

**UNIVERSIDADE REGIONAL DE BLUMENAU**  
**CENTRO DE CIÊNCIAS EXATAS E NATURAIS**  
**CURSO DE CIÊNCIAS DA COMPUTAÇÃO – BACHARELADO**

**FERRAMENTA DE APOIO À GERÊNCIA DE**  
**CONFIGURAÇÃO DE SOFTWARE BASEADO NO MODELO**  
**CMMI**

**ANDRÉ BOHN**

**BLUMENAU**  
**2005**

**2005/1-03**

**ANDRÉ BOHN**

**FERRAMENTA DE APOIO À GERÊNCIA DE  
CONFIGURAÇÃO DE SOFTWARE BASEADO NO MODELO  
CMMI**

Trabalho de Conclusão de Curso submetido à  
Universidade Regional de Blumenau para a  
obtenção dos créditos na disciplina Trabalho  
de Conclusão de Curso II do curso de Ciências  
da Computação — Bacharelado.

Prof. Everaldo Artur Grahl, Mestre – Orientador

**BLUMENAU**

**2005**

**2005/1-03**

**FERRAMENTA DE APOIO À GERÊNCIA DE  
CONFIGURAÇÃO DE SOFTWARE BASEADO NO MODELO  
CMMI**

Por

**ANDRÉ BOHN**

Trabalho aprovado para obtenção dos créditos  
na disciplina de Trabalho de Conclusão de  
Curso II, pela banca examinadora formada  
por:

Presidente: \_\_\_\_\_  
Prof. Everaldo Artur Grahl, Mestre – Orientador, FURB

Membro: \_\_\_\_\_  
Prof. Carlos Eduardo Negrão Bizzotto, Doutor, FURB

Membro: \_\_\_\_\_  
Prof. Joice Seleme Mota, Mestre, FURB

Blumenau, 20 de julho de 2005

Dedico este trabalho a minha esposa Karin e minha filha Helena que são as pessoas mais importantes da minha vida.

## **AGRADECIMENTOS**

A Deus, pelo seu imenso amor e graça.

Aos meus pais, que tornaram possível realizar esta graduação.

Ao meu sogro e sogra pela compreensão e ajuda no momento mais difícil.

Ao meu orientador, Everaldo Artur Grahl, por ter acreditado na conclusão deste trabalho e pela brilhante ajuda.

“De tudo ficam três coisas: a certeza de que estamos sempre começando, a certeza de que precisamos continuar e a certeza de que seremos interrompidos antes de terminar...”

Fernando Pessoa

## RESUMO

Com o mercado de desenvolvimento de sistemas cada vez mais competitivo, as empresas produtoras de software necessitam de um processo gerenciado e qualitativo. Para isso é preciso recorrer a modelos de qualidade que disciplinam o desenvolvimento de produtos. Um modelo muito utilizado nos dias de hoje é o CMMI (*Capability Maturity Model Integration*), que define metas e práticas que conduzem a empresa a produzir softwares com qualidade. Uma destas metas é a gerência da configuração de software que provê recursos para identificação, controle da evolução e auditoria de artefatos produzidos durante o desenvolvimento de um projeto de software. Este trabalho teve como objetivo a implementação de uma ferramenta para auxiliar uma empresa a gerenciar a configuração de software. As atividades básicas da gerência de configuração foram implementadas e a ferramenta atendeu a maioria das metas e práticas definidas no CMMI.

Palavras-chave: Qualidade de software. Gerência de configuração. CMMI.

## **ABSTRACT**

With the market of systems development more and more competitive, the producing companies of software need a process that managed and qualitative. For this is necessary to appeal the quality models that discipline the development of products. A model very used nowadays is the CMMI (Capability Maturity Model Integration), that it defines practical goals and that they lead the company to produce softwares with quality. One of these goals is the management of the software configuration that to provide resources for identification, control of the evolution and devices audit produced during the development of a software project. This work had as objective the implementation of a tool to assist a company to manage the software configuration. The basic activities of the management configuration had been implemented and the tool ansucered the majority of the defineds goals and practices from CMMI.

**Key-Words:** Quality of software. Software management configuration. CMMI.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Níveis de capacidade representação continua .....	17
Quadro 1 – Exemplos de Itens de configuração .....	26
Figura 2 – Processo de controle da configuração .....	28
Figura 3 – Tela principal do Team Coherence .....	32
Figura 4 – Tela de gravação de nova versão do item de configuração.....	33
Figura 5 – Tela principal do WinCVS.....	34
Figura 6 - Diagrama de casos de uso primário .....	38
Figura 7 - Diagrama de casos de uso secundários.....	40
Figura 8 - Diagrama de classes da camada de entidade .....	43
Figura 9 - Diagrama de classes da camada de controle.....	44
Figura 10 - Diagrama de atividades.....	45
Figura 11 - Diagrama de estados do pedido de modificação.....	46
Figura 12 - Diagrama de estados do item de configuração .....	47
Figura 13 - Diagrama de entidade-relacionamento .....	48
Quadro 2 – Dicionário de dados .....	52
Quadro 3 – Código de conexão com o banco de dados.....	54
Quadro 4 – Código para manutenção de tabelas .....	55
Quadro 5 – Código da camada de controle da entidade gerente de projeto .....	56
Quadro 6 – Código da definição da Classe de gerente de projeto.....	56
Figura 14 – Tela de edição padrão.....	57
Quadro 7 – Código do botão gravar da tela de gerente de projeto .....	58
Quadro 8 – Exemplo de arquivo com versão anterior e atual .....	58
Quadro 9 – Comandos resultantes .....	59
Quadro 10 – Cabeçalho do arquivo de versões .....	59
Quadro 11 – Janela de versão do arquivo de versões .....	59
Quadro 12 – Arquivo de versões completo .....	60
Figura 15 – Tela de cadastro de gerente de projeto.....	61
Figura 16 – Tela de <i>login</i> .....	61
Figura 17 – Tela de cadastro de projeto .....	62
Figura 18 – Tela de cadastro de versão de configuração.....	62
Figura 19 – Tela de cadastro de Tipos de item de configuração .....	63

Figura 20 – Tela principal do sistema .....	64
Figura 21 – Tela de registro de item de configuração .....	64
Figura 22 – Tela de cadastro de analista de sistemas .....	65
Figura 23 – Tela de cadastro de auditor .....	66
Figura 24 – Tela de cadastro de clientes.....	66
Figura 25 – Tela principal com visualização dos pedido de modificação.....	67
Figura 26 – Tela de registro de pedido de modificação .....	67
Figura 27 – Tela de registro de avaliação de pedido de modificação .....	68
Figura 28 – Tela de registro de decisão .....	69
Figura 29 – Tela de registro de auditoria do pedido de modificação .....	70
Figura 30 – Relatório de itens de configuração .....	71
Figura 31 – Relatório de pedidos de modificação .....	71
Figura 32 – Relatório de auditorias .....	72
Figura 33 – Relatório de tipos de itens de configuração .....	72
Figura 34 – Relatório de versões da configuração .....	73
Figura 35 – Relatório do projeto.....	73
Quadro 13 – Resultado análise de ferramentas de gerência de configuração .....	74

## **LISTA DE SIGLAS**

ME - Meta Específica

MG - Meta Genérica

PE - Prática Específica

PG - Prática Genérica

CE - Compromissos a Executar

HE - Habilidades a Executar

DE - Direcionando a Execução

VA - Verificando a Execução

# SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>13</b>
1.1 OBJETIVOS.....	14
1.2 ORGANIZAÇÃO DO TEXTO.....	15
<b>2 FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>16</b>
2.1 CAPABILITY MATURITY MODEL INTEGRATION (CMMI).....	16
2.1.1 Áreas de Processo.....	16
2.1.2 Representação contínua.....	17
2.1.3 Representação por estágios.....	18
2.1.4 Gerência de configuração no CMMI.....	19
2.1.4.1 Metas e Práticas.....	21
2.2 GERÊNCIA DE CONFIGURAÇÃO.....	24
2.2.1 Linhas básicas (baselines).....	25
2.2.2 Itens da configuração de software.....	25
2.2.3 O processo de gerência de configuração.....	26
2.2.3.1 Identificação de itens de configuração.....	26
2.2.3.2 Controle da configuração.....	27
2.2.3.2.1 Controle de Manutenção.....	29
2.2.3.2.2 Controle de Versão.....	30
2.2.4 Relato da situação.....	30
2.2.5 Auditoria de configuração.....	31
2.2.6 Trabalhos correlatos.....	31
2.2.7 Ferramentas para gerência de configuração.....	31
<b>3 DESENVOLVIMENTO DO TRABALHO.....</b>	<b>35</b>
3.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO.....	35
3.1.1 Requisitos funcionais.....	35
3.1.2 Requisitos não funcionais.....	37
3.2 ESPECIFICAÇÃO.....	37
3.2.1 Diagrama de casos de uso.....	37
3.2.2 Diagramas de Classes.....	42
3.2.3 Diagrama de Atividades.....	44
3.2.4 Diagramas de Estados.....	46

3.2.5 Diagramas de Entidade-Relacionamento .....	47
3.2.6 IMPLEMENTAÇÃO .....	52
3.2.7 Técnicas e Ferramentas Utilizadas.....	53
3.2.8 Estrutura do código .....	53
3.2.8.1 Camada de persistência.....	54
3.2.8.2 Camada de controle .....	55
3.2.8.3 Camada de entidade .....	56
3.2.8.4 Camada de fronteira.....	57
3.2.9 Formato do arquivo para armazenamento de diferenças.....	58
3.2.10 Operacionalidade da ferramenta .....	60
3.3 RESULTADOS E DISCUSSÃO .....	74
3.3.1 Análise de ferramentas existentes no mercado .....	74
3.3.2 Análise da ferramenta implementada em relação ao CMMI .....	75
<b>4 CONCLUSÕES.....</b>	<b>79</b>
4.1 EXTENSÕES .....	80
<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>81</b>

## 1 INTRODUÇÃO

O mercado de software atual tem exigido processos de desenvolvimento de software que tenham como resultado um produto com qualidade, respeitando os prazos e custos determinados. Para alcançar estes objetivos é necessário que a organização implante um processo de desenvolvimento de software que estabeleça etapas bem definidas, e seja adequado à cultura organizacional da empresa (ROCHA; MALDONADO; WEBER, 2001, p. 22).

Existem modelos de referência criados por institutos de pesquisa que fornecem uma seqüência de disciplinas, auxiliando a organização a construir um processo de desenvolvimento adequado. Uma das disciplinas abordadas por estes modelos, é a gerência de configuração de software, que é responsável pelo controle de todos os artefatos gerados durante o ciclo de vida do software.

A gerência de configuração é uma disciplina básica em vários modelos da qualidade e uma atividade independente de qualquer outra etapa que ocorre ao longo de todo o processo de desenvolvimento. As modificações realizadas em um projeto, podem gerar perda de funcionalidades e erros.

Para evitar estes problemas as atividades de gerência de configuração são desenvolvidas para identificar modificações, controlar modificações, garantir que as modificações sejam adequadamente implementadas e relatar as modificações para que outros membros do projeto não venham produzir inconsistências (PRESSMAN, 2002, p. 219). Para facilitar a implantação da gerência de configuração, é preciso escolher um modelo de referência adequado.

O modelo escolhido para este trabalho é o Capability Maturity Model Integration (CMMI), proposto pela Software Engineering Institute (SEI), que vem sendo um modelo

adotado por algumas organizações. O CMMI auxilia na melhoria de processos organizacionais e a habilidade de gerenciar o desenvolvimento e manutenção de produtos e serviços. Ele é um modelo baseado em níveis de maturidade onde a gerência de configuração encontra-se no nível 2 (SEI CMMI Product Team, 2001).

De acordo com Pressman (2002, p. 18), a utilização de ferramentas de engenharia de software apoiadas por computador, fornecem apoio automatizado para os modelos de qualidade e servem como depósito importante de informações sobre análise, projeto, construção de programas e teste.

Na área da gerência de configuração existem ferramentas proprietárias como o Rational ClearCase que faz parte da suíte Rational da IBM (IBM, 2004), e o StarTeam da Borland (BORLAND CORPORATION, 2004), que oferecem um suporte completo e customizável a todos os itens da gerência de configuração. Existem também ferramentas baseadas em software livre como o Concurrent Versions System (CVS, 2003) que atende apenas à parte de controle de versão, no entanto é largamente utilizado para gerenciar projetos *open-source*.

## 1.1 OBJETIVOS

O objetivo deste trabalho é desenvolver uma ferramenta de suporte ao processo de gerência de configuração de software.

Os objetivos específicos são:

- a) Incorporar funcionalidades na ferramenta para atender diretrizes previstas no modelo CMMI no que se refere ao processo de gerência de configuração;
- b) Comparar a ferramenta criada com outras ferramentas existentes no mercado.

## 1.2 ORGANIZAÇÃO DO TEXTO

O trabalho está dividido em quatro capítulos, cuja descrição segue.

O primeiro capítulo faz uma introdução, abordando em linhas gerais o trabalho, sua relevância, objetivo e o contexto no qual está inserido.

O segundo capítulo fundamenta teoricamente o trabalho, através de uma revisão bibliográfica sobre gerência de configuração e o modelo de qualidade CMMI.

O terceiro capítulo trata sobre a especificação e implementação da ferramenta, através de seus requisitos, diagramas de casos de uso, descrição dos casos de uso, diagrama de seqüência, diagrama de estados, diagrama entidade relacionamento (DER) e dicionário de dados.

Por fim, tem-se a conclusão, focando os resultados do trabalho e as limitações da ferramenta.

## 2 FUNDAMENTAÇÃO TEÓRICA

O objetivo deste capítulo é apresentar a fundamentação teórica sobre o modelo CMMI e a gerência de configuração de software.

### 2.1 CAPABILITY MATURITY MODEL INTEGRATION (CMMI)

O CMMI é um *framework* que possui os elementos necessários para tornar um processo de desenvolvimento de software mais eficiente e controlado e foi construído para integrar os diferentes modelos criados a partir do sucesso alcançado pelo CMM para software. O termo disciplina foi adotado para representar os diferentes corpos de conhecimento integrados no CMMI. Cada disciplina possui áreas de processo para descrever suas práticas e metas a serem alcançadas. O CMMI introduziu duas representações para apresentar seu conteúdo, a representação contínua e por estágios. A representação contínua usa níveis de capacidade para medir o processo de melhoria, enquanto a representação em estágio usa níveis de maturidade (BARTIÉ, 2002).

#### 2.1.1 Áreas de Processo

Uma área de processo é um conjunto de práticas relacionadas em uma determinada área que, quando executadas coletivamente, satisfazem um conjunto de objetivos considerados importantes para executar significantes melhorias naquela área (SEI CMMI Product Team, p. 14, 2002).

Dentro de cada área existem metas específicas que são implementadas pelas práticas específicas e metas genéricas implementadas pelas práticas genéricas. Práticas e metas específicas são únicas para cada área de processo, enquanto que metas e práticas genéricas

aplicam-se a múltiplas áreas. Cada prática pertence a um único nível de capacidade.

### 2.1.2 Representação contínua

Segundo Agnol e Herbert (p. 109, 2004), na representação contínua os componentes principais são áreas de processo. A principal diferença entre níveis de maturidade e capacidade é a representação a qual eles pertencem e como eles são aplicados. Níveis de capacidade pertencentes à representação contínua, aplicam-se a cada área de processo no estabelecimento do processo de melhoria da organização. Existem seis níveis de capacidade, numerados de 0 a 5. Cada nível de capacidade corresponde a uma meta genérica e um conjunto de práticas genéricas e específicas. A figura 5 faz a representação da relação entre as áreas de processo e níveis de capacidade.

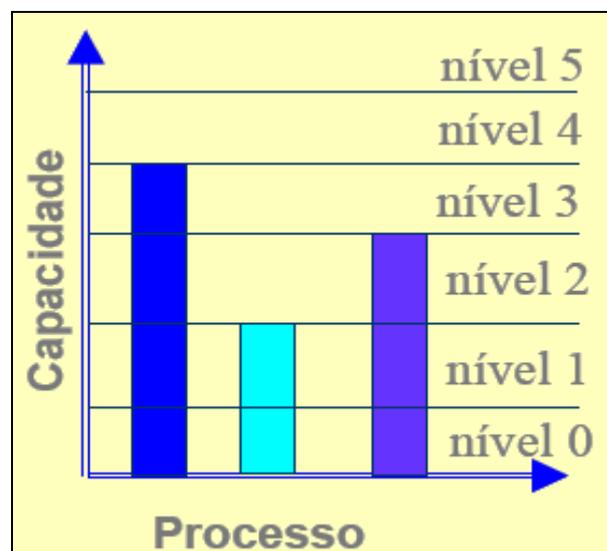


Figura 1 – Níveis de capacidade representação contínua

Os níveis de capacidades são descritos a seguir:

- a) 0 Incompleto: processo não executado ou executado parcialmente;
- b) 1 Executável: satisfaz metas específicas da área de processo;
- c) 2 Controlado: processo executado e também planejado, monitorado e controlado

para atingir um objetivo;

- d) 3 Definido: processo gerenciado, adaptado de um conjunto de processos padrão da organização;
- e) 4 Gerenciado quantitativamente: processo definido, controlado utilizando estatísticas ou outras técnicas quantitativas;
- f) 5 Otimizado: processo gerenciado quantitativamente para a melhoria contínua do desempenho do processo.

Cada nível de capacidade forma uma base sobre a qual será estabelecido o próximo nível. Usando a representação contínua, a organização pode produzir um perfil de nível de capacidade, ou seja, uma lista das áreas de processo e seus correspondentes níveis de capacidade.

Nesta representação as áreas de processo além de possuir a classificação por níveis, são agrupadas em quatro categorias: gerenciamento de processo, gerenciamento de projeto, engenharia e apoio. A gerência de configuração é uma prática da área de processo de apoio (CHAN; BIANCHINI, p. 2, 2005).

### 2.1.3 Representação por estágios

Segundo Agnol e Herbert (p. 109, 2004), a representação por estágio provém uma seqüência comprovada de melhorias, iniciando com práticas básicas e evoluindo através de um caminho pré-definido de níveis sucessivos, permite comparações no âmbito de uma organização, ou entre organizações e é a representação mais utilizada por organizações de software segundo os resultados de avaliação do CMMI reportados ao SEI.

Na representação por estágios os níveis de maturidade são uma forma de priorizar as ações de melhoria de tal forma que se aumente a maturidade do processo de software. Cada

nível indica quais áreas do processo devem ser implantadas para que este seja alcançado.

Existem cinco estágios e estes estão descritos da seguinte forma:

- a) nível 1 – inicial: este nível é caracterizado pela ausência de processos, sendo que as principais conseqüências são o abandono do projeto, não cumprimento de prazos e aumento de custos de projeto;
- b) nível 2 – gerenciado: o ponto principal deste nível é que a organização começa a fazer o planejamento e gerenciamento do projeto, onde os controles sobre procedimentos, atividades e compromissos, são bem fundamentados. É neste nível que acontece a gerência de configuração do software;
- c) nível 3 – definido: neste nível os processos são bem definidos e compreendidos, descritos em padrões, procedimentos e métodos;
- d) nível 4 – quantitativamente gerenciado: processos para medir a produtividade e qualidade são implantados, baseados nas necessidades dos clientes, usuários finais, da organização e dos implementadores do processo;
- e) nível 5 – otimizado: este nível é caracterizado pela existência processos para melhoria contínua. Os processos implantados são avaliados para prevenir defeitos e não cometer erros repetidos em projetos futuros.

Com exceção do nível 1 (considerado imaturidade organizacional), todos os níveis restantes (gerenciado, definido, quantitativamente gerenciado e em otimização) são compostos de áreas de processos que tratam aspectos de cada componente do processo.

#### 2.1.4 Gerência de configuração no CMMI

O objetivo da gerência de configuração no CMMI é estabelecer e manter a integridade dos artefatos, usando a identificação da configuração, controle da configuração, relato do status da configuração, e auditorias da configuração. Abaixo segue uma explicação mais

detalhada sobre o que cada um desses itens que formam a base da norma CMMI (SEI CMMI Product Team, p. 187, 2002).

