em linhas gerais o planejamento para o projeto do software FAGP – Ferramenta de Acompanhamento e Gerência de Projeto.

A motivação do projeto surgiu da demanda por um software não-comercial de apoio à gerência de projetos no ambiente empresarial. A partir da idéia inicial de utilizá-lo no mercado, foi percebido que o software seria útil até mesmo em disciplinas da universidade que envolvessem a realização de projetos, tais como: "Projeto Integrado", "Sistemas Operacionais", "Linguagens de Programação", dentre outras incluindo o Projeto de Fim de Curso do qual nasceu a ferramenta, mesmo subutilizando as funções do software.

#### **2.1.2 Escopo e Objetivo do Projeto**

##### **2.1.2.1 Escopo**

O software se destina a auxiliar na gerência de projeto de maneira que seu escopo será o mais abrangente possível para que possa ser utilizado em qualquer tipo de projeto.

##### **2.1.2.2 Principais Funções**

As funcionalidades macro do software consistem em cadastrar e acompanhar a execução de um projeto. Alguns dos parâmetros utilizados na identificação e acompanhamento da evolução do projeto são citados na tabela a seguir:

E n t r a d a s	Cadastro de Usuários	T o t a l  = 2 4
	Cadastro de Projetos	
	Cadastro de Riscos	
	Cadastro de Pontos de Atenção	
	Cadastro de Lições Aprendidas	
	Cadastro de Solicitação de Mudanças	
	Cadastro de Relatório de Status	
	Cadastro de Fichas de Projeto	
	Consulta ao Cadastro de Usuários	
	Consulta ao Cadastro de Projetos	
	Consulta aos Riscos	
	Consulta aos Pontos de Atenção	
	Consulta de Lições Aprendidas	
	Consulta às Solicitações de Mudanças	
	Consulta aos Relatório de Status	
	Consulta de Fichas de Projeto	
	Edição do Cadastro de Usuários	
	Edição do Cadastro de Projetos	
	Edição do Cadastro de Riscos	
	Edição de Pontos de Atenção	
Edição de Lições Aprendidas		
Edição do Cadastro de Solicitação de Mudanças		
Edição do Relatório de Status		
Edição de Fichas de Projeto		
S a í d a s	Confirmação do Cadastro/Edição de Usuários	T o t a l  = 8
	Confirmação do Cadastro/Edição de Projetos	
	Confirmação do Cadastro/Edição de Riscos	
	Confirmação do Cadastro/Edição de Pontos de Atenção	
	Confirmação do Cadastro/Edição de Lições Aprendidas	
	Confirmação do Cadastro/Edição de Solicitação de Mudanças	
	Confirmação do Cadastro/Edição de Relatório de Status	
	Confirmação do Cadastro/Edição de Fichas de Projeto	

C o n s u l t a s	Resultado da Consulta ao Cadastro de Usuários	T o t a l  =  8
	Resultado da Consulta ao Cadastro de Projetos	
	Resultado da Consulta aos Riscos	
	Resultado da Consulta aos Pontos de Atenção	
	Resultado da Consulta às Lições Aprendidas	
	Resultado da Consulta às Solicitações de Mudanças	
	Resultado da Consulta ao Relatório de Status	
	Resultado da Consulta de Fichas de Projeto	

Cabe ressaltar a importância da questão da segurança de acesso ao sistema. Para tanto, será implementada a autenticação dos usuários, com diferentes níveis de permissão de acesso.

Este software é baseado em perfis – as interfaces para os usuários de cada nível de acesso são diferentes entre si. Por exemplo, o diretor do projeto pode acessar todas as funções enquanto funcionários membros de equipe, considerados de hierarquia mais baixa, não possuem permissão para acesso a todas as funções do sistema.

### 2.1.2.3 Itens de Desempenho

Este software foi feito para rodar em uma máquina servidora com Windows. Apesar disso, ele possui um rápido tempo de resposta para suas funções, tanto para cadastros quanto para consultas.

Em execução, necessita de baixa capacidade de armazenamento.

Levando-se em conta as considerações feitas acima, verifica-se que o software FAGP (Ferramenta de Acompanhamento e Gerência de Projeto) não necessitará de uma máquina com recursos especiais de hardware ou mesmo de última geração para operar. Como será utilizada uma máquina servidora com um SGBD instalado, será necessário apenas um espaço

em disco e memória RAM (random access memory) suficientes para a instalação dos softwares e armazenamento dos dados em tempo de execução respectivamente.

#### 2.1.2.4 Gerenciamento e Restrições Técnicas

O plano será gerenciado e acompanhado por meio de um cronograma e estimativas realizadas com o software COCOMO II.

As principais restrições técnicas identificadas são listadas a seguir:

- A disponibilidade de um servidor com a configuração mínima necessária;
- A instalação dos softwares no servidor.

## 2.2 Estimativas

### 2.2.1 Dados Históricos Usados nas Estimativas

As estimativas utilizadas foram obtidas através do software Cocomo II. Este software estabelece relações entre indicadores com base em dados históricos de estudos de casos de projetos de software em empresas norte-americanas.

### 2.2.2 Técnicas de Estimativa

Para as estimativas dos parâmetros do projeto foi utilizada a técnica “Ponto por Função”.

### 2.2.3 Estimativa do Esforço, Custo e Duração

Seguem os resultados obtidos do software COCOMO II para os parâmetros do projeto:

**SLOC Input Dialog - FAGP**

Sizing Method  
 SLOC  
 Function Points  
 Adaptation

Breakage  
 % of code thrown away due to requirements volatility  
 BRAK

Module Size in Function Points  
 Language

Function Type	# of Function Points			SubTotal
	Low	Average	High	
Inputs	<input type="text" value="24"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	72
Outputs	<input type="text" value="8"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	32
Files	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	0
Interfaces	<input type="text" value="30"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	150
Queries	<input type="text" value="8"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	24
Total Unadjusted Function				278
Equivalent Total in SLOC				5560

OK Cancel

Figura 1 – Dados de entrada no COCOMO para estimativa

C:\Arquivos Pessoais\Projeto Final\FAGP\_PROJETO.est - COCOMOIL.1997.1

File View Edit Parameters Calibrate Phase Help

Project Name: FAGP Scale Factor Schedule

X	Module Name	Module Size	LABOR Rate (\$/month)	EEF	NOM PM DEV	EST PM DEV	PROD	COST	INST COST	FSWP	RISK
	FAGP	F: 5560	3500.00	1.04	17.7	18.4	303.0	64233.43	11.6	2.4	0.0

	EST	Sched	PROD	COST	INST	FSWP	RISK
<b>Total SLOC:</b> <input type="text" value="5560"/>	<b>Optimistic</b>	14.7	7.0	378.7	51386.75	9.2	2.1
<b>Effort (PM):</b> <input type="text" value="17.7"/>	<b>Most Likely</b>	18.4	7.6	303.0	64233.43	11.6	2.4
<b>Productivity:</b> <input type="text" value="313.9"/>	<b>Pessimistic</b>	22.9	8.2	242.4	80291.79	14.4	2.8

Project Is Saved To File : C:\Arquivos Pessoais\Projeto Final\FAGP\_PROJETO.est

Figura 2 - Estimativa de esforço usando o software COCOMO

## 2.3 Estratégia para o Gerenciamento dos Riscos

### 2.3.1 Tabela de Riscos

RISCOS	PROBAB.	CONSEQÜÊNCIAS	CLASSIFICAÇÃO	RMM
Atraso do cronograma	50%	Moderada	Equipe	Plano 1
Problemas de hardware nos servidores	20%	Catastrófica	Tecnológica	Plano 2
Problemas de Segurança	20%	Catastrófica	Tecnológica	Plano 3
Problemas no Sistema Operacional	20%	Moderada	Tecnológica	Plano 4
Insatisfação com o produto	10%	Moderada	Usuário	Plano 5
Problemas com o browser	10%	Marginal	Tecnológica	Plano 6

### 2.3.2 Discussão dos Riscos a Serem Gerenciados

Segue uma breve discussão dos riscos a serem gerenciados:

- Atraso do cronograma:  
Passível de ser negociado para pequenos atrasos. Esse risco é crítico uma vez que os prazos para entrega do projeto são rígidos e que o projeto não é a única atividade em que a equipe está envolvida.
  
- Problemas de hardware nos servidores:  
Consistem em quaisquer problemas no hardware do servidor que impeçam o seu uso pleno. Apesar de possuir hardware redundante, a eventual perda do HD levaria a perda dos dados do trabalho, caso não seja implementada uma política de backup.
  
- Problemas de Segurança:  
O sistema é naturalmente vulnerável a ataques de hacker devido a estar conectado à Internet. A questão da segurança no local físico do servidor também deve ser levada em

consideração, já que existe a possibilidade de pessoas alheias ao projeto tentarem utilizá-lo.

➤ Problemas no Sistema Operacional:

Não pode ser ignorado o risco de problemas no Sistema Operacional, já que deste depende o pleno funcionamento dos servidores Web e de Banco de Dados.

### 2.3.3 Plano de RMMM para cada Risco

Plano 1	<b>Mitigar</b>	➤ Acompanhar a execução das tarefas pelos membros da equipe.
	<b>Monitorar</b>	➤ Verificar por qual motivo a equipe não está conseguindo realizar suas tarefas nos prazos estabelecidos; ➤ Acompanhar o cumprimento das tarefas estabelecidas nos marcos do cronograma do projeto.
	<b>Gerenciar</b>	➤ Discutir com a Banca Examinadora a possibilidade de adiar a apresentação do projeto; ➤ Aumentar e redistribuir a carga de trabalho dos membros da equipe; ➤ Apresentar um protótipo de programa que atenda parcialmente as especificações.
Plano 2	<b>Mitigar</b>	➤ Utilizar hardware confiável; ➤ Manter hardware redundante acessível; ➤ Manter backup dos dados para o caso de defeito no HD.
	<b>Monitorar</b>	➤ Monitorar constantemente o funcionamento do hardware do servidor;
	<b>Gerenciar</b>	➤ Tomar ações corretivas imediatas quando da ocorrência de qualquer problema de hardware; ➤ Substituir o hardware defeituoso pelo hardware reserva; ➤ Utilizar o backup em caso de perda dos dados do servidor.
Plano 3	<b>Mitigar</b>	➤ Restringir o acesso de pessoas alheias ao projeto à sala do servidor; ➤ Configurar o servidor com o máximo de restrições de segurança.
	<b>Monitorar</b>	➤ Acompanhar advisories de novos bugs dos softwares do servidor.
	<b>Gerenciar</b>	➤ Corrigir imediatamente os bugs dos softwares anunciados em advisories; ➤ Tomar ações imediatas quando da ocorrência de ataques de hackers ao sistema.
Plano 4	<b>Mitigar</b>	➤ Não sobrecarregar a configuração do servidor; ➤ Instalar somente os softwares necessários ao projeto do produto; ➤ Manter os CDs de instalação acessíveis.
	<b>Monitorar</b>	➤ Monitorar constantemente o funcionamento do Sistema Operacional do servidor.
	<b>Gerenciar</b>	➤ Tomar ações corretivas imediatas quando da ocorrência de qualquer problema no Sistema Operacional; ➤ Reinstalar o Sistema Operacional assim que este apresentar instabilidade.

## 2.4 Cronograma

### 2.4.1 Estrutura do Projeto

Este projeto é composto de apenas um módulo.

### 2.4.2 Rede de Tarefas

A seguir é apresentada uma rede PERT com as tarefas a serem desenvolvidas neste projeto. A rede apresenta nós, que representam uma data, um marco, ou um evento; apresenta, também, ramos, que representam as atividades ou tarefas desenvolvidas entre dois marcos. As caixas de texto retangulares que apontam para cada marco indicam a descrição de tal marco.

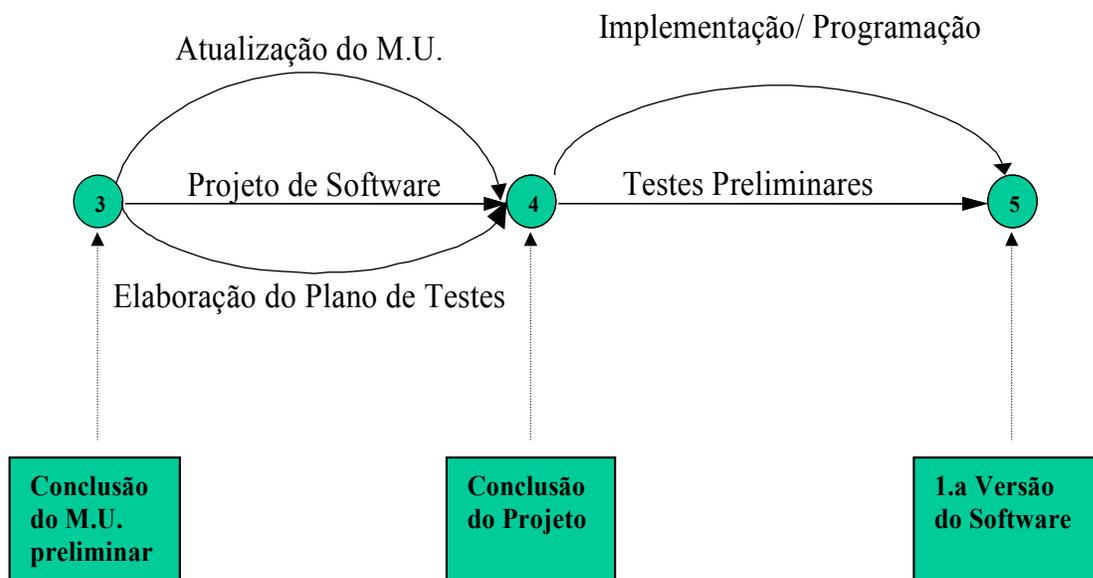


Figura 4 – Rede PERT (continuação)

**Figura 5 – Rede PERT (continuação)**

### **2.4.3 Cartas de Tempo (Gantt)**

A tabela a seguir apresenta os eventos numerados e suas respectivas datas, juntamente com os principais acontecimentos e realizações destas datas:

<b>Eventos</b>	<b>Datas</b>	<b>Realizações</b>
1	01-Fev-04	Início do Projeto e da Documentação
2	01-Mar-04	Conclusão da Análise do Projeto
3	01-Abr-04	Conclusão do M.U. Preliminar
4	01-Mai-04	Conclusão do Projeto
5	01-Jul-04	Versão Preliminar do Software
6	01-Ago-04	Resultado dos Testes de Erros

7	10-Ago-04	Versão Final do Software e Manual
8	16-Ago-04	Defesa do Projeto

#### 2.4.4 Tabela de Recursos

O total de recursos necessário à realização do projeto estão listados no próximo item deste plano de projeto e podem ser divididos em três grandes grupos: recursos de hardware, recursos de software e recursos pessoais.

