

Júnia Gaudereto Carvalho
junia@dcc.ufmg.br

Praxis Mentor - Uma Ferramenta de Apoio à Utilização de um Processo de Desenvolvimento de Software

Dissertação apresentada ao Departamento de
Ciência da Computação do Instituto de Ciências
Exatas da Universidade Federal de Minas Gerais
como requisito parcial para a obtenção do grau de
Mestre em Ciência da Computação.

Belo Horizonte

27 de março de 2001

Agradecimentos

Não poderia deixar de agradecer àqueles que, de alguma forma, contribuíram para a realização deste trabalho. Em primeiro lugar agradeço a Deus, simplesmente por tudo permitir. Agradeço também a meus pais, por sempre terem incentivado meus estudos e por terem me mostrado a sua importância. Agradeço e, ao mesmo tempo, peço desculpas aos meus irmãos, Diana e Juninho, e aos meus amigos e familiares, que sempre me apoiaram e entenderam minhas constantes ausências nos últimos dois anos.

Em especial agradeço ao Arnaldo, meu noivo, não só pelo apoio incondicional, mas pelo carinho, amor e paciência oferecidos de forma tão sincera durante todo este período. Jamais conseguiria colocar em poucas palavras o que isso significou para mim. Por isso, deixarei simplesmente o meu “muito obrigada, meu amor”.

Gostaria de agradecer também, de forma especial, à amiga Gisele, por reservar um tempo que não tinha para oferecer contribuições e realizar revisões neste trabalho. Agradeço a ela ainda, pela amizade e pelo apoio, que tantas vezes deram-me forças para terminar este projeto. Às amigas Fa e Pat, companheiras de jornada, também deixo o meu muito obrigada pelas inúmeras vezes em que me ouviram e se fizeram presentes para me ajudar a seguir este caminho.

Aos professores João Dantas e Roberto Bigonha gostaria de deixar o meu sincero agradecimento por terem me recebido no mestrado e me oferecido as primeiras diretrizes. Infelizmente, nosso projeto original não pôde ser realizado, mas a experiência de trabalhar com cada um de vocês já foi muito proveitosa e compensadora.

Ao meu orientador, professor Wilson de Pádua Paula Filho, também tenho muito a agradecer. Em primeiro lugar, por ter me recebido de forma tão aberta e, principalmente, por ter apostado no meu trabalho. Gostaria de agradecer-lhe ainda pela maneira com que conduziu este projeto. Também agradeço pelo incentivo dado a cada conquista e pelas exigências feitas de forma sutil e motivadora. Agradeço, enfim, pela confiança em mim depositada, que me permitiu assumir a direção do trabalho e expor minhas idéias, contando sempre com as orientações que se faziam necessárias. Ao meu ver, o papel de orientação deste trabalho foi cumprido de forma precisa e isto foi fundamental para o sucesso do mesmo.

Finalmente, gostaria de agradecer às empresas que, de uma forma ou de outra, apoiaram este trabalho. À MSA Infor, agradeço pela gentileza em ceder-me uma máquina para que eu realizasse estudos sobre o RUP. À BMS Sistemas, agradeço pela paciência e apoio com relação às minhas inúmeras ausências justificadas pelo mestrado durante o tempo em que estive na empresa. Ao pessoal do grupo Synergia, também agradeço pelo apoio dado às tarefas do mestrado e pelo tempo sempre cedido para que eu conseguisse finalizar meus estudos. E, a todas as pessoas envolvidas com as empresas e com o grupo citado, agradeço pela amizade e pela torcida a mim demonstradas.

Resumo

Este trabalho apresenta uma ferramenta de apoio à utilização do processo Praxis denominada PRAXIS MENTOR. O Praxis é um processo completo de desenvolvimento de software adequado a projetos com duração de seis meses a um ano, em um contexto educacional ou de treinamento. Ele cobre métodos técnicos (requisitos, análise, desenho, implementação e testes) e gerenciais (gestão de compromissos, gestão de projetos, gestão da qualidade e engenharia de processos). A ferramenta aqui apresentada atende a todas estas fases, fornecendo informações sobre cada uma delas e auxiliando o usuário do processo na aplicação do mesmo. Ela ainda possibilita o controle do fluxo de trabalho dentro da equipe do projeto. Recursos como comunicação interna, controle de fases e tarefas do projeto e um material de referência do processo também são suportados pela ferramenta.

Abstract

This work shows a tool named PRAXIS MENTOR for supporting Praxis process. Praxis is a complete software development process adequate to projects that take from six months to one year, in an educational or training context. It covers both technical methods (requirements, analysis, design, implementation, and tests) and management methods (contract, project and quality management and project engineering). The tool presented here covers all these phases of the process, giving information about each one and helping the user on applying the process. In addition, it allows controlling the workflow of the project team. Features like internal communication, management of tasks and checkpoints, and reference documents for the process are supported by the tool as well.

Sumário

INTRODUÇÃO	1
1.1 MOTIVAÇÃO	1
1.2 AMBIENTAÇÃO	4
1.2.1 O PROCESSO PRAXIS	4
1.2.1.1 Características básicas	5
1.2.1.2 Arquitetura	5
1.2.1.3 Artefatos	7
1.2.1.4 Procedimentos de controle	9
1.2.2 FERRAMENTAS CASE	10
1.2.2.1 Definição	10
1.2.2.2 Vantagens	10
1.2.2.3 Taxonomia	11
1.3 PROPOSTA DA FERRAMENTA CASE PRAXIS MENTOR	13
1.3.1 O PRAXIS MENTOR	13
1.3.2 RESULTADOS ESPERADOS	13
1.4 ORGANIZAÇÃO DESTE DOCUMENTO	14
CONCEPÇÃO	15
2.1 DEFINIÇÃO	15
2.2 CONCEPÇÃO DO PRAXIS MENTOR	16
2.2.1 MODELAGEM DE PROCESSOS DE NEGÓCIO	16
2.2.2 PROPOSTA DE ESPECIFICAÇÃO DO SOFTWARE	21
2.2.2.1 Missão do produto	22
2.2.2.2 Lista de funções	22
2.2.2.3 Outros aspectos	23
ELABORAÇÃO	24
3.1 DEFINIÇÃO	24
3.1.1 LEVANTAMENTO DOS REQUISITOS	24
3.1.2 ANÁLISE DOS REQUISITOS	24
3.2 ELABORAÇÃO DO PRAXIS MENTOR	26
3.2.1 ESPECIFICAÇÃO DOS REQUISITOS DO SOFTWARE	26
3.2.1.1 Definição do escopo	26
3.2.1.2 Definição dos requisitos	28
3.2.1.3 Detalhamento dos requisitos de interface	32
3.2.1.4 Requisitos adiados	40
3.2.1.5 Detalhamento dos requisitos funcionais	41
3.2.1.6 Detalhamento dos requisitos não funcionais	43
3.2.1.7 Modelo de Análise	44
3.2.1.8 Classificação dos requisitos	48

3.2.1.9	Decisões de especificação	49
3.2.2	REVISÃO DA ESPECIFICAÇÃO DOS REQUISITOS DO SOFTWARE	50
3.2.3	PLANO DE DESENVOLVIMENTO DO SOFTWARE	51
3.2.3.1	Análise de Pontos de Função	51
3.2.3.2	Dimensionamento do Praxis Mentor	52
3.2.3.3	Análise de riscos do Praxis Mentor	55
3.2.4	PLANO DA QUALIDADE DO SOFTWARE	55
3.2.4.1	Responsabilidades no projeto	56
3.2.4.2	Ferramentas de Garantia da Qualidade	56
3.2.4.3	Política de controle de mídia	59

CONSTRUÇÃO **60**

4.1	DEFINIÇÃO	60
4.1.1	DESENHO INICIAL	60
4.1.2	LIBERAÇÕES	61
4.1.3	TESTES ALFA	62
4.2	CONSTRUÇÃO DO PRAXIS MENTOR	62
4.2.1	DESCRIÇÃO DO DESENHO DO SOFTWARE	63
4.2.1.1	Desenho das interfaces de usuário	63
4.2.1.2	Desenho interno	82
4.2.1.3	Desenho das liberações	98
4.2.2	CÓDIGOS FONTES E EXECUTÁVEIS DO SOFTWARE	100
4.2.3	MANUAL DO USUÁRIO DO SOFTWARE	102
4.2.3.1	Introdução	102
4.2.3.2	Conceitos básicos do produto	103
4.2.3.3	Funções do produto	104

TRANSIÇÃO **109**

5.1	DEFINIÇÃO	109
5.2	TRANSIÇÃO DO PRAXIS MENTOR	109

CONCLUSÃO E TRABALHOS FUTUROS **110**

6.1	CONCLUSÃO	110
6.2	TRABALHOS FUTUROS	112
6.2.1	REQUISITOS ADIADOS	113
6.2.2	REQUISITOS OPCIONAIS DA VERSÃO 1.0 NÃO IMPLEMENTADOS	113
6.2.3	REQUISITOS IDENTIFICADOS AO LONGO DO PROJETO	114

Lista de figuras

Figura 1 - O “Triângulo do Sucesso”	2
Figura 2 - Diagrama de Atividades – Fluxo Principal	17
Figura 3 - Diagrama de Atividades – Subfluxo Procedimentos de Controle.....	18
Figura 4 - Diagrama de Atividades – Subfluxo Criação de Linha de Base	18
Figura 5 - Diagrama de Atividades – Subfluxo Replanejamento	19
Figura 6 - Diagrama de Atividades – Fluxo Alternativo Relatório de Acompanhamento	20
Figura 7 - Diagrama de Atividades – Fluxo Alternativo Alteração de Linha de Base	20
Figura 8 - Diagrama de Atividades – Fluxo Alternativo Alteração de Equipe.....	21
Figura 9 - Diagrama de Atividades – Fluxo Alternativo Resolução de Problemas	21
Figura 10 - Diagrama Consulta ao Processo.....	28
Figura 11 - Diagrama Comunicação.....	28
Figura 12 - Diagrama Gestão do Planejamento do Projeto.....	30
Figura 13 - Diagrama Controle de Projetos	31
Figura 14 - Diagrama de Classes Persistentes	43
Figura 15 - Diagrama de Classes Aplicação.....	45
Figura 16 - Diagrama de Classes Comunicação	45
Figura 17 - Diagrama de Classes Consulta ao Processo.....	46
Figura 18 - Diagrama de Classes Controle de Projetos	46
Figura 19 - Diagrama de Classes Planejamento de Projetos	47
Figura 20 - Realizações do Caso de Uso Criação de Projetos – Fluxo Principal	48
Figura 21 - Realizações do Caso de Uso Controle de Mensagens Recebidas – Fluxo Alternativo Responder ao Remetente.....	48
Figura 22 - Diagrama de Gantt do Praxis Mentor	54
Figura 23 - Estrutura Dinâmica do Praxis Mentor.....	65
Figura 24 - Diagrama de Pacotes Lógicos do Praxis Mentor	87
Figura 25 - Diagrama de Classes Persistentes do Praxis Mentor	88
Figura 26 - Diagrama de Desenho do Banco de Dados do Praxis Mentor	89
Figura 27 - Diagrama de Classes da Camada de Persistência do Praxis Mentor.....	90
Figura 28 - Diagrama de Interfaces do Usuário do Processo.....	91
Figura 29 - Diagrama de Interfaces do Administrador	91
Figura 30 - Diagrama de Interfaces do Gerente de Projeto	92
Figura 31 - Diagrama de Classes Aplicação.....	92
Figura 32 - Diagrama de Classes Consulta ao Processo.....	93
Figura 33 - Diagrama de Classes Comunicação	93
Figura 34 - Diagrama de Classes Controle de Projetos	93
Figura 35 - Diagrama de Classes Planejamento de Projetos	94
Figura 36 - Diagrama de Seqüência Criação de Projetos – Roteiro Principal	95
Figura 37 - Diagrama de Colaboração Criação de Projetos – Roteiro Principal	95
Figura 38 - Diagrama de Seqüência Controle de Mensagens Recebidas – Roteiro Responder ao Remetente	96

Figura 39 - Diagrama de Colaboração Controle de Mensagens Recebidas – Roteiro Responder ao Remetente	96
Figura 40 - Diagrama de Componentes Físicos do Praxis Mentor	97
Figura 41 - Diagrama de Desdobramento do Praxis Mentor	97

Lista de tabelas

Tabela 1 - Fases do Praxis	6
Tabela 2 - Fluxos do Praxis	6
Tabela 3 - Detalhamento das fases do Praxis	7
Tabela 4 - Documentos permanentes do Praxis.....	8
Tabela 5 - Modelos permanentes do Praxis	8
Tabela 6 - Relatórios do Praxis.....	9
Tabela 7 - Caso de uso de negócio “Gerir Projeto (Construção e Transição)”	16
Tabela 8 - Missão do Praxis Mentor.....	22
Tabela 9 - Funções do Praxis Mentor	23
Tabela 10 - Outros aspectos do Praxis Mentor	23
Tabela 11 - Missão do Praxis Mentor.....	26
Tabela 12 - Limites do Praxis Mentor	27
Tabela 13 - Benefícios do Praxis Mentor	27
Tabela 14 - Atores e casos de uso do Diagrama Consulta ao Processo.....	28
Tabela 15 - Atores e casos de uso do Diagrama Comunicação	29
Tabela 16 - Atores e casos de uso do Diagrama Gestão do Planejamento do Projeto.....	30
Tabela 17 - Atores e casos de uso do Diagrama Controle de Projetos	32
Tabela 18 - Interfaces de usuário do Praxis Mentor	34
Tabela 19 - Requisitos adiados do Praxis Mentor	40
Tabela 20 - Amostra do cadastro inicial dos requisitos do Praxis Mentor	49
Tabela 21 - Decisão sobre o ambiente de desenvolvimento do Praxis Mentor	49
Tabela 22 - Decisão sobre a nomenclatura utilizada no Modelo de Análise do Praxis Mentor	49
Tabela 23 - Resultado da revisão técnica informal do Praxis Mentor	50
Tabela 24 - Regras do COCOMO aplicadas ao Praxis Mentor	52
Tabela 25 - Tabela de McConnell aplicada ao Praxis Mentor	53
Tabela 26 - Distribuição do esforço de desenvolvimento do Praxis Mentor por fases.....	53
Tabela 27 - Orçamento do Praxis Mentor.....	54
Tabela 28 - Cronograma tabular do Praxis Mentor	54
Tabela 29 - Principais riscos ao projeto Praxis Mentor	55
Tabela 30 - Procedimentos de análise dos riscos do Praxis Mentor	55
Tabela 31 - Responsabilidades da equipe do Praxis Mentor	56
Tabela 32 - Ferramentas de Garantia da Qualidade do Praxis Mentor	56
Tabela 33 - Política de controle de mídia do Praxis Mentor	59
Tabela 34 - Características dos usuários do Praxis Mentor	64
Tabela 35 - Estrutura estática do Praxis Mentor.....	65
Tabela 36 - Funções do Praxis Mentor	66
Tabela 37 - Convenções gerais do Praxis Mentor	66
Tabela 38 - Estratégias de arquitetura do Praxis Mentor.....	82
Tabela 39 - Lista das liberações do Praxis Mentor	98

Tabela 40 - Liberação 1 do Praxis Mentor – Gestão de Projetos.....	99
Tabela 41 - Liberação 2 do Praxis Mentor – Comunicação da Equipe	100
Tabela 42 - Código fonte da classe ControladorBancoDados do Praxis Mentor	101
Tabela 43 - Público alvo do manual do Praxis Mentor.....	102
Tabela 44 - Aplicabilidade do Praxis Mentor.....	103
Tabela 45 - Missão do Praxis Mentor.....	103
Tabela 46 - Convenções utilizadas no manual do Praxis Mentor.....	103
Tabela 47 - Conceitos básicos do Praxis Mentor.....	104

Capítulo 1

Introdução

1.1 Motivação

Um sistema bem sucedido é aquele que satisfaz ou supera as expectativas do cliente, é desenvolvido de forma adequada e econômica e pode ser facilmente estendido ou adaptado. Diversos fatores, entretanto, tornam a produção de um sistema com tais características um desafio fascinante. Dentre esses fatores pode-se citar:

- sistemas cada vez mais complexos;
- alterações freqüentes nos requisitos e na modelagem;
- prazos para desenvolvimento mais apertados;
- aspectos econômicos, organizacionais e psicológicos mais importantes a cada dia;
- mercado mais competitivo, causando a necessidade de maior flexibilidade, produtividade e qualidade;
- menor tolerância a falhas;
- maior exigência com relação a performance, qualidade e facilidade de uso;

Tendo em vista esses fatores, é preciso organizar, planejar e controlar a produção de um sistema para que se consiga um resultado positivo ao final do projeto. Segundo Quatrani [Quatrani98], para que o projeto de um sistema seja bem sucedido, ele precisa de três aspectos: uma notação, um processo e uma ferramenta.