- a) Identificação de Configuração: tem como objetivo permitir a distinção das características do software através da identificação e documentação das mesmas. Dessa maneira é possível rastrear toda mudança durante o ciclo de vida do desenvolvimento. São atividades da Identificação de Configuração:
- identificar a estrutura do sistema de software,
  - identificar de maneira unívoca cada item de configuração,
  - definir inter-relacionamento entre os itens de configuração,
  - identificar o conjunto de linhas base a serem criadas e mantidas,
  - caracterizar o conteúdo de cada linha base, identificando os elementos básicos que a compõem,
  - criar o mecanismo de nomeação dos itens de configuração e linha base,
  - definir o mecanismo de identificação de versão dos itens de configuração e linhas base,
  - caracterizar os meios de acesso aos itens de configuração;
- b) Controle de Configuração: quando uma mudança em algum Item de Configuração ou linha base requisitada, a gerência de configuração diz como ela deve ser realizada. Ele define os procedimentos utilizados para realizar as mudanças, fazendo-as de modo que o software possa evoluir sem perda de informação e que todas as conseqüências das mudanças do software sejam capturadas;
- c) Administração de Estado: como o próprio nome sugere, a administração de estado diz respeito a estabelecer registros do estado das linhas base e itens de configuração atuais, incluindo as ações que foram tomadas até o momento;
- d) Auditagem de Configuração: a auditagem permite determinar se a evolução do

software esta seguindo o esperado. Deve haver documentação para todos os itens de configuração e permitir a validação de cada um dos produtos de trabalho.

#### 2.1.4.1 Metas e Práticas

As metas e práticas definidas para a área de processo de gerência de configuração são (SEI CMMI Product Team, p. 190, 2002):

- a) ME 1 Estabelecer Linhas base: linhas base de produtos de trabalhos identificados são estabelecidos. Esta meta cobre as práticas específicas para estabelecer linhas base;
  - PE 1.1 Identificar Itens de Configuração: identificação dos itens de configuração, componentes, e produtos de trabalho relacionados que serão colocados sobre a gerência de configuração,
  - PE 1.2 Estabelecer um Sistema de Gerenciamento de Configuração: estabelecer e manter sistema de gerenciamento de configuração e sistema de gerenciamento de mudanças para controlar produtos de trabalho,
  - PE 1.3 Criar ou Liberar as Linhas Base: crie ou libere linhas base para uso interno e para entregar ao cliente;
- b) ME 2 Rastrear e Controlar Mudanças: mudanças nos produtos de trabalho sob a gerência de configuração são rastreadas e controladas;
  - PE 2.1 Rastrear Pedidos de Mudança: rastreie os pedidos de mudança para os itens de configuração,
  - PE 2.2 Controlar Itens de Configuração: controle mudanças nos itens de configuração;
- c) ME 3 Estabelecer Integridade: integridade da linha base é estabelecida e mantida;

- PE 3.1 Estabelecer Registros de Gerência de Configuração: estabeleça e mantenha registros descrevendo os itens de configuração,
  - PE 3.2 Realizar Auditorias de Configuração: realize auditorias de configuração para manter a integridade das baselines de configuração;
- d) MG 2 Institucionalizar um Processo Controlado: o processo é institucionalizado como um processo controlado;
- PG 2.1 Estabelecer uma Política Organizacional: estabelecer e manter uma política organizacional para planejamento e execução do processo de gerência de configuração,
  - PG 2.2 (HE 1) Planejar o Processo: estabelecer e manter o plano para a execução do processo de gerência de configuração,
  - PG 2.3 (HE 2) Fornecer Recursos: forneça recursos adequados para a execução do processo da gerência de configuração, desenvolvimento dos produtos de trabalho e fornecimento de serviços dos processos,
  - PG 2.4 (HE 3) Atribuir Responsabilidades: atribua responsabilidades e autoridades para a execução do processo, desenvolvimento de produtos de trabalho, e fornecimento de serviços do processo de gerência de configuração,
  - PG 2.5 (HE 4) Treinar Pessoas: treine as pessoas que executam ou dão suporte ao processo de gerência de configuração quando necessário,
  - PG 2.6 (DE 1) Gerencie as Configurações: coloque os produtos de trabalho identificados do processo de gerência de configuração sob níveis apropriados de gerência de configuração,
  - PG 2.7 (DE 2) Identificar e Envolver Stakeholders Relevantes: identifique e envolva os Stakeholders relevantes do processo de gerência de configuração como planejado,

- PG 2.8 (DE 3) Monitorar e Controlar o Processo: monitore e controle o processo de gerência de configuração de acordo com o plano de execução do processo e tome ações corretivas apropriadas,
  - PG 2.9 (VE 1) Avaliar Adesão: avalie objetivamente a adesão do processo de gerência de configuração de acordo com sua descrição, padrões e procedimentos, e relacione as não-conformidades,
  - PG 2.10 (VE 2) Reveja o Status com a Alta Gerência: reveja as atividades, status e resultados do processo de gerência de configuração com a alta gerência e resolva problemas;
- e) MG 3 Institucionalizar um Processo Definido: o processo é institucionalizado como um processo definido;
- PG 3.1 Estabeleça um Processo Definido Estabeleça e mantenha a descrição de um processo definido de gerência de configuração,
  - PG 3.2 Colete Informações de Melhoria: colete produtos de trabalho, medidas, resultados de medidas e informações de melhoria derivadas do planejamento e execução do processo de gerência de configuração para dar suporte ao uso e melhoria futuros dos processos e recursos do processo da organização.

## 2.2 GERÊNCIA DE CONFIGURAÇÃO

De acordo com Sommerville (2003, p. 550), gerência de configuração é a utilização de modelos e padrões para gerenciar um software em desenvolvimento. Mudanças nas funcionalidades, correções de defeitos e adaptações para diferentes hardwares e sistemas operacionais, geram diferentes versões do sistema, e estas podem estar sendo desenvolvidas ao mesmo tempo. A gerência da configuração serve para evitar conflitos nos itens de configuração modificados.

A gerência de configuração funciona como uma ferramenta para comunicar os requisitos do cliente e garante que estes requisitos possam ser rastreados até o produto final. A principal entidade definida no processo de gerência de configuração é o item de configuração, onde o software é visto como um conjunto de itens de configuração (PETTERS; PEDRYCZ, 2001, p. 68).

Segundo Paula Filho (2001), mesmo um projeto pequeno gera diversos itens de configuração. Ao longo do ciclo de vida de um produto, é importante que resultados sejam guardados para realizar eventuais modificações. Organizar e controlar a proliferação dos itens de configurações é o objetivo da gerência de configuração.

Para manter a integridade dos itens de configuração durante o ciclo de vida do software, a gerência de configuração estabelece algumas disciplinas que são:

- f) identificação da configuração de software;
- g) controle da configuração;
- h) relato da situação;
- i) auditoria da configuração.

### 2.2.1 Linhas básicas (baselines)

Uma linha básica é um marco de referência no desenvolvimento de software. A revisão formal de um item de configuração, feito por uma equipe de projeto, gera uma linha básica. Uma vez definida uma linha básica, as modificações em qualquer um dos itens de configuração, devem ser avaliadas e aprovadas para então fazer parte da linha básica (PRESSMAN, 2002, p. 221).

Todo item de configuração integrante da linha básica deverá ser depositado em um repositório de software, ou seja, um banco de dados contendo os itens do projeto, sendo que para qualquer membro da equipe de projeto realizar uma modificação, deverá fazer uma cópia do item de configuração para seu ambiente particular de trabalho. Terminada a alteração deverá fazer uma descrição das atividades e o item será avaliado e aprovado, para assim fazer parte definitivamente do repositório de software.

Durante a alteração, o item de configuração em questão ficará bloqueado para os outros membros da equipe, evitando assim que seja usado um item de configuração não revisado e aprovado.

### 2.2.2 Itens da configuração de software

Itens de configuração de software são qualquer informação gerada a partir do processo de engenharia de software. Pode ser um documento, definições de casos de teste, uma biblioteca de funções, arquivo fonte de programa, modelos de dados, entre outros. É importante ressaltar que as ferramentas utilizadas no desenvolvimento, como compiladores, ferramentas CASE e editores, também são considerados itens de configuração, uma vez que a adoção de uma nova versão possa gerar resultados diferentes (PRESSMAN, 2002, p. 222).

Os itens de configuração de software podem ser organizados em conjunto a fim de

formar objetos de configuração, e estes são catalogados na base de dados do projeto com um único nome. Um objeto de configuração é relacionado com outros objetos, com o objetivo de visualizar o impacto da alteração de um determinado item de configuração.

No quadro 1, são apresentados alguns exemplos de itens de configuração.

<b>Item de Configuração</b>	<b>Exemplo</b>
Plano de gestão	Plano do processo, plano de gerência de configuração, plano de testes, plano de manutenção
Especificação	Requisitos, projeto, especificação de teste
Projeto	Código-fonte
Teste	Projeto dos testes, casos, procedimentos, dados e geração de testes
Software de suporte	Documentos de planejamento
Dicionário de dados	Especificação dos requisitos de software
Código	Fonte, executável, requisito
Bibliotecas	Componentes, bibliotecas reutilizáveis
Banco de dados	Banco de dados de auditoria
Manutenção	Listagem, descrição detalhada de projeto, medidas

Fonte: Pedrycz e Petters (2001, p. 68)

Quadro 1 – Exemplos de Itens de configuração

### 2.2.3 O processo de gerência de configuração

Para alcançar os objetivos da gerência de configuração são estabelecidos quatro atividades principais, descritas nas próximas seções: identificação de itens de configuração, controle da configuração, relato da situação e auditoria da configuração.

#### 2.2.3.1 Identificação de itens de configuração

Segundo Sommerville (2003, p. 552), durante o desenvolvimento de um software, vários documentos são gerados. Para um determinado membro da equipe, o histórico das

modificações desses documentos pode ser interessante. A elaboração de um planejamento de gerenciamento de configuração descreve os itens que devem ser controlados.

Esta etapa é importante, pois a equipe de projeto escolhe os itens que serão gerenciados, e como estes serão organizados. Existem alguns passos a seguir com o objetivo de chegar a um resultado satisfatório.

O primeiro passo é selecionar os itens a serem gerenciados, evitando itens não relevantes, pois o excesso de documentação onera a gerência. Os itens geralmente escolhidos são os mais usados durante o ciclo de vida do software, os mais genéricos, os mais importantes para a segurança e os que são alterados por vários desenvolvedores. Depois de selecionados, é feita uma descrição dos relacionamentos entre os itens, para facilitar na identificação de itens afetados em uma determinada alteração. Feito isto, é feita a atribuição de nomes exclusivos para cada componente, e por fim, com base nos dados levantados devem ser planejadas as linhas de referência (ROCHA, MALDONADO e WEBER, 2001, p. 61).

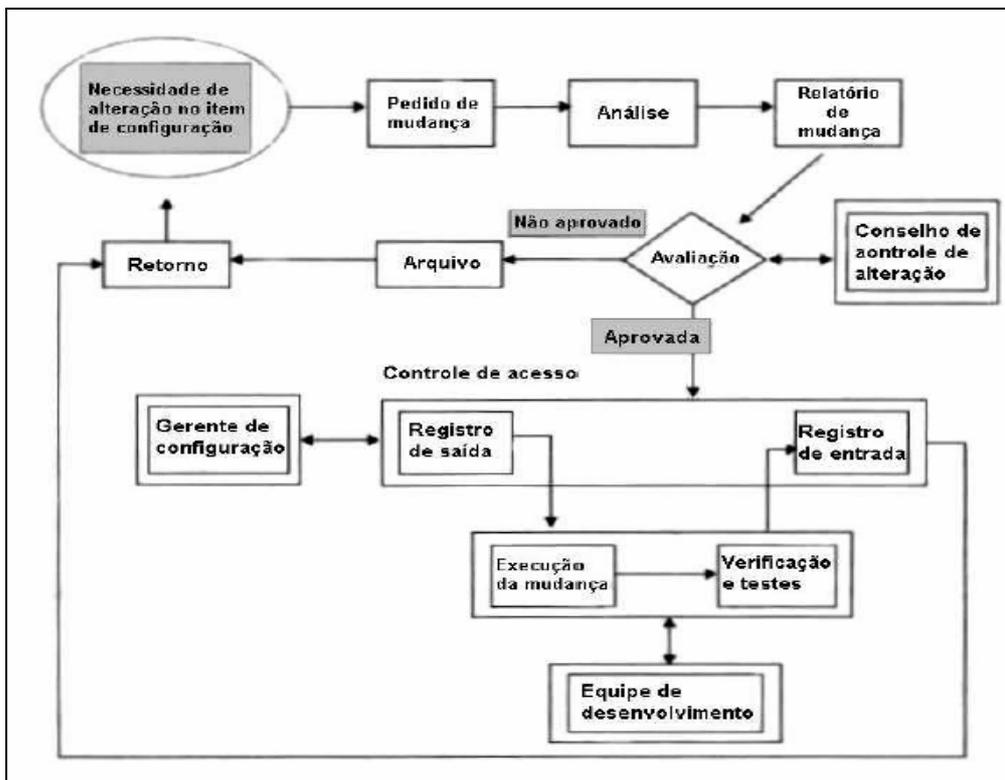
#### 2.2.3.2 Controle da configuração

Esta atividade é descrita por dois controles específicos que são o controle de manutenção e o controle de versão. Mudanças descontroladas podem levar rapidamente ao caos. Definir processos que combinem procedimentos humanos e ferramentas automatizadas é muito importante para um bom controle de mudanças. Este controle só pode ser aplicado após a definição da linha básica (PRESSMAN, 2002, p. 226).

Para Sommerville (2003, p. 554), gerência de versão consiste no processo de identificar e acompanhar o desenvolvimento de diferentes versões de um sistema. A pessoa responsável pela gerência pode a qualquer momento acessar versões diferentes sem que estas sofram modificações acidentais.

Estas duas atividades estão presentes em todo o ciclo de vida do software e garantem que novas versões sejam implementadas sem que estas percam as funcionalidades descritas no documento de requisitos.

O fluxo de controle de manutenção é exemplificado na figura 1.



Fonte: Rocha, Maldonado e Weber (2001, p. 64)

Figura 2 – Processo de controle da configuração

O controle de manutenção começa com a necessidade de alteração em um item de configuração. É feito a análise de um pedido de mudança para averiguar a importância da mudança e o seu impacto na configuração. Se for aprovada, os itens de configuração são copiados para área de trabalho da equipe de desenvolvimento. Depois de terminada a implementação é feita a verificação e se estiver tudo certo os itens alterados são copiados para o repositório principal, gerando uma nova versão do item de configuração.

#### 2.2.3.2.1 Controle de Manutenção

Segundo Pressman (2002, p. 228), o processo do controle de modificação inicia com um pedido de modificação, que é avaliado para verificar se a modificação é viável de ser implementada. O resultado da análise é apresentado para o responsável para um gerente de projeto, ou qualquer outra autoridade designada para administrar o projeto, que irá tomar a decisão final sobre o estado e a prioridade da modificação.

Para cada modificação aprovada, é gerada uma ordem de modificação que descreve como a modificação será implementada, as restrições que devem ser respeitadas e os critérios da revisão e auditoria. O item de configuração a ser modificado, é copiado para a área de trabalho do analista que irá implementar a modificação. Terminada a implementação, um auditor verifica o item de configuração modificado para garantir que a modificação foi implementada conforme o descrito na solicitação de modificação. Estando o item avaliado e aprovado, volta a fazer parte da linha base da configuração.

O processo de saída e entrada da linha base da configuração é a implementação de dois elementos importantes do controle de modificação que é o controle de acesso e controle de sincronização. O controle de acesso garante que apenas o analista responsável pela modificação terá acesso ao item de configuração relacionado. O controle de sincronização ajuda a garantir que modificações paralelas, feitas por pessoas diferentes, não se anulem.

Sendo assim, o controle de acesso estará verificando a propriedade do item de configuração solicitado e o controle de sincronização bloqueia no repositório do projeto, para que não seja feita outra modificação até que a versão atual retirada tenha sido reintroduzida. O fato do item estar bloqueado não impede que outros analistas façam uma cópia do item de configuração, no entanto não poderá atualizar a linha base. Somente depois de avaliado e aprovado o item alterado é atualizado na linha base e desbloqueado.

#### 2.2.3.2.2 Controle de Versão

A cada ciclo de modificação do item de configuração, é gerada uma versão (revisão) do item. Todas as versões devem ser identificadas e armazenadas, a fim de obter o controle sobre as diversas versões. É conveniente que o esquema de identificação seja feito em forma de árvore, pois ao mesmo tempo que se mantém um histórico das versões dos itens, permite a identificação única e ramificações a partir de qualquer versão (ROCHA, MALDONADO e WEBER, 2001, p. 62).

Um grande número de versões pode ser gerados, e conseqüentemente o espaço de armazenamento aumenta consideravelmente. Para minimizar este problema existe o conceito de delta, onde são armazenadas uma versão completa e as diferenças entre as versões. Existem variações deste conceito que são o delta negativo, onde é armazenado integralmente a versão mais recente e as diferenças existentes até então, e o delta positivo, que consiste em armazenar a versão mais antiga e, para montar as versões mais recentes, processam-se as diferenças armazenadas.

#### 2.2.4 Relato da situação

Conforme Rocha, Maldonado e Weber (2001, p. 63), esta tarefa tem como função deixar ao alcance de todos os membros de um projeto, as alterações e modificações realizadas. Cada desenvolvedor poderá acessar as informações perante autorização do gerente de configuração. Relatórios de alterações de grande impacto deverão ser emitidos para todas as pessoas envolvidas, a fim de agilizar o desenvolvimento e otimizar a comunicação. Isto evita que um mesmo item de configuração seja alterado por outro membro com intenções diferentes ou conflitantes.

### 2.2.5 Auditoria de configuração

A abordagem de Pressman (2002, p. 231) diz que a auditoria de configuração vem a garantir que as implementações foram feitas adequadamente. Esta complementa a revisão técnica formal, avaliando apenas os pontos que não foram considerados durante a revisão. Esta etapa é geralmente realizada por uma equipe de garantia de qualidade.

### 2.2.6 Trabalhos correlatos

Em Barbaresco (2000), é descrito em profundidade os conceitos da gerência de configuração e foi feito um estudo detalhado dos padrões e modelos de qualidade de software. A ferramenta desenvolvida atende as atividades básicas do processo de gerência de configuração .

Em Andrade (2002), fala-se sobre os processos de ciclo de vida de um software e descreve a gerência de configuração e o modelo CMMI. Neste trabalho é avaliado o processo de gerência de configuração de uma empresa sediada em São José dos Campos - SP chamada COMPSIS, apontando as dificuldades encontradas e possíveis soluções.

### 2.2.7 Ferramentas para gerência de configuração

Entre as ferramentas para gerência de configuração disponíveis no mercado, foram pesquisadas três para a elaboração deste trabalho: Team Coherence, TortoiseCVS e WinCVS.

O Team Coherence (QSC, 2003) é uma ferramenta de gerência de configuração da Quality Software Company. É voltada para desenvolvedores que utilizam o ambiente de programação das ferramentas Borland, como Delphi, ModelMaker e C++ Builder. A figura 2 apresenta a tela principal da ferramenta.