A tabela seguinte relaciona as tarefas entre dois eventos com os recursos necessários a sua realização:

Eventos	Tarefas	Recursos
1-2	Planejamento	Pessoal
	Elaboração do Plano de Trabalho	Pessoal, computador, Windows e MS Office
	Idealização do Banco de Dados	Pessoal, computador, Erwin, SGBD SQL Server
	Idealização das Interfaces Gráficas	Pessoal
	Modelagem do Código	Pessoal
2-3	Instalação e Configuração dos Softwares no Servidor	Pessoal, computador, Windows Server, SGBD SQL Server, FrontPage
	Elaboração do M.U. preliminar	Pessoal, computador, Windows, MS Office
3-4	Atualização do Manual do Usuário	Pessoal, computador, Windows, MS Office
	Projeto de Software	Pessoal, consulta a material didático
	Elaboração do Plano de Testes	Pessoal, computador, MS Office
4-5	Implementação/ Programação	Pessoal, computador, Windows Server, SQL Server, Browser, FrontPage
	Testes Preliminares	Pessoal, computador, Windows Server, MS Office, Browser
5-6	Elaboração do Manual de Instalação	Pessoal, computador, Windows, MS Office
	Elaboração do M.U. Final	Pessoal, computador, Windows, MS Office
	Testes Sistemáticos	Pessoal, computador, Windows Server, SQL Server, Browser
6-7	Correção dos Possíveis Erros	Pessoal, computador, Windows Server, SGBD SQL Server, Browser, FrontPage
7-8	Documentação Final e Apresentação do Projeto e Manuais	Pessoal, computador, Windows, MS Office

## 2.5 Recursos do Projeto

### **2.5.1 Pessoal**

- Desenvolvedor: Rodrigo Cardoso Pereira
- Orientador: Professor Aloysio de Castro P. Pedroza

### **2.5.2 Software e Hardware**

#### *SOFTWARE*

- Microsoft Windows 2000;
- MS IIS Server 5.0;
- MS SQL Server 7.0;
- Internet Explorer 6.0;
- Microsoft FrontPage 2000;
- Microsoft Word 2000;
- Microsoft PowerPoint 2000;
- COCOMO II;
- ERwin 4,1;
- AxiomSys V6 Demo;
- UltraEdit-32.

#### *HARDWARE*

As necessidades de hardware serão supridas por um laptop COMPAQ M700, 196 Mb de RAM, 700 Mhz de clock, 20 Gb de HD.

### **2.5.3 Recursos Especiais**

Entre os principais recursos utilizados podem ser citados livros, tutoriais, páginas na Internet e conhecimentos de projetos de consultoria realizados ao longo de dois anos trabalhando em empresas de telecomunicações.

## **2.6 Organização da Equipe**

- Rodrigo Cardoso Pereira: Modelagem e implementação das interfaces e do Banco de Dados, programação, instalação e configuração dos softwares, elaboração da documentação;
- Professor Aloysio de Castro P. Pedroza: orientar o andamento do projeto, revisar documentações.

## **2.7 Mecanismos de Controle e Acompanhamento**

### **2.7.1 Controle e Garantia da Qualidade**

A garantia de qualidade será assegurada pela validação de técnicas de desenvolvimento de projeto em cada uma das etapas definidas na Carta de Tempo. Esta validação avaliará técnicas de modelagem de dados, manutenibilidade do software, reusabilidade de código, modularidade, integridade, testabilidade, interface amigável, entre outros.

Caso venha a ser utilizado comercialmente será utilizado um programa para controle de qualidade baseado em três objetivos principais: clientes satisfeitos, pessoas motivadas e desenvolvimento equilibrado do produto.

O fundamento para a gestão de qualidade para o cliente deve basear-se num modelo que forneça uma abordagem e linguagem comum na identificação das expectativas, planejamento de qualidade, medida de desempenho e aperfeiçoamento contínuo. Para isso deve ser identificado cinco etapas críticas aos esforços de gestão de qualidade: Identificação das Expectativas do Cliente (Expectativas), Planejamento para a Qualidade (Planejamento), Execução do Plano (Execução), Avaliação do Progresso (Conferência) e Aperfeiçoamento Contínuo (Customização).

Na etapa das **Expectativas**, a equipe de comprometimento deve identificar continuamente todas as necessidades e expectativas dos clientes. A etapa do **Planejamento** é onde as expectativas do cliente são traduzidas em objetivos da equipe. O plano deve identificar a métrica de desempenho e medir os procedimentos para processos-chave e resultados finais. A etapa da **Execução** concentra-se na implementação do Plano de Qualidade como uma parte integral da prestação de serviços ao cliente. Os componentes-chave desta etapa são a utilização de metodologias detalhadas e da cobrança de métricas apropriadas. Isto fornece à equipe uma medida em vigor do que esperar no que diz respeito aos seus objetivos e metas de qualidade. A etapa de **Conferência** assegura que os objetivos de qualidade sejam atingidos continuamente enquanto um comprometimento durar. A informação obtida durante a Execução é utilizada para analisar quaisquer brechas entre métricas objetivadas e reais. A etapa final, **Customização**, está voltada para a implementação dos aperfeiçoamentos aos processos de comprometimentos comerciais. As contribuições da equipe são também reconhecidas nesta etapa.

Os seguintes benefícios são atingidos por esse modelo:

- Esforços da equipe concentrados onde são mais necessários;
- Todos os integrantes da equipe trabalhando em prol de uma meta comum;
- Especialização retida pelo cliente;
- Crescente satisfação profissional dos membros da equipe;
- Operação de acertos freqüentemente e,
- Crescente satisfação do cliente (baseado em resultados de pesquisa).

## **2.7.2 Controle e Gerenciamento das Mudanças**

O gerenciamento da mudança em um projeto de fim de curso é um fator crítico de sucesso. As mudanças de escopo devem ser controladas e os procedimentos de tratamento acordado entre as partes para evitar desgastes e problemas no trabalho conjunto, re-trabalho e atraso na entrega do produto final.

É entendida por mudança evolutiva a alteração solicitada em soluções já implantadas cuja responsabilidade do desenvolvedor seja de manutenção da solução. Já a mudança de escopo, são

alterações que ocorrem durante o desenvolvimento de uma determinada solução. Todavia, o gerenciamento de mudança solicitada seguirá um mesmo processo de tratamento.

Qualquer solicitação de mudança no decorrer do projeto, de escopo ou cronograma, será devidamente documentada e discutida com o orientador do projeto.

Nos marcos do projeto será verificado se as fases recém concluídas apresentaram inconsistências em relação aos parâmetros estipulados previamente no projeto.

### **3. Especificação de Requisitos do FAGP**

#### **3.1 Introdução**

##### **3.1.1 Finalidade**

Este capítulo tem como objetivo apresentar a Especificação dos Requisitos do Software FAGP – Ferramenta de Acompanhamento e Gestão de Projetos. A audiência para o qual é dirigido abrange inicialmente a banca de projeto final. Em médio prazo, o aluno responsável pelo projeto pretende apresentar esse mesmo documento a clientes em potencial.

A Especificação de Requisitos de Software pretende descrever as principais funcionalidades do programa, requisitos de instalação e utilização, detalhes da lógica das funções, fluxo de dados, seqüência de estados, esquema do armazenamento de dados, público alvo, dentre outros aspectos.

##### **3.1.2 Escopo**

O software se destina a auxiliar na gerência de projeto de maneira que seu escopo será o mais abrangente possível para que possa ser utilizado em qualquer tipo de projeto. Entre os principais benefícios do sistema podem ser destacados:

- Facilidade para que todos os membros de uma equipe acompanhem o andamento e desenvolvimento das ações tomadas durante a execução de um projeto;
- Facilidade para o gerente de projeto controlar a resolução de riscos, pontos de atenção ou a aprovação de uma solicitação de mudança;
- Base de dados com lições aprendidas nos projetos;
- Sinalizar resoluções de riscos, pontos de atenção que estiverem fora do prazo de solução;
- Geração de relatórios de status com a possibilidade de realizar upload de arquivos gerados em outras ferramentas como: MS WORD, MS Project.

### **3.1.3 Definições, Acrônimos e Abreviaturas**

- HTML – Hyper Text Markup Language. HTML é a linguagem tradicional usada para criar páginas WEB com programação de hipertexto (documento com palavras ou imagens que levam para outras páginas com um clique).
- HTTP – Hyper Text Transfer Protocol. HTTP é um protocolo da camada de aplicação do modelo OSI utilizado para transferência de dados na World Wide Web.
- SGBD – Sistema de Gerenciamento de Banco de Dados.
- INTRANET – Termo usado para caracterizar uma rede local de computadores (restritos a uma determinada área, prédio ou instituição) que utiliza a mesma tecnologia desenvolvida para a Internet.
- QUERY – Termo utilizado para referenciar acesso ao Banco de Dados ou para leitura ou para escrita.
- ASP – Active Server Pages. Tecnologia proprietária da Microsoft comumente utilizada para desenvolvimento WEB.
- JAVASCRIPT – Linguagem criada pela Netscape que serve basicamente para dar mais recursos ao navegador de Internet.
- ODBC – Open DataBase Connectivity.
- DFD – Diagrama de Fluxo de Dados.
- DER – Diagrama de Entidades e Relacionamentos
- ERS – Especificação de Requisitos de Software.

### **3.1.4 Referências**

- PRESSMAN, R.S. Software engineering: a practioner's approach. 5a.ed. McGraw-Hill (series in computer science), 2001

### **3.1.5 Resumo**

Segue os itens abordados nessa parte da documentação:

- Diagrama de Fluxo de Dados:  
Descrição em níveis do fluxo de informação entre os processos do sistema.
  
- Dicionário de Dados:  
Detalhamento de agregados de dados referidos no sistema.
  
- Especificação da Lógica dos Processos:  
Descrição da lógica de todas as funções do software.
  
- Diagrama de Estados:  
Seqüência de Estados possíveis do software.
  
- Diagrama de Entidades e Relacionamentos:  
Representação gráfica do relacionamento entre diferentes dados, entre dados e seus atributos, e a organização dos agregados de dados do sistema.

## **3.2 Descrição Geral**

### **3.2.1 Perspectiva do Produto**

O sistema foi desenvolvido em ASP o que significa que deverá ficar numa máquina servidora de http. O aplicativo fará acesso a Banco de Dados utilizando o MS SQL Server 7.0. Como a linguagem ASP não roda no browser do usuário o sistema será independente de tipos ou versões de browser (a restrição poderia estar por conta da utilização de JavaScript para validação de preenchimento de formulários. No entanto, tomou-se o cuidado de utilizar a versão 1.0 do JavaScript garantindo que rodará em qualquer browser.). Como restrição ao usuário o sistema

necessita que este instale uma máquina virtual interpretadora de applet em sua máquina para que possa visualizar o menu da ferramenta.

### **3.2.2 Funções do Produto**

As principais funções do FAGP (Ferramenta Gerência e Acompanhamento de Projetos) são enumeradas a seguir separadas por módulo:

#### **Usuários**

- Cadastro
- Alteração de dados cadastrais
- Consulta
- Alteração de senha

#### **Projetos**

- Cadastro
- Alteração de dados
- Consulta
- Inserção de novas atividades
- Anexo de arquivos

#### **Riscos**

- Cadastro
- Alteração de dados
- Consulta
- Remoção
- Inserção de histórico

#### **Ponto de Atenção**

- Cadastro

- Alteração de dados
- Consulta
- Remoção

### **Lições Aprendidas**

- Cadastro
- Alteração de dados
- Consulta
- Remoção
- Anexo de arquivos

### **Solicitações de Mudança**

- Cadastro
- Alteração de dados
- Consulta
- Remoção

## **3.2.3 Características do Usuário**

Definição do perfil básico do usuário do software:

- |                                      |   |                                      |
|--------------------------------------|---|--------------------------------------|
| ➤ Idade                              | → | faixa etária jovem/ adulta           |
| ➤ Escolaridade                       | → | nível médio/ superior/ pós-graduação |
| ➤ Frequência de uso                  | → | casual/ permanente                   |
| ➤ Experiência com sistemas similares | → | média/ alta                          |

## **3.2.4 Restrições**

Como o software desenvolvido rodará numa máquina servidora, as restrições se aplicam somente em relação a esta máquina. Essa máquina deverá conter processador e memória RAM suficiente

para rodar aplicações servidoras em Banco de Dados SQL Server. São sugeridos como requisitos mínimos: Pentium II (ou equivalente de outros fabricantes) com 128 Mb de RAM.

Essa máquina deverá possuir um sistema operacional servidor Windows rodando ou então Windows 2000 ou superior com IIS instalado.