Segundo Ambler [Ambler97] e Fowler [Fowler+97], uma notação é um conjunto de símbolos que compõem um sistema de representação a ser utilizado durante a análise e o desenho de um software. A seleção apropriada de uma notação permite que os requisitos e os aspectos particulares da arquitetura do sistema sejam descritos de forma mais precisa e concisa do que se descritos em linguagem natural.

O uso de uma notação facilita o trabalho do engenheiro de software, uma vez que disponibiliza uma linguagem comum e facilmente entendida por desenvolvedores, gerentes e usuários. A utilização dessa “linguagem comum” torna mais fácil a comunicação entre todos os envolvidos no projeto e proporciona um melhor entendimento sobre os requisitos do sistema.

Já o processo de desenvolvimento descreve “o que fazer”, “como fazer”, “quando fazer” e “quem deve fazer”. Esse processo precisa favorecer a criatividade e a inovação, mas, ao mesmo tempo, deve ser controlado e medido para garantir que o projeto seja realmente finalizado. O grande desafio na proposta de um processo é conseguir um conjunto de regras que guiem o desenvolvimento sem comprometer a criatividade e a motivação dos desenvolvedores.

Em seu uso normal, a palavra “processo” significa uma relação de atividades que devem ser executadas em uma certa ordem para se atingir um objetivo. No processo de desenvolvimento de software, é necessário saber o objetivo final do processo, definir regras a serem seguidas e adotar um método fixo de desenvolvimento. Neste caso, este objetivo é produzir um sistema que atenda às necessidades do negócio de forma eficiente e previsível.

Ao aumentar a produtividade do trabalho e facilitar o controle e o cálculo do progresso de um projeto, a utilização de um processo de desenvolvimento de software torna possível a produção de um sistema com as características citadas anteriormente. A implantação e utilização de um processo de desenvolvimento de software, porém, não é uma tarefa simples. Ela exige treinamento e disciplina por parte de todos os integrantes da equipe. Exige também a existência de um Grupo de Garantia da Qualidade que assegure que o processo esteja sendo não só seguido, mas seguido da forma correta.

Para facilitar o aprendizado e a utilização dos processos de desenvolvimento, são utilizadas ferramentas que automatizam cada vez mais tarefas do ciclo de desenvolvimento de software. Essas ferramentas, conhecidas como ferramentas CASE¹, oferecem suporte não apenas a linguagens de modelagem e programação, mas, principalmente, ao processo que controla essas atividades. Elas geralmente possuem descrições de cada fase do processo, ajuda *on-line* e aconselhamentos sobre o que fazer em cada fase, suporte a desenvolvimento iterativo e fácil comunicação com outras ferramentas.

A integração dos três aspectos descritos anteriormente pode ser visualizada através da Figura 1, que mostra o chamado “Triângulo do Sucesso”.

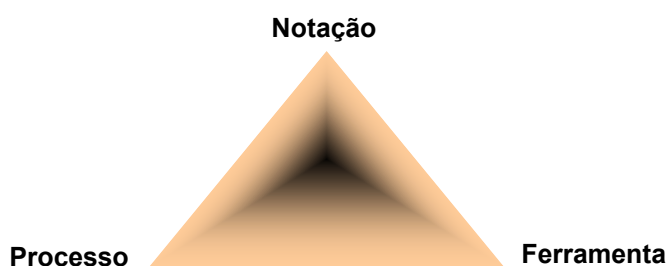


Figura 1 - O “Triângulo do Sucesso”

A idéia do “Triângulo do Sucesso” é a seguinte: pode-se aprender e utilizar uma notação, mas se não existir um processo no qual a notação seja aplicada, provavelmente os

¹ Do inglês *Computer Aided Software Engineering*, CASE é um termo que designa ferramentas que oferecem auxílio automatizado à Engenharia de Software.

desenvolvedores se perderão no caminho, não atingindo por completo seu objetivo inicial. Por outro lado, se for adotado um excelente processo de desenvolvimento, mas não existir uma forma fácil e padronizada de se comunicar esse processo (o que é feito através de uma notação), o projeto também não conseguirá evoluir. Finalmente, se os desenvolvedores não puderem documentar e armazenar seu trabalho de maneira eficiente, e se não conseguirem trabalhar com a notação e com o processo adotado com certa agilidade (ferramenta), o projeto também ficará extremamente prejudicado e fadado a falhar.

Uma notação que está sendo muito adotada atualmente é a UML (*Unified Modeling Language*), definida em [Rumbaugh+99] e [Booch+99]. Trata-se de uma notação unificada que combina os principais conceitos dos métodos de *Booch*, *Rumbaugh* (OMT) e *Jacobson* (OOSE), propostos anteriormente em [Booch91], [Rumbaugh91] e [Jacobson94], respectivamente. Ela se tornou um padrão OMG² em 1997 e, desde então, tem sido largamente utilizada.

A idéia central da UML é fazer com que os arquitetos de sistemas trabalhem com análise e desenho orientados por objetos utilizando uma linguagem consistente para especificação, visualização, construção e documentação dos aspectos do software e modelagem do negócio.

Em se tratando de processos, muitos têm sido propostos e adotados por empresas de software que pretendem atingir um bom nível de qualidade em seus produtos. E alguns deles utilizam a UML como notação. Alguns exemplos de processos são: o Processo Unificado, o Processo Pessoal de Software (*Personal Software Process - PSP*), o Processo de Software para Times (*Team Software Process - TSP*), o Processo Orientado a Objetos para Software Extensível (PROSE) e o Processo para Aplicativos Extensíveis Interativos (Praxis).

O Processo Unificado foi proposto em [Jacobson+99] por Booch, Jacobson e Rumbaugh, que também propuseram a UML. Ele descende de métodos anteriores propostos pelos autores em [Booch94], [Booch96], [Jacobson94], [Jacobson+94a], [Jacobson+97] e [Rumbaugh91]. Trata-se de um processo iterativo e incremental, dirigido por casos de uso e centrado na arquitetura do sistema, que pretende viabilizar a produção de sistemas de alta qualidade que atendam às necessidades dos usuários finais e sejam bastante previsíveis.

O PSP consiste de uma série de processos pessoais que podem ser aprendidos em uma disciplina de Engenharia de Software. Ele foi proposto por Watts Humphrey em [Humphrey95]. Os processos que compõem o PSP são aprendidos através de uma seqüência de pequenos projetos contidos nesse livro. Os projetos devem ser realizados seguindo rigorosamente os processos que incluem um conjunto de formulários, scripts e relatórios predefinidos. Os projetos são individuais, com duração típica de cerca de 10 horas.

O TSP surgiu como uma seqüência natural do PSP e foi proposto por Humphrey em [Humphrey99]. De acordo com este processo, os participantes do time de desenvolvedores são organizados de tal forma que cada desenvolvedor desempenhe um ou dois papéis gerenciais bem definidos, além de dividir a carga de desenvolvimento. Os papéis suportados pelo processo são os de gerente de desenvolvimento, de planejamento, de qualidade, de processo e de suporte, além do líder do time.

O PROSE foi criado no Departamento de Ciência da Computação da Universidade Federal de

² O OMG, *Object Management Group*, é um grupo que fornece diretrizes para a aplicação da tecnologia de objetos na indústria de software através de especificações de padrões como a própria UML.

Minas Gerais pelo professor Wilson de Pádua Paula Filho. Foi concebido como um processo padrão que visava cobrir todo o ciclo de vida dos produtos de software, especialmente de aplicativos extensíveis, através de sucessivas versões produzidas em um ciclo de vida de produto com duração de vários anos. Estes produtos normalmente seriam aplicativos gráficos interativos baseados na tecnologia orientada por objetos. A última versão publicada deste processo está contida em [Paula+98a], [Paula+98b], [Paula+98c] e [Paula+98d].

O Praxis também foi criado no Departamento de Ciência da Computação da Universidade Federal de Minas Gerais pelo mesmo professor. É um processo destinado a suportar projetos didáticos em disciplinas de Engenharia de Software de cursos de Informática. Ele se aplica especialmente ao desenvolvimento de aplicativos gráficos interativos baseados na tecnologia orientada por objetos. O Praxis foi desenhado para suportar projetos de seis meses a um ano de duração e abrange material relativo tanto a métodos gerenciais como técnicos. Dentro deste objetivo, ele procura aproveitar algumas das contribuições do Processo Unificado, Prose, PSP e TSP. A definição completa do processo pode ser encontrada em [Paula01].

Com relação às ferramentas de apoio aos processos de desenvolvimento, alguns exemplos podem ser citados. Dos cinco processos mostrados anteriormente, quatro possuem uma ferramenta. Cada ferramenta possui características diferentes e fornece um determinado tipo de suporte. O *Rational Unified Process (RUP)*, ferramenta da empresa *Rational Software Corporation* de apoio ao Processo Unificado, fornece orientação on-line aos desenvolvedores. Maiores informações sobre o produto podem ser encontradas em [RUP00]. O SAPP (ferramenta de apoio ao PROSE) fornece suporte gerencial e as ferramentas disponíveis para o PSP e TSP, *PSP Studio* e *TSPi Support Tool*, respectivamente, oferecem suporte para coleta de métricas.

O Praxis é o único processo citado que ainda não possui uma ferramenta. Ele é um processo relativamente novo, mas que já está sendo aplicado com grande sucesso em projetos da Universidade Federal de Minas Gerais. O presente trabalho consiste exatamente na proposta e no desenvolvimento de uma ferramenta CASE de apoio ao processo Praxis, denominada Praxis Mentor. O trabalho contempla não só a implementação da ferramenta, mas também uma avaliação do processo, visto que ele foi utilizado durante todo o projeto.

As próximas seções fornecem uma breve descrição sobre o processo Praxis, uma discussão sobre Ferramentas CASE e a proposta da ferramenta descrita neste trabalho, o Praxis Mentor.

1.2 Ambientação

1.2.1 O Processo Praxis

O Praxis, Processo para Aplicativos Extensíveis Interativos, é um processo proposto pelo professor Wilson de Pádua Paula Filho, da Universidade Federal de Minas Gerais, e visa integrar um conjunto de métodos e padrões cobrindo todos os principais aspectos da Engenharia de Software. Ele se aplica especialmente ao desenvolvimento de aplicativos gráficos interativos baseados na tecnologia orientada por objetos e foi desenhado para suportar projetos de seis meses a um ano de duração em um contexto educacional ou de treinamento.

Apesar do Praxis ter sido desenhado para projetos em contextos educacionais ou de

treinamento, ele já está sendo utilizado em projetos reais do Departamento de Ciência da Computação da Universidade Federal de Minas Gerais com grande êxito. A apresentação completa do processo pode ser encontrada em [Paula01].

1.2.1.1 Características básicas

As principais características do Praxis são:

- é iterativo e incremental.
- utiliza a UML como notação em todos os passos em que for aplicável;
- cobre tanto aspectos técnicos quanto gerenciais;
- possui material que inclui gabaritos (*templates*) de documentos e roteiros de revisão, que facilitam a elaboração dos documentos requeridos e a revisão dos mesmos, respectivamente.
- possui padrões incluídos que procuram ser conformes com os correspondentes do IEEE [IEEE94].
- possui práticas gerenciais inspiradas nas práticas chaves do SW-CMM³, descrito em [Paulk+93], [Paulk+93a] e [Paulk+95];
- pode ser complementado e personalizado para passar à prática de uma organização produtora de software.

1.2.1.2 Arquitetura

1.2.1.2.1 Fluxos e Fases

Segundo Paula [Paula01], as fases e os fluxos do Praxis são inspirados nos elementos correspondentes do Processo Unificado, tendo em vista compatibilizar a nomenclatura com um processo que possui grande aceitação na indústria de software. Mas as definições das fases e fluxos são mais específicas em relação às opções de desenho de processo adotadas no Praxis.

³ *Capability Maturity Model*. Modelo de capacitação da área de software que serve para avaliar a maturidade dos processos de uma organização.

Fase	Definição
Concepção	Fase na qual necessidades dos usuários e conceitos da aplicação são analisados o suficiente para justificar a especificação de um produto de software, resultando em uma proposta de especificação.
Elaboração	Fase na qual a especificação do produto é detalhada o suficiente para modelar conceitualmente o domínio do problema, validar os requisitos em termos deste modelo conceitual e permitir um planejamento acurado da fase de construção.
Construção	Fase na qual é desenvolvida (desenhada, implementada e testada) uma liberação completamente operacional do produto, que atende aos requisitos especificados.
Transição	Fase na qual o produto é colocado à disposição de uma comunidade de usuários para testes finais, treinamento e uso inicial.

Tabela 1 - Fases do Praxis

Fluxo	Definição
Requisitos	Fluxo que visa obter um conjunto de requisitos de um produto, acordado entre cliente e fornecedor.
Análise	Fluxo que visa detalhar, estruturar e validar os requisitos, em termos de um modelo conceitual do problema, de forma que estes possam ser usados como base para o planejamento e acompanhamento detalhados da construção do produto.
Desenho	Fluxo que visa formular um modelo estrutural do produto, que sirva de base para a implementação, definindo os componentes a desenvolver e a reutilizar, assim como as interfaces entre eles.
Implementação	Fluxo que visa detalhar e implementar o desenho através de componentes de código e de documentação associada.
Testes	Fluxo que visa verificar os resultados da implementação, através do planejamento, desenho e realização de baterias de testes.