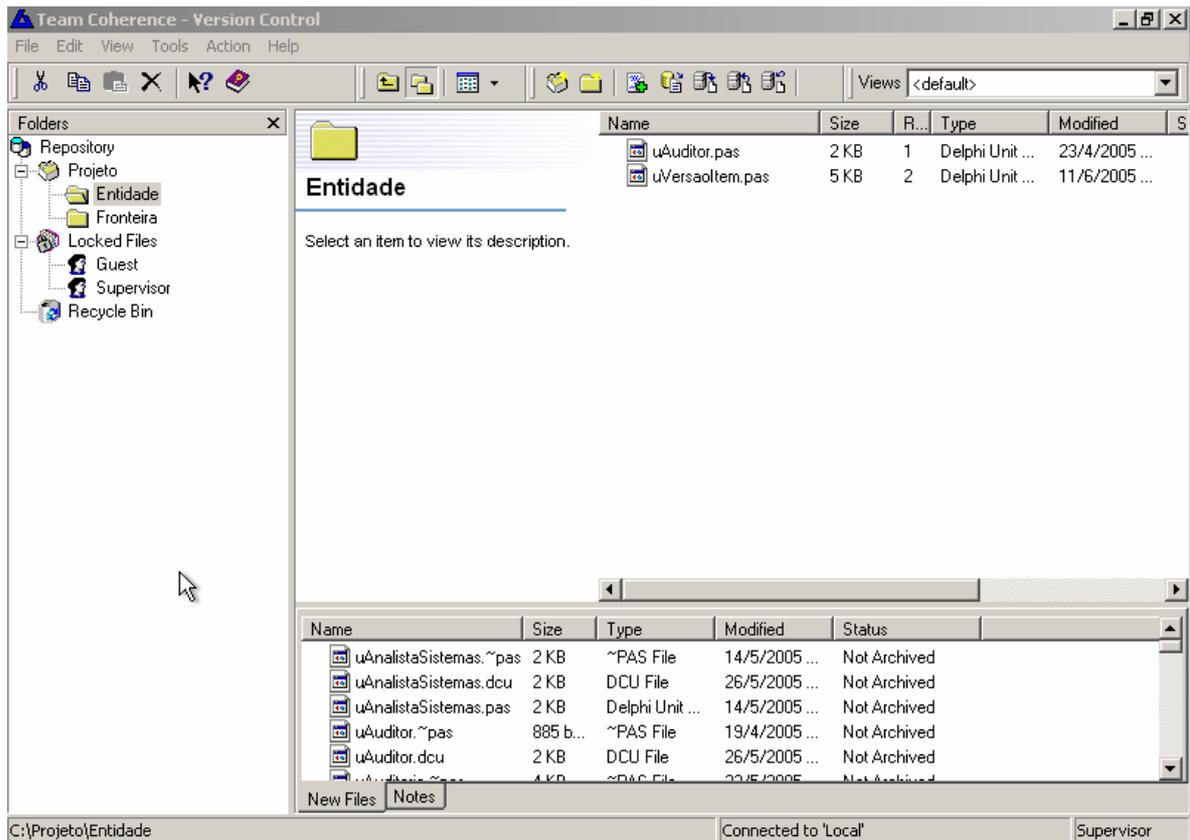


Figura 3 – Tela principal do Team Coherence

O TortoiseCVS (TORTOISE, 2003) é uma ferramenta baseada no projeto *open-source Concurrent Version System* (CVS, 2003), que fornece serviço para gestão de configuração. O TortoiseCVS é integrado com a *Shell* do Windows sendo bem prática de ser utilizado. Na figura 3 apresenta uma tela da ferramenta.

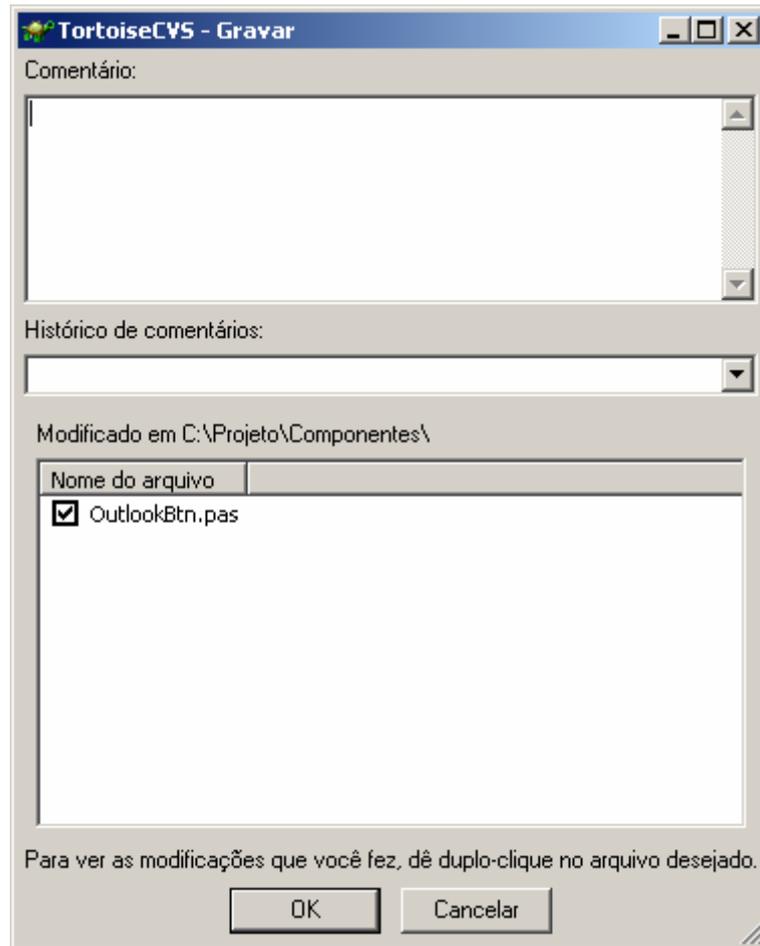


Figura 4 – Tela de gravação de nova versão do item de configuração

O WinCVS (CVSGUI, 2002) é uma ferramenta com a mesmo conceito do TortoiseCVS, com a diferença que esta não é integrado no *Shell* do Windows. A figura 4 apresenta uma tela desta ferramenta.

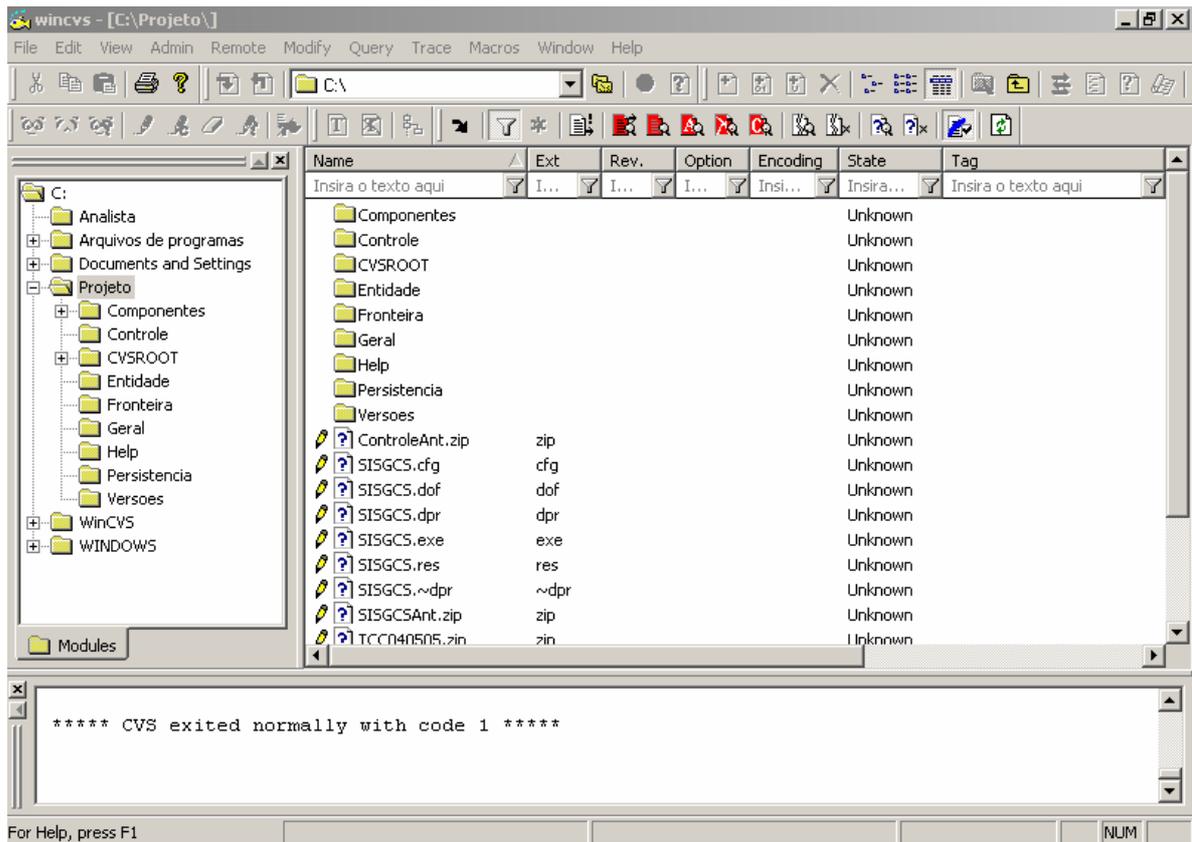


Figura 5 – Tela principal do WinCVS

O Team Coherence é uma ferramenta com licença comercial e as outras duas são estão sobre licença livre. Todas as três ferramentas estudadas permitem o controle de versão dos itens de configuração, criar e recuperar linhas base, controlar e recuperar versões e visualizar diferenças em itens de configuração.

### 3 DESENVOLVIMENTO DO TRABALHO

Nas próximas seções serão apresentados os requisitos da ferramenta e o desenvolvimento da ferramenta.

#### 3.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

Os requisitos funcionais foram elaborados com base no modelo CMMI, funcionalidades comuns percebidas em ferramentas de gerência de configuração do mercado e outros trabalhos pesquisados.

##### 3.1.1 Requisitos funcionais

Os requisitos funcionais que foram implementados na ferramenta são:

- a) cadastro de gerente de projeto. O sistema deve cadastrar os gerentes de projeto;
- b) cadastro de projetos. O sistema deve cadastrar os projetos;
- c) cadastro de clientes. O sistema deve cadastrar os clientes ;
- d) cadastro de analistas de sistemas. O sistema deve cadastrar os analistas;
- e) cadastro de auditores. O sistema deve cadastrar os auditores;
- f) cadastro de tipos de itens de configuração. O sistema deve cadastrar os tipos de itens de configuração;
- g) registro de itens de configuração. O sistema deve registrar os itens de configuração;
- h) registro do pedido de modificação. O sistema deve registrar os pedidos de modificação;
- i) registro da avaliação. O sistema deve permitir registrar a avaliação de um pedido

de modificação;

- j) registro da decisão. O sistema deve registrar a decisão do gerente de projeto para um determinado pedido de modificação;
- k) registro do início do desenvolvimento. O sistema deve permitir que o analista de sistemas registre o início do desenvolvimento de um determinado pedido de modificação;
- l) registro do término do desenvolvimento. O sistema deve permitir que o analista de sistemas registre o término de um desenvolvimento;
- m) registro de auditoria. O sistema de permitir que o auditor registre suas auditorias das linhas base e dos pedidos de modificação;
- n) verificação das diferenças entre os itens de configuração. O sistema deve verificar as diferenças entre as diferentes versões dos itens de configuração;
- o) relatório sobre os tipos de configuração. O sistema de listar todos os tipos de itens de configuração do projeto;
- p) relatório de pedidos de modificação. O sistema deve listar todos os pedidos de modificação gerados para um determinado projeto;
- q) relatório de linhas base. O sistema deve listar todas as linhas base de um determinado projeto;
- r) relatório de itens de configuração. O sistema deve listar as informações sobre a evolução de todos os itens de configuração;
- s) relatório de auditagem. O sistema deve listar as auditorias;
- t) relatório de projetos. O sistema deve apresentar informações relacionadas à configuração do projeto específico, com todas as suas linhas base e itens de configuração.

### 3.1.2 Requisitos não funcionais

Os requisitos não funcionais da ferramenta são:

- a) controle de acesso: o sistema deve controlar o acesso de todos os usuários;
- b) arquivamento em banco de dados relacional: O sistema deve usar o gerenciador de banco de dados MySQL;
- c) facilidade de uso: o sistema deve utilizar recursos gráficos para melhor usabilidade por parte dos usuários finais;

## 3.2 ESPECIFICAÇÃO

A especificação da ferramenta será apresentada através dos diagramas de casos de uso primário e secundário, diagrama de classes, diagrama de seqüência, diagrama de estados e diagrama de entidade-relacionamento. Com exceção do diagrama de entidade-relacionamento, os diagramas serão apresentados seguindo a notação da UML. Assim como as classes implementadas na ferramenta seguirão os conceitos de orientação a objetos.

Os diagramas de casos de uso, seqüência e estados, foram elaborados na ferramenta Rational Rose 2002 versão demo da Rational Rose Corporation. Os diagramas de classes e entidade-relacionamento foram feitos com a ferramenta Power Designer 9 versão demo da Sybase Inc.

### 3.2.1 Diagrama de casos de uso

Nas figuras 6 e 7 são representados os diagramas de casos de uso primários e secundários.

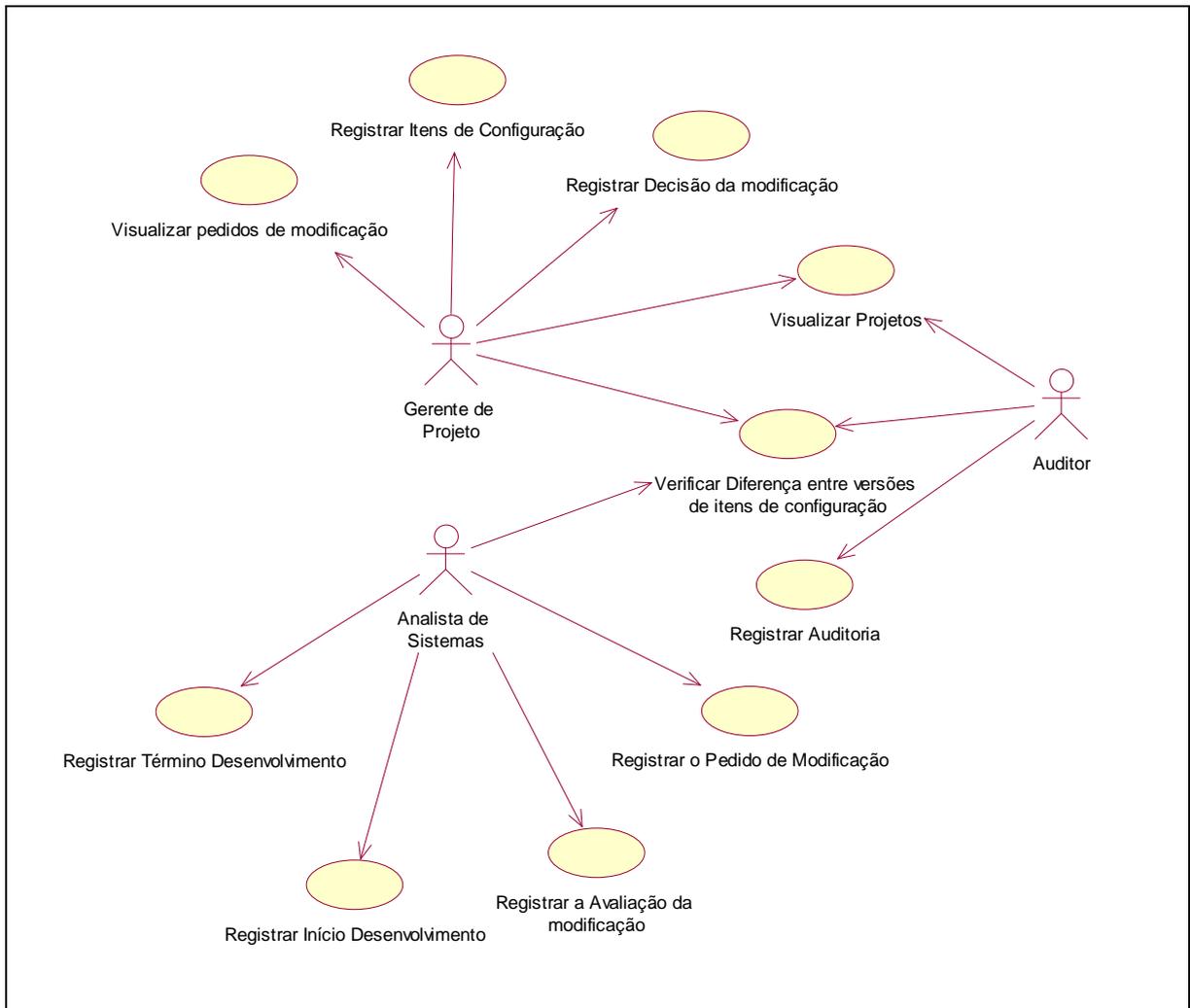


Figura 6 - Diagrama de casos de uso primário

A seguir são descritos brevemente os casos de uso primários:

- a) registrar itens de configuração: o gerente do projeto deve registrar os itens de configuração para que estes possam ser controlados informando o nome físico em disco e sua descrição. Os itens de configuração podem ser incluídos e excluídos e só poderão ser alterados pelo processo de pedido de modificação;
- b) registrar pedido de modificação: o analista de sistemas deve registrar os pedidos de modificação para serem implementados informando o cliente, a data e a descrição da modificação. Os pedidos podem ser incluídos, alterados, excluídos e consultados;
- c) registrar a avaliação da modificação: o analista deve registrar a avaliação da

modificação, descrevendo os aspectos relevantes, grau de dificuldade, tempo estimado e itens de configuração que serão afetados;

- d) registrar a decisão da modificação: o gerente de projeto deve registrar sua decisão informando se a modificação será realizada ou não e uma justificativa de sua decisão;
- e) registrar o início do desenvolvimento: o analista deve registrar o início do desenvolvimento, sendo neste momento feito a saída dos itens de configuração do repositório principal e copiado para a pasta de trabalho do analista responsável;
- f) registrar o término do desenvolvimento: o analista deve registrar o término do desenvolvimento, permitindo assim que seja feita a auditoria da modificação;
- g) registrar auditoria: o auditor deve registrar o resultado dos testes feitos com os novos itens de configuração informando uma descrição da auditoria e o resultado da mesma, aprovado ou reprovado. Se for aprovado é feita a cópia dos itens de configuração alterado para o repositório principal gerando uma nova versão de cada item, caso contrário a situação do pedido volta para “em desenvolvimento” sendo que o analista responsável deve corrigir os problemas encontrados;
- h) verificar diferenças: os usuários podem verificar as diferenças entre os itens de configuração. Deverá apresentar uma tela com a versão atual do item de configuração na linha base e a versão selecionada;
- i) visualizar pedidos de modificação: o gerente de projeto pode visualizar todos os pedidos de modificação registrados emitindo um relatório com descrição do pedido, nome do analista que o registrou, grau de prioridade, esforço estimado e situação do pedido;
- j) visualizar projeto: o gerente de projeto pode visualizar a situação de um determinado projeto emitindo um relatório com as linhas base e seus itens de configuração e o histórico das modificações.