### **3.2.5 Pressupostos e Dependências**

O desenvolvimento do software depende da disponibilidade de uma estação de trabalho servidora com os seguintes softwares instalados:

- Windows 2000 Server;
- MS IIS Server 3.0 ou superior;
- MS SQL Server 7.0;
- Internet Explorer;
- UltraEdit-32;
- ERwin 4.1;
- Microsoft Office 2000;
- COCOMO II;
- AxiomSys V6 Demo.

Quanto às restrições de hardware temos aquelas descritas no item anterior.

## **3.3 Requisitos Específicos**

### **3.3.1 Requisitos Funcionais**

Abaixo é realizada uma descrição funcional do sistema:

O software FAGP deverá permitir que um projeto seja gerenciado/ acompanhado via rede Intranet/ Internet de computadores. Nele deverá ser possível o cadastro/ consulta de Riscos,

Pontos de Atenção, Solicitação de Mudança, Relatórios de Status do projeto, Projetos em andamento, Usuários participantes dos projetos, Reuniões de projeto, além da alteração e exclusão destes.

De maneira automatizada os membros de uma equipe terão acesso às informações dos projetos que estão participando. Para uma maior interação entre sistema/ usuário o software ainda sinalizará atrasos na resolução de Riscos, Solicitações de Mudanças e Pontos de Atenção.

Os acessos às funções de cadastro e alteração de informações do sistema serão restritos a determinados perfis associados a cargos empresariais (Gerente, Diretor) ou ao administrador do sistema.

O desenvolvimento do software foi o mais abrangente possível de maneira que possa ser utilizado em qualquer tipo de projeto, desde projetos em empresas como projetos acadêmicos, neste último com a evidente subutilização de algumas funções.

Depois de instalado o software, toda a administração poderá ser realizada de modo gráfico pelo próprio sistema, não necessitando inserção em tabela via DDL ou outros acessos a Banco de Dados por modo texto.

Foram utilizadas as ferramentas CASE AxiomSys v6 Demo e ERwin 4.1 para a análise estruturada do software e modelagem de dados respectivamente. A seguir são apresentados os diagramas de contexto, fluxo e a modelagem de dados do sistema:

**Figura 6 - Diagrama de  
Contexto**

**Figura 7** - Diagrama de fluxo de dados nível 1 (Ferramenta de Gerência de Projetos)

**Figura 8** - Diagrama de fluxo de dados nível 2 (Relatório de Status)

**Figura 9** - Diagrama de fluxo de dados nível 2  
(Administração)

Os diagramas de fluxo de dados de Risco, Pontos de Atenção, Lições Aprendidas e Solicitação de Mudanças seguem o mesmo modelo do diagrama do Relatório de Status apenas alterando para a entidade de dados correspondente aquele módulo.

A seguir é apresentada a modelagem de dados do sistema ou DER (diagrama de entidades-relacionamento):

**Figura 10** - Modelagem de dados do sistema (DER)

### **3.3.2 Interfaces Externas**

➤ Interface dos Usuários:

Interface WEB. O usuário utilizará um browser para interagir com o sistema através de formulários HTML.

➤ Interfaces de Comunicação:

Redes baseadas no protocolo TCP/ IP (Internet/ intranet)

### **3.3.3 Requisitos de Desempenho**

O principal requisito de desempenho do projeto se refere ao tempo de resposta do software que deverá ser avaliado de maneira qualitativa. Desta maneira será utilizado como forma de mensurar este requisito o ritmo de trabalho normal do usuário. Portanto, deverão ser realizados testes junto à área usuária para a validação do tempo de resposta e obtenção da satisfação no resultado final.

### **3.3.4 Restrições de Projeto**

As restrições de projeto do software FAGP no que concerne o software e o hardware são apresentadas no item deste documento. Do ponto de vista do Usuário, o software FAGP é

acessado via WEB usando qualquer browser, não tendo dependência em relação a outros aplicativos.

Já do ponto de vista do Servidor, é necessário um PC configurado com Windows 2000 Server ou Windows NT Server, MS IIS e SGBD MS SQL Server, na sua configuração essencial.

Em relação ao hardware do servidor, na falta de critérios de desempenho quantitativos, a performance das aplicações será avaliada experimentalmente ao longo do projeto.

### **3.3.5 Atributos**

Os principais atributos de qualidade do software são listados abaixo:

- Rápido – o tempo de resposta do software deve ser imperceptível para o usuário;
- Amigável – por se basear em formulário HTML para WEB o sistema foi desenhado de maneira a ser de fácil utilização pelos usuários;
- Confiável – utilizando criptografia para armazenamento de senhas no Banco de Dados, baseando o acesso às funções a partir de perfis de usuários e, não deixando o acesso a nenhuma das funções do sistema sem que antes seja realizado o login são algumas das medidas de segurança tomada no sistema;
- Expansível – por se basear em ASP o desenvolvimento do software e, devido às facilidades apresentadas por essa linguagem, o sistema torna-se facilmente customizável, podendo ser inclusive instalado e adaptado à necessidade do usuário no site que for utilizá-lo;

- Manutível – devido à experiência do desenvolvedor em diversos projetos de programação, o software foi codificado de maneira organizada e intuitiva para que qualquer programador com o mínimo de experiência nas ferramentas utilizadas possa dar manutenção.

## **4. Descrição de Projeto do FAGP**

### **4.1 Introdução**

#### **4.1.1 Finalidade**

Este capítulo tem como finalidade apresentar a Descrição do Projeto do Software FAGP baseado no apêndice exemplificativo da norma ANSI/IEEE 1016 simplificada.

A audiência para a qual este texto é dirigido abrange inicialmente a banca de Projeto Final do Departamento de Engenharia Eletrônica. A equipe responsável pelo projeto planeja, a médio prazo, apresentar esta documentação a clientes em potencial.

O nome do produto – FAGP se refere à Ferramenta de Acompanhamento e Gerência de Projetos. A escolha desse título se deveu principalmente à necessidade de seguir uma tendência de mercado em que muitos softwares são denominados por acrônimos. No processo de escolha do nome do produto, ao mesmo tempo em que títulos por extenso eram discutidos, seus respectivos acrônimos eram analisados. A questão do impacto causado pelo acrônimo, principalmente do ponto de vista fonético foi bastante discutida.

O ciclo de vida do projeto tem sido desenvolvido segundo o método Cascata, ou seja, com a sucessão de fases bem definidas: Planejamento, Especificação de Requisitos e Descrição do Projeto do Software. Com relação ao ciclo de vida do produto, a equipe responsável planeja duas linhas de ação mutuamente exclusivas, dependentes do cenário futuro do projeto. Caso se confirme a demanda pelo produto, a equipe pretende implementar Versões sucessivas, baseadas em erros identificados pelos usuários. Caso contrário, a equipe não pretende investir em novas versões após a conclusão do projeto final. No entanto, devido à expansibilidade do software espera-se que o seu ciclo de vida seja bastante longo.

### **4.1.2 Escopo**

O escopo do FAGP foi definido previamente na seção .

### **4.1.3 Definições e Acrônimos**

Principais termos utilizados neste capítulo:

- PK – Primary Key. Também conhecido como chave primária, indica um atributo ou conjunto de atributos que identificam um determinado registro da tabela.
- FK – Foreign Key. Também conhecido como chave estrangeira, indica um atributo de uma tabela que referencia a primary key de outra tabela.

## **4.2 Referências**

- PRESSMAN, R.S. Software engineering: a practioner's approach. 5ª edição McGraw-Hill (series in computer science), 2001.

## **4.3 Decomposição**

### **4.3.1 Decomposição em módulos**

O software desenvolvido neste projeto é composto de apenas um módulo, conforme já mencionado no Plano de Projeto. Esse módulo é composto de várias funções, que por sua vez também podem ser consideradas módulos. A figura a seguir mostra a arquitetura do módulo e suas funções organizadas hierarquicamente:

**Figura 11** – Arquitetura do módulo

Para iniciar o uso do programa o usuário deverá digitar seu login e senha. Essas informações são conferidas nos dados cadastrais da tabela de usuários afim de autenticá-lo. Caso o sistema não valide essas informações (informações correspondentes não encontradas na base de dados), as funcionalidades do software não estarão disponíveis e um novo para login/ senha será solicitado.

Após a autenticação ter sido confirmada, será mostrado um menu contendo as funções gerenciais que deverão estar disponíveis de acordo com os diferentes perfis de usuário. Na figura mostrada anteriormente estão mostradas todas as funções, considerando que aquelas que estão explícitas para Riscos deverão se repetir para SM (Solicitação de Mudança), RS (Relatório de Status), PA (Ponto de Atenção), LA (Lições Aprendidas) e Administração. Sendo que para o módulo de Administração as funções estão associadas com o cadastro/ consulta/ edição/ exclusão de projetos e usuários.

A seguir é mostrada uma descrição esquemática do menu:

**Funções:**

- Risco
  - Novo Risco
  - Consultar
  - Editar
    - Inserir Histórico
    - Fechar
  - Excluir
- Ponto de Atenção
  - Novo Ponto de Atenção
  - Consultar
    - Editar
      - Fechar
    - Excluir
- Lição Aprendida
  - Nova Lição Aprendida
  - Consultar
    - Editar
    - Excluir
- Solicitação de Mudança
  - Nova Solicitação de Mudança
  - Consultar
    - Editar
    - Excluir
- Relatório de Status
  - Novo Relatório de Status
  - Consultar

- Editar
- Excluir
- Administração
  - Novo Usuário
  - Consultar Usuário
    - Editar Usuário
  - Novo Projeto
    - Consultar Projeto
    - Editar Projeto

Como pode ser observado neste esquema, todas as funções permitem fazer o mesmo tipo de manipulação com os dados. Desse modo, a arquitetura de cada uma delas. Por isso, e para poupar espaço, a figura com a arquitetura do programa só faz o detalhamento hierárquico da função Risco.

Na função Risco, é apresentado um menu para escolha entre cadastro e consulta de riscos. A opção consulta só será válida caso existam dados cadastrados. O usuário fará uma opção que será lida e o despachante de menu o levará a função escolhida.

A função Novo Risco é dividida em entrada, validação, manipulação e saída de dados. O usuário entra com os dados de um novo risco (entrada de dados), estes dados são validados por um código JavaScript, caso os dados estejam preenchidos corretamente estes deverão ser armazenados no Banco de Dados (manipulação) e uma tela de confirmação deverá ser mostrada ao usuário confirmando o cadastro das informações do risco (saída de dados).

A função consulta é dividida em três partes bem definidas: entrada, manipulação e saída de dados. Como entrada o usuário escolherá entre várias opções aquelas que serão usadas para filtrar a consulta. Na parte de manipulação, estes dados serão pesquisados dentre os registros existentes e será mostrada uma saída com os Riscos que satisfaçam os critérios da busca realizada. No formulário que mostrará os dados de saída, o Risco poderá ser selecionado para edição ou poderá ser excluído da Base de Dados.

A função de edição de Riscos é dividida em entrada dos novos dados, atualização do Banco de Dados com esses dados e será mostrada uma tela de confirmação das alterações realizadas.

A função de exclusão da Base de Dados resume-se em selecionar no formulário de resultado da consulta aqueles Riscos que deverão ser excluídos (podendo-se escolher a opção todos) e confirmar a exclusão desses do Banco de Dados. Após a confirmação deverá ser mostrada uma página HTML de saída confirmando a exclusão.

Como já havia sido mencionado, as funções: Solicitações de Mudanças, Relatório de Status, Lições Aprendidas, Pontos de Atenção e Administração possuem estrutura hierárquica idêntica à explicada para riscos.

#### **4.3.2 Decomposição em processos concorrentes**

Apesar do software ser composto de dois processos (cliente e servidor), apenas a parte servidora foi desenvolvida neste projeto já que o cliente consistia apenas de um browser. Contudo, o software pode ser utilizado por vários usuários simultâneos via Web utilizando concorrentemente os servidores HTTP ou o servidor de Banco de Dados. Mas o controle desses processos deverá ser realizado pelo sistema operacional e o servidor de Banco de Dados correspondente, portanto, não fazendo parte do escopo do software desenvolvido.

#### **4.3.3 Decomposição de dados**

A seguir serão apresentados todas as entidades utilizadas no software, seus atributos e descrição de cada um deles:

- tab\_tipo\_solicitacao\_mudanca

Esta entidade armazena os tipos de solicitações de mudanças que poderão ser cadastradas no software, como, por exemplo: escopo ou cronograma.

<b>Atributos</b>	<b>Tipo</b>	<b>Descrição</b>
cd tipo_solicitacao	int	Código identificador para tipo de solicitação de mudança
de_solicitacao	char(20)	Descrição da solicitação de mudança

- tab\_status

Esta entidade armazena o status das ações relacionadas com o tratamento das informações gerenciais cadastradas na ferramenta. Por exemplo: Iniciado, Aguardando Ocorrência, Realizado.

<b>Atributos</b>	<b>Tipo</b>	<b>Descrição</b>
cd_status	int	Código identificador do status da ação.
status	char(25)	Descrição das opções de status das ações.

- tab\_tipo\_resposta\_risco

Esta entidade armazena os tipos de respostas relacionadas à ação que deverá ser tomada para a resolução do Risco. Essas respostas poderão ser mitigar, eliminar ou aceitar o Risco, por exemplo.

<b>Atributos</b>	<b>Tipo</b>	<b>Descrição</b>
cd_tipo_resposta	int	Código identificador do tipo de resposta dada ao Risco.
de_tipo_resposta	char(20)	Descrição dos tipos de resposta dados.

- tab\_atividades

Esta entidade armazena os tipos de atividades desenvolvidas no período de realização do relatório de status do projeto. As atividades poderão ser classificadas em: atividades em andamento, iniciadas no período ou realizadas, por exemplo.

<b>Atributos</b>	<b>Tipo</b>	<b>Descrição</b>
cd atividade	int	Código identificador do tipo de atividade desenvolvida.
de atividade	char(40)	Descrição do tipo de atividade desenvolvida.