Tabela 2 - Fluxos do Praxis

1.2.1.2.2 Iterações

A divisão das fases do Praxis obedece ao modelo de ciclo de vida de entrega evolutiva [McConnell96]. Do ponto de vista técnico, o desenvolvimento do software é visto como uma sucessão de iterações, nas quais vislumbra-se o software tomar forma. Em cada iteração, um subconjunto de funções do produto vai evoluindo até chegar ao produto final. A tabela a seguir, extraída de [Paula01], apresenta o detalhamento das fases e iterações do Praxis:

Fase	Iteração	Descrição
Concepção	Ativação	Levantamento e análise das necessidades dos usuários e conceitos da aplicação, em nível de detalhe suficiente para justificar a especificação de um produto de software.
Elaboração	Levantamento dos Requisitos	Levantamento detalhado das funções, interfaces e requisitos não funcionais desejados para o produto.
	Análise dos Requisitos	Modelagem conceitual dos elementos relevantes do domínio do problema e uso deste modelo para validação dos requisitos e planejamento detalhado da fase de Construção.
Construção	Desenho Inicial	Definição interna e externa dos componentes de um produto de software, em nível de detalhe suficiente para decidir as principais questões de arquitetura e tecnologia, e para permitir o planejamento detalhado das atividades de implementação.
	Liberação 1	Implementação de um subconjunto de funções do produto que será avaliado pelos usuários.
	Liberação ...	Idem.
	Liberação Final	Idem.
	Testes Alfa	Realização dos testes de aceitação, no ambiente dos desenvolvedores, juntamente com elaboração da documentação de usuário e possíveis planos de Transição.
Transição	Testes Beta	Realização dos testes de aceitação, no ambiente dos usuários.
	Operação Piloto	Operação experimental do produto em instalação piloto do cliente, com a resolução de eventuais problemas através de processo de manutenção

Tabela 3 - Detalhamento das fases do Praxis

1.2.1.3 Artefatos

Os resultados produzidos e os insumos consumidos nos passos do Praxis são chamados de artefatos do processo. Os artefatos podem ser documentos ou modelos, conforme seus consumidores primários sejam humanos ou ferramentas.

Um modelo é um artefato de uma ferramenta técnica específica (como uma ferramenta CASE, por exemplo), produzido e usado nas atividades de um dos fluxos do processo. Já um documento é um artefato produzido por um editor de textos, de relatórios ou de hipertexto, que pode ser consultado *on-line* ou em forma impressa, para fins de referência ou revisão.

Em relação ao processo, os artefatos podem ser classificados ainda como permanentes ou transitórios. Os artefatos permanentes são atualizados a cada iteração do processo, de acordo com procedimentos de gestão de configurações. Um conjunto de artefatos associados a um marco do projeto, consistentes entre si e conformes com os padrões do processo, constitui uma linha de base do projeto. As linhas de base são tipicamente montadas ao final de cada iteração, preservando-se assim um retrato completo do projeto em cada um destes instantes. Isto é muito importante para facilitar a localização posterior de defeitos.

As tabelas 4, 5 e 6 a seguir, também extraídas de [Paula01], apresentam os documentos, modelos e relatórios do Praxis:

Nome	Sigla	Descrição
Proposta de Especificação do Software	PESw	Documento que delimita preliminarmente o escopo de um projeto, contendo um plano da fase de Elaboração.
Especificação dos Requisitos do Software	ERSw	Documento que descreve, de forma detalhada, o conjunto de requisitos especificados para um produto de software.
Plano de Desenvolvimento do Software	PDSw	Documento que descreve, de forma detalhada, os compromissos que o fornecedor assume em relação ao projeto, quanto a recursos, custos, prazos, riscos e outros aspectos gerenciais.
Plano da Qualidade do Software	PQSw	Documento que descreve, de forma detalhada, os procedimentos de garantia da qualidade que serão adotados no projeto.
Descrição do Desenho do Software	DDSw	Documento que descreve, de forma detalhada, os aspectos mais importantes do desenho do software.
Descrição dos Testes do Software	DTSw	Documento que descreve, de forma detalhada, os planos e especificações dos testes que serão executados.
Manual do Usuário do Software	MUSw	Documento que serve de referência para uso do produto.

Tabela 4 - Documentos permanentes do Praxis

A Tabela 4 apresenta os documentos permanentes oficiais do Praxis. Os dois planos são considerados documentos gerenciais, enquanto os demais são documentos técnicos. Tipicamente, eles são produzidos através uma ferramenta de edição de textos. O formato destes documentos é conforme com os padrões do IEEE [IEEE94]. Estes padrões requerem documentos bastante detalhados, mas o processo inclui gabaritos (*templates*) que facilitam o preenchimento destes. Além disto, em alguns gabaritos são preenchidas previamente as partes que não variam muito entre projetos conformes com este processo; assim, só as exceções precisam ser documentadas.

Nome	Sigla	Descrição	Ferramentas aplicáveis
Cadastro dos Requisitos do Software	CRSw	Modelo que contém os requisitos levantados, assim como referências aos itens correspondentes dos modelos seguintes.	Planilha, banco de dados
Modelo de Análise do Software	MASw	Modelo que detalha os conceitos do domínio do problema a resolver, que sejam relevantes para a validação dos requisitos.	Ferramenta de modelagem orientada por objetos
Memória de Planejamento do Projeto do Software	MPPSw	Modelo que contém a informação necessária para o planejamento e acompanhamento de tamanhos, esforços, custos, prazos e riscos do projeto.	Planilha, ferramenta de gestão de projetos
Modelo de Desenho do Software	MDSw	Modelo que detalha a estrutura lógica e física do produto em termos de seus componentes.	Ferramenta de modelagem orientada por objetos
Bateria de Testes de Regressão do Software	BTRSw	Conjunto dos scripts dos testes de regressão.	Ferramenta de desenvolvimento, ferramenta de testes
Códigos Fontes do Software	CFSw	Conjunto dos códigos fontes produzidos.	Ferramenta de desenvolvimento
Códigos Executáveis do Software	CESw	Conjunto dos códigos executáveis produzidos.	Ferramenta de desenvolvimento

Tabela 5 - Modelos permanentes do Praxis

A Tabela 5 apresenta os modelos permanentes do Praxis. A Memória de Planejamento do Projeto do Software é o único modelo gerencial. A última coluna indica o tipo de ferramenta necessário para processamento de cada modelo. Em alguns casos, são apresentadas alternativas de menor ou maior sofisticação tecnológica.

Nome	Sigla	Descrição
Relatórios dos Testes do Software	RTSw	Conjunto dos relatórios que descrevem os resultados dos testes realizados.
Relatórios de Revisão do Software	RRSw	Conjunto dos relatórios que descrevem as conclusões das revisões realizadas.
Relatórios das Auditorias da Qualidade do Software	RAQSw	Conjunto dos relatórios que descrevem as conclusões das auditorias da qualidade realizadas.
Relatórios de Acompanhamento do Projeto do Software	RAPSw	Conjunto dos relatórios de acompanhamento do projeto, que relatam esforços, custos, prazos e riscos do período relatado, comparados com o que foi planejado.
Relatório Final do Projeto do Software	RFPSw	Relatório de balanço final do projeto.

Tabela 6 - Relatórios do Praxis

A Tabela 6 apresenta os relatórios do Praxis. Os dois primeiros são de caráter técnico, enquanto os demais de caráter gerencial. O plano da qualidade prevê as datas de emissão dos relatórios de testes, revisões e auditorias. Os relatórios de acompanhamento são produzidos com a periodicidade especificada no plano de desenvolvimento (geralmente mensal).

1.2.1.4 Procedimentos de controle

Os procedimentos de controle são executados em diferentes iterações do ciclo de vida de desenvolvimento de software. A conclusão destes procedimentos é uma condição necessária para que uma iteração ou fase seja considerada como finalizada. Os principais tipos de procedimentos de controle são: revisões técnicas, revisões gerenciais, auditorias da qualidade, avaliações por parte dos clientes e avaliações por parte dos usuários chaves. Maiores detalhes sobre os tipos de revisões existentes podem ser encontrados em [Paula01].

As revisões técnicas funcionam como principal forma de controle da qualidade quanto aos aspectos técnicos. Os pontos para realização das revisões técnicas foram definidos levando-se em consideração vários aspectos envolvidos, como o volume de material a ser submetido para revisão, os riscos que envolvem as atividades posteriores e o próprio custo de realização das revisões.

As revisões gerenciais determinam se uma atividade pode ser dada como concluída. Para tal determinação, o gerente do projeto ouve os membros da equipe envolvidos na atividade ou que possam ser afetados por ela. Em caso de rejeição, o gerente do projeto solicita que a equipe refaça a atividade e, em caso de aprovação, o gerente do projeto conduz um balanço da iteração e toma as providências necessárias para dar início às próximas atividades. O balanço da iteração se destina à determinação das lições mais importantes aprendidas até o momento, que servirão de base para a proposição de melhorias do processo em projetos futuros.

As auditorias da qualidade são geralmente feitas por um grupo independente de Garantia da

Qualidade. Este grupo verifica principalmente a conformidade das atividades realizadas com os padrões definidos pelo processo. Este grupo também verifica a conformidade dos procedimentos de gestão de configurações, a consistência entre os artefatos do processo e a rastreabilidade entre os requisitos e os demais artefatos do processo.

As aprovações dos clientes geralmente são necessárias em momentos que envolvem a decisão de se continuar ou não o projeto, como no fim das fases de Concepção e Elaboração. São necessárias também no momento da aceitação formal do produto, que se dá ao final das fases de Construção e Transição. Já as aprovações por usuários chaves geralmente são feitas para verificar se, em um determinado estágio da construção, o produto atende às necessidades dos usuários. Nestas avaliações, pretende-se garantir que os requisitos tenham sido totalmente compreendidos pelos desenvolvedores.

1.2.2 Ferramentas CASE

1.2.2.1 Definição

O termo *CASE*, *Computer Aided Software Engineering*, é utilizado para designar ferramentas de auxílio automatizado à Engenharia de Software. Estas ferramentas são empregadas de forma a reduzir significativamente o custo de produção e o tempo de desenvolvimento dos sistemas, além de possibilitar um aumento na qualidade e produtividade dos mesmos.

A história das ferramentas CASE começou no início dos anos 70 com simples processadores de textos usados para gerar a documentação dos projetos. Na década seguinte, apareceram as primeiras ferramentas gráficas para desenho de diagramas. No entanto, rapidamente surgiu a necessidade de se guardar informações a respeito dos dados e da lógica do sistema. Além disso, recursos de verificação e consistência de modelos eram desejados. No final da década de 80, apareceram as primeiras ferramentas que já possuíam essas características e realmente automatizavam parte das tarefas de desenvolvimento.

Atualmente, as ferramentas CASE existentes no mercado já abrangem uma parte bem maior das tarefas de modelagem e desenvolvimento de sistemas. Elas cobrem desde a geração de especificação a partir do modelo até a geração de código. Maiores informações sobre o assunto podem ser encontradas em [McClure89].

1.2.2.2 Vantagens

O uso de ferramentas CASE no desenvolvimento de sistemas pode trazer as seguintes vantagens:

- aumento da praticidade no uso dos processos de desenvolvimento: a utilização de um processo, apesar de importante, tem um nível de dificuldade alto que prejudica sua correta aplicação. As ferramentas CASE levam este fato em consideração e visam sempre aumentar a praticidade no uso do processo;
- aumento da qualidade dos produtos finais: as ferramentas CASE diminuem a probabilidade de erros, uma vez que podem ajudar no controle da consistência dos dados. Também

proporcionam maior eficácia dos produtos, ao auxiliarem os fluxos de Análise e Testes do produto;

- aumento da produtividade: ao ajudar na realização de tarefas e até mesmo automatizar algumas delas, as ferramentas CASE contribuem para uma maior agilidade no desenvolvimento de software;
- eliminação de trabalho monótono, com a conseqüente liberação dos desenvolvedores para trabalhos mais criativos: as ferramentas CASE podem realizar algumas tarefas cansativas para os desenvolvedores, tais como procurar informações e desenhar símbolos de um diagrama e deixá-los liberados para executar as tarefas mais criativas e subjetivas do processo de desenvolvimento de sistemas;
- flexibilidade e agilidade para mudanças: as ferramentas CASE permitem que os dados e diagramas sejam alterados de maneira mais rápida e fácil, o que ajuda o desenvolvedor no trabalho de atender às necessidades do usuário;
- melhoria da documentação: por armazenarem dados e diagramas, as ferramentas CASE também contribuem para uma melhor documentação do sistema, gerando relatórios, agilizando a busca e a alteração das informações;
- simplificação na manutenção de sistemas: por conseqüência do item anterior, fica mais fácil obter informações sobre o problema a ser resolvido na hora de realizar sua manutenção (correção, atualização ou expansão). Além disso, a atualização automática da documentação (através de recursos como engenharia reversa) faz com que ela corresponda sempre ao estado corrente do sistema desenvolvido. Por último, pode-se destacar recursos como redesenho, que possibilitam a algumas ferramentas CASE atualizarem o código com base em alterações no desenho;
- facilidade para o desenvolvimento incremental de sistemas: através da documentação e do redesenho de sistemas, as ferramentas CASE facilitam bastante o desenvolvimento incremental e iterativo. Deve-se destacar que este aspecto está completamente de acordo com as tendências mais atuais, conforme pode-se ver em [Jacobson97];

1.2.2.3 Taxonomia

As diversas ferramentas CASE existentes suportam diferentes fases do ciclo de vida do desenvolvimento de sistemas. Segundo McClure [McClure89], para facilitar o entendimento, a distinção e a avaliação destas ferramentas, elas podem ser classificadas da seguinte forma:

- *CASE Toolkits*: tipo mais simples de ferramentas CASE. Representam um conjunto de ferramentas integradas que automatizam uma tarefa/fase no ciclo de desenvolvimento de software, seja no desenho, seja na implementação;
- *CASE Workbenches*: conjunto de ferramentas que automatizam tarefas durante todo o processo de desenvolvimento de software, não compreendendo apenas uma área localizada;

- *CASE Methodology companion*: oferecem assistência computadorizada para um processo particular de desenvolvimento, que pode ser o Objectory [Jacobson97] ou o Processo Unificado [Jacobson+99], por exemplo.

1.2.2.3.1 CASE Toolkits

Os *CASE Toolkits* podem ser descritos como ferramentas em nível de fases por se concentrarem em uma determinada fase do ciclo de desenvolvimento de sistemas. Existem *CASE Toolkits* específicas para análise, programação e manutenção. Costumam, inclusive, participar de famílias de produtos para desenvolvedores, sendo produzidas e comercializadas em separado por opção mercadológica. Normalmente, são soluções para arquiteturas abertas, não exigindo hardware específico. Mesmo como produtos, pode-se tratá-las, em sua maioria, como produtos abertos: normalmente são integráveis entre si, mesmo quando produzidas por diferentes empresas. (A integração com outros produtos é uma característica louvável em qualquer produto e com ferramentas CASE isto não é diferente).

Alguns exemplos de ferramentas CASE nesta categoria são os geradores puros de código e as ferramentas para diagramação na análise de sistemas. A ferramenta de extração da documentação *javadoc* da linguagem Java também é um bom exemplo de *CASE Toolkits*.

1.2.2.3.2 CASE Workbenches

Os *CASE Workbenches* são conjuntos integrados de ferramentas para diversas tarefas na área de desenvolvimento de sistemas. Também conhecidos como I-CASE (*Integrated CASE*), costumam abranger e automatizar todas as tarefas numa determinada abordagem sistêmica (exemplo: abordagem dos dados que o sistema manipula).

Alguns exemplos clássicos de ferramentas desta categoria são produtos que abrangem desde a modelagem dos dados do sistema (via diagramas da UML, por exemplo) até a geração de código e implementação de banco de dados em uma determinada plataforma. Um exemplo deste tipo de ferramentas é o *Rational Rose*, da *Rational Software Corporation*. O *Rose* compõe o *Rational Suíte*, uma família de ferramentas integradas da *Rational* que oferecem suporte para todo o ciclo de desenvolvimento de software [RationalSuite00]. O *Rose* permite a modelagem dos dados, a geração automática de código em várias linguagens e de bancos de dados em diversas plataformas. Permite ainda, para o caso do código, o caminho inverso: a engenharia reversa do código de algumas linguagens para seu modelo.