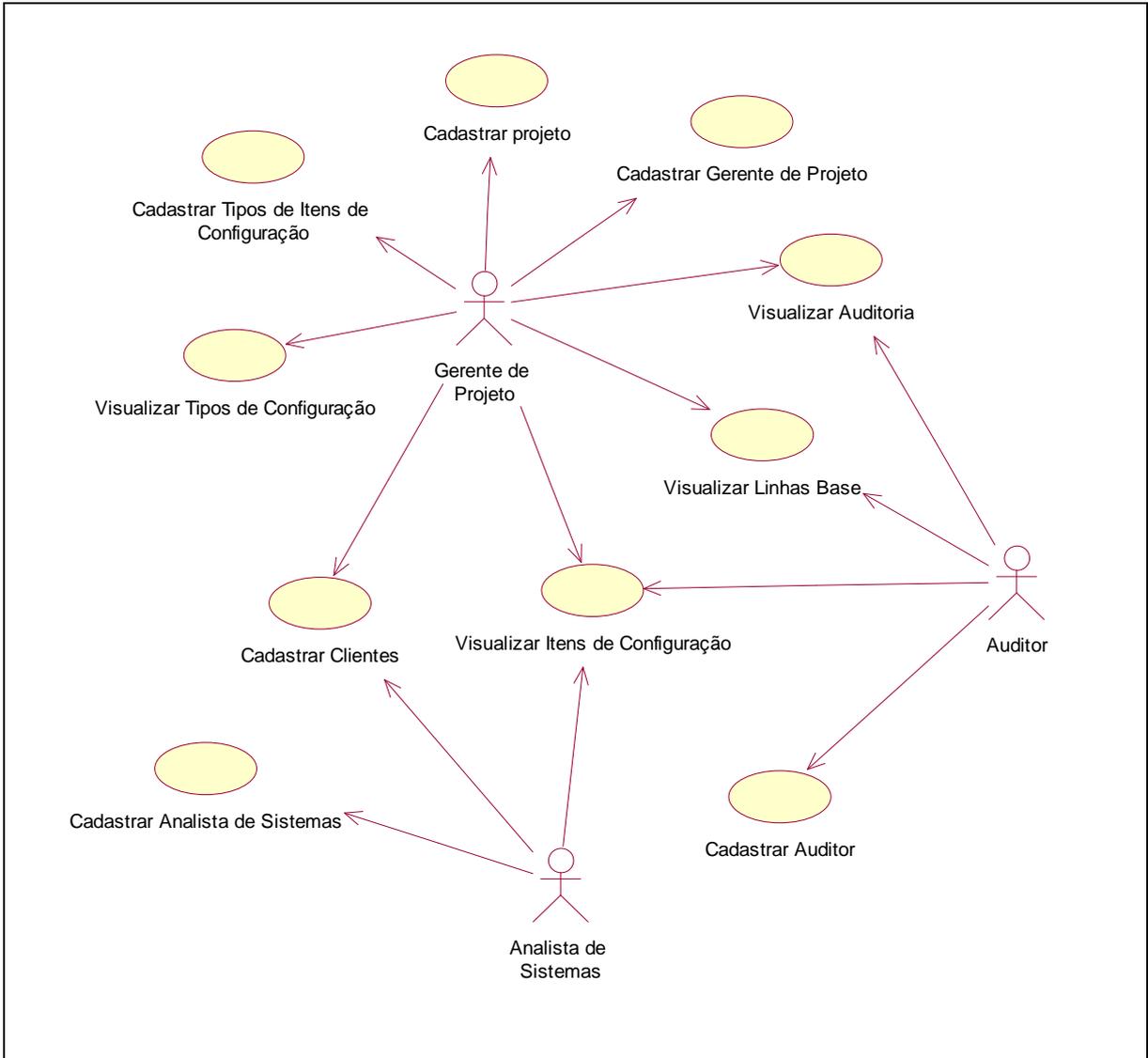


Figura 7 - Diagrama de casos de uso secundários

A seguir são descritos brevemente os casos de uso secundários:

- a) cadastrar gerente de projeto: o gerente de projeto registrar-se para ter acesso a todas as funções da ferramenta informando o seu nome e senha. Os gerentes de projeto podem ser incluídos, alterados, excluídos ou consultados;
- b) cadastrar projeto: o gerente de projeto deve registrar o projeto para identificação na ferramenta, informando o nome do projeto e uma descrição. Os projetos podem ser incluídos, alterados, excluídos ou consultados;
- c) cadastrar analista de sistemas: o analista de sistemas deve registrar-se para ter

acesso às funções que lhe são permitidas pelo sistema informando o seu nome, senha e diretório de trabalho. Os analistas de sistemas podem ser incluídos, alterados, excluídos ou consultados;

- d) cadastrar auditor: o auditor deve registrar-se para ter acesso às funções de sua responsabilidade informando o seu nome e senha. Os auditores podem ser incluídos, alterados, excluídos ou consultados;
- e) cadastrar cliente: o analista de sistemas ou o gerente de projeto devem registrar os clientes de um determinado projeto informando seus dados cadastrais como nome, endereço, bairro, cep, cidade, estado, tipo de pessoa, cpf, cnpj e telefone. Os clientes podem ser incluídos, alterados, excluídos ou consultados;
- f) cadastrar tipos de itens de configuração: o gerente de projeto deve cadastrar os tipos de itens de configuração para classificar os itens de configuração do projeto. Os tipos de itens de configuração podem ser incluídos, alterados, excluídos ou consultados;
- g) visualizar itens de configuração: os usuários da ferramenta podem visualizar os itens de configuração de um determinado projeto emitindo um relatório com o nome dos itens de configuração, a descrição e o histórico de modificações;
- h) visualizar os tipos de itens de configuração: o gerente de projeto pode visualizar os tipos de itens de configuração cadastrados no projeto emitindo um relatório com o nome dos tipos de itens de configuração e o diretório no repositório;
- i) visualizar linhas base: o gerente de projeto e o auditor podem visualizar os dados das linhas base de um determinado projeto emitindo um relatório com os itens de configurações agrupados por tipo de itens de configuração e linhas base;
- j) visualizar auditorias: o gerente de projeto e o auditor podem visualizar todas as auditorias realizadas em um determinado projeto emitindo um relatório com o

nome do auditor, descrição da auditoria, data e o resultado.

### 3.2.2 Diagramas de Classes

A ferramenta foi desenvolvida utilizando o conceito de quatro camadas de classes. A camada de entidade contendo os atributos das classes de negócio, a camada de persistência que realiza a gravação dos dados fisicamente, a camada de controle que integra a camada de entidade e a camada de persistência e por último a camada de fronteira que são as telas de entrada de dados.

A figura 8 apresenta o diagrama de classes da camada de entidade, onde são definidas as classes e os seus atributos. A figura 9 apresenta o diagrama de classes da camada de controle, onde a classes de persistência possui todos os métodos para inserção, alteração, exclusão e recuperação dos dados no banco de dados. Nesta mesma figura são apresentadas as classes de controle, que basicamente possuem os métodos para carregar os objetos com os dados vindos da camada de fronteira e gravar os dados utilizando os métodos da camada de persistência.

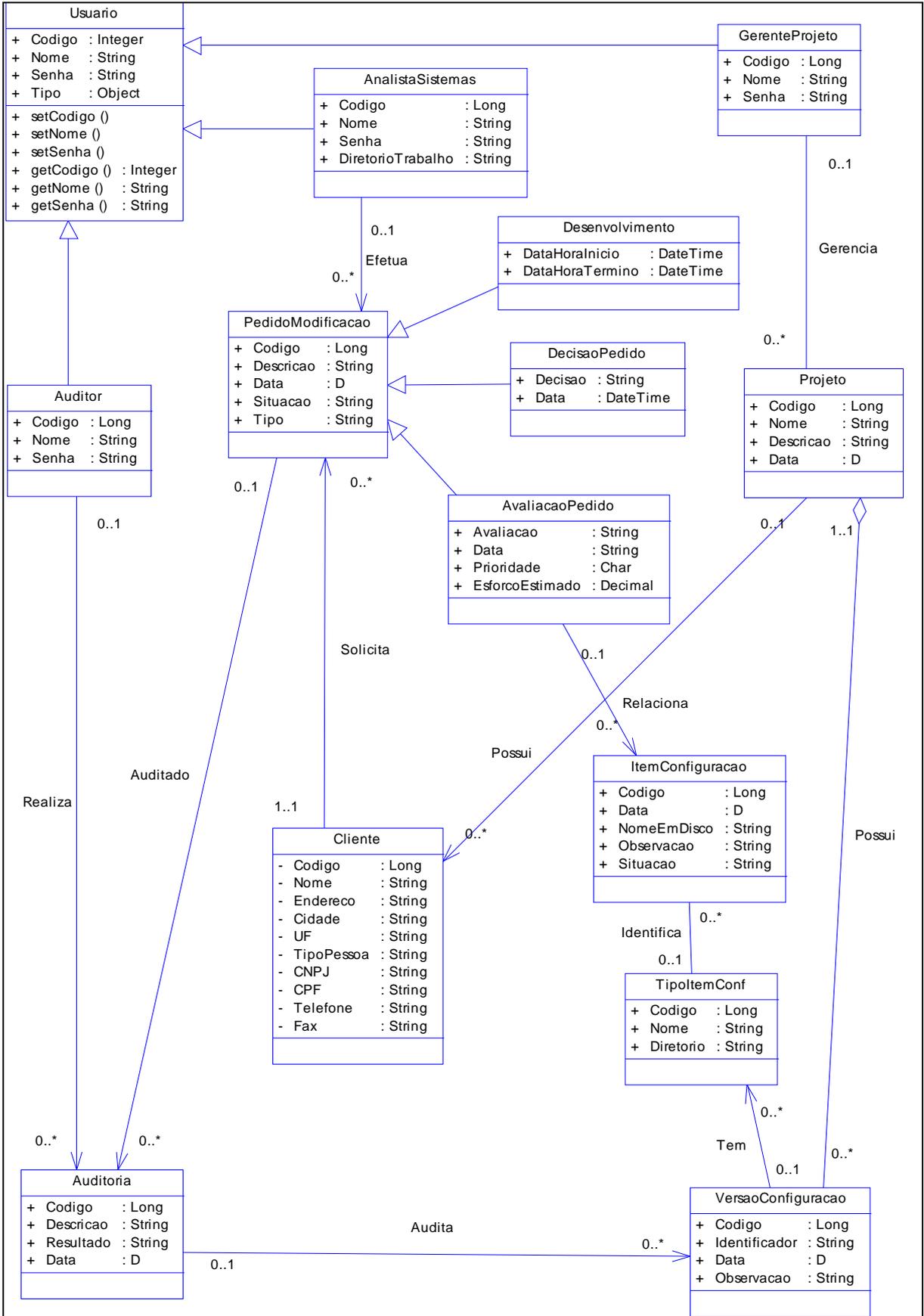


Figura 8 - Diagrama de classes da camada de entidade

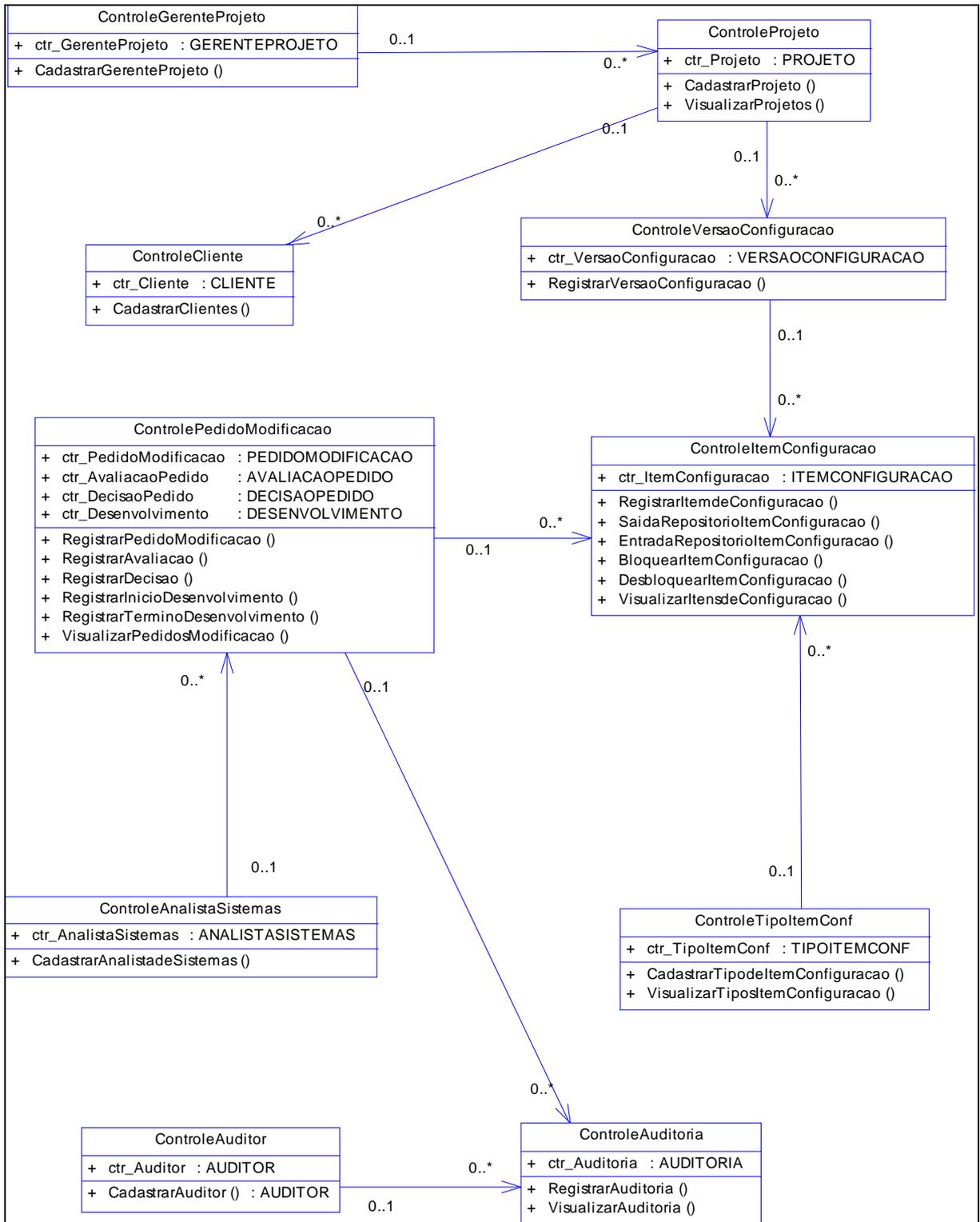


Figura 9 - Diagrama de classes da camada de controle

### 3.2.3 Diagrama de Atividades

Optou-se por fazer o diagrama de atividades apenas do processo de pedido de

modificação, por este possuir o um grande número de atores e classes envolvidas. A figura 10 apresenta o diagrama de atividades do processo de um pedido de modificação.

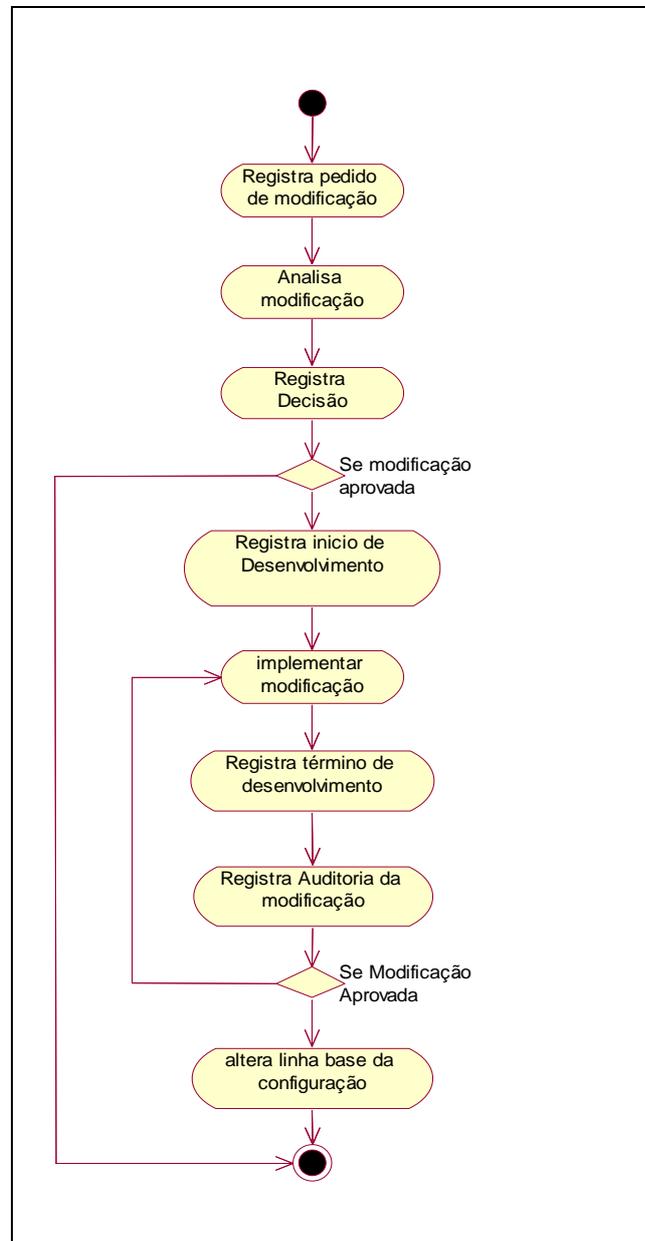


Figura 10 - Diagrama de atividades

O processo de pedido de modificação inicia com o analista de sistemas efetuando o registro do pedido e sua avaliação, informando os itens de configuração envolvidos na modificação. Com base na avaliação do analista de sistemas, o gerente de projeto registra sua decisão. Se for decidido que a modificação será implementada o analista de sistemas registra o inicio do desenvolvimento, sendo neste momento bloqueado os itens de configuração e

copiado os mesmos para a área de trabalho do analista de sistemas.

Concluída a modificação o analista responsável registra o término da implementação. O auditor irá realizar os testes nos itens modificados e registrar sua auditoria. Se nenhum problema foi encontrado o pedido é considerado concluído e os itens de configuração são incluídos no repositório do projeto gerando uma nova versão de cada item alterado.

### 3.2.4 Diagramas de Estados

A figura 11 apresenta o diagrama de estados do pedido de modificação durante o processo de execução de uma determinada modificação.

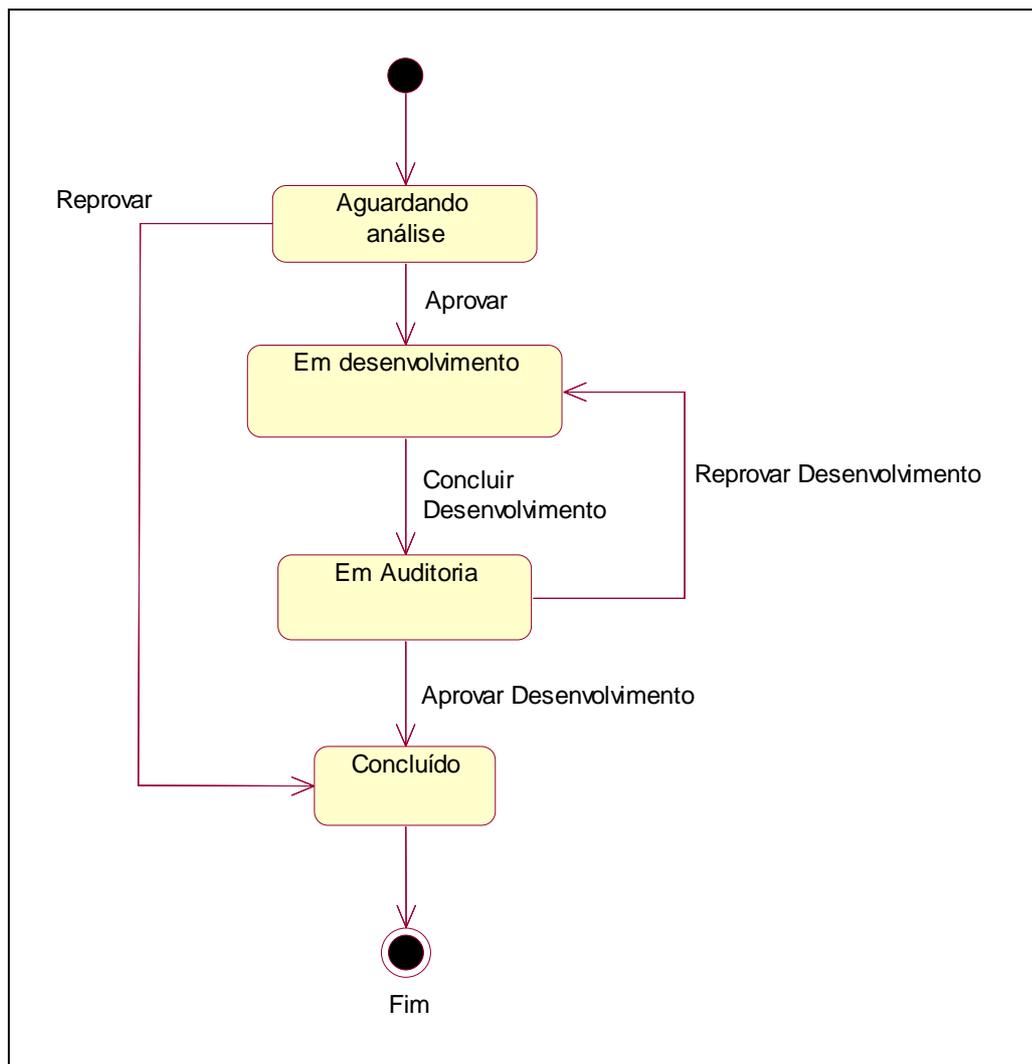


Figura 11 - Diagrama de estados do pedido de modificação

Quando um pedido é registrado este fica com o estado de “aguardando análise”. Se for reprovado seu esta muda para “concluído”, do contrário após o analista registrar o início do desenvolvimento, este passa para o estado de “em desenvolvimento”. Após o registro do término de desenvolvimento, seu estado passa para “em auditoria”. O auditor faz o registro da auditoria e se for reprovada passa novamente para o estado de “em desenvolvimento”, caso contrário passa para o estado de concluído.

A figura 12 apresenta o diagrama de estados dos itens de configuração durante o processo de modificação na linha base do projeto.