- tab\_criticidade

Esta entidade armazena as criticidades dos pontos de atenção cadastrados no sistema. A criticidade de um ponto de atenção pode ser classificada em Alta, Média ou Baixa.

<b>Atributos</b>	<b>Tipo</b>	<b>Descrição</b>
cd criticidade	int	Código identificador da criticidade do ponto de atenção.
criticidade	char(20)	Descrição da criticidade dos pontos de atenção.

- tab\_diretoria

Esta entidade armazena as diretorias ao qual deverão estar associado os projetos. Aplica-se mais no caso do sistema estar sendo utilizado em ambiente corporativo empresarial. Essa diretoria poderia ser de Sistemas, Qualidade ou Marketing, por exemplo.

<b>Atributos</b>	<b>Tipo</b>	<b>Descrição</b>
cd diretoria	int	Código identificador da diretoria.
no diretoria	char(30)	Nome da diretoria.

- tab\_empresa

Esta entidade armazena os nomes das empresas associadas aos membros da equipe do projeto levando-se em consideração que poderão ser contratadas empresas terceirizadas ou realizado parceria com outras empresas durante a execução de um mesmo projeto.

<b>Atributos</b>	<b>Tipo</b>	<b>Descrição</b>
cd_empresa	int	Código identificador da empresa.
no_empresa	char(30)	Nome da empresa associada o usuário da ferramenta.

- tab\_fase

Esta entidade armazena a fase em que o projeto está no momento do cadastro do Risco, Ponto de Atenção ou Lição Aprendida. Pode ser: fase de planejamento, concepção ou execução, por exemplo.

<b>Atributos</b>	<b>Tipo</b>	<b>Descrição</b>
cd_fase	int	Código identificador da fase do projeto
fase	char(20)	Nome da fase em está o projeto no momento do cadastro.

- tab\_impacto\_risco

Esta entidade armazena o impacto no projeto associado ao Risco caso ele não venha a ser solucionado. Esse impacto pode ser Alto, Médio ou Baixo.

<b>Atributos</b>	<b>Tipo</b>	<b>Descrição</b>
cd_impacto	int	Código identificador do impacto, caso o Risco não seja solucionado.
impacto	char(20)	Descrição do impacto do Risco.

- tab\_nivel\_autoridade

Esta entidade armazena os níveis de autoridades empresariais necessárias para o fechamento de um Risco. Esse níveis podem ser Presidencial, Gerencial, Operacional entre outros.

<b>Atributos</b>	<b>Tipo</b>	<b>Descrição</b>
cd_nivel_autoridade	int	Código identificador associado aos níveis de autoridades existentes.
nivel_autoridade	char(20)	Descrição dos níveis de autoridades.

- tab\_perfil

Esta entidade armazena os perfis associados aos usuários da ferramenta. As funções do software estão associadas a esses perfis que pode ser Administrador, Gerente de Projeto, Líder de Programa ou Membro de Equipe, por exemplo.

<b>Atributos</b>	<b>Tipo</b>	<b>Descrição</b>
cd_perfil	int	Código identificador do perfil.
perfil	char(20)	Descrição dos nomes dos perfis existentes.

- tab\_usuario

Esta entidade armazena os dados cadastrais dos usuários da ferramenta além das informações para acesso ao sistema como login, senha e perfil.

<b>Atributos</b>	<b>Tipo</b>	<b>Descrição</b>
login	char(10)	Atributo identificador do usuário. É utilizado no momento de acesso ao sistema juntamente com a senha.
senha	varbinary(255)	Senha criptografada do usuário. É validada juntamente com o login do usuário.
no_usuario	char(50)	Nome completo do usuário.
cd_perfil	int	Foreign Key associada a tabela tab_perfil indicando o perfil do usuário.
cd_empresa	int	Foreign Key associada à tabela tab_empresa indicando a empresa do usuário.
email	char(50)	E-mail do usuário caso ele possua.
telefone	char(10)	Telefone para contato do usuário caso ele possua.
celular	char(10)	Telefone móvel do usuário caso ele possua.

- tab\_probabilidade\_ocorrencia\_risco

Esta entidade armazena as probabilidades de ocorrência dos riscos dos projetos acompanhados pela ferramenta.

<b>Atributos</b>	<b>Tipo</b>	<b>Descrição</b>
cd_probabilidade_ocorrencia	int	Código identificador da probabilidade de ocorrência.
probabilidade_ocorrencia	char(10)	Descrição da probabilidade de ocorrência dos riscos.

- tab\_programa

Esta entidade armazena os programas ao qual deverão estar associados os projetos acompanhados pela ferramenta. Por exemplo: um programa implantação de um sistema de

gestão corporativa poderia ter projeto em diversas áreas, como por exemplo: engenharia, sistemas, faturamento, contas a pagar, contas a receber, marketing etc.

<b>Atributos</b>	<b>Tipo</b>	<b>Descrição</b>
cd_programa	int	Código identificador do programa.
no_programa	char(20)	Descrição do programa ao qual deverão estar associados os projetos.
lider	char(10)	Foreign key associada à tabela tab_usuario indicando o login do Líder de Programa.

- tab\_projeto

Esta entidade armazena as informações relacionadas ao projeto que estará sendo cadastrado na ferramenta para ser acompanhado ou gerenciado pelo FAGP.

<b>Atributos</b>	<b>Tipo</b>	<b>Descrição</b>
cd_projeto	Int	Código identificador do projeto.
no_projeto	char(20)	Nome do projeto cadastrado na ferramenta.
cd_diretoria	int	Foreign Key associada à tabela tab_diretoria indicando a diretoria responsável pelo projeto
cd_programa	int	Foreign key associada à tabela tab_programa indicando o programa ao qual o projeto está associado.
dt_inicio	smalldatetime	Data de Início do projeto.
dt_termino	smalldatetime	Data prevista para o fim do projeto.
dt_registro_ficha	smalldatetime	Data de registro da ficha do projeto.
de_projeto	char(500)	Descrição do projeto.
de_escopo	char(500)	Descrição do escopo do projeto.
de_beneficios	char(500)	Descrição dos benefícios trazidos com a realização do projeto.
vr_orcado	money	Custo estimado para o projeto.
vr_utilizado	money	Ao final do projeto deverá ser inserido o valor utilizado no projeto para que possa ser realizada uma comparação com o valor estimado.

gerente	char(10)	Foreign Key associada a tabela tab_usuario indicando o login do Gerente responsável pelo projeto.
---------	----------	---

- tab\_ponto\_atencao

Esta entidade é a responsável por armazenar os Pontos de Atenção que serão cadastrados na ferramenta.

<b>Atributos</b>	<b>Tipo</b>	<b>Descrição</b>
cd_ponto_atencao	int	Código identificador do ponto de atenção.
cd_projeto	int	Foreign Key associada à tabela tab_projeto indicando o projeto ao qual o ponto de atenção está associado.
de_ponto_atencao	char(500)	Descrição do ponto de atenção.
de_impacto	char(500)	Descrição do impacto previsto para o ponto de atenção caso não seja tomada alguma ação gerencial corretiva.
de_acao	char(500)	Descrição da ação corretiva sugerida pelo usuário que cadastrou o ponto de atenção na ferramenta.
responsavel	char(10)	Campo associado ao login na tabela tab_usuario indicando o usuário responsável indicado para solucionar o ponto de atenção.
solucionador	char(10)	Campo associado ao login na tabela tab_usuario indicando o usuário que efetivamente solucionou o ponto de atenção.
dt_prevista_solucão	smalldatetime	Data prevista para a solução do ponto de atenção.
cd_fase	int	Foreign Key associada a tabela tab_fase indicando a fase em que se encontra o projeto no momento da identificação do ponto de atenção.

dt_registro	smalldatetime	Data em que foi cadastrado o ponto de atenção na ferramenta.
de_resolucao	char(500)	Descrição da resolução dada para o ponto de atenção.
dt_resolucao	smalldatetime	Data em que o ponto de atenção foi fechado ou solucionado.
cd_status	int	Foreign Key associada à tabela tab_status indicando o status do tratamento do ponto de atenção.
cd_criticidade	int	Foreign Key associada à tabela tab_criticidade indicando a criticidade do ponto de atenção.
registradopor	char(10)	Login do usuário que cadastrou o ponto de atenção na ferramenta.

- tab\_licao\_aprendida

Esta entidade é a responsável por armazenar as Lições Aprendidas que serão cadastradas na ferramenta.

<b>Atributos</b>	<b>Tipo</b>	<b>Descrição</b>
cd_licao	int	Código identificador da lição aprendida.
cd_projeto	int	Foreign Key associada à tabela tab_projeto indicando o projeto ao qual a lição aprendida está associada.
dt_registro	smalldatetime	Data em que foi cadastrada a lição aprendida na ferramenta.
de_licao_aprendida	char(500)	Descrição da lição aprendida.
de_fato_gerador	char(500)	Descrição do fato que levou a esta lição aprendida.
cd_fase	int	Foreign Key associada à tabela tab_fase indicando a fase em que se encontra o projeto no momento da identificação da lição aprendida.
de_acao_planejadas	char(500)	Descrição das ações tomadas quando ocorreu o fato que gerador da lição aprendida.
registradopor	char(10)	Login do usuário que cadastrou a lição aprendida na ferramenta.

- tab\_rel\_usuario\_projeto

Entidade de relacionamento associando os usuários aos projetos em que estarão fazendo parte da equipe.

<b>Atributos</b>	<b>Tipo</b>	<b>Descrição</b>
login	char(10)	Campo identificador do usuário.
cd_projeto	int	Campo identificador do projeto.

- tab\_relatorio\_status

Esta entidade é a responsável por armazenar as informações de cadastro dos relatórios de status na ferramenta

<b>Atributos</b>	<b>Tipo</b>	<b>Descrição</b>
cd_projeto	int	Foreign Key associada à tabela tab_projeto indicando o projeto ao qual o relatório de status está associado.
dt_inicio_relatorio	smalldatetime	Data de início do relatório de status.
dt_fim_relatorio	smalldatetime	Data fim do relatório de status.
dt_registro	smalldatetime	Data em que o relatório de status foi registrado na ferramenta.
registradopor	char(10)	Login do usuário que cadastrou o relatório de status na ferramenta.

- tab\_rel\_status\_tipo\_risco

Entidade de relacionamento associando o tipo de resposta que deverá ser dada ao Risco com o status da ação proposta para a solução deste. Por exemplo: se o status da ação está “Em Progresso” não pode ser associado com o tipo de resposta “Aceitar”.

<b>Atributos</b>	<b>Tipo</b>	<b>Descrição</b>
cd_status	int	Campo identificador do status da ação proposta ao Risco.
cd_tipo_resposta	int	Campo identificador do tipo de resposta que deverá ser dado ao Risco.

- tab\_rel\_relatorio\_atividades

Entidade que relaciona o relatório de status com as atividades no período do relatório assim como armazena os dados relativos a essas atividades.

<b>Atributos</b>	<b>Tipo</b>	<b>Descrição</b>
cd_projeto	int	Foreign Key associada à tabela tab_projeto indicando o projeto ao qual o relatório de status está associado.
dt inicio relatorio	smalldatetime	Data de início do relatório de status.
dt fim relatorio	smalldatetime	Data fim do relatório de status.
cd_atividade	int	Foreign Key associada à tabela tab_atividades indicando o tipo de atividade associada a esse registro.
ic_atividade	int	Devido a podermos ter mais de uma atividade de um mesmo tipo associada ao mesmo relatório de status, esse campo funciona como um identificador dessas diferentes atividades de um mesmo tipo.
dt inicio previsto	smalldatetime	Data de início prevista para aquela atividade.
dt termino previsto	smalldatetime	Data de término prevista para aquela atividade.
pc previsto dt atual	int	Percentual previsto até a data atual.
pc realizado dt atual	int	Percentual realizado até a data atual.

pc_variacao_real_previsto	int	Percentual de variação entre o previsto e o real.
de_motivo	char(500)	Descrição do motivo de ocorrência do desvio do real em relação ao previsto, caso este seja negativo.
de_acao_prazo	char(500)	Descrição da ação corretiva e estabelecimento de um novo prazo.
flg_em_dia	char(1)	Indica, para as atividades em andamento, se elas estão em dia ou não.
dt_inicio_real	smalldatetime	Data em que se iniciou efetivamente a execução daquela atividade.

- tab\_criticidade\_risco

Esta entidade armazena as criticidades dos Riscos cadastrados no software FAGP associando estas a probabilidade de ocorrência do risco ao impacto que este causará no projeto caso não seja tomada a ação para solucioná-lo.

<b>Atributos</b>	<b>Tipo</b>	<b>Descrição</b>
cd_probabilidade_ocorrencia	int	Campo identificador da probabilidade de ocorrência do Risco.
cd_impacto	int	Campo identificador do impacto causado pelo Risco.
criticidade	char(20)	Descrição da criticidade do Risco.
cd_criticidade	int	Código identificador da combinação criticidade, cd probabilidade ocorrência, cd impacto.
sg_criticidade	char(2)	Sigla descritiva para a criticidade.

- tab\_risco

Esta entidade é a responsável por armazenar os Riscos que serão cadastrados na ferramenta.