1.2.2.3.3 CASE Methodology Companions

Os *CASE Methodology Companions* não são mais ou menos abrangentes que as categorias citadas anteriormente: são apenas especializações das mesmas. Pode-se ter *CASE Methodology Companions* integrados (I-CASE) ou não.

As ferramentas desta categoria estruturam o desenvolvimento de sistemas de acordo com os passos previstos em algum processo de desenvolvimento. Informações sobre este processo estão sempre disponíveis no produto e os passos são conduzidos por ele.

Um exemplo desse tipo de ferramenta é o RUP (*Rational Unified Process*), também da

empresa *Rational Software Corporation*, que oferece apoio ao Processo Unificado e faz parte da família *Rational Suite*. O RUP consiste numa documentação *on-line* que inclui práticas de desenvolvimento de software que guiam a equipe em todas as etapas do desenvolvimento. Ele provê diretrizes (*guidelines*), gabaritos (*templates*) e *Tool Mentors* (instruções sobre como aplicar as diretrizes utilizando as ferramentas específicas da *Rational*) para cada fase do ciclo de vida do desenvolvimento de software. Informações sobre este produto podem ser encontradas em [RUP00].

1.3 Proposta da ferramenta CASE Praxis Mentor

1.3.1 O Praxis Mentor

A proposta deste trabalho é a concepção de uma ferramenta CASE classificada como um *CASE Methodology Companion*. Ela consiste em uma ferramenta de apoio à utilização do processo Praxis em projetos de desenvolvimento de software. Esta ferramenta atende a todas as fases do processo utilizado, fornecendo informações sobre cada uma delas e auxiliando o usuário do processo na utilização do mesmo. Os módulos principais da ferramenta são: Consulta ao Processo, Composição da Equipe, Comunicação, Controle de Projeto, Proposição de Melhorias e Gestão de Problemas.

A Especificação dos Requisitos da ferramenta é parte deste trabalho e define todas as características da mesma. Uma das ferramentas que foram analisadas durante a fase de especificação e serviram de modelo para o Praxis Mentor foi o RUP. Foram analisadas também as ferramentas de suporte ao PROSE, PSP e TSP. Estas ferramentas apresentam os seguintes pontos fortes:

- o RUP fornece orientação on-line aos desenvolvedores;
- o SAPP (ferramenta de apoio ao PROSE) fornece suporte gerencial;
- as ferramentas disponíveis para o PSP e TSP oferecem suporte para coleta de métricas.

Todo o trabalho de concepção, elaboração e construção do Praxis Mentor foi feito utilizando-se o processo Praxis. Esse fato foi de extrema importância para o projeto pois possibilitou um conhecimento profundo das características do processo e das maiores dificuldades enfrentadas por seus usuários.

1.3.2 Resultados esperados

Ao final desse projeto, espera-se obter uma ferramenta visual que auxilie os usuários do Praxis. Ela deverá oferecer suporte a todas as fases do processo:

- instruindo o desenvolvedor no uso do processo;
- fornecendo acesso fácil a todos os documentos e modelos utilizados em cada fase;

- facilitando a gestão do projeto e do processo;

Além dessa ferramenta, espera-se obter uma avaliação do processo Praxis, uma vez que ele está sendo utilizado durante o desenvolvimento deste próprio projeto. A análise desses resultados poderá ser vista no Capítulo 6 (Conclusão e Trabalhos Futuros).

1.4 Organização deste documento

Para proporcionar uma visão temporal do desenvolvimento da ferramenta e explicitar as atividades e fases do processo, este documento foi organizado de acordo com a estrutura do mesmo. Os capítulos foram divididos entre as fases do processo e as conclusões do trabalho. A estrutura está definida a seguir:

- **Capítulo 2 - Concepção:** este capítulo contém uma descrição da fase de concepção do projeto da ferramenta Praxis Mentor. Ele apresenta o conteúdo da Proposta de Especificação do Software (PESw) da ferramenta, com o detalhamento do levantamento de requisitos elaborado nesta fase;
- **Capítulo 3 - Elaboração:** este capítulo apresenta as atividades realizadas durante a fase de Elaboração da ferramenta. Tais atividades consistem no levantamento detalhado dos requisitos e na análise dos mesmos. Nele são mostrados os itens mais importantes dos artefatos gerados durante esta fase: a Especificação dos Requisitos do Software (ERSw), o Plano de Desenvolvimento do Software (PDSw) e o Plano da Qualidade do Software (PQSw);
- **Capítulo 4 - Construção:** este capítulo apresenta as atividades realizadas durante a fase de Construção da ferramenta. Ele contém partes do artefato Descrição do Desenho do Software (DDSw), que consiste no desenho detalhado da ferramenta no qual se encontram todas as decisões de implementação deste projeto. Outro artefato apresentado parcialmente neste capítulo é o Manual do Usuário do Software (MUSw), que provê os conceitos básicos do Praxis Mentor e a descrição detalhada de algumas funções do produto. Este capítulo mostra também uma breve descrição das tecnologias envolvidas nesta fase do projeto, para que se tenha uma visão melhor da importância e conseqüente escolha das mesmas;
- **Capítulo 5 - Transição:** este capítulo descreve a fase de Transição da ferramenta. Na verdade, ele contemplará as atividades que deverão ser realizadas em um trabalho futuro, visto que esta fase não chegou a ser realizada neste projeto;
- **Capítulo 6 - Conclusão e Trabalhos Futuros:** este último capítulo traz as conclusões do trabalho, uma discussão sobre os assuntos abordados e sobre a ferramenta desenvolvida e os trabalhos futuros que poderão ser feitos na área.

Capítulo 2

Concepção

2.1 Definição

A fase de Concepção é composta de apenas uma iteração: a Ativação. Nesta iteração, são realizados o levantamento e a análise das necessidades dos usuários e dos conceitos da aplicação, em nível de detalhe suficiente para justificar a especificação de um produto de software.

Durante a fase de Concepção, é estabelecida uma Proposta de Especificação do Software (PESw) e delimitado o escopo do projeto. Para a elaboração desta proposta, algumas atividades são realizadas:

- definição do escopo do projeto;
- levantamento preliminar dos requisitos;
- levantamento das metas gerenciais (limites de prazo e custo aceitáveis para o cliente);
- análise de riscos a serem assumidos;
- levantamento dos recursos necessários para o projeto;
- estimativas de custo e prazo para a fase de Elaboração.

Durante esta fase, é comum também a construção de protótipos para validar o modelo conceitual feito para o sistema. Pode ser interessante ainda a criação de um esboço da arquitetura do produto, com a definição da estrutura do mesmo e uma proposta de tecnologias candidatas para o projeto.

A Proposta de Especificação do Software é um artefato produzido de acordo com o Padrão de Proposta de Especificação do Software apresentado em [Paula01]. Ela é baseada em documentos externos ao projeto e deve definir o escopo do mesmo, as taxas de risco, a estimativa de recursos necessários, o plano de trabalho mostrando as datas principais de entrega de resultados e o planejamento detalhado da fase de Elaboração. No caso de ter sido feito um estudo aprofundado da arquitetura do projeto e das tecnologias candidatas, a proposta deve conter também os resultados do estudo.

Ao término desta fase, são examinados os objetivos, prazos, custos e riscos do projeto para

se decidir sobre a continuidade do desenvolvimento. No caso do Praxis, para que o projeto tenha continuação, é necessária a aprovação da Proposta de Especificação do Software pelo cliente.

2.2 Concepção do Praxis Mentor

2.2.1 Modelagem de Processos de Negócio

Na fase de Concepção do Praxis Mentor, foi realizado um estudo sobre o processo para facilitar o levantamento de requisitos. A partir deste estudo, foi gerada uma Modelagem de Processos de Negócio para o Praxis, que revelou os aspectos mais importantes do processo. Esta modelagem foi feita a partir do caso de uso de negócio “Gerir projeto (Construção e Transição)”, que representa o fluxo de trabalho principal do Praxis após a fase de Elaboração. A descrição deste caso de uso é apresentada na Tabela 7:

<p>Precondição:</p> <ul style="list-style-type: none">✓ O projeto deve ter ERSw aceita pelo cliente, além do PDSw e PQSw já elaborados. <p>Fluxo Principal:</p> <ul style="list-style-type: none">✓ O gerente de projeto cria o projeto.✓ O gerente de projeto designa a equipe inicial do projeto.✓ O gerente de projeto cria a primeira linha de base (com ERSw, PDSw e PQSw).✓ Para cada iteração do processo:<ul style="list-style-type: none">• O gerente de projeto aciona os desenvolvedores.• Os desenvolvedores produzem os resultados da iteração.• O gerente de projeto aciona os procedimentos de controle.• Se a iteração for aprovada nos procedimentos de controle, o gerente de projeto aciona criação de linha de base.• Se for necessário um replanejamento, o gerente de projeto aciona o replanejamento.✓ O gerente de projeto aciona criação de linha de base de produto.✓ O gerente de projeto aciona elaboração do relatório final do projeto. <p>Subfluxos:</p> <ul style="list-style-type: none">✓ Procedimentos de controle<ul style="list-style-type: none">• Revisão técnica• Revisão gerencial• Aprovação pelo cliente• Auditoria da qualidade✓ Criação de linha de base✓ Replanejamento <p>Fluxos Alternativos:</p> <ul style="list-style-type: none">✓ Relatório de acompanhamento (acionado ao fim de cada mês).✓ Alteração de linha de base✓ Alteração de equipe✓ Resolução de problemas
--

Tabela 7 - Caso de uso de negócio “Gerir Projeto (Construção e Transição)”

Os diagramas de atividades apresentados a seguir contemplam o fluxo de trabalho do Praxis descrito neste caso de uso. Eles foram elaborados de forma a refletir todas as atividades de cada fluxo, subfluxo e fluxo alternativo do caso de uso em questão. Dessa forma, foi feito um diagrama de atividades para o fluxo principal e outros diagramas para cada subfluxo e fluxo alternativo existente no caso de uso.

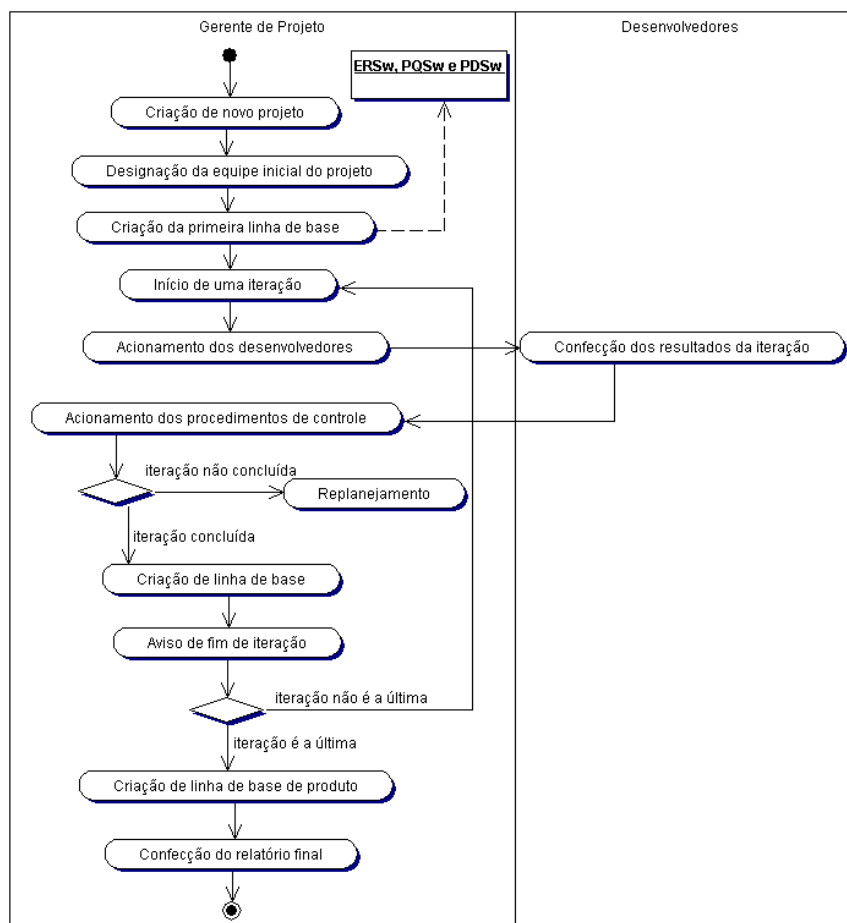


Figura 2 - Diagrama de Atividades – Fluxo Principal

O diagrama da Figura 2 apresenta as atividades do Fluxo Principal do Praxis. O fluxo é iniciado através da **Criação de novo projeto**. Em seguida, o gerente compõe a equipe do projeto através da atividade **Designação da equipe inicial do projeto**.

A **Criação da primeira linha de base** submete os primeiros artefatos produzidos à Gestão de Configurações do projeto. A partir daí, se tem o **Início de uma iteração**. Através do **Acionamento dos desenvolvedores**, tem-se o início das atividades da iteração. Após a **Confecção dos resultados da iteração**, acontece o **Acionamento dos procedimentos de controle**. Se os resultados não forem aprovados pelos procedimentos de controle, dá-se início a um **Replanejamento** e o Gerente solicita que os desenvolvedores refaçam o trabalho. Se os resultados forem aprovados, tem-se a **Criação de linha de base** com os resultados da iteração e o **Aviso de fim de iteração**. Se a iteração não for a última, tem-se o início da próxima iteração e as atividades deste ciclo se repetem. Se a iteração for a última, faz-se a **Criação de linha de base de produto** e a **Confecção do relatório final** com a síntese dos resultados do projeto e as lições

aprendidas para serem usadas na melhoria do processo em projetos futuros.

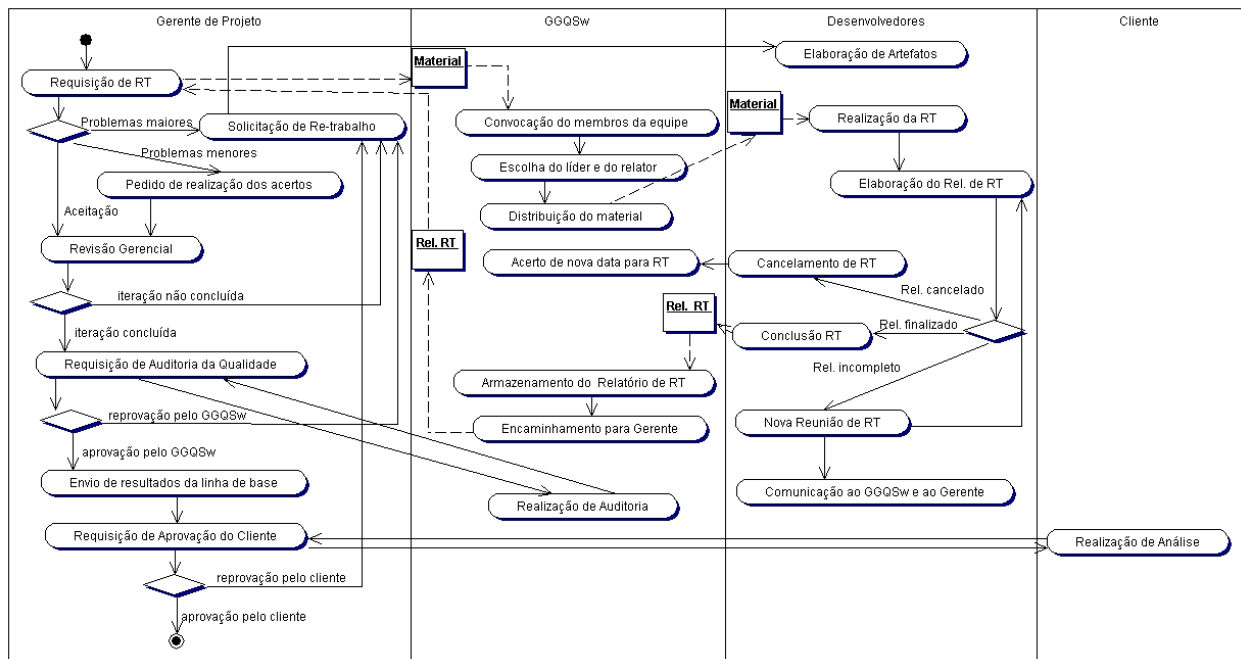


Figura 3 - Diagrama de Atividades – Subfluxo Procedimentos de Controle

A Figura 3 apresenta as atividades do subfluxo Procedimentos de Controle. Neste diagrama, é mostrada a ativação dos quatro tipos de procedimentos de controle existentes no Praxis: revisão técnica, revisão gerencial, auditoria da qualidade e aprovação pelo cliente ou usuários chaves.