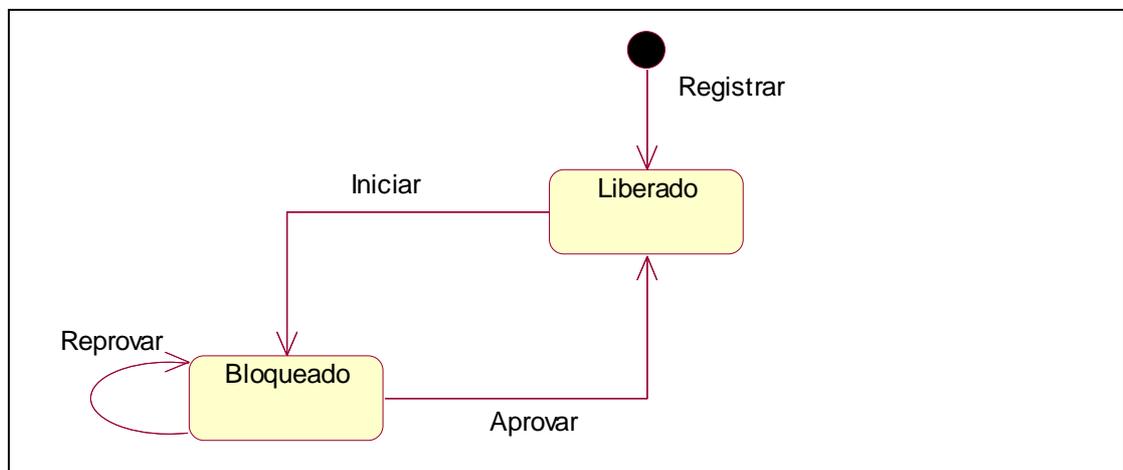


Figura 12 - Diagrama de estados do item de configuração

Para manter o controle do acesso aos itens de configuração é preciso gerenciar os estados dos mesmos. Quando o analista de sistemas registra o início do desenvolvimento, o estado dos itens de configuração que serão modificados passa para “bloqueado”. Após o término do desenvolvimento, o auditor irá registrar a auditoria da modificação. Se for aprovada os itens passam para o estado de liberado.

### 3.2.5 Diagramas de Entidade-Relacionamento

Para um visualizar o armazenamento dos dados, a figura 13 apresenta o DER físico da ferramenta, com todas as tabelas que constituem o banco de dados.

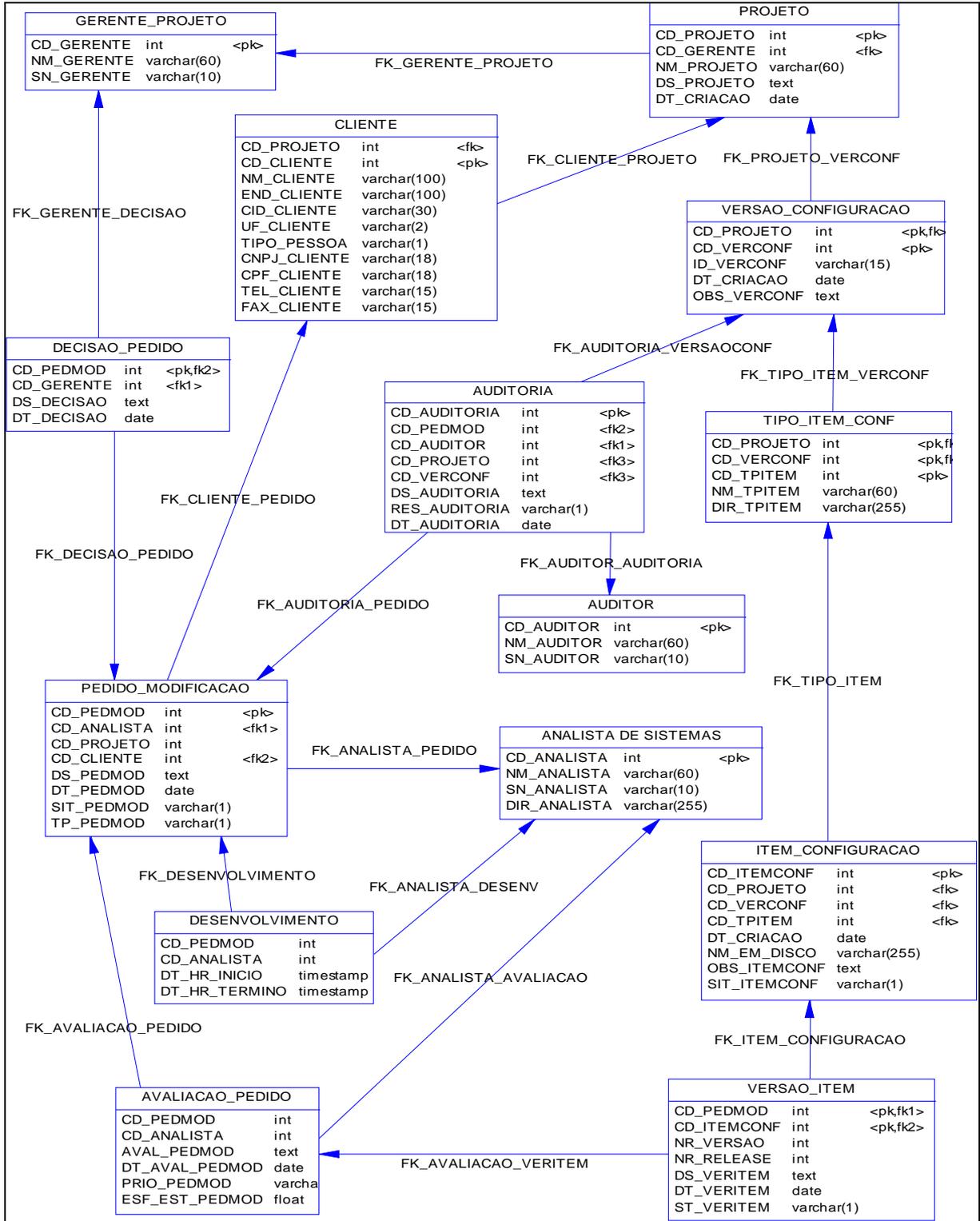


Figura 13 - Diagrama de entidade-relacionamento

Complementando o DER físico, no quadro 2 está representado o dicionário de dados, detalhando cada campo das tabelas. Na descrição constam o nome, descrição, tipo e se o campo é chave primária.

<b>GERENTE_PROJETO</b>			
<b>Nome</b>	<b>Descrição</b>	<b>Tipo</b>	<b>Chave Primária</b>
CD_GERENTE	Código do gerente de projeto	Int	Sim
NM_GERENTE	Nome do gerente de projeto	Varchar(60)	
SN_GERENTE	Senha de acesso do gerente de projeto	Varchar(10)	

<b>PROJETO</b>			
<b>Nome</b>	<b>Descrição</b>	<b>Tipo</b>	<b>Chave Primária</b>
CD_PROJETO	Código do projeto	Int	Sim
CD_GERENTE	Código do gerente de projeto	Int	
NM_PROJETO	Nome do projeto	Varchar(60)	
DS_PROJETO	Descrição do projeto	Text	
DT_CRIACAO	Data de criação do projeto	Date	

<b>CLIENTE</b>			
<b>Nome</b>	<b>Descrição</b>	<b>Tipo</b>	<b>Chave Primária</b>
CD_PROJETO	Código do projeto	Int	
CD_CLIENTE	Código do cliente	Int	Sim
NM_CLIENTE	Nome do cliente	Varchar(100)	
END_CLIENTE	Endereço do cliente	Varchar(100)	
CID_CLIENTE	Cidade do cliente	Varchar(30)	
UF_CLIENTE	Estado do cliente	Varchar(2)	
TIPO_PESSOA	Se cliente é pessoa jurídica ou física	Varchar(1)	
CNPJ_CLIENTE	CNPJ do cliente	Varchar(18)	
CPF_CLIENTE	CPF do cliente	Varchar(18)	
TEL_CLIENTE	Telefone do cliente	Varchar(15)	
FAX_CLIENTE	Fax do cliente	Varchar(15)	

<b>VERSAO_CONFIGURACAO</b>			
<b>Nome</b>	<b>Descrição</b>	<b>Tipo</b>	<b>Chave Primária</b>
CD_PROJETO	Código do projeto	Int	Sim
CD_VERCONF	Código da versão da configuração	Int	Sim
ID_VERCONF	Identificador da versão da configuração	Varchar(15)	
DT_CRIACAO	Data de criação da versão da configuração	Date	
OBS_VERCONF	Comentário da versão da configuração	Text	

<b>TIPO_ITEM_CONF</b>			
<b>Nome</b>	<b>Descrição</b>	<b>Tipo</b>	<b>Chave Primária</b>
CD_PROJETO	Código do projeto	Int	Sim
CD_VERCONF	Código da versão da configuração	Int	Sim
CD_TPITEM	Código do tipo de item de configuração	Int	Sim
NM_TPITEM	Nome do tipo de item de configuração	Varchar(60)	
DIR_TIPITEM	Diretório contendo os itens de configuração do tipo de item de configuração	Varchar(255)	

<b>ITEM_CONFIGURACAO</b>			
<b>Nome</b>	<b>Descrição</b>	<b>Tipo</b>	<b>Chave Primária</b>
CD_PROJETO	Código do projeto	Int	
CD_VERCONF	Código da versão da configuração	Int	
CD_TPITEM	Código do tipo de item de configuração	Int	
CD_ITEMCONF	Código do item de configuração	Int	Sim
DT_CRIACAO	Data de criação do item de configuração	Date	
NM_EM_DISCO	Nome do item de configuração em disco	Varchar(255)	
OBS_ITEMCONF	Observação do item de configuração	Text	
SIT_ITEMCONF	Situação do item de configuração 0 - desbloqueado 1 - bloqueado		

<b>VERSÃO_ITEM</b>			
<b>Nome</b>	<b>Descrição</b>	<b>Tipo</b>	<b>Chave Primária</b>
CD_PEDMOD	Código do pedido de modificação	Int	Sim
CD_ITEMCONF	Código do item de configuração	Int	Sim
NR_VERSAO	Número da versão do item de configuração	Int	
NR_RELEASE	Número do release da versão do item de configuração	Varchar(60)	
DS_VERITEM	Descrição da versão do item de configuração	Text	
DT_VERITEM	Data em que a versão do item de configuração foi criada	Date	
ST_VERITEM	Situação da versão do item de configuração 0 – registrado 1 – em desenvolvimento	Varchar(1)	

<b>PEDIDO_MODIFICACAO</b>			
<b>Nome</b>	<b>Descrição</b>	<b>Tipo</b>	<b>Chave Primária</b>
CD_PEDMOD	Código do pedido de modificação	Int	Sim
CD_PROJETO	Código do projeto	Int	
CD_ANALISTA	Código do analista de sistemas que registrou o pedido de modificação	Int	
CD_CLIENTE	Código do cliente que solicitou a modificação	Int	
DS_PEDMOD	Descrição do pedido de modificação	Text	
DT_PEDMOD	Data de registro do pedido de modificação	Date	
SIT_PEDMOD	Situação do pedido de modificação 0 – em análise 1 – em desenvolvimento 2 – em auditoria 3 – concluído	Varchar(1)	
TP_PEDMOD	Tipo do pedido de modificação 0 – Corretiva 1 – Adaptativa 2 – Nova funcionalidade	Varchar(1)	

<b>DECISAO_PEDIDO</b>			
<b>Nome</b>	<b>Descrição</b>	<b>Tipo</b>	<b>Chave Primária</b>
CD_PEDMOD	Código do pedido de modificação	Int	Sim
CD_GERENTE	Código do gerente de projeto	Int	
DS_DECISAO	Descrição da decisão	Text	
DT_DECISAO	Data da decisão	Date	

<b>AVALIACAO_PEDIDO</b>			
<b>Nome</b>	<b>Descrição</b>	<b>Tipo</b>	<b>Chave Primária</b>
CD_PEDMOD	Código do pedido de modificação	Int	Sim
CD_ANALISTA	Código do analista de sistemas	Int	
AVAL_PEDMOD	Avaliação do pedido de modificação	Text	
DT_AVAL_PEDMOD	Data da avaliação	Date	
PRIO_PEDMOD	Prioridade do pedido de modificação 0 – baixa 1 – média 2 – alta	Varchar(1)	
ESF_EST_PEDMOD	Esforço estimado para desenvolvimento do pedido de modificação		

<b>DESENVOLVIMENTO</b>			
<b>Nome</b>	<b>Descrição</b>	<b>Tipo</b>	<b>Chave Primária</b>
CD_PEDMOD	Código do pedido de modificação	Int	Sim
CD_ANALISTA	Código do analista de sistemas	Int	
DT_HR_INICIO	Data e hora de início do desenvolvimento	TimeStamp	
DT_HR_TERMINO	Data e hora de término do desenvolvimento	TimeStamp	

<b>AUDITOR</b>			
<b>Nome</b>	<b>Descrição</b>	<b>Tipo</b>	<b>Chave Primária</b>
CD_AUDITOR	Código do auditor	Int	Sim
NM_AUDITOR	Nome do auditor	Varchar(60)	
SN_AUDITOR	Senha de acesso do auditor	Text	

<b>AUDITORIA</b>			
<b>Nome</b>	<b>Descrição</b>	<b>Tipo</b>	<b>Chave Primária</b>
CD_AUDITORIA	Código da auditoria	Int	Sim
CD_PEDMOD	Código do pedido de modificação	Int	
CD_AUDITOR	Código do auditor	Int	
CD_PROJETO	Código do projeto	Int	
CD_VERCONF	Código da versão da configuração	Int	
DS_AUDITORIA	Descrição da auditoria	Text	
RES_AUDITORIA	Resultado da auditoria 0 – Aprovado 1 – Reprovado	Varchar(1)	
DT_AUDITORIA	Data da auditoria	Date	

Quadro 2 – Dicionário de dados

### 3.2.6 IMPLEMENTAÇÃO

Serão apresentadas a seguir as técnicas e ferramentas utilizadas, assim como a estrutura do código e a operacionalidade da ferramenta.

### 3.2.7 Técnicas e Ferramentas Utilizadas

Para realizar a implementação da ferramenta utilizou-se a linguagem de programação *Object Pascal* no ambiente de desenvolvimento Borland Delphi, na versão 7.0 e o sistema gerenciador de banco de dados *MySQL* versão 4.0. Para a construção da interface, foram utilizados os componentes visuais que acompanham o ambiente Delphi.

O acesso ao banco de dados foi implementado com o pacote *open-source* *Zeoslib*, que provém acesso ao banco de dados através de uma DLL. Para fazer os relatórios utilizou-se o gerador de relatórios *ReportBuilder* da Metaphors Digital.

Para armazenar as versões dos itens de configuração foi seguido o formato de arquivo pela ferramenta TortoiseCVS. A seguir será apresentada a estrutura dos códigos fonte da ferramenta.

### 3.2.8 Estrutura do código

A ferramenta foi implementada com o conceito de quatro camadas que são:

- a) persistência: esta camada possui todos os procedimentos para manutenção da base de dados do sistema;
- b) controle: esta camada define os procedimentos de integração entre a camada de fronteira e a camada de persistência;
- c) entidade: define todas as classes de negócio da ferramenta;
- d) fronteira: esta camada possui todas as telas para entrada de dados da ferramenta.

Foi escolhido este método para que futuras alterações nos requisitos não funcionais, sejam facilmente implementados, uma vez que a camada de fronteira não está diretamente dependente da camada de persistência. Assim se houver a necessidade de trocar o banco de dados, é necessário apenas mudar a camada de persistência sem ter que fazer modificações

nas outras camadas.

### 3.2.8.1 Camada de persistência

Esta camada foi implementada em uma única classe, e contém todos os procedimentos para manutenção da base de dados. Para realizar a comunicação com o banco de dados foi utilizado o pacote *zeoslib*. O componente responsável pela conexão com o banco de dados é *tzconnection*. A quadro 3 apresenta a implementação dos procedimentos para a conexão com o banco de dados.

```
procedure Persistencia.ConectaDB;  
begin  
  try  
    BancoDados := TZConnection.Create(Application);  
    ExecutaSQL := TZSQLProcessor.Create(BancoDados);  
    SeleccionaDados := TZQuery.Create(BancoDados);  
    ExecutaSQL.Connection := BancoDados;  
    SeleccionaDados.Connection := BancoDados;  
  
    BancoDados.Database := 'SIGGCS';  
    BancoDados.HostName := '127.0.0.1';  
    BancoDados.Protocol := 'mysql-4.0';  
    BancoDados.User := 'root';  
  
    BancoDados.Connect;  
  except  
    ShowMessage('Não foi possível conectar com Banco de Dados');  
  end;  
end;
```

Quadro 3 – Código de conexão com o banco de dados

Para realizar a conexão com o banco de dados, é necessário carregar os seguintes atributos do componente *tzconnection*:

- a) *database*: neste atributo é informado o nome do banco de dados;
- b) *hostname*: é o endereço do servidor do banco de dados;
- c) *protocol*: é o nome do *driver* do banco de dados utilizado;
- d) *user*: é o nome do usuário do banco de dados.

Após informado este atributos é preciso chamar o método connect, que realizará a conexão entre a aplicação e o banco de dados. Para fazer uma atualização nas tabelas foi utilizado o componente *tzsqlprocessor*. O quadro 4 mostra um exemplo de como é realizada a manutenção de uma tabela.

```

function Persistencia.InsereGerenteProjeto(
    obj_GerenteProjeto: GerenteProjeto): boolean;
begin
    result := true;
    ExecutaSQL.Script.Clear;
    ExecutaSQL.Script.Add('INSERT INTO gerente_projeto (NM_GERENTE, SN_GERENTE)');
    ExecutaSQL.Script.Add('VALUES ('''+obj_GerenteProjeto.getNome+''','''+
        obj_GerenteProjeto.getSenha+''')');

    try
        ExecutaSQL.Execute;
    except
        raise EZSQLException.Create('Não foi possível inserir este Gerente de Projeto');
        result := false;
    end;
end;

```

Quadro 4 – Código para manutenção de tabelas

Neste é necessário informar o código SQL que se deseja executar e chamar o método *execute*. Para todas as tabelas existe métodos para inclusão, alteração, exclusão e recuperação de registros.

### 3.2.8.2 Camada de controle

Esta camada realiza a comunicação entre a camada de persistência e a camada de fronteira. Para todas as classes da camada de entidade, foi implementado uma classe de controle. A classe de controle basicamente recebe os dados informado na camada de fronteira e carrega um objeto da classe de entidade relativo, e realiza a ação informada na camada de fronteira. O quadro 5 apresenta o código da camada de controle para a entidade de gerente de projeto.

```

procedure ControleGerenteProjeto.CarregaObjetoGerenteProjeto(pNome, pSenha : String);
begin
    ctr_GerenteProjeto.setNome(pNome);
    ctr_GerenteProjeto.setSenha(pSenha);
end;

function ControleGerenteProjeto.ValidaGerenteProjeto(pOperacao: TOperacao) : boolean;
begin
    result := False;
    case pOperacao of
        opIncluir : Begin
            if ctr_GerenteProjeto.getNome = '' then
                ShowMessage('Digite o nome do Gerente de Projeto')
            else
                result := obj_Persistencia.InsereGerenteProjeto(ctr_GerenteProjeto);
            end;
        opAlterar : result := obj_Persistencia.AlterarGerenteProjeto(ctr_GerenteProjeto);
        opExcluir : if MessageDlg('Excluir este Gerente de Projeto ?', mtWarning, [mbYes, mbNo], 0) = mrYes then
            result := obj_Persistencia.ExcluiGerenteProjeto(ctr_GerenteProjeto);
    end;
end;

```

Quadro 5 – Código da camada de controle da entidade gerente de projeto

A camada de fronteira da classe gerente de projeto utiliza o método para *CarregaObjetoGerenteProjeto()* para carregar o objeto de negócio e o método *ValidaGerenteProjeto()* que irá executar a operação selecionada da camada de fronteira.

### 3.2.8.3 Camada de entidade

A camada de entidade possui a definição de todas as classes descritas no diagrama de classe. O quadro 6 apresenta a definição da classe gerente de projeto.

```

Type
GerenteProjeto = class
private
    Codigo : Longint;
    Nome : String;
    Senha : String;
public
    procedure setCodigo(pCodigo : Longint);
    procedure setNome(pNome : String);
    procedure setSenha(pSenha : String);
    Function getCodigo : Longint;
    Function getNome : String;
    Function getSenha : String;
end;

```

Quadro 6 – Código da definição da Classe de gerente de projeto

#### 3.2.8.4 Camada de fronteira

O objetivo desta camada é disponibilizar as telas para entrada de dados da ferramenta. Para isso foi criada uma tela de edição padrão para efetuar os cadastros. Basicamente a tela de edição padrão define o comportamento das telas de cadastro. A figura 14 apresenta a tela de edição padrão.