<b>Atributos</b>	<b>Tipo</b>	<b>Descrição</b>
cd_risco	int	Código identificador do risco.
cd_projeto	int	Foreign Key associada à tabela tab_projeto indicando o projeto ao qual o risco está associado.
de_acao_proposta	char(500)	Descrição da ação corretiva proposta para solucionar o risco.
cd_probabilidade_ocorrencia	int	Foreign Key associada à tabela tab_probabilidade_ocorrencia_risco indicando a probabilidade de ocorrência do risco cadastrado.
de_risco	char(500)	Descrição do Risco.

responsavel	char(10)	Campo associado ao login na tabela tab_usuario indicando o usuário responsável indicado para solucionar o risco.
cd_nivel_autoridade	int	Foreign Key associada à tabela tab_nivel_autoridade indicando o nível de autoridade necessária para a resolução do risco.
dt_registro	smalldatetime	Data em que foi cadastrado o risco na ferramenta.
dt_prevista_conclusao_acao	smalldatetime	Data prevista para a solução do risco.
cd_impacto	int	Foreign Key associada à tabela tab_impacto_risco indicando o impacto causado pelo risco caso não seja tomada nenhuma ação corretiva.
cd_tipo_resposta	int	Foreign Key associada à tabela tab_tipo_resposta_risco indicando o tipo de resposta que deverá ser dados ao risco cadastrado.
solucionador	char(10)	Campo associado ao login na tabela tab_usuario indicando o usuário que efetivamente solucionou o risco.
cd_status	Int	Foreign Key associada à tabela tab_status indicando o status do tratamento do risco.
dt_resolucao	smalldatetime	Data em que o risco foi fechado ou solucionado.
de_resolucao	char(500)	Descrição da resolução dada para o risco.
cd_fase	int	Foreign Key associada a tabela tab_fase indicando a fase em que se encontra o projeto no momento da identificação do risco.
registradopor	char(10)	Login do usuário que cadastrou o risco na ferramenta.

- tab\_status\_risco

Entidade responsável por armazenar o andamento das ações tomadas em relação à solução do Risco. Funciona como uma tabela de histórico do Risco.

<b>Atributos</b>	<b>Tipo</b>	<b>Descrição</b>
cd_risco	int	Campo identificador do risco ao qual estará associado este histórico.
de_status_risco	int	Descrição de como está o Risco atualmente.
de_acao	char(500)	Descrição da última ação proposta para solucionar o Risco.
dt_registro	smalldatetime	Data de registro da linha de histórico.
registradopor	char(10)	Login associado ao usuário que registrou essa linha de histórico.

- tab\_sistema\_projeto

Entidade responsável por armazenar informações relacionadas aos sistemas que serão utilizados durante a execução do projeto.

<b>Atributos</b>	<b>Tipo</b>	<b>Descrição</b>
cd_sistema	int	Campo identificador do sistema que será utilizado no projeto.
cd_projeto	int	Campo identificador do projeto ao qual o sistema estará associado.
de_funcionalidade	char(500)	Descrição das funcionalidades do sistema.
sistema_operacional	char(500)	Descrição do sistema operacional utilizado.
banco_dados	char(500)	Descrição do sistema gerenciador de Banco de Dados utilizado.
linguagem_programacao	char(500)	Descrição das linguagens de programação utilizadas.
hardware	char(500)	Descrição dos hardwares utilizados.

- tab\_solicitacao\_mudanca

Esta entidade é a responsável por armazenar as Solicitações de Mudança que serão cadastradas na ferramenta.

<b>Atributos</b>	<b>Tipo</b>	<b>Descrição</b>
------------------	-------------	------------------

cd_solicitacao	int	Código identificador da solicitação de mudança.
cd_projeto	int	Foreign Key associada à tabela tab_projeto indicando o projeto ao qual a solicitação de mudança está associada.
de_solicitacao	smalldatetime	Descrição da solicitação da mudança.
de_risco	char(500)	Descrição da lição aprendida.
de_impacto	char(500)	Descrição do fato que levou a esta lição aprendida.
dt_limite_aprovacao	int	Data limite para a aprovação da solicitação de mudança
dt_implantacao	char(500)	Data prevista para implantação da mudança solicitada.
cd_tipo_solicitacao	char(10)	Foreign Key associada à tabela tab_tipo_solicitacao_mudanca indicando o tipo de mudança que está sendo solicitada. Podendo ser de escopo, cronograma ou outros.
de_resolucao	char(500)	Descrição da resolução dada para a solicitação de mudança.
cd_status	int	Foreign Key associada à tabela tab_status indicando o status da implantação da solicitação de mudança.
dt_registro	smalldatetime	Data em que foi cadastrada a solicitação de mudança na ferramenta.
requisitadopor	char(10)	Login do usuário que cadastrou a solicitação de mudança na ferramenta.

- tab\_status\_solicitacao

Esta entidade armazena o status das solicitações de mudança. Os possíveis status de uma solicitação de mudança são os seguintes: ‘Aberta’, ‘Em Análise’ e ‘Fechada’.

Atributos	Tipo	Descrição
cd_status	int	Código identificador do status da solicitação de mudança.
status	char(25)	Descrição das opções de status de uma solicitação.

- tab\_projeto\_arquivo\_anexo

Esta entidade armazena os dados relacionados aos arquivos que são anexados no módulo ‘Ficha de Projetos’ do software. Os dados armazenados nessa entidade são o nome do arquivo anexado, código do módulo e o projeto ao qual esse módulo se refere.

<b>Atributos</b>	<b>Tipo</b>	<b>Descrição</b>
no_arquivo_anexo	char(50)	Nome do arquivo anexado.
cd_modulo	int	Foreign Key associada à tabela tab_modulo indicando a qual módulo esse arquivo se refere.
cd_projeto	int	Foreign Key associada à tabela tab_projeto indicando o projeto ao qual a ação de anexar está relacionada.

- tab\_licao\_arquivo\_anexo

Esta entidade armazena os dados relacionados aos arquivos que são anexados no módulo ‘Licao Aprendidas’ do software. Os dados armazenados nessa entidade são o nome do arquivo anexado, código do módulo e o projeto ao qual esse módulo se refere.

<b>Atributos</b>	<b>Tipo</b>	<b>Descrição</b>
no_arquivo_anexo	char(50)	Nome do arquivo anexado.
cd_modulo	int	Foreign Key associada à tabela tab_modulo indicando a qual módulo esse arquivo se refere.
cd_projeto	int	Foreign Key associada à tabela tab_projeto indicando o projeto ao qual a ação de anexar está relacionada.

- tab\_modulo

Esta entidade associa o código dos módulos a uma string com o nome do módulo.

<b>Atributos</b>	<b>Tipo</b>	<b>Descrição</b>
cd_modulo	int	Código identificador do modulo.
no_modulo	char(20)	Campo contendo o nome do módulo.

## **4.4 Descrição das Dependências**

### **4.4.1 Dependência entre módulos**

O software FAGP desenvolvido nesse projeto é composto de apenas um módulo (a parte cliente não foi desenvolvida, consistindo apenas de um browser). As dependências entre as funções (sub-módulos) que compõem o módulo foram descritas previamente na seção .

### **4.4.2 Dependência entre processos**

Como já mencionado no item , o software desenvolvido é composto de apenas um módulo, o aplicativo servidor e as ocorrências de acessos simultâneos por vários usuários deverão ser gerenciadas pelo Sistema Operacional e o Sistema Gerenciador de Banco de Dados.

### **4.4.3 Dependência entre dados**

Os relacionamentos entre os dados são descritos pelo Diagrama de Entidades-Relacionamento (DER) apresentado na especificação de requisitos de software e reproduzido a seguir por conveniência:

**Figura 12** - Modelagem de dados do sistema (DER)

## **4.5 Descrição das Interfaces**

### **4.5.1 Interfaces dos Módulos**

O programa será utilizado pela Web. Sendo assim, as interfaces com o usuário serão feitas através de formulários HTML e visualizadas pelo usuário em um browser qualquer. As funções serão definidas dentro do código HTML com alguma linguagem que pode alterar dinamicamente o conteúdo de uma página, como por exemplo, Java Script.

Todas as entradas de dados serão feitas via formulários. O usuário ora entrará com os dados digitando-os manualmente, ora escolhendo-os entre opções existentes carregadas do Banco de Dados no momento do acesso à página. O envio das informações será ativado pelo usuário pressionando-se um botão.

Quando for realizada uma consulta, os dados retornados serão mostrados na tela na forma de links. Clicando-se em um link, um formulário será aberto para que se possa fazer a edição dos dados ou simplesmente visualizar detalhes que não são mostrados no formulário de retorno da consulta.

O menu principal conterà as opções disponíveis na forma de links para os formulários das funções correspondentes. Este menu sempre estará sendo mostrado no canto esquerdo da tela.

No que diz respeito às interfaces internas, cada função sempre estará fazendo consultas ou cadastrando dados em um banco de dados. Então, na máquina servidora deverá estar instalado um servidor de Banco de Dados. As funções de Lição Aprendida e Relatório de Status permitirão ao usuário anexar um arquivo a essas informações. Esse arquivo deverá ficar armazenado em um diretório específico na máquina servidora.

Este software não possui interfaces externas nem com outros softwares nem interface de hardware.

#### **4.5.2 Interfaces entre Processos**

Sendo o software composto por apenas um processo, não se aplica qualquer descrição neste item.

### **4.6 Projeto Detalhado**

#### **4.6.1 Projeto Detalhado dos Módulos**

Nesta seção será descrito a lógica dos módulos bem como suas interações com as entidades de dados:

- Módulo *Login*

*boolean ler\_e\_testar\_login* (*texto login, texto senha*)

**begin**

- lê do formulário o par login/senha;
- consulta no BD a entidade de dados *tab\_usuario*;
- enquanto (login/senha não é válido) faça
- begin
  - envia mensagem de erro;
  - pede nova entrada de dados;
- end
- return true;

**end**

- Módulo Fichas de Projeto

*texto* **cria\_ficha\_de\_projeto**

**begin**

- lê do formulário os dados\_nova\_ficha\_de\_projeto;
- insere no BD os dados\_nova\_ficha\_de\_projeto nas entidades de dados *tab\_projeto* e *tab\_sistema\_projeto*;
- retorna resultado da operação para o usuário;

**end**

*texto* **consulta\_ficha\_de\_projeto**

**begin**

- seleciona o projeto ao qual deseja ser visualizada a ficha;
- consulta no BD as entidades de dados *tab\_projeto* e *tab\_sistema\_projeto*;
- retorna a ficha do projeto selecionado com os dados preenchidos;

- permite a edição das fichas de projeto consultadas;

**end**

*texto* **edita\_ficha\_de\_projetos** (*texto* ficha\_de\_projetos\_consultada)

**begin**

- lê do formulário os resultados da consulta\_fichas\_de\_projeto;
- lê do formulário os parâmetros\_fichas\_de\_projeto\_alteradas;
- se for clicado no botão de anexar arquivos chama método de upload;
- atualiza no BD os parâmetros\_fichas\_de\_projeto\_alteradas nas entidades de dados *tab\_projeto* e *tab\_sistema\_projeto*;
- retorna resultado da operação para o usuário;

**end**

- Módulo Pontos de Atenção (*PA*)

*texto* **novo\_ponto\_de\_atenção** (*texto* dados\_novo\_PA)

**begin**

- lê do formulário os dados\_novo\_PA;
- insere no BD os dados\_nova\_PA na entidade de dados *tab\_ponto\_atencao*;
- retorna resultado da operação para o usuário;

**end**

*texto* **consulta\_ponto\_de\_atenção** (*texto* parâmetros\_consulta\_PA)

**begin**

- lê do formulário os parâmetros\_consulta\_PA;

- consulta no BD a entidade de dados *tab\_ponto\_atencao*;
- retorna os PA contendo os parâmetros *\_consulta\_PA*;
- permite a edição dos PA consultados;

**end**

*texto* **edita\_ponto\_atencao** (*texto* PAs consultados)

**begin**

- lê do formulário os resultados da consulta *\_PA*;
- lê do formulário os parâmetros *\_PA\_alterados*;
- atualiza no BD os parâmetros *\_PA\_alterados* na entidade de dados *tab\_ponto\_atencao*;
- retorna resultado da operação para o usuário;

**end**

*texto* **deleta\_ponto\_atencao** (*texto* PAs consultadas)

**begin**

- lê do formulário os resultados da consulta *\_PA*;
- lê do formulário os parâmetros *\_PA\_selecionados*;
- apaga no BD os parâmetros *\_PA\_selecionados* na entidade de dados *tab\_ponto\_atencao*;
- retorna confirmação da operação para o usuário;

**end**

- Módulo *Riscos*

*texto novo\_risco* (*texto dados\_novo\_risco*)

**begin**

- lê do formulário os dados\_novo\_risco;
- insere no BD os dados\_novo\_risco na entidade de dados *tab\_risco*;
- retorna resultado da operação para o usuário;

**end**

*texto consulta\_risco* (*texto parâmetros\_consulta\_risco*)

**begin**

- lê do formulário os parâmetros\_consulta\_risco;
- consulta no BD a entidade de dados *tab\_risco*;
- retorna os riscos contendo os parâmetros\_consulta\_risco;
- permite a edição dos riscos consultados;

**end**

*texto edita\_risco* (*texto riscos consultados*)

**begin**

- lê do formulário os resultados da consulta\_risco;
- lê do formulário os parâmetros\_riscos\_alterados;
- atualiza no BD os parâmetros\_riscos\_alterados na entidade de dados *tab\_risco*;
- retorna resultado da operação para o usuário;

**end**

*texto deleta\_riscos* (*texto riscos consultados*)

**begin**

- lê do formulário os resultados da consulta\_risco;
- lê do formulário os parâmetros\_riscos\_selecionados;
- apaga no BD os parâmetros\_riscos\_selecionados na entidade de dados *tab\_risco*;
- retorna confirmação da operação para o usuário;

**end**

- Módulo *Solicitação de Mudanças (SM)*

*texto nova\_solicitação\_de\_mudanças (texto dados\_nova\_SM)*

**begin**

- lê do formulário os dados\_nova\_SM;
- insere no BD os dados\_nova\_SM na entidade de dados *tab\_solicitacao\_mudanca*;
- retorna resultado da operação para o usuário;