O diagrama mostra todas as atividades envolvidas quando os artefatos submetidos aos procedimentos de controle são aprovados e rejeitados. Mostra também como é organizada uma revisão técnica e quais membros da equipe estão envolvidos na sua organização.

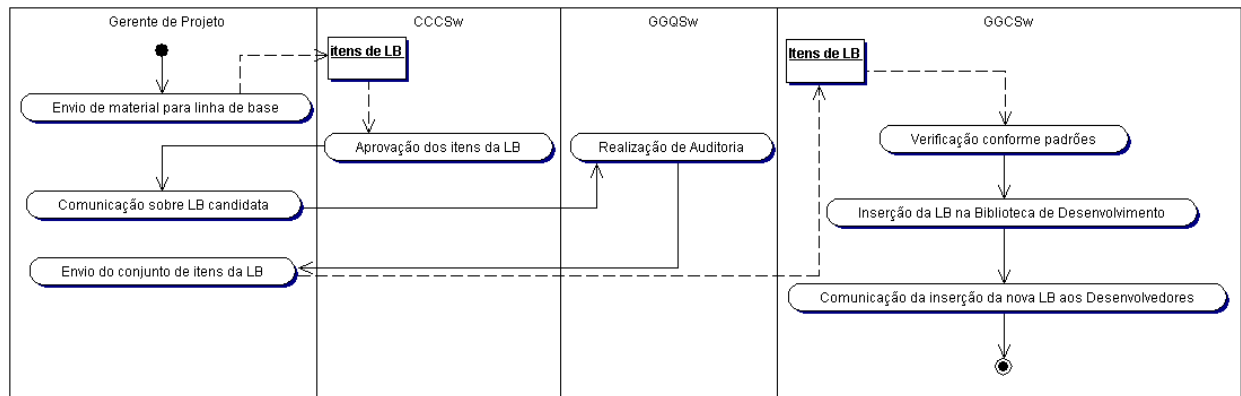


Figura 4 - Diagrama de Atividades – Subfluxo Criação de Linha de Base

A Figura 4 apresenta o diagrama referente ao subfluxo Criação de Linha de Base. Ele contempla todas as atividades envolvidas no momento da criação de uma linha de base. Tais atividades consistem no pedido de criação a partir do **Envio de material para a linha de base**; na aprovação dos itens pelos membros da Comissão de Controle de Configurações de Software (CCCSw), do Grupo da Garantia da Qualidade de Software (GGQSw) e do Grupo de Gestão de

Configurações de Software (GGCSw), através da **Aprovação dos itens da LB**, da **Realização de Auditoria** e da **Verificação conforme os padrões**, respectivamente. Finalmente, o fluxo é concluído com a **Inserção da LB na Biblioteca de Desenvolvimento** e com a **Comunicação da inserção da nova LB aos Desenvolvedores**.

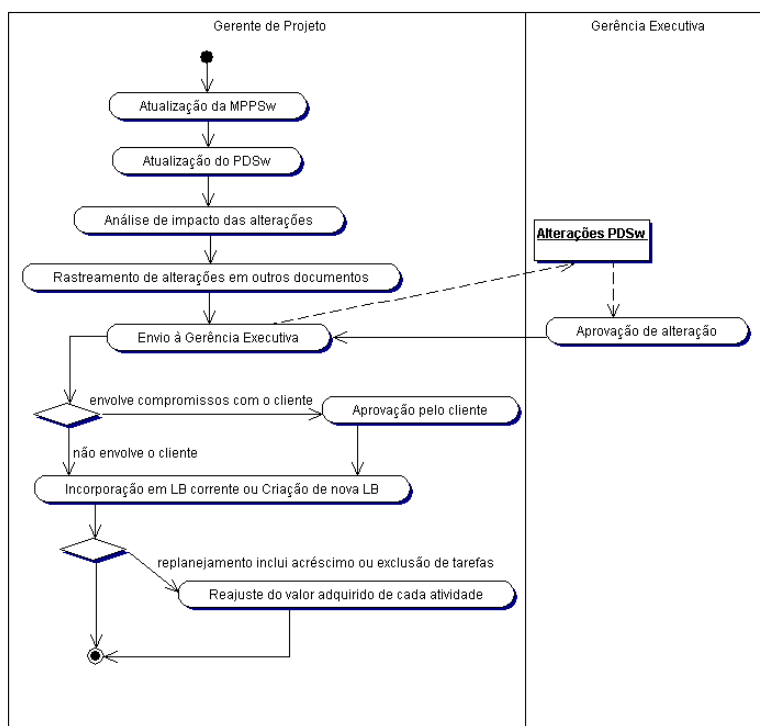


Figura 5 - Diagrama de Atividades – Subfluxo Replanejamento

A Figura 5 apresenta o subfluxo Replanejamento. Neste diagrama, são mostradas todas as atividades envolvidas no subfluxo. Ele se inicia com o novo cálculo das métricas do software e a **Atualização da MPPSw** (Memória de Planejamento do Projeto do Software) e a **Atualização do PDSw** (Plano de Desenvolvimento do Software). A **Análise de impacto das alterações** e o **Rastreamento de alterações em outros documentos** garantem a verificação da viabilidade e a manutenção da consistência das alterações, respectivamente. Após o **Envio à Gerência Executiva**, deve ocorrer a **Aprovação de alteração** por parte da gerência e a **Aprovação pelo cliente**, caso as alterações envolvam compromissos com o mesmo. Depois, deve ser feita a **Incorporação em Linha de Base corrente ou Criação de nova Linha de Base**. Se o replanejamento acrescentar ou excluir tarefas, deve ser realizado o **Reajuste do valor adquirido de cada atividade** para que este valor possa refletir a nova porcentagem do esforço total.

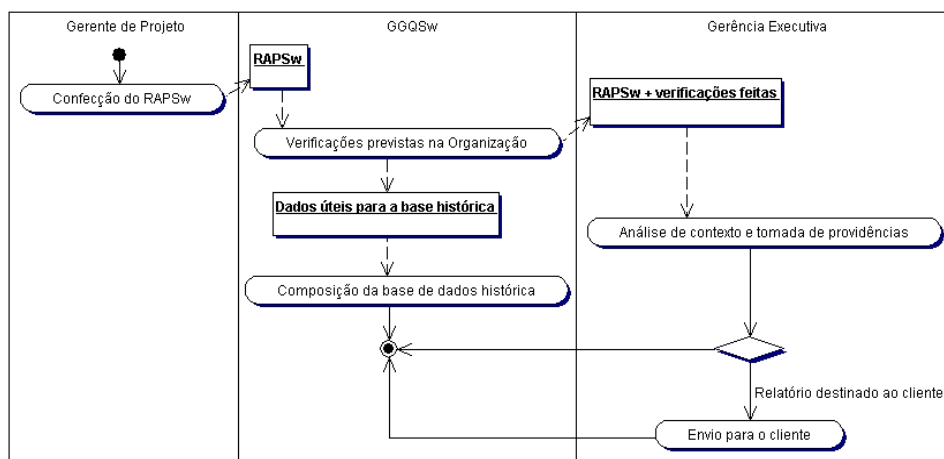


Figura 6 - Diagrama de Atividades – Fluxo Alternativo Relatório de Acompanhamento

A Figura 6 apresenta o fluxo alternativo Relatório de Acompanhamento. O diagrama contempla as atividades previstas quando da confecção do relatório. Após a **Confecção do RAPS w** (Relatório de Acompanhamento do Projeto do Software), devem ser feitas as **Verificações previstas na Organização** pelo GGQSw. Após tais verificações, o GGQSw deve selecionar os dados úteis para a base histórica da Organização e proceder com a **Composição da base de dados histórica**. As verificações e o RAPS w devem ser submetidos à Gerência Executiva para que seja realizada uma **Análise de contexto e tomada de providências**. Se o relatório for destinado ao cliente, deve ser feito o **Envio para o cliente** para que ele fique a par do andamento do projeto.

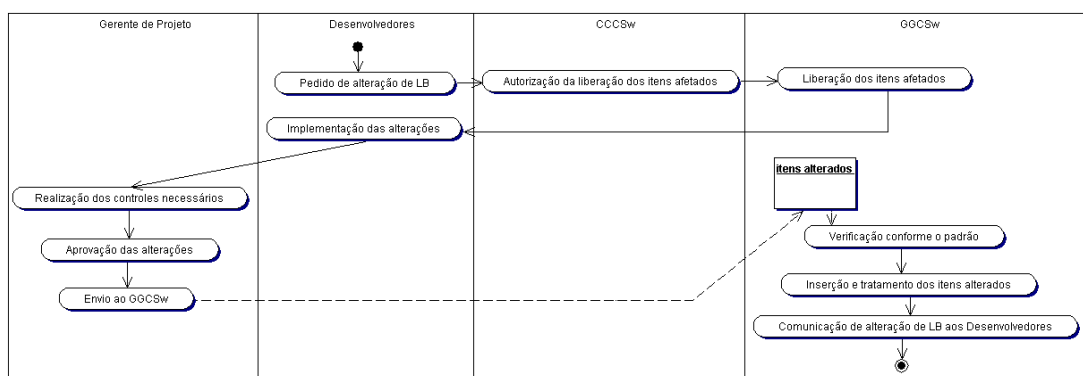


Figura 7 - Diagrama de Atividades – Fluxo Alternativo Alteração de Linha de Base

A Figura 7 apresenta o fluxo alternativo Alteração de Linha de Base. O diagrama contém todas as atividades que devem ser executadas para que a alteração de uma linha de base seja realizada. Este fluxo se inicia quando os Desenvolvedores fazem o **Pedido de Alteração de LB** para a CCCSw. Esta comissão dá a **Autorização da liberação dos itens afetados** e o GGCSw executa a **Liberação dos itens afetados**. A partir daí, os Desenvolvedores podem dar início à **Implementação das alterações**. O Gerente de Projeto deve fazer a **Realização dos controles necessários** e a **Aprovação das alterações**. Após esta aprovação, ocorre o **Envio ao GGCSw** para que este grupo realize a **Verificação conforme o padrão** e a **Inserção e tratamento dos itens alterados**. Após o encerramento da alteração, este grupo deve proceder com a

Comunicação de alteração de LB aos Desenvolvedores.

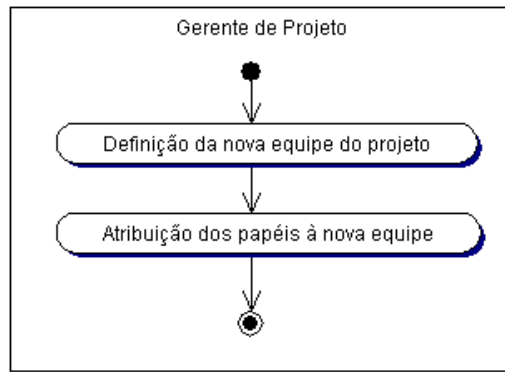


Figura 8 - Diagrama de Atividades – Fluxo Alternativo Alteração de Equipe

A Figura 8 apresenta o fluxo alternativo Alteração de Equipe. Este fluxo é bem simples e contempla as atividades envolvidas na composição da equipe do projeto. Tais atividades consistem na **Definição da nova equipe do projeto** pelo Gerente, com a inclusão ou exclusão de membros, e na **Atribuição dos papéis à nova equipe**, com uma redistribuição de papéis caso necessário.

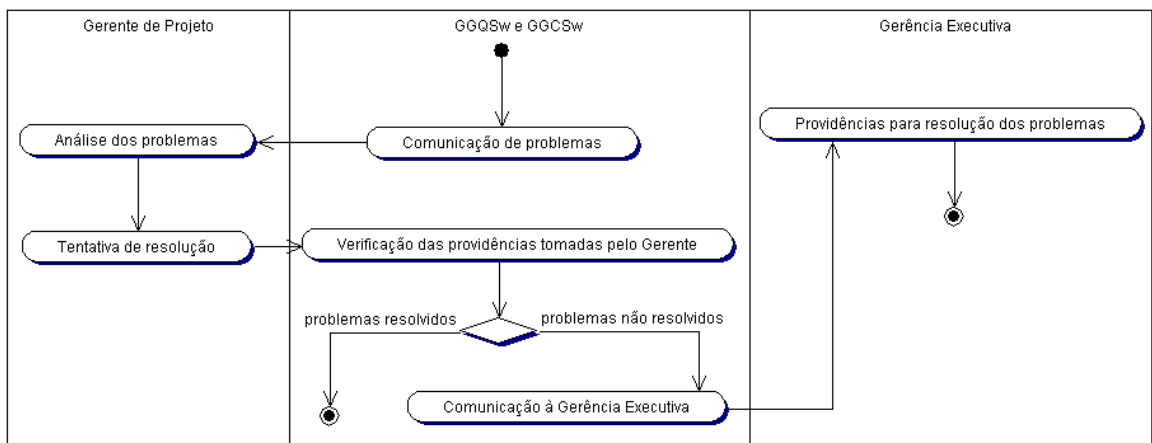


Figura 9 - Diagrama de Atividades – Fluxo Alternativo Resolução de Problemas

A Figura 9 apresenta o fluxo alternativo Resolução de Problemas. O diagrama mostra as atividades envolvidas no fluxo que se inicia com a **Comunicação de problemas** ao Gerente de Projeto. O Gerente faz a **Análise dos problemas** e a **Tentativa de resolução**. O grupo que levantou o problema realiza uma **Verificação das providências tomadas pelo Gerente** e repassa os problemas considerados como não resolvidos para a Gerência Executiva que toma as **Providências para resolução dos problemas**.

2.2.2 Proposta de Especificação do Software

O levantamento de requisitos do Praxis Mentor foi elaborado após a finalização da modelagem do negócio. A partir da análise dos requisitos encontrados, foi criada a Proposta de Especificação do Software para o Praxis Mentor, conforme os padrões de proposta do Praxis. Os principais tópicos deste documento são apresentados nas seções que se seguem.

2.2.2.1 Missão do produto

Segundo Paula [Paula01], a missão descreve os objetivos do produto a ser desenvolvido. Ela sintetiza o valor que o produto acrescenta para o cliente e para os usuários. A definição da missão deve delimitar as responsabilidades e o escopo do produto e sintetizar o compromisso assumido com o cliente.