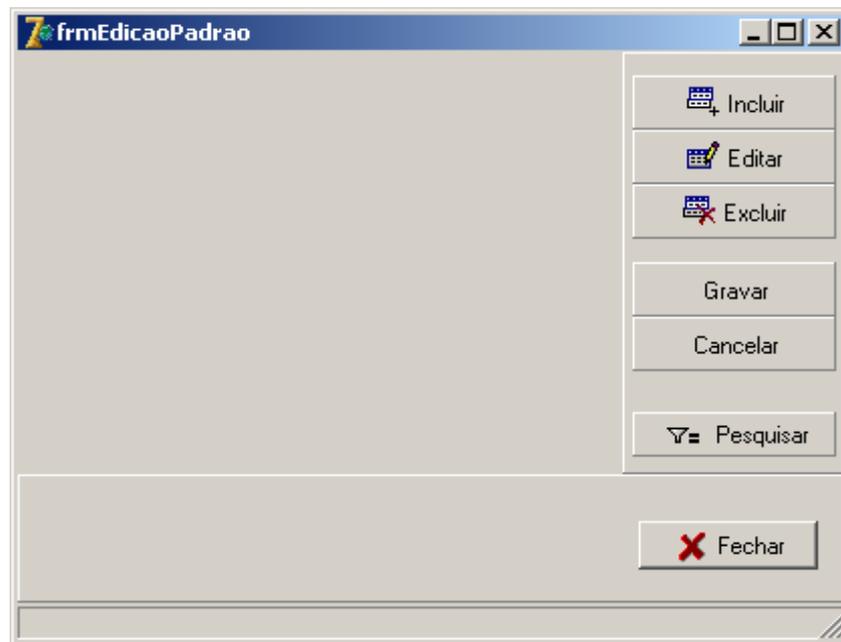


Figura 14 – Tela de edição padrão

Assim todas as telas de cadastro foram herdadas desta tela, facilitando a implementação da ferramenta. A operação principal desta tela, é o botão gravar. É nele que está implementado a comunicação da camada de fronteira com a camada de controle. O quadro 7 apresenta o código do botão gravar da tela de cadastro de gerente de projeto.

```

procedure TfrmEdicaoGerenteProjeto.btnGravarClick(Sender: TObject);
begin
  with obj_ControleGerenteProjeto do
    Begin
      CarregaObjetoGerenteProjeto(edNome.Text,edSenha.Text);
      if not ValidaGerenteProjeto(self.getOperacaoAtual) then
        Begin
          self.FocusParaPrimeiroControle;
          exit;
        end;
      end;
    end;
inherited;
end;

```

Quadro 7 – Código do botão gravar da tela de gerente de projeto

### 3.2.9 Formato do arquivo para armazenamento de diferenças

Baseado na ferramenta de estudo TortoiseCVS, foi utilizado o formato de arquivo para armazenar as diferenças entre as versões dos itens de configuração. Este formato é composto por um conjunto de *tags* que determinam as modificações que devem ser feitas a partir da versão atual de um determinado item de configuração, para que seja possível recuperar o conteúdo de uma versão anterior. Essas *tags* têm o seguinte formato:

- a) <ADD> - adiciona uma linha no arquivo
- b) <TROCA> <Lx> *arg* – troca o conteúdo de *arg1* da linha *x*
- c) <DELE> <Lx> - exclui a linha *x*

Para exemplificar, o quadro 8 apresenta o conteúdo da versão anterior de um determinado item de configuração e o conteúdo da versão atual.

Ln	Arquivo da versão Anterior	Ln	Arquivo da versão Atual
1	Trabalho de conclusão de curso	1	Trabalho de conclusão de curso
2	FURB	2	Centro de ciências exatas
3	BCC	3	BCC
4		4	Prof Everaldo
5	Diagrama de classes	5	Diagrama de classes
6	Classe Cliente		

Quadro 8 – Exemplo de arquivo com versão anterior e atual

A idéia é criar uma seqüência de comandos de forma que se aplicado na versão atual do item de configuração, o resultado seja a versão anterior deste mesmo item. Os comandos referentes ao exemplo do quadro 8 são apresentados no quadro 9.

```
<TROCA> <L2> FURB
<DELE> <L4>
<ADD> Classe Cliente
```

Quadro 9 – Comandos resultantes

O arquivo resultante é composto por um cabeçalho e janelas de conteúdo. O cabeçalho contém as versões presente no arquivo, o quadro 10 mostra um exemplo.

```
<HIST_VERSAO=1.1>
<HIST_VERSAO=1.0>
```

Quadro 10 – Cabeçalho do arquivo de versões

As janelas de conteúdo possuem dados para identificar cada versão presente no arquivo. Sendo assim cada janela é composta pelo identificador da versão, o autor, a data e a descrição da versão. O quadro 11 apresenta um exemplo de uma janela de conteúdo.

```
[VERSAO=1.0]
[AUTOR= Iam Sommerville]
[DATA=29/5/2005]
[DESCRICAO=Inserido no projeto]

[@BOF]
<TROCA> <L2> FURB
<DELE> <L4>
<ADD> Classe Cliente

[@EOF]
```

Quadro 11 – Janela de versão do arquivo de versões

O quadro 12 apresenta o arquivo de diferenças contendo todos os seus elementos.

```

<HIST_VERSAO=1.1>
<HIST_VERSAO=1.0>

[VERSAO=1.0]
[AUTOR=Iam Sommerville]
[DATA=29/5/2005]
[DESCRICAO=Inserido no projeto]

[@BOF]
<TROCA> <L2> FURB
<DELE> <L4>
<ADD> Classe Cliente
[@EOF]

[VERSAO=1.1]
[AUTOR=Iam Sommerville]
[DATA=29/5/2005]
[DESCRICAO=Modificação para entrega do TCC]

[@BOF]
Trabalho de conclusão de curso
Centro de ciências exatas
BCC
Prof Everaldo
Diagrama de classes
[@EOF]

```

Quadro 12 – Arquivo de versões completo

### 3.2.10 Operacionalidade da ferramenta

A ferramenta implementada, tem como objetivo gerenciar os artefatos que compõe um projeto de software, usando o conceito de gerência de configuração e assim auxiliando a equipe de desenvolvimento a alcançar as práticas e metas descritas no CMMI.

Cada usuário tem sua função e deve estar cadastrado no sistema. O gerente de projeto é

o usuário com acesso total a todas as funcionalidades da ferramenta. Sendo assim este precisa estar cadastrado para que um projeto possa ser iniciado. Para isso pode-se *logar* no sistema com um usuário em branco e acessar a tela de cadastro de gerente de projeto pelo menu principal na opção arquivos e sub-opção gerente de projeto. A figura 15 mostra a tela onde é cadastrado o gerente de projeto.

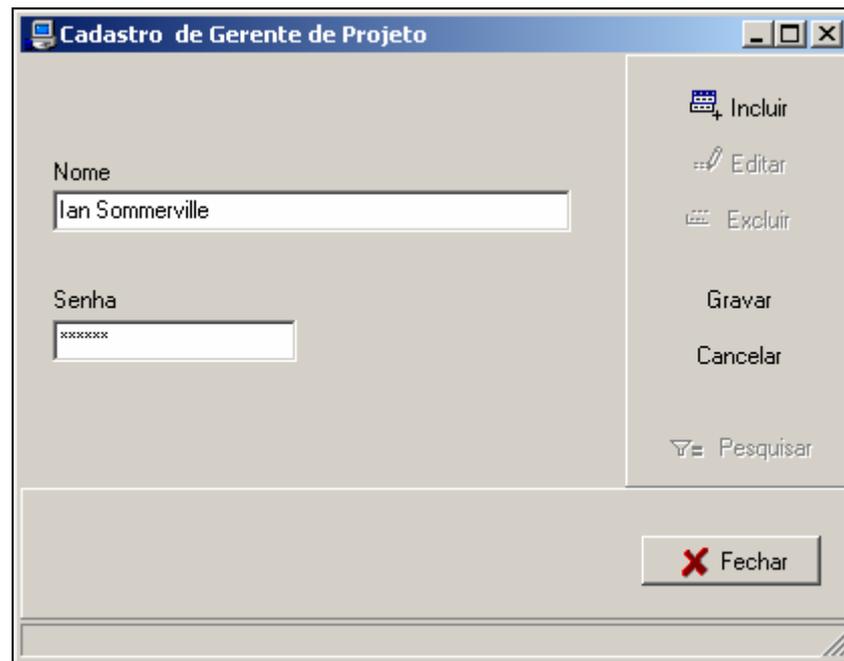
A imagem mostra uma janela de software com o título "Cadastro de Gerente de Projeto". No lado esquerdo, há dois campos de entrada: "Nome" com o texto "Ian Sommerville" e "Senha" com caracteres ocultos por "xxxxxx". No lado direito, há um menu vertical com as opções "Incluir", "Editar", "Excluir", "Gravar", "Cancelar" e "Pesquisar". No canto inferior direito, há um botão "Fechar" com um ícone de uma cruz vermelha.

Figura 15 – Tela de cadastro de gerente de projeto

Para iniciar o cadastro, pressiona-se o botão incluir, informa o nome e senha do gerente de projeto e pressiona o botão gravar para confirmar o registro. Feito isto, é necessário reiniciar a ferramenta para acessar a ferramenta como gerente de projeto.

Ao reiniciar, aparecerá a tela da figura 16, para efetuar o *login* na ferramenta.

A imagem mostra uma janela de software com o título "Login". No topo, há o texto "SIGGCS 0.10" e "TCC André Bohn 2005/1". Abaixo, há dois campos de entrada: "Usuario" com o texto "Ian Sommerville" e "Senha" com caracteres ocultos por "xxxxxx". No canto inferior direito, há dois botões: "OK" com um ícone de uma seta verde e "Cancelar" com um ícone de uma cruz vermelha.

Figura 16 – Tela de login

Para acessar o sistema é preciso selecionar o usuário e digitar a senha. Agora é preciso que o gerente de projeto registre os elementos básicos como projeto, versão da configuração, tipos de itens de configuração e itens de configuração do projeto. Para acessar a tela de cadastro de projeto, é usado a opção arquivos e sub-opção cadastro de projeto. A figura 17, apresenta a tela para cadastro de projeto.

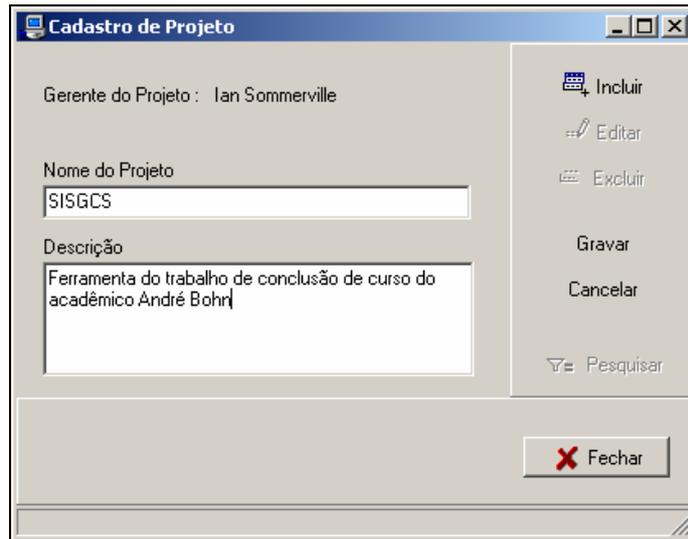


Figura 17 – Tela de cadastro de projeto

Após o cadastramento do projeto, é necessário cadastrar a versão da configuração inicial do projeto. Para acessar o cadastro, selecionar a opção operações e sub-opção versão da configuração. A figura 18 mostra como realizar este registro.

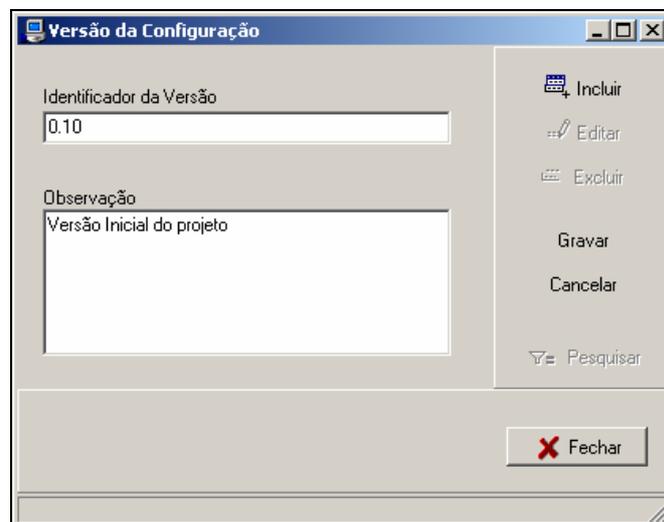


Figura 18 – Tela de cadastro de versão de configuração

Feito isto, agora é necessário registrar os tipos de itens de configuração, para isto é selecionado a opção arquivos e sub-opção tipos de item de configuração. A figura 19 apresenta a referida tela.

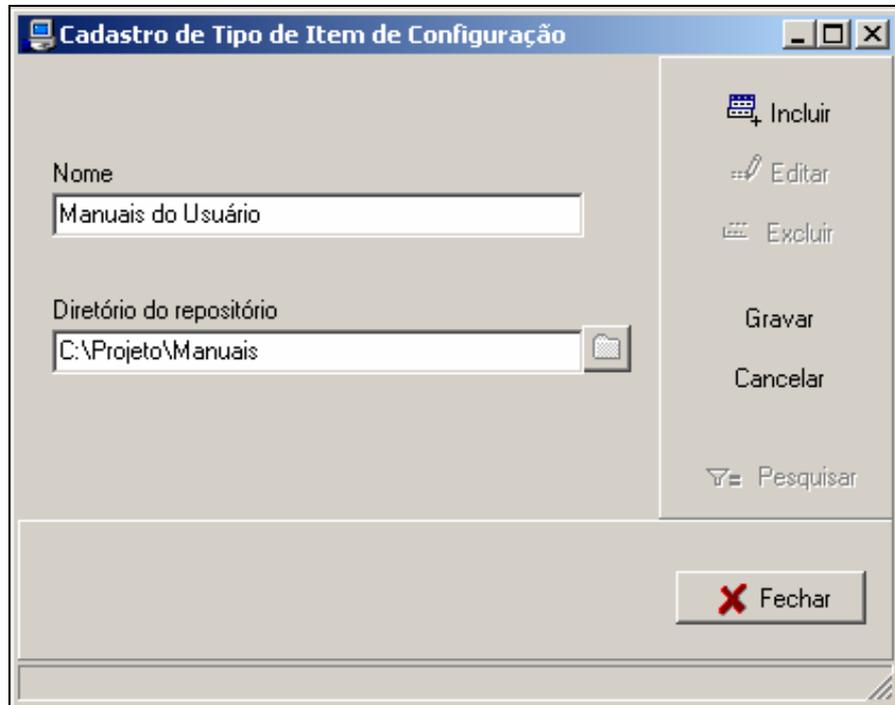


Figura 19 – Tela de cadastro de Tipos de item de configuração

O diretório do repositório, é o caminho de onde se encontram os arquivos do tipo de item de configuração sendo cadastrado. No exemplo da figura 19, é cadastrado o tipo '*manuais do usuário*', e os arquivos poderão ser encontrados no diretório *C:\Projeto\Manuais*. Esta informação será relevante no momento em que o gerente de projeto for registrar os itens de configuração do projeto. A próxima ação será registrar os itens de configuração do projeto. Esta etapa é feita na tela principal do sistema (figura 20).

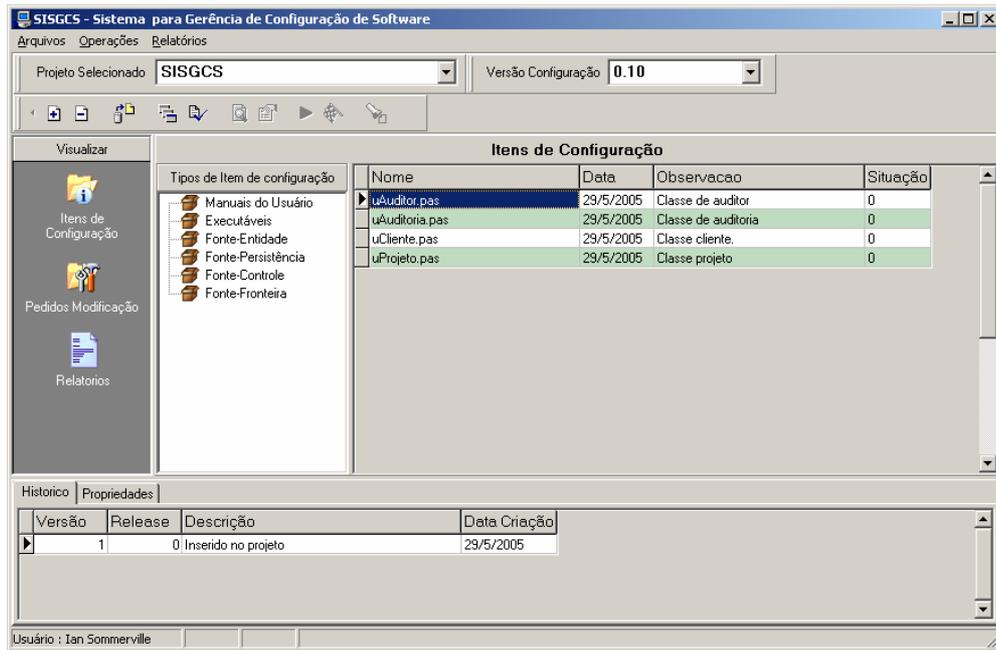


Figura 20 – Tela principal do sistema

Nesta tela é feito todo o processo de gerência de configuração. Para adicionar itens de configuração ao projeto é necessário selecionar o ícone *itens* na área de acesso rápido *visualizar*. Será apresentado os tipos de item de configuração onde é preciso selecionar o tipo de item de configuração que será adicionado. Após isso, pressiona o botão com o símbolo de adição, que se encontra acima da área central da tela. A figura 21 apresenta a tela para registro de item de configuração.



Figura 21 – Tela de registro de item de configuração

As informações necessárias são o nome do arquivo referente ao item de configuração e uma observação para identificar o item de configuração. Após adicionar o item de configuração este aparecerá na tela principal ao lado dos tipos de item de configuração. Ao selecionar um item de configuração, pode-se visualizar o histórico de modificações na aba histórico que se encontra na parte inferior da tela principal, conforme figura 20.

Após registrar todos os itens de configuração o sistema está pronto para que os processo de controle de mudança e controle de versão possa ser usado na ferramenta. Para realizar as operações sobre os itens de configuração, é necessário cadastrar os outros usuários que farão parte do projeto. O analista de sistemas, que conforme o diagrama de caso de uso registrará o pedido de modificação dos clientes, deverá estar cadastrado. A figura 22 apresenta a tela de cadastro de analista de sistemas.