**end**

*texto consulta\_solicitação\_de\_mudanças (texto parâmetros\_consulta\_SM)*

**begin**

- lê do formulário os parâmetros\_consulta\_SM;
- consulta no BD a entidade de dados *tab\_solicitacao\_mudanca*;
- retorna as SM contendo os parâmetros\_consulta\_SM;
- permite a edição das SM consultadas;

**end**

*texto edita\_solicitação\_de\_mudanças (texto SMs consultadas)*

**begin**

- lê do formulário os resultados da consulta\_SM;
- lê do formulário os parâmetros\_SM\_alterados;
- atualiza no BD os parâmetros\_SM\_alterados na entidade de dados *tab\_solicitacao\_mudanca*;
- retorna resultado da operação para o usuário;

**end**

*texto* **deleta\_solicitacao\_de\_mudancas** (*texto* SMs consultadas)

**begin**

- lê do formulário os resultados da consulta\_SM;
- lê do formulário os parâmetros\_SM\_selecionadas;
- apaga no BD os parâmetros\_riscos\_selecionados na entidade de dados dados *tab\_solicitacao\_mudanca*;
- retorna confirmação da operação para o usuário;

**end**

- Módulo *Relatório de Status (RS)*

*texto* **novo\_relatorio\_de\_status** (*texto* dados\_novo\_RS)

**begin**

- lê do formulário os dados\_novo\_RS;
- insere no BD os dados\_novo\_RS nas entidades de dados *tab\_relatorio\_status* e *tab\_rel\_relatorio\_atividade*;
- retorna resultado da operação para o usuário;

**end**

*texto* **consulta\_relatório\_de\_status** (*texto* parâmetros\_consulta\_RS)

**begin**

- lê do formulário os parâmetros\_consulta\_RS;
- consulta no BD as entidades de dados *tab\_relatorio\_status* e *tab\_rel\_relatorio\_atividade*;
- retorna os RSs contendo os parâmetros\_consulta\_RS;
- permite a edição dos RSs consultados;

**end**

*texto* **edita\_relatório\_de\_status** (*texto* relatórios\_de\_status\_consultados)

**begin**

- lê do formulário os resultados da consulta\_relatórios\_de\_status;
- lê do formulário os parâmetros\_RS\_alterados;
- caso tenha sido selecionada a opção salvar relatório atualiza no BD os parâmetros\_RS\_alterados nas entidades de dados *tab\_relatorio\_status* e *tab\_rel\_relatorio\_atividade*, senão, se tiver sido selecionada a opção salvar como novo relatório insere os dados os dados no BD como um novo relatório;
- retorna resultado da operação para o usuário;

**end**

- Módulo *Solicitação de Mudanças (SM)*

*texto* **nova\_lição\_aprendida** (*texto* dados\_nova\_LA)

**begin**

- lê do formulário os dados\_nova\_LA;
- insere no BD os dados\_nova\_LA na entidade de dados *tab\_licao\_aprendida*;
- retorna resultado da operação para o usuário;

**end**

*texto* **consulta\_licões\_aprendidas** (*texto* parâmetros\_consulta\_LA)

**begin**

- lê do formulário os parâmetros\_consulta\_LA;
- consulta no BD a entidade de dados *tab\_licao\_aprendida*;
- retorna as LA contendo os parâmetros\_consulta\_LA;
- permite a edição das LA consultadas;

**end**

*texto* **edita\_licões\_aprendidas** (*texto* LAs consultadas)

**begin**

- lê do formulário os resultados da consulta\_LA;
- lê do formulário os parâmetros\_LA\_alteradas;
- se for clicado no botão de anexar arquivos chama método de upload;
- atualiza no BD os parâmetros\_LA\_alteradas na entidade de dados *tab\_licao\_aprendida*;
- retorna resultado da operação para o usuário;

**end**

*texto* **deleta\_licões\_aprendidas** (*texto* LAs consultadas)

**begin**

- lê do formulário os resultados da consulta\_LA;
- lê do formulário os parâmetros\_LA\_selecionadas;
- apaga no BD os parâmetros\_LA\_selecionados na entidade de dados dados *tab\_licao\_aprendida*;
- retorna confirmação da operação para o usuário;

**end**

- Módulo Administração

*texto* **novo\_usuario** (*texto* dados\_novo\_usuario)

**begin**

- lê do formulário os dados\_novo\_usuario;
- insere no BD os dados\_novo\_usuario na entidade de dados *tab\_usuario*;
- retorna resultado da operação para o usuário;

**end**

*texto* **consulta\_usuario** (*texto* parâmetros\_consulta\_usuario)

**begin**

- lê do formulário os parâmetros\_consulta\_usuario;
- consulta no BD a entidade de dados *tab\_usuario*;
- retorna os usuários contendo os parâmetros\_consulta\_usuario;

**end**

*texto* **novo\_projeto** (*texto* dados\_novo\_projeto)

**begin**

- lê do formulário os dados\_novo\_projeto;
- insere no BD os dados\_novo\_projeto na entidade de dados *tab\_projeto*;
- retorna resultado da operação para o usuário;

**end**

*texto* **consulta\_projeto** (*texto* parâmetros\_consulta\_projeto)

**begin**

- lê do formulário os parâmetros\_consulta\_projeto;
- consulta no BD a entidade de dados *tab\_projeto*;
- retorna os projetos contendo os parâmetros\_consulta\_projeto;

**end**

#### 4.6.2 Projeto Detalhado das Entidades de Dados

O detalhamento das entidades de dados é dado pelo modelo físico das tabelas no Microsoft SQL Server 7.0.

- **tab\_atividades**

```
tab_atividades (
    cd_atividade    int IDENTITY,
    de_atividade    char(40) NULL
)
```

- **tab\_criticidade**

```
tab_criticidade (
    cd_criticidade  int IDENTITY,
```

```
    criticidade      char(20) NULL
)
```

- **tab\_criticidade\_risco**

```
tab_criticidade_risco (
    cd_probabilidade_ocorrenci int NOT NULL,
    cd_impacto                int NOT NULL,
    criticidade               char(20) NOT NULL,
    cd_criticidade            int IDENTITY,
    sg_criticidade            char(2) NOT NULL
)
```

- **tab\_diretoria**

```
tab_diretoria (
    cd_diretoria            int IDENTITY,
    no_diretoria            char(30) NULL
)
```

- **tab\_empresa**

```
tab_empresa (
    cd_empresa              int IDENTITY,
    no_empresa              char(30) NULL
)
```

- **tab\_fase**

```
tab_fase (  
    cd_fase      int IDENTITY,  
    fase        char(20) NULL  
)
```

- **tab\_impacto\_risco**

```
tab_impacto_risco (  
    cd_impacto   int IDENTITY,  
    impacto     char(10) NULL  
)
```

- **tab\_licao\_aprendida**

```
tab_licao_aprendida (  
    cd_licao     int IDENTITY,  
    cd_projeto   int NOT NULL,  
    dt_registro  smalldatetime NULL DEFAULT CURRENT_TIMESTAMP,  
    de_licao_aprendida char(500) NULL,  
    de_fato_gerador char(500) NULL,  
    cd_fase      int NOT NULL,  
    de_acao_planejada char(500) NULL,  
    registradopor char(10) NOT NULL  
)
```

```
tab_licao_arquivo_anexo
```

```
tab_licao_arquivo_anexo (  
    no_arquivo_anexo  char(50) NOT NULL,  
    cd_modulo         int NOT NULL,  
    cd_licao          int NOT NULL  
)
```

- **tab\_modulo**

```
tab_modulo (  
    cd_modulo         int IDENTITY,  
    no_modulo        char(20) NULL  
)
```

- **tab\_nivel\_autoridade**

```
tab_nivel_autoridade (  
    cd_nivel_autoridade int IDENTITY,  
    nivel_autoridade  char(20) NULL  
)
```

- **tab\_perfil**

```
tab_perfil (  
    cd_perfil         int IDENTITY,  
    perfil           char(20) NULL  
)
```

- **tab\_ponto\_atencao**

```
tab_ponto_atencao (  
    cd_ponto_atencao    int IDENTITY,  
    cd_projeto          int NOT NULL,  
    de_ponto_atencao   char(500) NULL,  
    de_impacto         char(500) NULL,  
    de_acao            char(500) NULL,  
    responsavel        char(10) NULL,  
    solucionador       char(10) NULL,  
    dt_prevista_solucao smalldatetime NULL,  
    cd_fase            int NOT NULL,  
    dt_registro        smalldatetime NULL DEFAULT CURRENT_TIMESTAMP,  
    de_resolucao       char(500) NULL,  
    dt_resolucao       smalldatetime NULL,  
    cd_status          int NOT NULL,  
    cd_criticidade     int NOT NULL,  
    registradopor     char(10) NOT NULL  
)
```

- **tab\_probabilidade\_ocorrencia\_risco**

```
tab_probabilidade_ocorrencia_risco (  
    cd_probabilidade_ocorrencia int IDENTITY,  
    probabilidade_ocorrencia char(10) NULL  
)
```

- **tab\_programa**

```
tab_programa (  
    cd_programa      int IDENTITY,  
    no_programa      char(20) NULL,  
    lider            char(10) NULL  
)
```

- **tab\_projeto**

```
tab_projeto (  
    cd_projeto       int IDENTITY,  
    no_projeto       char(50) NULL,  
    cd_diretoria     int NOT NULL,  
    cd_programa      int NOT NULL,  
    dt_inicio        smalldatetime NULL,  
    dt_termino       smalldatetime NULL,  
    dt_registro_ficha smalldatetime NULL,  
    de_projeto       char(500) NULL,  
    de_escopo        char(500) NULL,  
    de_beneficios    char(500) NULL,  
    vr_orcado        money NULL,  
    vr_utilizado     money NULL,  
    vr_beneficio_projeto money NULL,  
    gerente          char(10) NOT NULL  
)
```

```
tab_projeto_arquivo_anexo
tab_projeto_arquivo_anexo (
    no_arquivo_anexo char(50) NOT NULL,
    cd_modulo        int NOT NULL,
    cd_projeto       int NOT NULL
)
```

- **tab\_rel\_relatorio\_atividade**

```
tab_rel_relatorio_atividade (
    cd_projeto        int NOT NULL,
    dt_inicio_relatorio smalldatetime NOT NULL,
    dt_fim_relatorio  smalldatetime NOT NULL,
    cd_atividade      int NOT NULL,
    ic_atividade      int NOT NULL,
    dt_inicio_previsto smalldatetime NULL,
    dt_termino_previsto smalldatetime NULL,
    pc_previsto_dt_atual int NULL,
    pc_realizado_dt_atual int NULL,
    pc_variacao_real_previsto int NULL,
    de_motivo         char(500) NULL,
    de_acao_prazo     char(500) NULL,
    flg_em_dia        char(1) NULL,
    dt_inicio_real     smalldatetime NULL
)
```

- **tab\_rel\_status\_tipo\_risco**

```
tab_rel_status_tipo_risco (  
    cd_status      int NOT NULL,  
    cd_tipo_resposta int NOT NULL  
)
```

- **tab\_rel\_usuario\_projeto**

```
tab_rel_usuario_projeto (  
    login          char(10) NOT NULL,  
    cd_projeto     int NOT NULL  
)
```

- **tab\_relatorio\_status**

```
tab_relatorio_status (  
    cd_projeto     int NOT NULL,  
    dt_inicio_relatorio smalldatetime NOT NULL,  
    dt_fim_relatorio  smalldatetime NOT NULL,  
    dt_registro     smalldatetime NULL DEFAULT CURRENT_TIMESTAMP,  
    registradopor   char(10) NOT NULL  
)
```

- **tab\_risco**

```
tab_risco (  

```

```

cd_risco          int IDENTITY,
cd_projeto        int NOT NULL,
de_acao_proposta char(500) NULL,
cd_probabilidade_ocorrencia int NOT NULL,
de_risco          char(500) NULL,
responsavel       char(10) NULL,
cd_nivel_autoridade int NOT NULL,
dt_registro       smalldatetime NULL DEFAULT CURRENT_TIMESTAMP,
dt_prevista_conclusao_acao smalldatetime NULL,
cd_impacto        int NOT NULL,
cd_tipo_resposta  int NOT NULL,
solucionador      char(10) NULL,
cd_status         int NOT NULL,
dt_resolucao      smalldatetime NULL,
de_resolucao      char(500) NULL,
cd_fase           int NOT NULL,
registradopor     char(10) NOT NULL
)

```

- **tab\_sistema\_projeto**

```

tab_sistema_projeto (
  cd_sistema      int IDENTITY,
  cd_projeto       int NOT NULL,
  de_funcionalidade char(500) NULL,
  sistema_operacional char(500) NULL,

```

```
banco_dados      char(500) NULL,  
linguagem_programacao char(500) NULL,  
middleware       char(500) NULL,  
hardware        char(500) NULL  
)
```

- **tab\_solicitacao\_mudanca**

```
tab_solicitacao_mudanca (  
  cd_solicitacao    int IDENTITY,  
  cd_projeto       int NOT NULL,  
  de_solicitacao   char(500) NULL,  
  de_risco        char(500) NULL,  
  de_impacto      char(500) NULL,  
  dt_limite_aprovacao smalldatetime NULL,  
  dt_implantacao  smalldatetime NULL,  
  cd_tipo_solicitacao int NOT NULL,  
  de_resolucao    char(500) NULL,  
  dt_registro     smalldatetime NULL DEFAULT CURRENT_TIMESTAMP,  
  cd_status       int NOT NULL,  
  requisitadopor  char(10) NOT NULL  
)
```

- **tab\_status**

```
tab_status (  
  cd_status      int IDENTITY,
```

```
status      char(25) NULL
)
```