A missão do Praxis Mentor está definida na Tabela 8.

O produto Praxis Mentor 1.0 visa oferecer apoio informatizado à utilização do processo Praxis em projetos de desenvolvimento de software.

Tabela 8 - Missão do Praxis Mentor

2.2.2.2 Lista de funções

A partir da modelagem de processos de negócio e da análise das tarefas exercidas pelos membros da equipe do projeto, foram extraídas as funções que o produto deveria conter. Foram identificadas funções para auxílio ao aprendizado do processo, à organização dos artefatos gerados, à realização de algumas tarefas gerenciais e às propostas de melhorias para o processo.

É fácil notar que foi dado um enfoque bem gerencial para o produto. A decisão por este enfoque gerencial foi tomada pelo fato de existir uma necessidade maior por uma ferramenta que auxiliasse no aprendizado e aplicação do processo e na organização do projeto. Ferramentas para suporte técnico existem muitas e a maioria delas atende de forma satisfatória às necessidades técnicas do projeto. Inclusive, a idéia para uma próxima versão do Praxis Mentor é a integração com as melhores ferramentas técnicas existentes no mercado e não a implementação das funcionalidades disponibilizadas por elas.

A lista das funções identificadas para o Praxis Mentor está resumida na Tabela 9. Esta tabela apresenta quais necessidades do cliente e dos usuários⁴ deram origem a determinada função e quais benefícios são gerados por ela.

⁴ O cliente deste projeto é o orientador do mesmo, professor Wilson de Pádua Paula Filho. Os usuários são todas as pessoas que hoje utilizam o processo Praxis e que foram consultadas e forneceram suas opiniões para este trabalho.

Número de ordem	Nome da função	Necessidades	Benefícios
1	Orientação on-line ao uso do Processo	Conhecimento do processo e de sua forma de utilização	<ul style="list-style-type: none"> - Capacitação do usuário no uso do processo - Agilidade na utilização do processo - Diminuição dos erros no uso do processo
2	Acesso aos documentos e modelos gerados pelo projeto	Organização dos artefatos do projeto Padronização de uma estrutura de pastas para projetos	<ul style="list-style-type: none"> - Agilidade na recuperação de qualquer artefato do projeto - Facilidade na realização de cópias de segurança do projeto
3	Suporte Gerencial (Gestão do Projeto e Cálculo de métricas)	Planejamento de Projetos Conhecimento das métricas do processo	<ul style="list-style-type: none"> - Capacitação dos gerentes no uso do processo - Maior agilidade na tarefa de planejamento de um projeto - Maior facilidade no cálculo das métricas do processo - Diminuição dos erros no cálculo de métricas
4	PIP (<i>Process Improvement Proposal</i>)	Histórico sobre o uso do processo	<ul style="list-style-type: none"> - Auxílio na personalização do processo - Contribuição para a melhoria do processo

Tabela 9 - Funções do Praxis Mentor

Todas as funções descritas, exceto as referentes à coleta de métricas, foram consideradas essenciais e devem ser desenvolvidas.

2.2.2.3 Outros aspectos

Segundo Paula [Paula01], esta seção deve incluir algumas informações de valor estratégico, tais como: limitações do escopo do produto, possíveis interfaces com outros produtos e questões pendentes a serem esclarecidas durante a especificação dos requisitos.

Nesta seção, foram identificadas algumas restrições para o Praxis Mentor impostas pelo cliente. A restrição quanto à tecnologia deve-se à necessidade de se ter um produto moderno, multiplataforma e que possa ser utilizado por várias pessoas da equipe ao mesmo tempo.

O produto deverá atender a todas as fases do processo, fornecendo informações sobre cada uma delas e auxiliando o usuário do processo na aplicação do mesmo. Ele poderá também automatizar algumas tarefas de cada fase.

O produto deverá ser desenvolvido utilizando HTML ou alguma outra tecnologia como, por exemplo, Java ou ASP. Um projeto piloto de avaliação deverá ser realizado para apoiar a escolha das tecnologias.

Tabela 10 - Outros aspectos do Praxis Mentor

Capítulo 3

Elaboração

3.1 Definição

A fase de Elaboração se inicia após a aprovação da Proposta de Especificação do Software pelo cliente. Ela é composta de duas iterações: o Levantamento e a Análise dos Requisitos. Na iteração de Levantamento dos Requisitos, é realizada a captura das necessidades do cliente em relação ao produto, que consiste num detalhamento das funções, interfaces e requisitos não funcionais desejados para o mesmo. Tal detalhamento é expresso na linguagem dos usuários. Já na Análise, é produzido um modelo conceitual do produto que serve para a validação dos requisitos levantados e para o planejamento detalhado da fase de Construção.

3.1.1 Levantamento dos Requisitos

O Levantamento dos Requisitos deve ser detalhado o suficiente para que clientes, usuários e desenvolvedores cheguem a um consenso quanto ao resultado. Durante esta iteração, devem ser elaborados o corpo do artefato Especificação dos Requisitos do Software (ERSw) e a descrição dos requisitos funcionais através dos casos de uso, o que formará a primeira versão do Modelo de Análise do Software (MASw).

O fechamento desta iteração normalmente se dá através de uma revisão gerencial, pois neste momento ainda não se tem o Modelo de Análise completo e é preferível realizar uma revisão técnica formal após a elaboração desse modelo. Os requisitos levantados (interfaces, casos de uso e requisitos não funcionais) são armazenados no Cadastro de Requisitos do Software (CRSw), que futuramente os amarrará com os respectivos elementos derivados nos demais fluxos.

3.1.2 Análise dos Requisitos

Enquanto o Levantamento dos Requisitos é voltado para a visão que os clientes e usuários têm dos requisitos do produto, a Análise dos Requisitos focaliza a visão dos desenvolvedores. Entretanto, o processo ainda está no escopo da definição do problema e não no escopo das soluções. Durante a iteração Análise dos Requisitos, deve ser finalizado o Modelo de Análise do Software (MASw), que utiliza a notação orientada por objetos para descrever de forma mais

precisa os conceitos da aplicação relevantes para o domínio do problema.

As atividades do fluxo de Análise desta iteração levam à identificação de classes, atributos e relacionamentos que representam precisamente os conceitos expressos nos requisitos. Estes elementos compõem o modelo lógico dos dados (como o modelo Entidade-Relacionamento) e podem corresponder também a um modelo conceitual de um banco de dados a ser usado pelo produto. As classes aqui levantadas devem ser cadastradas no Cadastro de Requisitos do Software.

Após o encerramento da identificação dos elementos citados, é então realizado o detalhamento das operações das classes. Tais operações serão usadas para produzir as realizações dos casos de uso, que definem os fluxos dos mesmos em função das interações entre as classes identificadas.

Em alguns casos, é interessante a elaboração de um protótipo para validar os requisitos levantados e o modelo conceitual feito para o sistema. Se este protótipo já tiver sido feito na fase de Concepção, ele pode ser complementado com os novos requisitos encontrados na fase de Elaboração. Pode ser interessante ainda a criação de um esboço da arquitetura do produto, com a definição da estrutura do mesmo e uma proposta das tecnologias candidatas para o projeto, caso isto não tenha sido feito na fase de Concepção. Caso contrário, pode-se complementar o estudo já realizado e fazer uma análise de viabilidade das soluções propostas.

Durante esta iteração, o artefato Especificação dos Requisitos do Software (ERSw) deve ser finalizado. Neste momento, ele deve ser confiável e completo o suficiente para servir de base para o planejamento da fase de Construção. Este planejamento é realizado através da produção da Memória de Planejamento do Projeto do Software (MPPSw), que contém a informação necessária para a definição e o acompanhamento de tamanho, esforço, custo, prazo e riscos do projeto. A partir desta memória de planejamento, é elaborado o artefato Plano de Desenvolvimento do Software (PDSw), que descreve, de forma detalhada, os compromissos assumidos com o cliente. Além deste plano, já existe informação o suficiente para a elaboração de um planejamento para o grupo de garantia da qualidade, a ser expresso no artefato Plano da Qualidade do Software (PQSw). Todos estes artefatos devem ser produzidos segundo os respectivos padrões apresentados em [Paula01].

O fechamento desta iteração, que também representa o encerramento da fase de Elaboração, é de extrema importância e responsabilidade, uma vez que a decisão sobre a continuação ou não do projeto é tomada com base nos artefatos gerados até esse momento. De acordo com o Praxis, os seguintes procedimentos de controle devem ser realizados:

- revisão técnica formal, na qual um grupo de revisores independentes (que não pertencem à equipe responsável pela Elaboração) realiza a verificação da qualidade técnica da Especificação;
- auditoria da qualidade, na qual o grupo da garantia da qualidade verifica a conformidade das atividades realizadas com o processo;
- revisão gerencial da equipe responsável pela Elaboração, que verifica se os desenvolvedores concordam com os compromissos assumidos com o cliente e realiza um balanço da iteração;

Uma vez encerradas essas revisões, uma proposta de desenvolvimento com a Especificação dos Requisitos do Software e com a informação do Plano de Desenvolvimento, completa ou parcial, deverá ser encaminhada para o cliente. A partir daí, ele decidirá sobre a continuação ou não do projeto.

3.2 Elaboração do Praxis Mentor

Na fase de Elaboração do Praxis Mentor, foram produzidos todos os artefatos do processo pertinentes à fase: MASw, CRSw, MPPSw, ERSw, PDSw e PQSw. Além destes artefatos, foi elaborado também um protótipo funcional utilizando a ferramenta Borland Delphi 4.0 para facilitar a captura e verificação dos requisitos da ferramenta. Informações sobre a ferramenta utilizada para criação do protótipo podem ser encontradas em [Delphi98].

As partes principais dos artefatos MASw e CRSw se encontram no documento de Especificação dos Requisitos do Software (ERSw). As telas do protótipo elaborado também se encontram neste documento. O conteúdo do MPPSw pode ser visto no documento Plano de Desenvolvimento do Software.

Para ilustrar as decisões tomadas nesta fase, serão apresentados neste trabalho alguns trechos representativos dos artefatos ERSw, PDSw e PQSw, uma vez que as seções interessantes dos demais artefatos citados já estão incluídas nestes três. O conteúdo dos artefatos a serem apresentados não será mostrado na íntegra, pois eles são muito extensos e detalhados. Além dos artefatos citados, será apresentada também uma síntese do resultado gerado por uma revisão técnica informal da ERSw realizada para o Praxis Mentor.

3.2.1 Especificação dos Requisitos do Software

A Especificação dos Requisitos do Software (ERSw) resulta do fluxo de Requisitos e descreve, de forma detalhada, um conjunto dos requisitos que devem ser satisfeitos por uma solução implementável para o problema.

O conteúdo da ERSw do Praxis Mentor que contempla as decisões mais importantes do projeto é apresentado a seguir.

3.2.1.1 Definição do escopo

Segundo Paula [Paula01], o ponto principal do escopo de um produto é a sua missão. Como explicado no capítulo anterior, a missão sintetiza o valor que o produto acrescenta para o cliente e para os usuários.

A missão do Praxis Mentor, já apresentada anteriormente, é mostrada na tabela a seguir:

O <u>Praxis Mentor 1.0</u> visa oferecer apoio informatizado à utilização do processo Praxis em projetos de desenvolvimento de software.
--

Tabela 11 - Missão do Praxis Mentor

Para se evitar falsas expectativas por parte do cliente e dos usuários, é interessante deixar claro o que o produto não faz. Essa definição dos limites do produto também pode explicitar funções do produto que serão implementadas em versões posteriores. Os limites do Praxis Mentor estão definidos na Tabela 12:

O Praxis Mentor 1.0 não irá garantir a consistência entre os dados do projeto (tabelas como as do Plano de Desenvolvimento do Software, por exemplo) e os artefatos do processo.

O Praxis Mentor 1.0 não fará integração com nenhuma ferramenta de Gestão de Configurações (como o Microsoft Visual SourceSafe, por exemplo), com nenhuma ferramenta de Modelagem (como o Together ou Rational Rose) nem com Editores de Texto (como o Microsoft Word). Isto significa que não será possível trabalhar com os artefatos do projeto a partir do Praxis Mentor na versão 1.0.

Atividades como backup e recuperação das bases de dados do sistema ficarão a cargo da administração de dados e não serão providas pelo Praxis Mentor 1.0.

O Praxis Mentor 1.0 não terá ajuda *on-line*. Esta será substituída pela funcionalidade de Consulta ao Processo.

Não haverá tolerância a falhas no Praxis Mentor 1.0.

Tabela 12 - Limites do Praxis Mentor

Outra importante tarefa é a identificação dos benefícios a serem proporcionados para o cliente e a importância destes para o mesmo. A associação posterior dos benefícios levantados com as funções do produto permitirá a priorização dos requisitos funcionais de forma mais eficaz. Para que isto seja possível, é importante definir quais benefícios são essenciais para o produto e quais podem ser considerados apenas desejáveis ou até opcionais. A tabela a seguir mostra os benefícios do Praxis Mentor e o respectivo valor para o cliente:

Número de ordem	Benefício	Valor para o Cliente
1	Capacitação do usuário no uso do processo	Essencial
2	Agilidade na utilização do processo	Essencial
3	Diminuição dos erros no uso do processo	Essencial
4	Melhoria da comunicação da equipe do projeto	Essencial
5	Contribuição para a melhoria do processo	Essencial
6	Agilidade na resolução dos problemas do projeto	Essencial
7	Capacitação dos gerentes no uso do processo	Essencial
8	Maior agilidade na tarefa de planejamento de um projeto	Essencial
9	Agilidade na recuperação de qualquer artefato do projeto	Essencial
10	Facilidade na realização de cópias de segurança do projeto	Desejável
11	Auxílio na personalização do processo	Desejável
12	Maior facilidade no cálculo das métricas do processo	Opcional
13	Diminuição dos erros no cálculo de métricas	Opcional

Tabela 13 - Benefícios do Praxis Mentor

3.2.1.2 Definição dos requisitos

Uma das principais tarefas desta fase é o levantamento dos requisitos através da identificação dos casos de uso e atores do sistema. Os casos de uso podem ser entendidos como uma representação das funções disponíveis no produto e os atores como os usuários ou outros sistemas que interagem com ele. Os relacionamentos entre os casos de uso e os atores são apresentados nos diagramas de caso de uso. Um importante diagrama deste tipo é o Diagrama de Contexto, que mostra as interfaces do produto com seu ambiente de aplicação, incluindo os diversos tipos de usuários e outros sistemas com os quais o produto deve interagir.

As figuras a seguir apresentam os diagramas de contexto do Praxis Mentor. A definição de cada ator e caso de uso identificado para o produto também é apresentada.

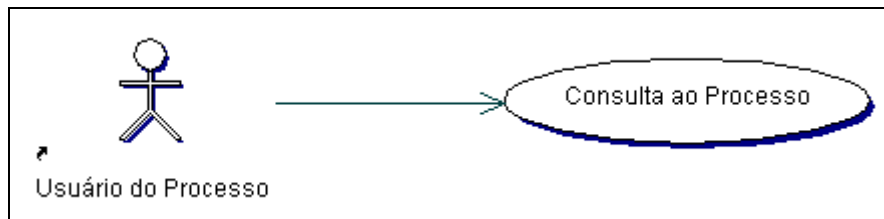


Figura 10 - Diagrama Consulta ao Processo

Este diagrama se refere ao requisito de Consulta ao Processo. O caso de uso aqui representado engloba os três primeiros benefícios da Tabela 13.