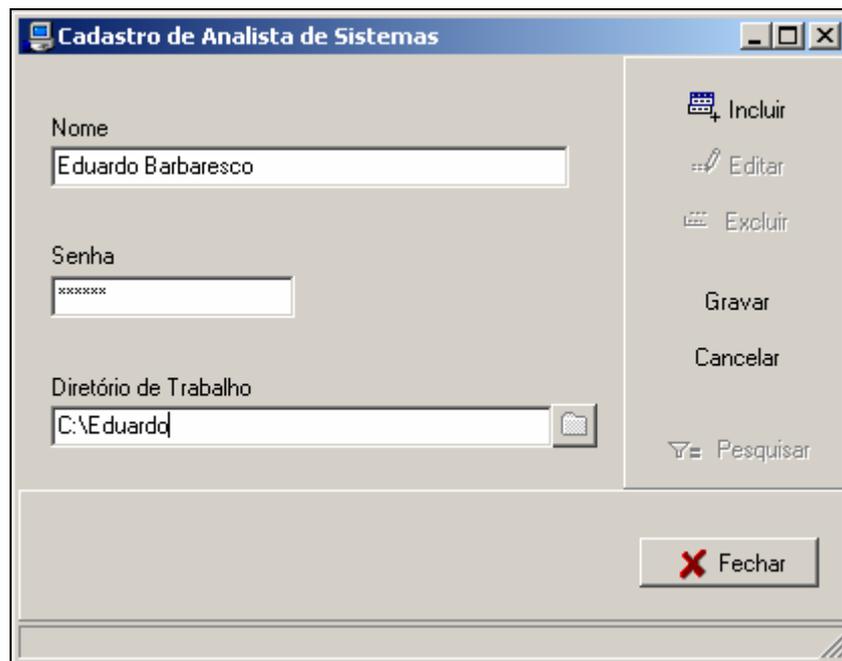
A imagem mostra uma janela de software intitulada "Cadastro de Analista de Sistemas". O formulário contém três campos de entrada: "Nome" com o texto "Eduardo Barbaresco", "Senha" com caracteres ocultos por pontos, e "Diretório de Trabalho" com o caminho "C:\Eduardo". À direita dos campos, há um menu de ações com ícones e o texto "Incluir", "Editar", "Excluir", "Gravar", "Cancelar" e "Pesquisar". Um botão "Fechar" com um ícone de uma cruz vermelha está na parte inferior direita da janela.

Figura 22 – Tela de cadastro de analista de sistemas

Esta tela é semelhante ao cadastro de gerente de projeto, com a diferença que neste é informado o *diretório de trabalho* que o local onde os itens de configuração serão copiados no momento que o pedido de modificação for aprovado para implementação.

Para realizar o registro das auditorias na versão da configuração e nas implementações

dos pedidos de modificação, o auditor também deve estar cadastrado. A tela de cadastro de auditor é ilustrada na figura 23.

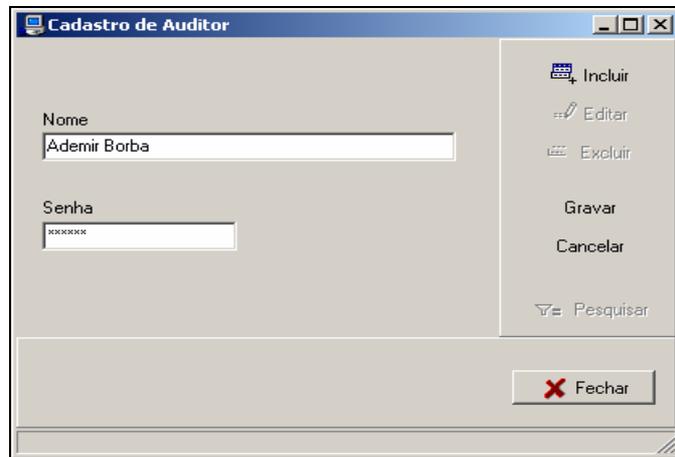


Figura 23 – Tela de cadastro de auditor

Esta tela é idêntica a tela de cadastro de gerente de projeto, sendo que nesta informa-se os auditores do projeto. Para registrar os pedidos de modificação, é necessário que os clientes do projeto sejam cadastrados e isto é feito acessando no menu principal a opção arquivos e sub-opção clientes. A figura 24 apresenta a tela de cadastro de clientes. Esta tela é usada para registrar os clientes do projeto, sendo informado seus dados básicos. O cliente será informado no momento em que o analista de sistemas estiver registrando um pedido de modificação.

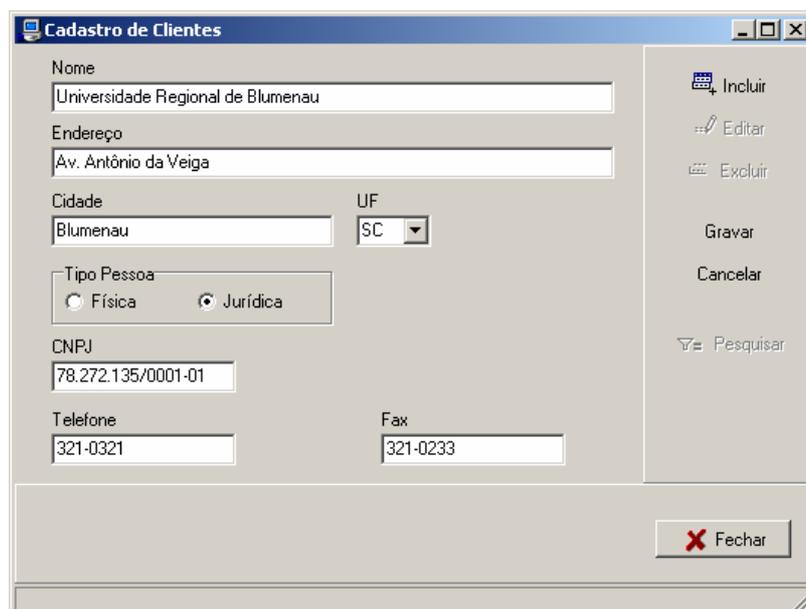


Figura 24 – Tela de cadastro de clientes

Para efetuar uma modificação na linha base ou versão da configuração, é iniciado com o registro de um pedido de modificação por parte do cliente. Este registro é feito na tela principal usando a visualização de pedidos de modificação (figura 25).

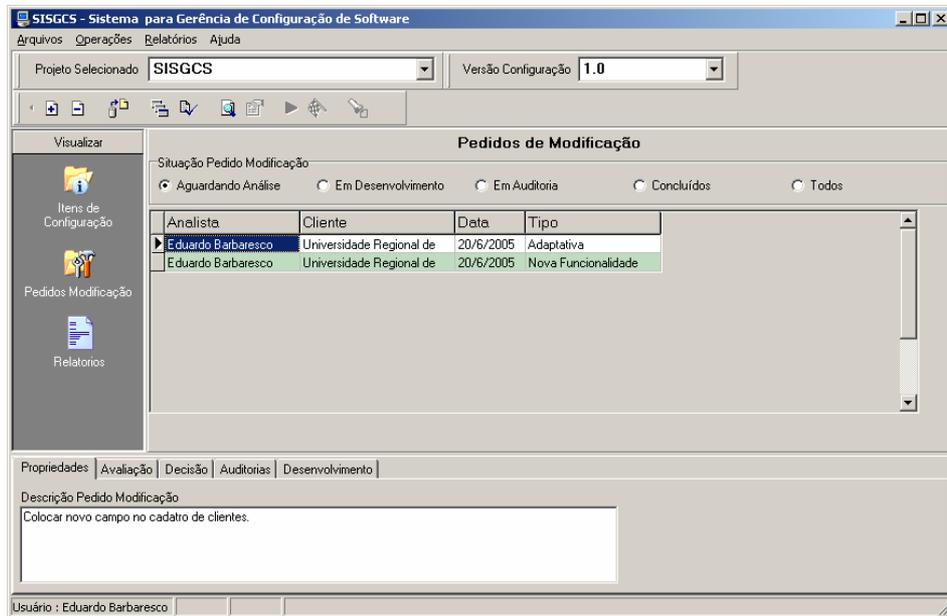


Figura 25 – Tela principal com visualização dos pedido de modificação

Na figura 25 são apresentados os pedidos de modificação registrados no sistema. Além disso pode-se visualizar os pedidos que estão aguardando análise, em desenvolvimento, em auditoria, concluídos e todos. Para registrar um pedido de modificação, pressiona-se o botão com um símbolo de adição. A figura 26 apresenta a tela para registro de um pedido de modificação.



Figura 26 – Tela de registro de pedido de modificação

Nesta tela são informados o cliente que está solicitando a mudança, a data do registro, o tipo de pedido e a descrição da modificação a ser implementada. Quando este registro for gravado, o pedido estará na situação *aguardando análise*. Um analista de sistemas ou o gerente de projeto poderá analisar este pedido de modificação e registrar a avaliação do mesmo. Isto é feito usando o botão direito do mouse, sendo que aparecerá um *popup menu*, que terá a opção para registrar a avaliação. A figura 27 apresenta a tela de registro de avaliação.

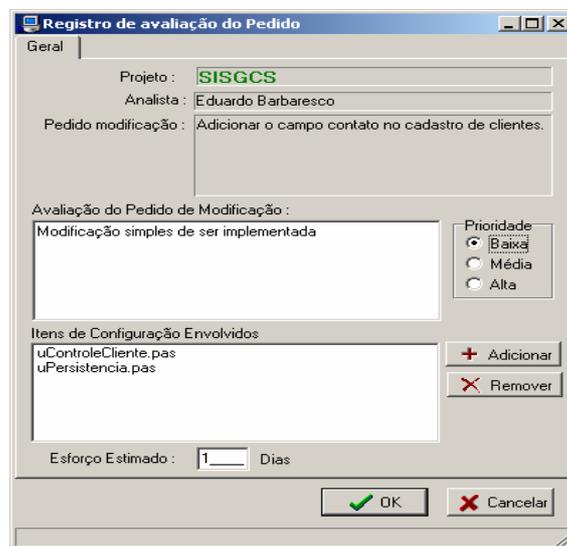
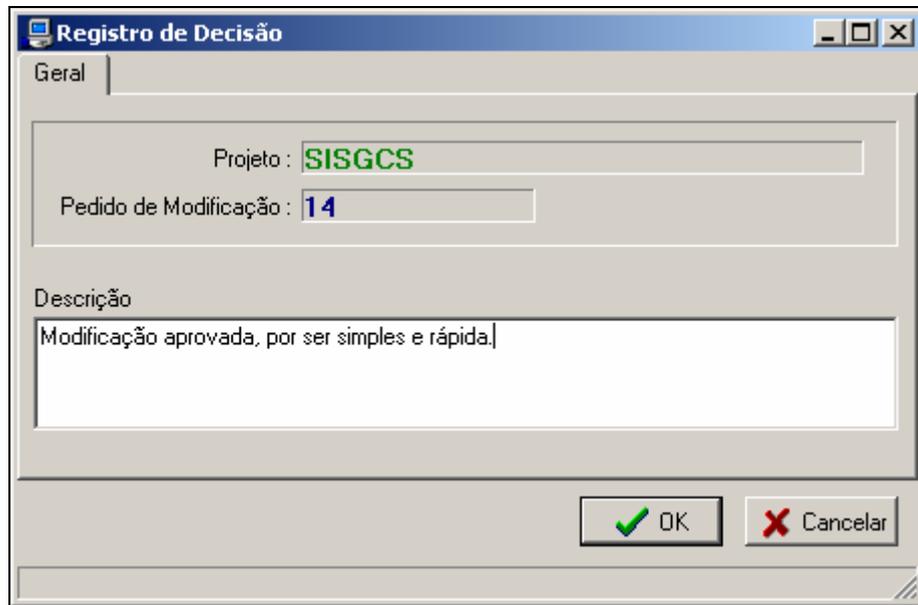


Figura 27 – Tela de registro de avaliação de pedido de modificação

Esta tela registra a análise do pedido de modificação feita por parte de um analista de sistemas ou gerente de projeto. São informados a avaliação do pedido, a prioridade da modificação, o esforço estimado em dias e os itens de configuração que serão afetados. A avaliação não altera a situação do pedido de modificação.

Após ter avaliado o pedido, o gerente de projeto poderá decidir se a modificação será implementada ou não. Isso é feito selecionando o pedido de modificação e pressionando o botão direito do *mouse* encima do referido pedido. Aparecerá um *popup menu* com as opções para aprovar ou reprovar a implementação do pedido de modificação. Caso seja selecionado a opção de reprovado a situação do pedido ficará como concluído. Se for aprovado, então o

gerente de projeto deverá registrar a decisão. A figura 28 apresenta a tela de registro da decisão.



The image shows a software dialog box titled "Registro de Decisão". It features a "Geral" tab. The "Projeto" field is filled with "SIGGCS" in green. The "Pedido de Modificação" field is filled with "14" in blue. The "Descrição" field contains the text "Modificação aprovada, por ser simples e rápida.". At the bottom right, there are two buttons: "OK" with a green checkmark icon and "Cancelar" with a red X icon.

Figura 28 – Tela de registro de decisão

Após o registro de decisão o pedido estará com a situação de em desenvolvimento. É neste momento, que os itens de configuração relacionados com o pedido de modificação aprovado, serão copiados para o diretório de trabalho do analista de sistemas que realizará a implementação, ficando estes itens bloqueados para outras modificações.

O analista de sistemas poderá informar o término da implementação, selecionando o pedido da modificação e usando o botão direito do *mouse* selecionar a opção de *finalizar desenvolvimento*, sendo que o pedido agora estará na situação de *em auditoria*.

Neste momento o auditor irá testar a modificação e verificar se a implementação está em conformidade com o que foi solicitado. A figura 29 apresenta a tela de registro de auditoria do pedido de modificação.

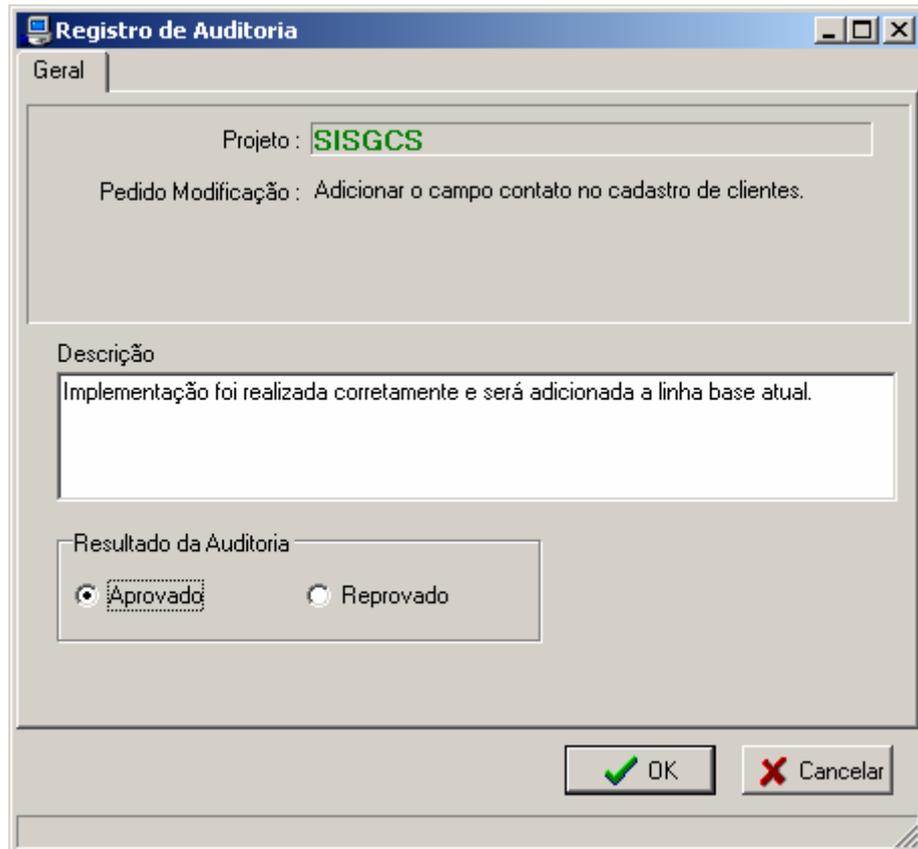


Figura 29 – Tela de registro de auditoria do pedido de modificação

Se a implementação for aprovada, a situação do pedido passará para concluído e os itens de configuração serão devolvidos ao repositório do projeto, ficando estes itens disponíveis para futuras implementações. Termina assim o processo de controle de modificação e controle de versão da ferramenta.

Para visualizar os itens de configuração, pode-se selecionar a opção relatórios do menu principal e sub-opção itens de configuração. Neste relatório é possível visualizar todos os itens de configuração registrados no projeto com seu histórico de modificações (figura 30).

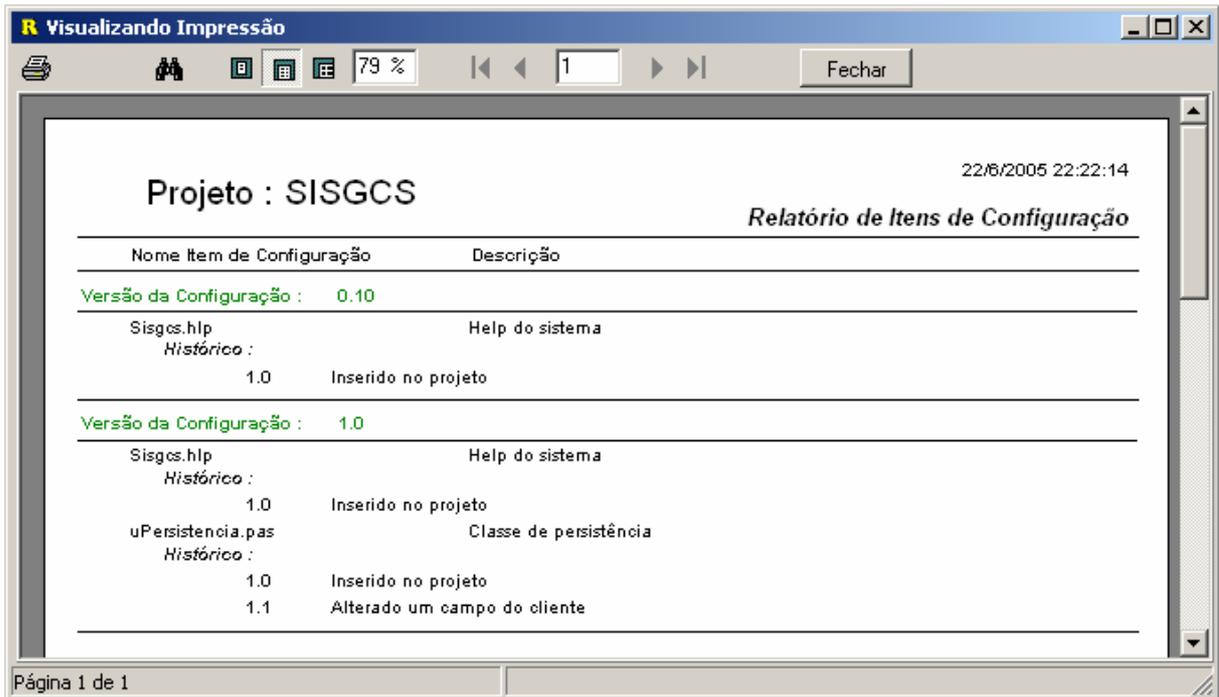


Figura 30 – Relatório de itens de configuração

Para visualizar os pedidos de modificação registrados seleciona-se a opção relatórios do menu principal e sub-opção itens de configuração. Neste relatório é possível visualizar todos os pedidos de modificação registrados no projeto (figura 31).

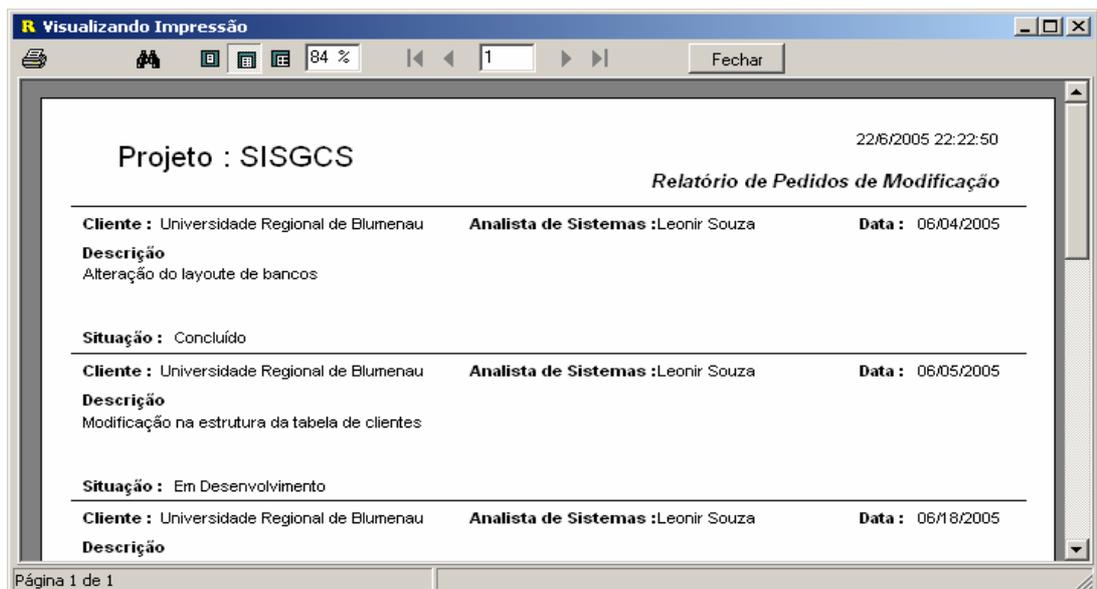


Figura 31 – Relatório de pedidos de modificação

A figura 32 apresenta o relatório de auditorias que pode ser acionado selecionando a opção relatórios do menu principal e sub-opção auditorias. Neste relatório é possível

visualizar todas as auditorias registradas no projeto.