- **tab\_status\_risco**

```
tab_status_risco (
  cd_risco      int NOT NULL,
  de_status_risco char(500) NOT NULL,
  de_acao       char(500) NOT NULL,
  dt_registro   smalldatetime NULL DEFAULT CURRENT_TIMESTAMP,
  registradopor char(10) NOT NULL
)
```

- **tab\_status\_solicitacao**

```
tab_status_solicitacao (
  cd_status      int IDENTITY,
  status_solicitacao char(25) NULL
)
```

- **tab\_tipo\_resposta\_risco**

```
tab_tipo_resposta_risco (
  cd_tipo_resposta int IDENTITY,
  de_tipo_resposta char(20) NULL
)
```

- **tab\_tipo\_solicitacao\_mudanca**

```
tab_tipo_solicitacao_mudanca (  
    cd_tipo_solicitacao int IDENTITY,  
    de_solicitacao     char(20) NULL  
)
```

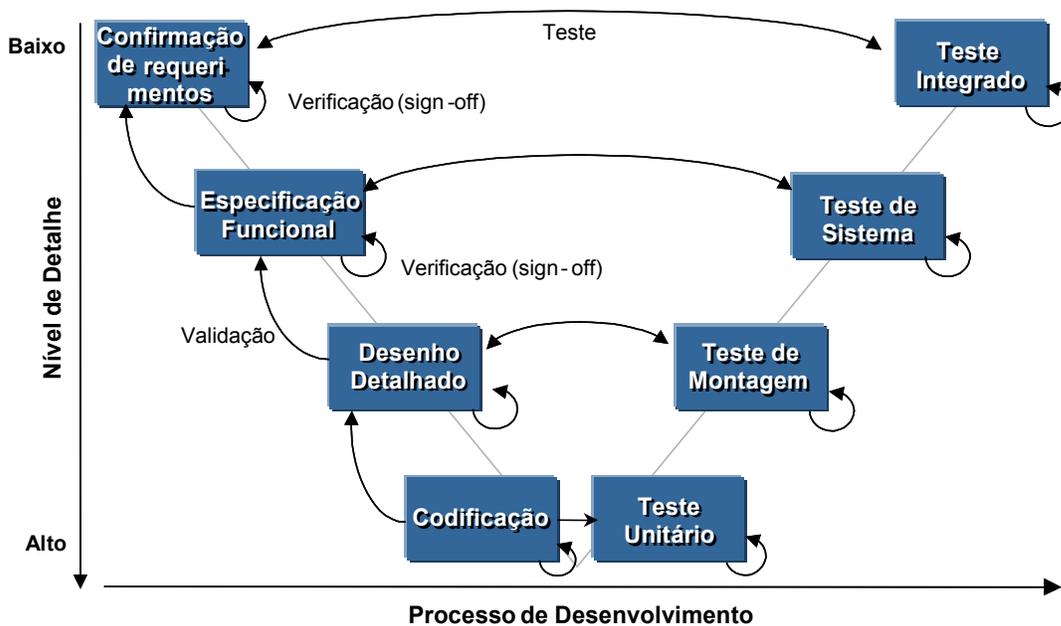
- **tab\_usuario**

```
tab_usuario (  
    login          char(10) NOT NULL,  
    senha          varbinary(255) NULL,  
    no_usuario     char(50) NULL,  
    cd_perfil      int NOT NULL,  
    cd_empresa     int NOT NULL,  
    email          char(50) NULL,  
    telefone       char(10) NULL,  
    celular        char(10) NULL  
)
```

## 5. Procedimento de Testes

A estratégia adotada neste projeto foi a realização de um volume significativo de testes em todos os níveis, objetivando aumentar a confiabilidade das soluções implementadas.

A metodologia que será apresentada é amplamente utilizada em projetos de consultoria em diversos ramos de mercado, com eficiência comprovada. Esta metodologia é denominada modelo V (V-Model):



Os testes a serem executados levam em consideração uma abordagem focada na identificação prematura e correção de erros na etapa ou estágio em que eles ocorrem. Esta abordagem aumenta a eficiência e a eficácia da implementação de pacotes de Software devido ao fato de os erros serem corrigidos num momento em que haja maior facilidade de correção trazendo assim menores custos.

- ↻ **Verificação:** Garante que um produto foi corretamente construído a partir das entradas do estágio correspondente e que é consistente internamente. Além disso, garante que as saídas e processos estejam de acordo com os padrões do projeto.



**Teste:** Durante cada estágio de desenho ou análise do lado esquerdo do modelo, será desenvolvido um plano de testes para documentar as condições de teste e resultados esperados para testar a implementação da especificação na atividade correspondente de teste no lado direito do modelo.



**Validação:** Garante que os produtos satisfaçam os requerimentos especificados na etapa anterior e que os requerimentos de negócios continuam sendo atendidos. Em outras palavras, assegura que os produtos do trabalho estejam dentro do escopo.

O **modelo “V”** de teste especifica que uma etapa deve estar validada antes de se passar para próxima. Antes de prosseguir para próxima, é importante conhecer o critério de saída da etapa atual. O teste para cada etapa deve ter sido executado com sucesso, desta forma assegurando o cumprimento dos objetivos do teste (foco primário) antes de passar para a próxima fase.

## Tipos de Testes

### Teste de Componente

- Um teste de componente envolve o teste de uma parte específica da solução.
- O objetivo do teste de componente é assegurar que o componente está de acordo com a especificação técnica.
- Este teste é considerado parte do esforço de construção e não parte do teste de integração, só envolve componentes novos ou modificados.
- É normalmente executado por programadores ou configuradores

### Teste de Montagem

- O teste de montagem (assembly) mede a integração técnica entre componentes fortemente relacionados.

- Este teste é considerado parte do esforço de construção e não parte do teste de Integração.
- É muito comum executar testes de componente e montagem ao mesmo tempo. Isto depende da complexidade dos módulos novos ou modificados e a relação de custo benefício entre simular a interação entre o módulo em questão e os módulos com que interage versus realmente testar a interação.

### **Teste de Produto**

- Teste de produto se concentra nos requisitos e nos processos de negócio de um único aplicativo.
- O teste de produto assegura que todos os requisitos definidos para um produto sejam atendidos.
- O teste de Produto de cada pacote que compõe uma solução deve ser completado antes de se passar para o próximo estágio.
- O teste de Produto deve ser executado por cada equipe de implantação de aplicativos.

### **Teste de Integração**

- O foco do teste de Integração é testar a integração de todos os componentes da nova arquitetura (pacotes e interfaces) entre si e com outros sistemas legados da empresa ou instituição em que se está instalando o software. Apesar de fazer parte da metodologia modelo V, este teste não se fez necessário devido a ferramenta não interagir com sistemas legados da empresa.

### **Teste Piloto**

- O foco do teste Piloto é testar os processos de negócio de ponta-a-ponta para confirmar que a solução atende os requisitos de negócio.

## 6. Autenticação

Por se tratar de um software destinado à gerência de projetos, o sistema deverá possuir uma etapa de autenticação como forma de controlar o acesso dos usuários. Além dessa primeira etapa de autenticação o acesso aos módulos é limitado por perfis de acesso que são verificados logo após serem validadas suas credenciais.

Foi utilizado para a segurança as funções “pwdencrypt” e “pwdcompare”, que são funções internas (e não documentadas) do Microsoft SQL Server 7.0. Pwdencrypt usa um método hash one-way que transforma uma string numa versão criptografada daquela string. Pwdcompare compara uma string não criptografada com sua criptografada representação para checar se elas “batem” uma com a outra.

Abaixo mostramos como foi utilizado essas funções no projeto e logo a seguir apresentamos a página ASP de criação de usuários e de validação de senhas:

*1) Montando a query de inserção dos usuários no Banco de Dados. Utilização da função pwdencrypt:*

```
sqltemp = "INSERT INTO tab_usuario "  
sqltemp = sqltemp & "( login, senha, no_usuario, cd_perfil, cd_empresa) "  
sqltemp = sqltemp & "VALUES ('" & vLogin & "',"  
sqltemp = sqltemp & "CONVERT(VARBINARY(255), PWDENCRYPT("'" & vNovaSenha & "')),"
```

```
sqltemp = sqltemp & "" & vUser & ","  
sqltemp = sqltemp & vCargo & ","  
sqltemp = sqltemp & vEmpresa & ")"
```

Como pode ser observado foi utilizado o comando CONVERT para converter a senha criptografada em um campo VARBINARY na qual deve corresponder ao campo na tabela tab\_usuario.

2) *Trecho de código para validar a senha entrada pelo usuário. Utilização da função pwdcompare:*

```
sql = "select PWDCOMPARE( "" & vSenha & ",senha, 0) as resultado,no_usuario, cd_perfil,login  
from tab_usuario where login = "" & vusuario & """
```

A função pwdcompare é a inversa da função pwdencrypt. Esta função realiza um processo de comparação, o SQL não descriptografa a senha, ele apenas compara, portando o único modo de alguém descobrir a senha é por tentativa e erro. Desta maneira caso um usuário esqueça sua senha ela deverá ser resetada pelo administrador para um valor default. Na query acima o valor retornado em resultado indicará a validade da senha. Quando o valor for igual a 1 a senha é igual, quando for 0 a senha é diferente.

Antes mesmo de realizar a autenticação de senhas o sistema verifica se o usuário existe no Banco de Dados. Caso o usuário exista e a senha seja válida, durante o login será identificado o perfil desse usuário para acesso aos módulos do sistema.

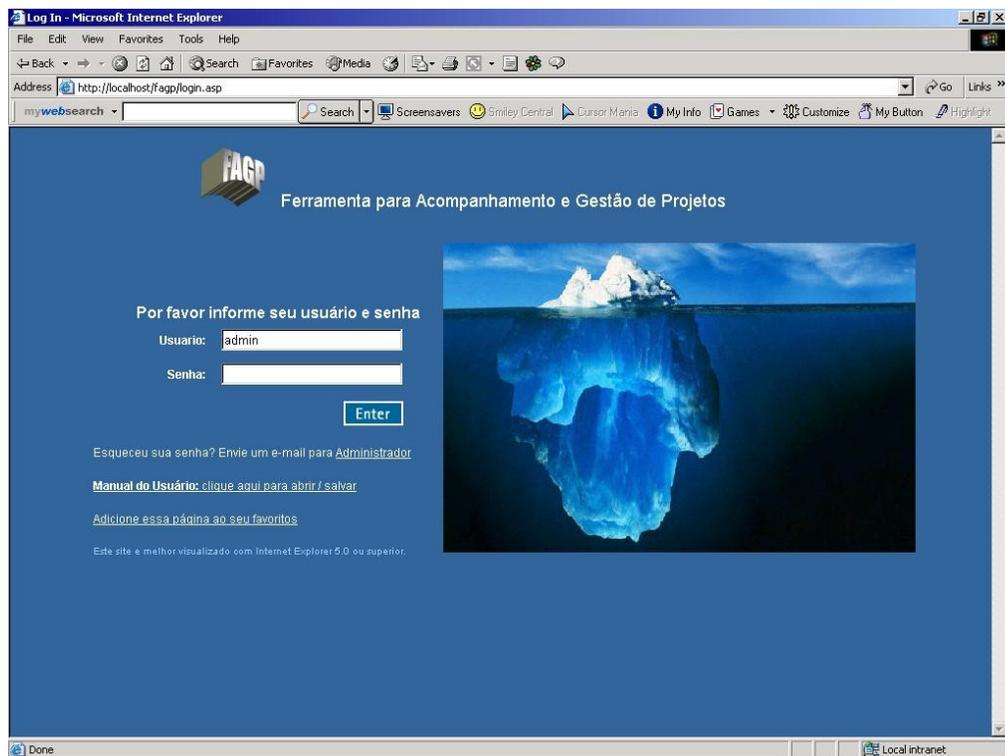


Figura 5 - Página de login

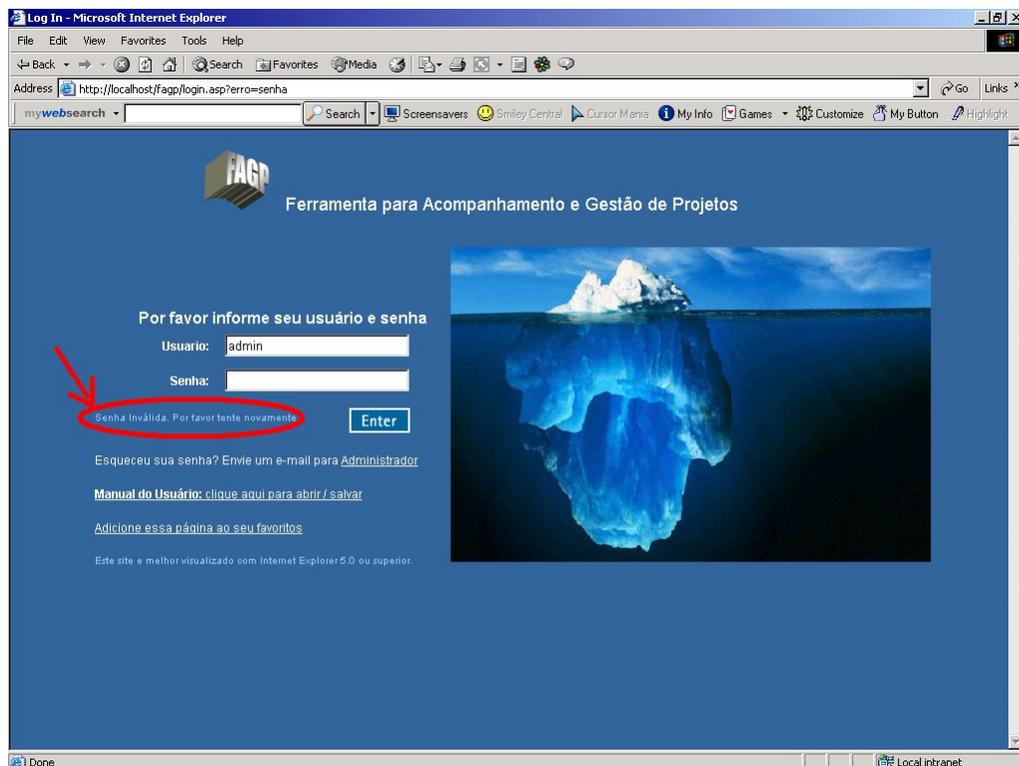


Figura 6 - Página de login apresentando erro de senha do usuário

A seguir são apresentados os códigos completos de criação de usuários e de validação de login:

### **Criação de Usuário**

```
<%
```

```
vUser =          ucase(Request.Form("txtUsername"))
```

```
vLogin =        Request.Form ("txtLogin")
```

```
vNovaSenha =    Request.Form("txtNewPassword")
```

```
vCargo =        Request.Form("txtCargo")
```

```
vEmpresa =      Request.Form("Empresa")
```

```
Set dbconn = Server.CreateObject("ADODB.Connection")
```

```
dbconn.Open = Application("dbconn_ConnectionString")
```

```
Set rsprojeto      = Server.CreateObject("ADODB.Recordset")
```

```
sqlprojeto = "SELECT * FROM tab_projeto"
```

```
rsprojeto.Open sqlprojeto, dbconn,3,3
```

```
sqltemp = "INSERT INTO tab_usuario "
```

```
sqltemp = sqltemp & "( login, senha, no_usuario, cd_perfil, cd_empresa) "
```

```
sqltemp = sqltemp & "VALUES ('" & vLogin & "',"
```

```
sqltemp = sqltemp & "CONVERT(VARBINARY(255), PWDENCRYPT('" & vNovaSenha & "')),"
```

```
sqltemp = sqltemp & "'" & vUser & "',"
```

```
sqltemp = sqltemp & vCargo & ", "
```

```
sqltemp = sqltemp & vEmpresa & ")"
```

```
Set rstemp = dbconn.execute(sqltemp)
```

```
for each projeto in request("idprojeto")
```

```
    while not rsprojeto.eof
```

```

        if( trim(rsprojeto("cd_projeto"))= trim(projeto)) then
            sqlrel_usuario_projeto = "INSERT INTO tab_rel_usuario_projeto (login,
cd_projeto)"
            sqlrel_usuario_projeto = sqlrel_usuario_projeto & " VALUES ('" & vLogin & "',"
& rsprojeto("cd_projeto") & ")")
            Set rsusuario = dbconn.execute(sqlrel_usuario_projeto)
        end if
        rsprojeto.movenext
    wend
    rsprojeto.close
    sqlprojeto = "SELECT * FROM tab_projeto"
    rsprojeto.Open sqlprojeto, dbconn,3,3
next

```

```

response.redirect "CadastraUsuario.asp?acao=Criado"

```

```

Set dbconn = Nothing
%>

```

## **Validação de Login**

```

<%
Session.LCID = 1046

vusuario = Request.Form("txtUsername")
vSenha = Request.Form("txtPassword")

Public vCargo

Public Sub definePerfil(cd_perfil)
    if cd_perfil = 1 then
        vCargo = "Administrador"
    elseif cd_perfil = 2 then

```

```

                vCargo = "Gerente de Projeto"
elseif cd_perfil = 3 then
                vCargo = "Membro de Equipe"
end if
End Sub

Set dbconn = Server.CreateObject("ADODB.Connection")
dbconn.Open = Application("dbconn_ConnectionString")

Set rs          = Server.CreateObject("ADODB.Recordset")
Set rs_valida_senha = Server.CreateObject("ADODB.Recordset")

sql = "select PWDCOMPARE( "" & vSenha & "",senha, 0) as resultado,no_usuario, cd_perfil,login from
tab_usuario where login = "" & vusuario & """
rs.Open sql,dbconn,3,3

if rs.eof then
        response.redirect "login.asp?erro=usuario"
end if

definePerfil(rs("cd_perfil"))

if rs("resultado")= 1 then
        response.cookies("pmtoolweb")("data") = now
        response.cookies("pmtoolweb")("usuario") = rs("login")
        response.cookies("pmtoolweb")("nome") = rs("no_usuario")
        response.cookies("pmtoolweb")("cargo") = vCargo

if vCargo = "Administrador" then
        response.redirect "indexAdministrador.htm"
elseif vCargo = "Membro de Equipe" then
        response.redirect "indexMembro.htm"
elseif vCargo = "Gerente de Projeto" then

```

```
        response.redirect "indexGerente.htm"
    else
        response.redirect "indexMembro.htm"
    end if
else
    response.redirect "login.asp?erro=senha"
end if

rs.Close
Set rs = Nothing
Set dbconn = Nothing

%>
```

## **7. Discussão**

Os capítulos anteriores abordaram formalmente os requisitos, critérios e funcionalidades utilizados durante o desenvolvimento do sistema não se preocupando com as dificuldades e decisões tomadas durante a evolução do projeto. Neste capítulo será dado foco a este tipo de informação, preocupando-se em explicar detalhes para o leitor que o ajudará na execução de um projeto semelhante ou na escolha de ferramentas para a realização de um determinado projeto.

### **7.1 Escolha das Ferramentas e Linguagens**

Por se tratar de um projeto com aplicações empresariais, durante a seleção das ferramentas foi levado em consideração aspectos como: difusão do conhecimento dos softwares utilizados pelo mercado, facilidade de suporte, ferramentas fáceis de aprender e de administrar, linguagens de programação e Banco de Dados que suportassem todos esse requisitos.

Para tanto o projeto foi desenvolvido em ambiente Windows 2000 Professional com Microsoft Internet Information Server (IIS) versão 5.0 instalado.

Sendo escolhido o ambiente para desenvolvimento, a linguagem de programação e o Banco de Dados a serem utilizados foram escolhidos de acordo com as facilidades oferecidas pelo Sistema Operacional. Dessa maneira escolheu-se desenvolver o projeto utilizando linguagem ASP com acesso a Banco de Dados SQL Server 7.0 via ODBC.

A seguir são apresentadas as linguagens (ASP, JavaScript e HTML) utilizadas durante o desenvolvimento:

### **O que é ASP?**

**R:** ASP (Active Server Pages – Páginas de Servidor Ativas) são um ambiente para programação por scripts no servidor, que você pode usar para criar páginas dinâmicas, interativas e de alta performance. Como as páginas ASP, os scripts rodam no servidor e não no cliente. É o próprio servidor que transforma os scripts em HTML padrão, fazendo com que qualquer browser do mercado seja capaz de acessar um site que usa ASP.

Entre os recursos que podem ser implementados via ASP, podem ser citados os seguintes:

- Programação em VBScript ou JScript
- Acesso a banco de dados
- Sessões (persistência de informações no servidor)

ASP surgiu juntamente com o lançamento do Internet Information Server 3.0. Esta é uma solução Microsoft, que exige que o seu servidor precisa rodar um sistema operacional da Microsoft (Windows 95 ou superior ou NT). Os seguintes servidores suportam o uso de páginas ASP:

- Microsoft Internet Information Server versão 3.0 ou superior no Windows NT Server
- Microsoft Peer Web Services versão 3.0 ou superior no Windows NT Workstation
- Microsoft Personal Web Server no Windows 95 ou superior

A grande vantagem, porém, é que existe esta exigência apenas do lado do servidor. No lado do cliente, você pode utilizar qualquer browser, mesmo os que não suportam VBScript (como os da Netscape).

### **Vantagens do ASP**

- *Independência do browser*: ASP pode rodar páginas complexas no servidor e enviar somente os resultados para o cliente.
- *Páginas x Bancos de Dados*: Permite visualizar, atualizar e adicionar informações nos servidores SQL
- *Segurança do código fonte*: Como o Servidor retorna somente o resultado html, o código fonte (lógica) fica preservada.
- *Linguagens*: O ASP pode utilizar de comandos em VBScript, JavaScript e HTML (todos esses três tipos de comandos foram utilizados durante o projeto, sendo o VBScript utilizado apenas numa situação específica de criação da janela de upload para o sistema).

## **JavaScript**

JavaScript é uma linguagem de script, isto é, texto ASCII interpretado sem necessidade de compilação que permite a construção de páginas Web interativas (utilizado em validações de formulários, popups de confirmação etc), assim como servir de plataforma de integração com Applets Java (no projeto esta funcionalidade foi utilizada na integração com o menu do sistema) entre outras funcionalidades que não foram utilizadas nesse projeto, como integração com ActiveX, plug-ins de browsers etc.

JavaScript foi desenvolvido pela Netscape, com o intuito de capacitar a linha de produtos desta empresa (browser e Web Server) de uma linguagem básica de scripting. Baseada na linguagem Java, que por sua vez tem as suas origens nas linguagens de programação C e C++, a sintaxe de programação de JavaScript é semelhante, senão idêntica, às utilizadas por estas linguagens de programação.

## **HTML**

Se o leitor utiliza a Internet então ele já viu páginas criadas em HTML (HyperText Markup Language). Essa linguagem é a responsável por criar as páginas Web no formato como as conhecemos.

Não é possível programar páginas dinâmicas em linguagem HTML (existe a DHTML para esse fim), pois ela é simplesmente uma linguagem de marcação: ela serve para indicarmos formatações para textos, inserir imagens e ligações de hipertexto.

Os browsers identificam as marcações em HTML e apresentam os documentos conforme o que foi especificado por essas marcações.

## 8. Sugestões para as próximas versões

### a. Notificação por E-mail:

Durante a criação dos módulos, direcionamento de um Ponto de Atenção, Solicitação de Mudança seria enviado um E-mail para os usuários impactados para que pudessem ser informados sem que necessitassem entrar na ferramenta.

### b. Acompanhamento Gráfico:

A partir de gráficos poder visualizar a quantidades de Riscos, Pontos de Atenção etc para um determinado projeto, por períodos, acompanhar quantidade deles solucionados num determinado intervalo de tempo etc.

### c. Parametrização de telas por perfis:

Nessa primeira versão, o acesso a ferramenta por perfis foi realizado através de customização do menu de navegação e de diferentes formas de visualização dos módulos.

Por exemplo:

- Administrador → Visualiza todos os projetos cadastrados.
- Líder de Programa → Visualiza todos os projetos associados ao seu programa
- Gerente de Projeto e Membro de Equipe → Visualizam todos os projetos que estão associados.

Numa próxima versão, devem-se restringir ainda mais esses acessos, de maneira que algumas telas possuam funcionalidades que não possam ser acessadas por todos os perfis.

Para realizar tal alteração na ferramenta deve-se remodelar o Banco de Dados associando num relacionamento *three-way many-to-many* telas, campos e perfis.

d. Inserção de dados de referência a partir do aplicativo:

Alguns parâmetros bases da ferramenta como: perfis de usuário, fases de Risco, Pontos de Atenção, Solicitações de mudança etc, status, diretorias que são utilizados no cadastro de um usuário, Riscos, Pontos de Atenção, Solicitações de Mudança e Projetos respectivamente necessitam estar previamente cadastrados no Banco de Dados. Caso seja necessária a inserção de novos registros bases deverá ser rodada uma query de inserção diretamente no Banco de Dados.

Numa próxima versão deverá ser acrescida essa funcionalidade de cadastramento de dados de referência ao perfil de Administrador.

e. Visualização de currículos dos envolvidos no projeto:

A partir dessa nova funcionalidade poderíamos visualizar os currículos daqueles envolvidos no projeto partir de cadastros realizados pelos próprios usuários em formulários padrão no módulo de edição de dados do usuário.

## **9. Conclusão**

O software desenvolvido neste projeto permite facilitar no acompanhamento e gerenciamento de projetos em diversas áreas através da possibilidade de cadastro, consulta, alteração e deleção de Riscos, Pontos de Atenção, Solicitações de Mudança e Lições Aprendidas, bem como cadastro, consulta e alteração de Relatório de Status, Fichas de Projeto, Projetos e Usuários.

Neste projeto foram aplicados diversos conceitos estudados ao longo do curso de Engenharia Eletrônica e Computação na UFRJ, tais como modelagem de dados (Banco de Dados), modelagem e documentação do sistema (Engenharia de Software), técnicas de programação (Linguagens de Programação, Algoritmos e Estruturas de Dados, Sistema Operacional).

Por se tratar de um projeto voltado não só para o meio acadêmico, mas também para empresas em geral que poderiam estar vindo a utilizar este software, diversos conceitos utilizados, bem como experiência nas ferramentas e regras de gerenciamento foram aproveitados de projetos de consultoria realizados em empresas de telecomunicações ao longo de 2 anos e meio antes da realização do projeto.

Ferramentas desse tipo vêm sendo cada vez mais solicitadas por equipes de PMO em projetos. Essas equipes são responsáveis junto aos gerentes em garantir um bom gerenciamento e documentação de todo o projeto

## 10. Referências Bibliográficas

- [1] PRESSMAN, “Software Engineering: A Practitioner's Approach”, 5a Ed. McGraw Hill, 2001
- [2] Barbieri, C, Modelagem de Dados, IBPI Press, 1994
- [3] Weissinger, a Keyton, Asp - O Guia Essencial, Editora Campus, 2000
- [4] Wille, Christoph, Aprenda em 24 Horas Active Server Pages Asp, Editora Campus, 1999
- [5] <http://msdn.microsoft.com/>
- [6] <http://www.soloasp.com.ar/>
- [7] <http://www.javascript.com/>
- [8] <http://javascript.internet.com/>
- [9] <http://www.felixgers.de/teaching/sql/>
- [10] <http://www.w3schools.com/html/default.asp>
- [11] <http://www.asptutor.com/sql/>
- [12] <http://www.asptutorial.info/>
- [13] <http://www.del.ufrj.br/~ac/eel873.htm>

# APÊNDICES