Atores	Definição
Usuário do Processo	Pessoa que está utilizando o processo.
Casos de Uso	Definição
Consulta ao Processo	Representa qualquer tipo de consulta realizada pelo usuário aos textos explicativos do processo. Engloba a função de orientação on-line ao uso do processo.

Tabela 14 - Atores e casos de uso do Diagrama Consulta ao Processo

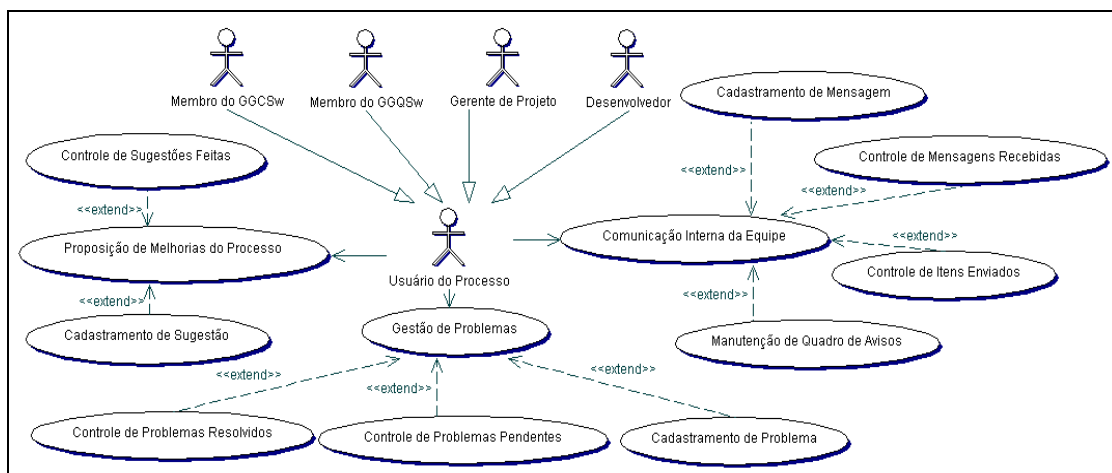


Figura 11 - Diagrama Comunicação

Este diagrama representa todos os requisitos que se referem à comunicação da equipe do

projeto. Os casos de uso deste diagrama correspondem aos benefícios 4, 5 e 6 da Tabela 13. Alguns dos requisitos aqui apresentados foram levantados com base no Diagrama de Atividades Fluxo Alternativo Resolução de Problemas, mostrado na Figura 9.

Atores	Definição
Usuário do Processo	Pessoa que está utilizando o processo.
Membro do GGCSw	Membro do grupo que planeja, coordena, e implementa ações para gerir um sistema centralizado de guarda de configurações de software.
Membro do GGQSw	Membro do grupo que planeja e implementa atividades que asseguram a qualidade do produto.
Gerente de Projeto	Pessoa que tem responsabilidade completa por um projeto, inclusive a sua direção, o seu controle e a sua administração.
Desenvolvedor	Profissional que exerce tarefas técnicas em um projeto de desenvolvimento de software.
Casos de Uso	Definição
Comunicação Interna da Equipe	Armazenamento de mensagens designadas a algum usuário do processo (gerentes, testadores, desenvolvedores, etc.).
Cadastramento de Mensagem	Criação de uma nova mensagem a ser enviada para algum(s) membro(s) da equipe do projeto.
Controle de Mensagens Recebidas	Relação das mensagens recebidas por um determinado membro da equipe do projeto.
Controle de Itens Enviados	Relação das mensagens enviadas por um determinado membro da equipe do projeto.
Manutenção de Quadro de Avisos	Relação de avisos visíveis a todos os membros da equipe do projeto.
Gestão de Problemas	Comunicação dos problemas ao gerente do projeto por parte dos membros do GGQSw e do GGCSw e controle das providências por ele tomadas.
Cadastramento de Problema	Criação e definição de um novo problema a ser enviado para algum(s) membro(s) da equipe do projeto.
Controle de Problemas Pendentes	Relação dos problemas ainda não resolvidos pelo membro da equipe do projeto.
Controle de Problemas Resolvidos	Relação dos problemas já resolvidos pelo membro da equipe do projeto.
Proposição de Melhorias do Processo	Criação do histórico do uso do processo e armazenamento das opiniões e funções agregadas ao processo.
Cadastramento de Sugestão	Criação de uma nova sugestão de melhoria do processo a ser enviada para todos os membros da equipe do projeto.
Controle de Sugestões Feitas	Relação de todas as sugestões de melhoria feitas pelos membros da equipe do projeto.

Tabela 15 - Atores e casos de uso do Diagrama Comunicação

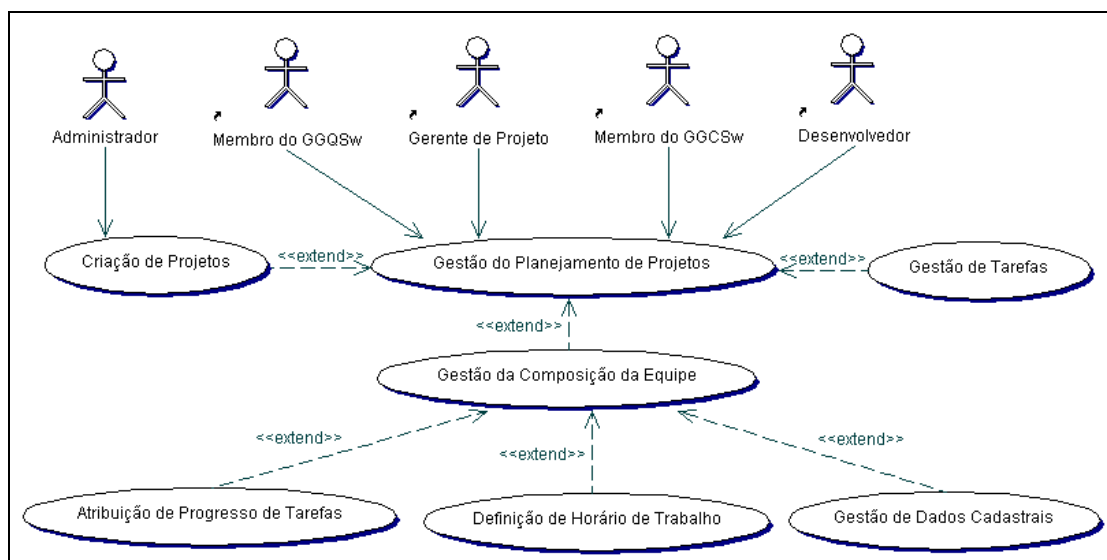


Figura 12 - Diagrama Gestão do Planejamento do Projeto

Este diagrama se refere ao requisito de Planejamento do Projeto. O caso de uso aqui representado engloba os benefícios 7 e 8 da Tabela 13. Alguns dos requisitos aqui apresentados foram levantados com base no Diagrama de Atividades Fluxo Alternativo Alteração de Equipe, mostrado na Figura 8.

Atores	Definição
Administrador	Usuário especial da ferramenta responsável pela tarefa de criação de projetos e pela entrada de dados no momento inicial do uso da ferramenta.
Membro do GGCSw	Membro do grupo que planeja, coordena, e implementa ações para gerir um sistema centralizado de guarda de configurações de software.
Membro do GGQSw	Membro do grupo que planeja e implementa atividades que asseguram a qualidade do produto.
Gerente de Projeto	Pessoa que tem responsabilidade completa por um projeto, inclusive a sua direção, o seu controle e a sua administração.
Desenvolvedor	Profissional que exerce tarefas técnicas em um projeto de desenvolvimento de software.
Casos de Uso	Definição
Gestão do Planejamento de Projetos	Aspectos quantitativos do planejamento dos projetos. Controle dos marcos estabelecidos no projeto (datas), etc.
Criação de Projetos	Tarefas de criação de um projeto: cadastramento dos dados do projeto (nome, data de início e fim etc.), criação da pasta do projeto e cópia dos artefatos para essa pasta.
Gestão de Tarefas	Procedimentos de controle de tarefas: criação, atribuição a usuários, etc.
Gestão da Composição da Equipe	Inclusão, remoção e alteração dos dados dos membros da equipe do projeto.
Gestão de Dados Cadastrais	Definição dos dados pessoais de um membro da equipe do projeto.
Definição de Horário de Trabalho	Definição do horário de trabalho de um membro da equipe do projeto.
Atribuição de Progresso de Tarefas	Definição do progresso das tarefas de um membro da equipe do projeto.

Tabela 16 - Atores e casos de uso do Diagrama Gestão do Planejamento do Projeto

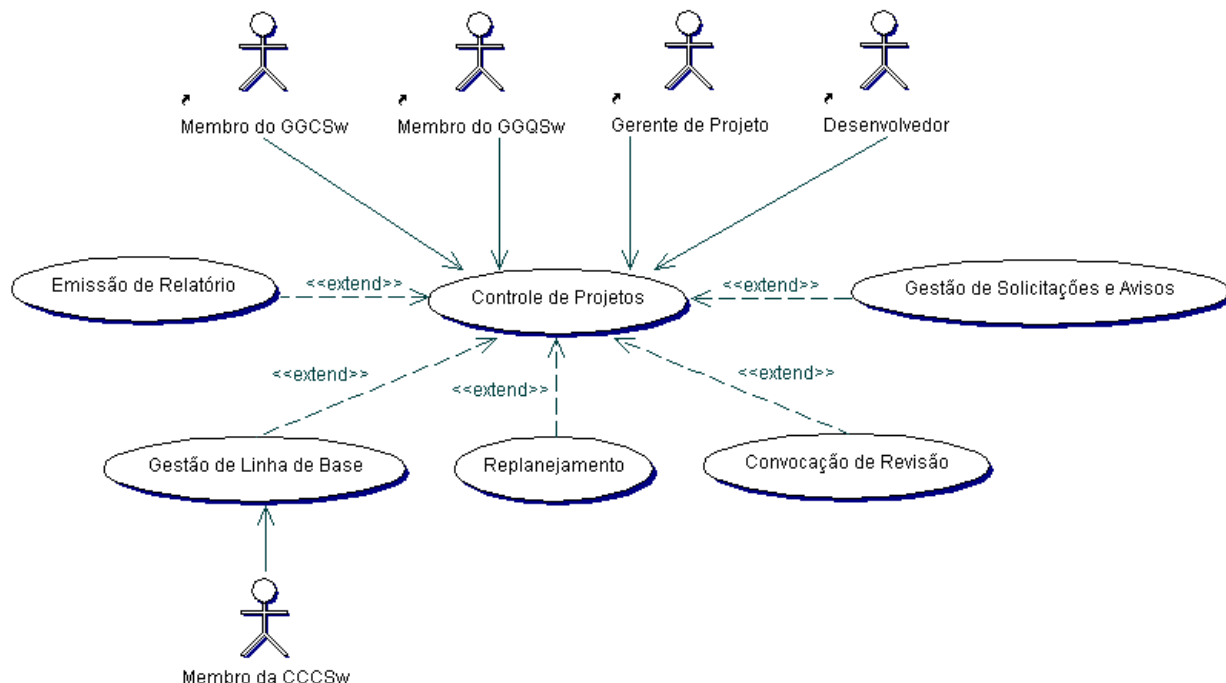


Figura 13 - Diagrama Controle de Projetos

Este diagrama se refere ao requisito de Controle de Projetos. O caso de uso aqui representado engloba os benefícios 9, 10, 11, 12 e 13 da Tabela 13. Alguns dos requisitos aqui apresentados foram levantados com base nos Diagramas de Atividades Subfluxo Procedimentos de Controle, Subfluxo Criação de Linha de Base, Subfluxo Replanejamento, Fluxo Alternativo Relatório de Acompanhamento e Fluxo Alternativo Alteração de Linha de Base, mostrados nas Figuras 3, 4, 5, 6 e 7, respectivamente.

Atores	Definição
Membro do GGCSw	Membro do grupo que planeja, coordena, e implementa ações para gerir um sistema centralizado de guarda de configurações de software.
Membro do GGQSw	Membro do grupo que planeja e implementa atividades que asseguram a qualidade do produto.
Gerente de Projeto	Pessoa que tem responsabilidade completa por um projeto, inclusive a sua direção, o seu controle e a sua administração.
Desenvolvedor	Profissional que exerce tarefas técnicas em um projeto de desenvolvimento de software.
Membro da CCCSw	Membro da comissão responsável pelas decisões relativas ao controle das Linhas de Base.
Casos de Uso	Definição
Controle de Projetos	Procedimentos de acompanhamento, supervisão e controle de projetos de software.
Replanejamento	Procedimentos de replanejamento do projeto. O replanejamento deve ser efetuado sempre que ocorrer eventos que justifiquem tal procedimento. Por exemplo, a ocorrência de uma discrepância entre o valor adquirido previsto e o valor adquirido realizado.
Gestão de Solicitações e Avisos	Procedimentos de controle das solicitações pendentes: aprovação de solicitação, rejeição ou encaminhamento, e visualização de avisos de emissão de relatórios.
Convocação de Revisão	Convocação de uma reunião que possibilite a análise de um material, efetuada por um grupo de pares dos autores, com a utilização de um processo formal, para identificar defeitos e problemas de conformidade com padrões.
Gestão de Linha de Base	Procedimento de criação de uma Linha de Base do projeto, tratando da inserção dos itens da Linha de Base na Biblioteca de Desenvolvimento do Software.
Emissão de Relatório	Emissão do Relatório Final do Projeto onde são recolhidas as métricas importantes coletadas no projeto e as lições que possam levar à melhoria do processo em projetos futuros.

Tabela 17 - Atores e casos de uso do Diagrama Controle de Projetos

3.2.1.3 Detalhamento dos requisitos de interface

Para detalhar os requisitos de interfaces gráficas, é interessante o uso de esboços gráficos das mesmas. Tais esboços ajudam a identificar mais claramente os requisitos e, muitas vezes, resultam da tarefa de prototipagem já realizada. É importante ressaltar que estes esboços devem representar apenas sugestões e que o detalhamento definitivo deverá ser feito dentro do fluxo de Desenho. Neste projeto, entretanto, foi desenvolvido um protótipo com interfaces gráficas bem avançadas e já próximas das definitivas. A decisão de se adiantar algumas decisões de desenho na Especificação dos Requisitos foi tomada em função das restrições de tempo existentes para o projeto.