**Projeto : SISGCS** 22/6/2005 22:35:49  
**Relatório de Auditorias**

**Auditor:** Ademar Alves

Versão	Descrição	Data	Resultado
0.10	Modificação testada e todas as funcionalidades foram corretamente implementadas.	05/06/2005	Aprovado

**Auditor:** Ademar Borba

Versão	Descrição	Data	Resultado
0.10	Versão da Configuração apresenta consistência e integridade. Alguns itens de configuração não foram inseridos.	22/06/2005	Aprovado

Página 1 de 1

Figura 32 – Relatório de auditorias

Os tipos de itens de configuração podem ser visualizados selecionando a opção relatórios do menu principal e sub-opção tipos de itens de configuração. Neste relatório é possível visualizar todos os tipos de itens de configuração registrados no projeto (figura 33).

**Projeto : SISGCS** 22/6/2005 22:36:21  
**Relatório de Tipos de Itens de Configuração**

Código	Nome	Local no Repositório
<b>Versão da Configuração : 0.10</b>		
1	Manuais do Usuário	C:\Projeto\Manuais
2	Executáveis	C:\Projeto\Executáveis
3	Fonte-Entidade	C:\Projeto\Entity
4	Fonte-Persistência	C:\Projeto\Persistencia
5	Fonte-Controle	C:\Projeto\Controle
6	Fonte-Fronteira	C:\Projeto\Fronteira

Página 1 de 1

Figura 33 – Relatório de tipos de itens de configuração

O relatório de versões da configuração apresenta os dados das linhas base do projeto. Na opção relatórios do menu principal e sub-opção versões da configuração acessa o referido

relatório. A figura 34 apresenta o relatório de versões da configuração.

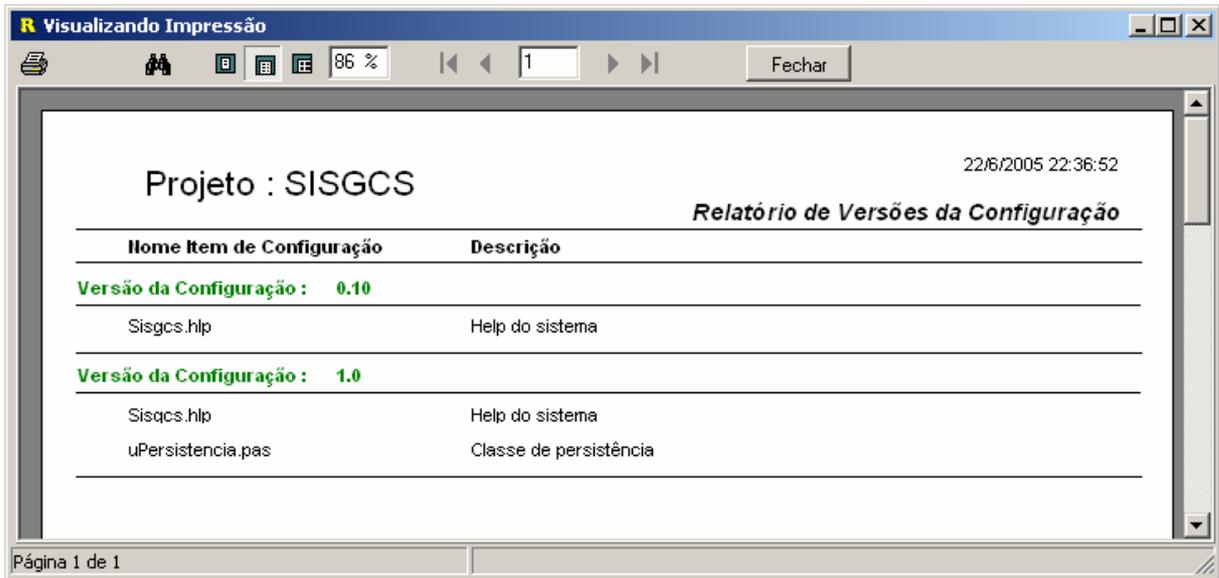


Figura 34 – Relatório de versões da configuração

Os dados do projeto podem ser visualizados no relatório de projeto (figura 35), que apresenta as versões da configuração registradas no projeto, os itens de configuração e os clientes do projeto.

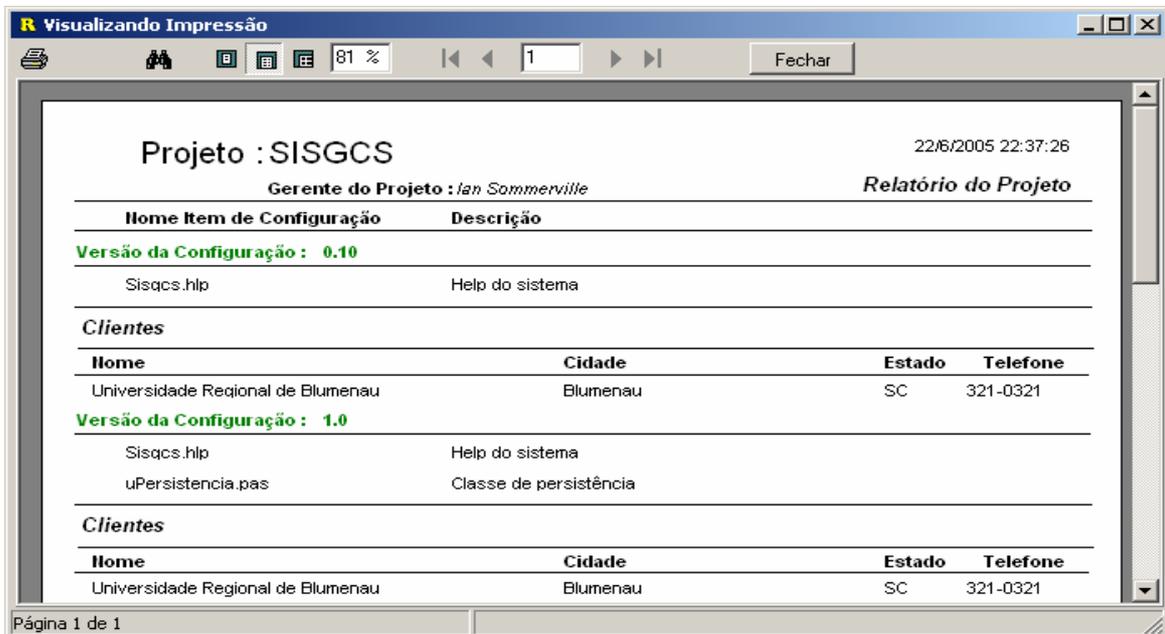


Figura 35 – Relatório do projeto

### 3.3 RESULTADOS E DISCUSSÃO

Nesta seção será apresentada uma análise de ferramentas já existentes para a gerência de configuração e análise da ferramenta implementada em relação ao CMMI.

#### 3.3.1 Análise de ferramentas existentes no mercado

Para esta análise foi utilizado a norma ISO/IEC 14102 que especifica um conjunto de processos bem definidos para a avaliação e seleção de ferramentas CASE. Esta análise foi baseada nas funcionalidades que são desejáveis em uma ferramenta de gerência de configuração da referida norma. Maiores informações sobre a norma em (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, 1999).

Para a análise foram escolhidas as seguintes ferramentas:

- a) team coherence: empresa Quality Softwares Components Ltd;
- b) tortoise cvs: projeto *open-source* baseado no CVS (*concurrent version control*);
- c) siggcs: ferramenta resultante do estudo do presente trabalho;
- d) wincvs: projeto *open-source* baseado no CVS (*concurrent version control*).

O quadro 13 apresenta a tabela com os resultados da análise das ferramentas.

	Team Coherence	Tortoise CVS	SISGCS	WinCVS
Controle de Acesso	Sim	Não	Sim	Sim
Registro de Rastreamento	Não	Não	Sim	Não
Definições e gerenciamento de múltiplas versões	Sim	Sim	Sim	Sim
Relato da situação da configuração	Sim	Não	Sim	Sim
Geração de versões	Sim	Sim	Sim	Sim
Capacidade de arquivamento	Sim	Sim	Sim	Sim

Quadro 13 – Resultado análise de ferramentas de gerência de configuração

A ferramenta *open-source* TortoiseCVS e WinCVS são uma opção interessante por ser licença gratuita, no entanto a questão da segurança é crítica uma vez que qualquer usuário pode alterar um item de configuração. A ferramenta Team Coherence da Quality Softwares Component, tem atende bem às atividades da gerência de configuração, é fácil de usar e tem boa estabilidade.

Outras ferramentas importantes do seguimento como o Rational ClearCase da IBM, MKS da empresa MKS e Star Team da Borland, não foram analisadas por serem versões comerciais e exigem a comprovação de possível uso comercial para disponibilizar uma cópia de demonstração.

Das ferramentas estudadas apenas o Team Coherence possui licença paga, mas é de baixo custo. O motivo de estudar ferramentas gratuitas e de baixo custo, é para que empresas que não possam investir grandes quantias em ferramentas possam implantar a gerência de configuração no seu processo de desenvolvimento de software.

### 3.3.2 Análise da ferramenta implementada em relação ao CMMI

O quadro 14 apresenta as metas e prática do CMMI para a gerência de configuração e se a ferramenta desenvolvida neste trabalho atende bem, atende parcial ou não atende a estas práticas.

Metas e Práticas do CMMI	Situação na ferramenta desenvolvida
ME 1 Estabelecer Linhas base	Atende bem
PE 1.1 Identificar Itens de Configuração	Atende bem
PE 1.2 Estabelecer um Sistema de Gerenciamento de Configuração	Não se aplica
PE 1.3 Criar ou Liberar as Linhas Base	Atende parcial
ME 2 Rastrear e Controlar Mudanças	Atende bem
PE 2.1 Rastrear Pedidos de Mudança	Atende bem
PE 2.2 Controlar Itens de Configuração	Atende bem
ME 3 Estabelecer Integridade	Atende parcial
PE 3.1 Estabelecer Registros de Gerência de Configuração	Atende bem
PE 3.2 Realizar Auditorias de Configuração	Atende bem
MG 2 Institucionalizar um Processo Controlado	Atende bem
PG 2.1 Estabelecer uma Política Organizacional	Atende parcial
PG 2.2 (HE 1) Planejar o Processo	Não Atende
PG 2.3 (HE 2) Fornecer Recursos	Não se aplica
PG 2.4 (HE 3) Atribuir Responsabilidades	Não se aplica
PG 2.5 (HE 4) Treinar Pessoas	Não se aplica
PG 2.6 (DE 1) Gerencie as Configurações	Atende bem
PG 2.7 (DE 2) Identificar e Envolver Relevantes Stakeholders	Atende parcial
PG 2.8 (DE 3) Monitorar e Controlar o Processo	Não se aplica
PG 2.9 (VE 1) Avaliar Adesão	Não se aplica
PG 2.10 (VE 2) Reveja o Status com a Alta Gerência	Atende bem
MG 3 Institucionalizar um Processo Definido	Não se aplica
PG 3.1 Estabeleça um Processo Definido	Não se aplica
PG 3.2 Colete Informações de Melhoria	Não se aplica

Quadro 14 – Grade de nível de atendimento das práticas do CMMI

No total são vinte e quatro práticas que foram levadas em consideração sendo que a ferramenta atende bem 41,66%, atende parcialmente 20,83%, não atende 4,16% e 33,33% não se aplica sobre a ferramenta.

A seguir foi feita uma breve justificativa das práticas que atendem bem:

- a) estabelecer linhas base: o sistema permite que sejam criadas varias linhas base para um projeto;

- b) identificar itens de configuração: o sistema permite o registro de itens de configuração;
- c) rastrear e controlar mudanças: o sistema permite que todas as mudanças sejam controladas pelo gerente de projeto;
- d) rastrear pedidos de mudança: o sistema permite que todas os pedidos de mudança sejam registrados, sendo possível rastreá-los;
- e) controlar itens de configuração: o sistema mantém o registro de todas as mudanças feitas para cada item de configuração;
- f) estabelecer registros de gerência de configuração: o sistema permite que todos os itens de configuração sejam mantidos e com suas respectivas descrições;
- g) realizar auditorias de configuração: o sistema permite que o auditor registre suas auditorias na configuração;
- h) institucionalizar um processo controlado: o sistema esta baseada em um sistema institucionalizado uma vez que cada autor tem seu papel no uso da ferramenta e só pode acessar as funcionalidades que lhe cabem;
- i) gerencie as configurações: o sistema permite que a configuração seja gerenciada;
- j) reveja o status com a alta gerência: o sistema permite a visualização da configuração através do relatório de projetos.

As práticas que atendem parcialmente são justificadas a seguir:

- a) criar ou liberar as linhas base: o sistema permite que sejam criadas linhas base, mas não foi implementada uma funcionalidade para liberar as linhas base;
- b) estabelecer integridade: o sistema mantém as informações gravadas no banco de dados com integridade, mas não implementado a integridade dos itens de configuração a nível físico;
- c) estabelecer uma política organizacional: o sistema ajuda a estabelecer uma política

uma vez que cada usuário tem funções determinadas;

- d) identificar e envolver stakeholders relevantes: o sistema permite apenas alguns dos *stakeholders* relevantes como o gerente de projeto, analista de sistemas, auditor e cliente;

Não foi implementada uma funcionalidade para registrar o planejamento do processo de configuração pois a ferramenta ficou direcionada especificamente para o processo de gerência de configuração. As práticas que foram consideradas como não aplicáveis, estão relacionadas a práticas relacionadas a decisões e tarefas da alta gerência.

## 4 CONCLUSÕES

A gerência de configuração é uma disciplina importante que faz parte do todo processo de desenvolvimento de um produto. O controle das modificações de um projeto é um fator a ser considerado para a garantia da qualidade de um produto de software. Para um controle efetivo, é necessário produzir uma documentação, que forneça a um líder de projeto uma visão ampla e confiável do projeto.

Aplicar a gerência de configuração utilizando um modelo de qualidade como o CMMI, ajuda uma organização a implantar este controle em seu processo de desenvolvimento, podendo visualizar com antecedência os pontos críticos, e assim estruturar uma solução evitando frustrações.

As ferramentas disponíveis no mercado para gerência de configuração de forma geral são completas mas complexas, o que deixam acadêmicos e desenvolvedores de pequenas empresas sem um acesso facilitado a esta tecnologia, além de possuir alto custo para aquisição da licença. A ferramenta desenvolvida se mostrou melhor no aspecto de controle de modificações por ser fácil de usar. Características como funcionamento na WEB e controle de versão de arquivos binários, presentes na maioria das ferramentas não foi implementado.

A ferramenta desenvolvida não atendeu a funcionalidade de planejamento da configuração, por ser uma atividade administrativa. O controle da versão dos itens de configuração foi realizado apenas para arquivos texto, devido a complexidade dos algoritmos envolvidos.

A principal diferença da ferramenta desenvolvida com as ferramentas estudadas, é o controle de pedidos de modificação. As ferramentas estudadas não possuem uma funcionalidade específica para este controle. Outra diferença é a questão de segurança dos itens de configuração que só são liberados através do processo formal de pedido de

modificação enquanto que nas outras ferramentas qualquer usuário pode liberar os itens de configuração e alterá-los.

A ferramenta alcançou o objetivo proposto. A ferramenta permite que itens de configuração sejam controlados e mantém o histórico das modificações nos mesmos. Mais de 50% das metas e práticas definidas no CMMI foram atendidas. A ferramenta foi desenvolvida utilizando termos encontrados na literatura, facilitando o aprendizado por parte de acadêmicos e outros usuários.

#### 4.1 EXTENSÕES

Para a ampliação deste trabalho sugere-se a implementação do processo de cópia de arquivos utilizando um protocolo de internet para usuários da ferramenta, possam realizar modificações na configuração qualquer lugar.

Outra sugestão é a implementação do armazenamento das versões dos itens de configuração em arquivo binário.

## REFERÊNCIAS BIBLIOGRÁFICAS

- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 14102:** tecnologia de informação – orientação para avaliação e seleção de ferramentas CASE. Rio de Janeiro, 1999.
- ANDRADE, Patrícia Inês. **Qualidade nos processos do ciclo de vida do produto.** 2002. 91 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Departamento de Ciências da Computação, Universidade Federal de Lavras, Lavras, 2002.
- AGNOL, Samuel Dall; HERBERT, Juliana Silva. Utilização do TSP para a gerência de equipes nível 2 do CMMI. In: SIMPÓSIO INTERNACIONAL DE MELHORIA DE PROCESSOS DE SOFTWARE, 6., 2004, São Paulo. **Anais...** São Leopoldo: ESICenter UNISINOS, 2004. p. 108-118.
- BARBARESCO, Eduardo Alexandre. **Software de apoio ao processo de gerência da configuração segundo normas e modelos da qualidade.** 2000. 64 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau, 2000.
- BARTIÉ, Alexandre. **Garantia da qualidade de software:** adquirindo maturidade organizacional. Rio de Janeiro. Campus, 2002.
- BORLAND CORPORATION. **BORLAND StarTeam:** automated configuration and change management systemno. Scotts Valley, 2004. Disponível em: <<http://www.borland.com/starteam>>. Acesso em: 02 nov. 2004.
- CHAN, Alessandra; BIANCHINI, Sandro Lopes. **Gerenciamento de configuração no CMMI.** 2005. Trabalho da disciplina de engenharia de software. 8 f. Curso de pós-graduação em Computação e Matemática Computacional, USP, São Carlos.
- CVS. **Concurrente version system:** the open standard for version control. Disponível em <<https://www.cvshome.org/>>. Acesso em: 04 abr. 2005.
- CVSGUI. **Wincvs.** Disponível em <<http://www.wincvs.org/>>. Acesso em: 04 abr. 2005.

IBM. **CLEAR Case Resources**. New York, 2004. Disponível em: <<http://www-136.ibm.com/developerworks/rational/products/clearcase>>. Acesso em: 02 nov. 2004.

PAULA FILHO, Wilson de Pádua. **Engenharia de software**: fundamentos, métodos e padrões. Rio de Janeiro: Livros Técnicos e Científicos, 2001. 581 p.

PETTERS, James; PEDRYCZ, Witold. **Engenharia de software**: teoria e prática. Tradução Ana Patrícia Machado de Pinho Garcia. Rio de Janeiro: Campus, 2001.

PRESSMAN, Roger S. **Engenharia de software**. 5. ed. Rio de Janeiro: McGraw-Hill, 2002. 843 p.

QSC: **Team coherence**. Disponível em <<http://www.qsc.co.uk/teamcoherence.htm>>. Acesso em: 04 abr. 2005.

ROCHA, Ana Regina Cavalcanti; MALDONADO, José Carlos; WEBER, Kival Chaves. **Qualidade de software**: teoria e prática. São Paulo: Prentice Hall, 2001. 301 p.

SEI CMMI Product Team. **Concept of operations for the CMMI**. Carnegie Mellon, jan. 2001. Disponível em: <<http://www.sei.cmu.edu/cmmi/background/conops.html>>. Acesso em: 02 nov. 2004.

\_\_\_\_\_. **Continuous representation**. Carnegie Mellon, ago. 2002a. Disponível em: <<http://www.sei.cmu.edu/publications/documents/02.reports/02tr028.html>>. Acesso em: 02 nov. 2004.

\_\_\_\_\_. **Staged representation**. Carnegie Mellon, ago. 2002b. Disponível em: <<http://www.sei.cmu.edu/publications/documents/02.reports/02tr029.html>>. Acesso em: 02 nov. 2004.

SOMMERVILLE, Ian. **Engenharia de software**. 6. ed. São Paulo: Addison Wesley, 2003. 592p.

TORTOISECVS. **Enjoyable version control**. Disponível em <<http://www.tortoisecvs.org/index.shtml>>. Acesso em: 04 abr. 2005.