A lista das interfaces gráficas identificadas para o Praxis Mentor é apresentada na tabela a seguir:

Número de ordem	Nome	Ator	Caso de uso	Descrição
1	Tela Identificação	Usuário do Processo	-	Solicitação do usuário e da senha para identificação das funções disponíveis ao usuário.
2	Tela Mudança de Senha	Usuário do Processo	-	Mudança de senha.
3	Tela Seleção de Projetos	Usuário do Processo	-	Seleção de um projeto existente ou criação de um novo.
4	Tela Propriedades de Projeto / Novo Projeto	Gerente de Projeto/ Administrador	Gestão do Planejamento de Projetos /Criação de Projetos	Consulta e alteração dos dados de um projeto/ Inserção de um novo projeto.
5	Tela Exclusão de Projeto	Administrador	-	Exclusão de um projeto existente.
6	Tela Lista de Itens em Edição	Usuário do Processo	-	Verificação das mensagens em aberto. Avisa o usuário e oferece a possibilidade de finalização dessas mensagens, por exemplo, antes de encerrar o sistema.
7	Tela Destinatários	Usuário do Processo	-	Atribuição dos membros da equipe do projeto em questão à lista de destinatários de uma mensagem ou solicitação.
8	Tela Principal	Usuário do Processo	-	Interface principal com opção de criar e abrir projeto, salvar, imprimir, etc. .
9	Tela Consulta ao Processo	Usuário do Processo	Consulta ao Processo	Pesquisa por tópicos ou direta a um material de referência ao processo.
10	Tela Controle de Projeto – Linha Mestra	Gerente de Projeto	Controle de Projetos / Replanejamento	Inclusão, consulta, alteração e exclusão dos marcos importantes do projeto e dos avisos periódicos.
11	Tela Controle de Projeto – Tarefas	Gerente de Projeto	Gestão de Tarefas	Inclusão, consulta, alteração e exclusão de tarefas. Atribuição de tarefas a usuários.
12	Tela Controle de Projeto – Solicitações e Avisos	Usuário do Processo	Gestão de Solicitações e Avisos	Lista dos avisos de emissão de relatórios e das solicitações do projeto com a possibilidade de aprovação, rejeição ou encaminhamento para outro membro da equipe.
13	Tela Controle de Projeto – Solicitar Revisão	Gerente de Projeto	Convocação de Revisão	Envio de pedido de revisão técnica ou gerencial com a lista do material para a revisão.
14	Tela Controle de Projeto – Solicitar Criação ou Alteração de Linha de Base	Gerente de Projeto	Gestão de Linha de Base	Envio de pedido de criação ou de alteração de Linha de Base com a lista dos itens a serem criados ou alterados.
15	Tela Controle de Projeto – Emitir Relatório	Gerente de Projeto	Emissão de Relatório	Envio de aviso de emissão do RAPS w ou do RFPS w com observações sobre o mesmo.

16	Tela Comunicação – Quadro de Avisos	Usuário do Processo	Manutenção de Quadro de Avisos	Inclusão e exclusão de avisos a que toda a equipe do projeto terá acesso.
17	Tela Comunicação – Quadro de Avisos – Edição	Usuário do Processo	Manutenção de Quadro de Avisos	Edição de um aviso.
18	Tela Comunicação – Caixa de Entrada	Usuário do Processo	Controle de Mensagens Recebidas	Visualização das mensagens recebidas com possibilidade de envio de resposta ao remetente, a todos os membros da equipe ou encaminhamento a um membro específico.
19	Tela Comunicação – Itens Enviados	Usuário do Processo	Controle de Itens Enviados	Visualização das mensagens enviadas com possibilidade de envio de resposta ao remetente, a todos os membros da equipe ou encaminhamento a um membro específico.
20	Tela Comunicação – Nova Mensagem	Usuário do Processo	Cadastramento de Mensagem	Elaboração e envio de nova mensagem.
21	Tela Composição da Equipe – Seleção de Usuários	Usuário do Processo	Gestão da Composição da Equipe	Seleção de usuários já cadastrados no sistema para compor a equipe de um projeto.
22	Tela Composição da Equipe – Dados Cadastrais	Usuário do Processo	Gestão de Dados Cadastrais	Inclusão, consulta, alteração e exclusão dos dados dos membros da equipe do projeto.
23	Tela Composição da Equipe – Horário de Trabalho	Usuário do Processo	Definição de Horário de Trabalho	Inclusão, consulta, alteração e exclusão do horário de trabalho para um membro da equipe do projeto.
24	Tela Composição da Equipe – Progresso de Tarefas	Gerente de Projeto	Atribuição de Progresso de Tarefas	Inclusão, consulta e alteração de papéis para um membro da equipe do projeto e de progresso para as tarefas do mesmo.
25	Tela Gestão de Problemas – Problemas Pendentes	Usuário do Processo	Controle de Problemas Pendentes	Lista dos problemas pendentes com a possibilidade de inclusão de uma solução e mudança de status do problema para resolvido ou do encaminhamento do problema para outros membros da equipe.
26	Tela Gestão de Problemas – Problemas Resolvidos	Usuário do Processo	Controle de Problemas Resolvidos	Lista dos problemas resolvidos com a identificação de quem o resolveu e quando. Possibilidade de rejeição da solução dada.
27	Tela Gestão de Problemas – Novo Problema	Usuário do Processo	Cadastramento de Problema	Inclusão e envio de um novo problema para o gerente do projeto.
28	Tela Proposição de Melhorias – Sugestões Feitas	Usuário do Processo	Controle de Sugestões Feitas	Lista de sugestões para a melhoria do processo com a possibilidade de se adicionar um comentário a uma sugestão dada.
29	Tela Proposição de Melhorias – Nova Sugestão	Usuário do Processo	Cadastramento de Sugestão	Inclusão de sugestão para melhoria do processo.

Tabela 18 - Interfaces de usuário do Praxis Mentor

A especificação detalhada de algumas interfaces do Praxis Mentor é mostrada nas seções seguintes.

A primeira interface apresentada consiste na tela principal do projeto. Ela possui dois estados: um em que um projeto está aberto e outro em que não existe nenhum projeto aberto. No primeiro, todos os recursos da ferramenta estão disponíveis de acordo com o papel do usuário corrente. No segundo, apenas a funcionalidade de Consulta ao Processo está disponível.

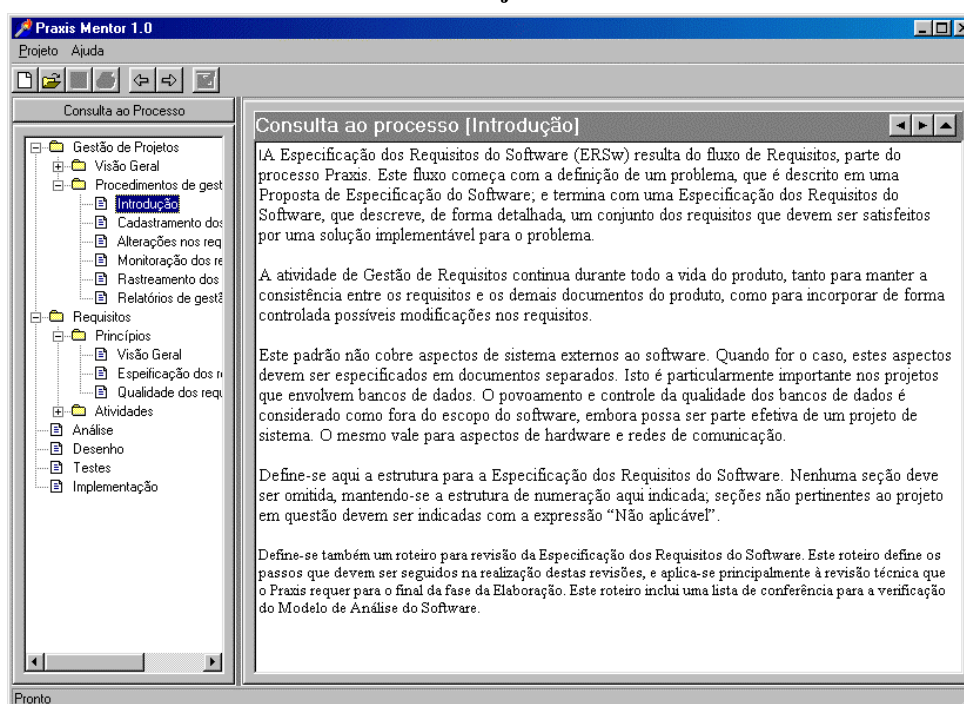
A segunda interface mostrada é a de criação ou propriedades de projeto. Ela é utilizada tanto no momento da criação de um projeto quanto nos momentos em que o usuário deseja alterar algum dado de definição do projeto.

A terceira e última interface mostrada é a de mensagens recebidas. Esta interface contém a caixa de entrada das mensagens do usuário e disponibiliza as funcionalidades de resposta ou encaminhamento da mensagem.

Maiores detalhes sobre a utilidade e o funcionamento das interfaces aqui apresentadas podem ser encontrados na seção 4.2.3 (Manual do Usuário do Software) do Capítulo 4.

3.2.1.3.1 Interface de usuário Tela Principal

Nenhum Projeto Aberto



Com um Projeto Aberto

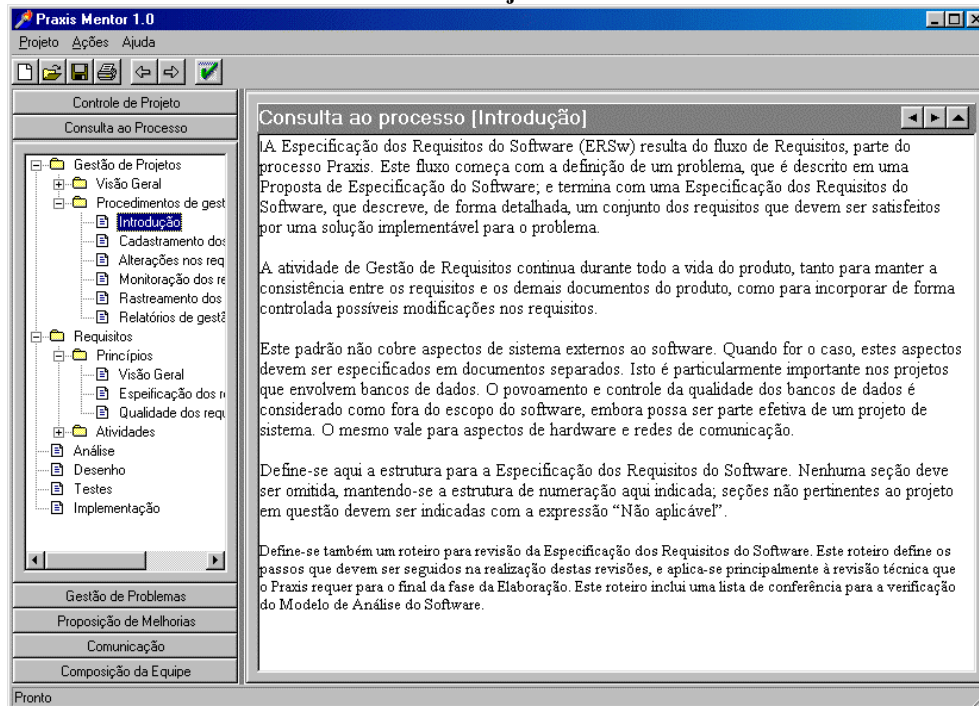
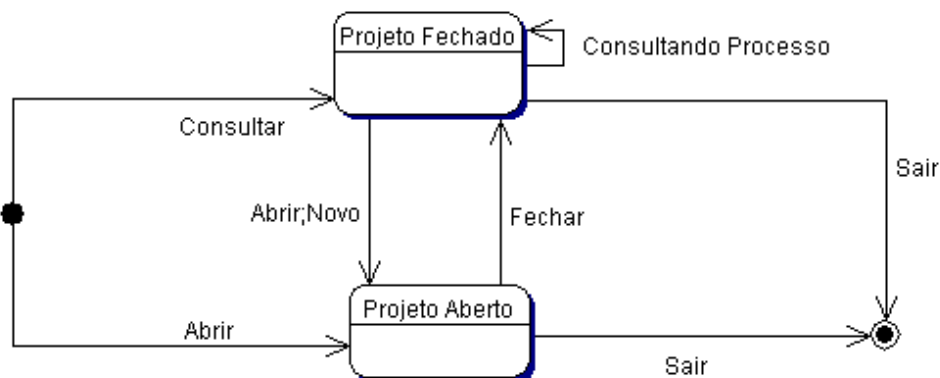


Diagrama de estados



Relacionamento com outras interfaces

A partir dessa tela, todas as outras podem ser acionadas, com exceção das telas de Identificação e Mudança de Senha.

Campos

Número	Nome	Valores válidos	Formato	Tipo	Restrições
1	Lista de Tópicos	Tópicos do processo previamente cadastrados	-	Texto	Preenchido pelo Praxis Mentor / não alterável.
2	Conteúdo	Conteúdos relativos aos tópicos também cadastrados previamente.	-	Texto	Preenchido pelo Praxis Mentor / não alterável.

Comandos

Número	Nome	Ação	Restrições
1	Novo (Menu Projeto)	Aciona interface para criação de um novo projeto.	Sempre habilitado.
2	Abrir (Menu Projeto)	Aciona interface de abertura de um novo projeto.	Sempre habilitado.
3	Fechar (Menu Projeto)	Fecha o projeto.	Habilitado quando houver algum projeto aberto.
4	Excluir (Menu Projeto)	Exclui um projeto a ser escolhido.	Sempre habilitado.
5	Exportar (Menu Projeto)	Exporta os dados do projeto para formato HTML.	Habilitado quando houver algum projeto aberto.
6	Imprimir (Menu Projeto)	Imprime dados do projeto.	Habilitado quando houver algum projeto aberto.
7	Propriedades (Menu Projeto)	Aciona interface de propriedades de um projeto.	Habilitado quando houver algum projeto aberto.
8	Sair (Menu Projeto)	Fecha a interface.	Sempre habilitado.
9	Voltar tópico	Posiciona no tópico anterior de consulta ao processo.	Sempre habilitado.
10	Avançar tópico	Posiciona no próximo tópico de consulta ao processo.	Sempre habilitado.
11	Verificar Pendências	Exibe a lista de mensagens iniciadas e não enviadas pelo usuário.	Habilitado quando houver algum projeto aberto.
12	Controle de Projeto (Menu Ações)	Aciona a interface com os tópicos de Controle de Projetos.	Sempre habilitado, com restrições dependendo do papel do usuário.
13	Consulta ao Processo (Menu Ações)	Aciona a interface com os tópicos de Consulta ao Processo.	Sempre habilitado.
14	Gestão de Problemas (Menu Ações)	Aciona a interface com os tópicos de Gestão de Problemas.	Sempre habilitado, com restrições dependendo do papel do usuário.
15	Proposição de Melhorias (Menu Ações)	Aciona a interface com os tópicos de Proposição de Melhorias.	Sempre habilitado.
16	Comunicação (Menu Ações)	Aciona a interface com os tópicos de Comunicação.	Sempre habilitado.
17	Composição da Equipe (Menu Ações)	Aciona a interface com os tópicos de Composição da Equipe.	Sempre habilitado, com restrições dependendo do papel do usuário..
18	Sobre (Menu Ajuda)	Aciona a interface Sobre o Praxis Mentor.	Sempre habilitado.

3.2.1.3.2 Interface de usuário Tela Propriedades de Projeto / Novo Projeto

Diagrama de estados

Não aplicável.

Relacionamentos com outras interfaces

O botão OK aciona a interface Tela Principal com o projeto aberto.
O botão Cancelar retorna à interface que chamou essa tela.

Campos

Número	Nome	Valores válidos	Formato	Tipo	Restrições
1	Sigla	Não vazio.	Até 15 caracteres.	Texto	Obrigatório / alterável.
2	Nome	Não vazio.	Até 50 caracteres.	Texto	Obrigatório / alterável.
3	Descrição	-	Até 255 caracteres.	Texto	Opcional / alterável.
4	Início	-	dd/mm/aaaa	Data	Obrigatório / alterável.
5	Término	Maior que início.	dd/mm/aaaa	Data	Obrigatório / alterável.
6	Gerente de Projeto (Nome + Usuário)	Não vazio.	Até 65 caracteres.	Texto	Obrigatório / alterável.
7	Pasta do Projeto	Não vazio.	Até 255 caracteres.	Texto	Obrigatório / alterável.

Comandos

Número	Nome	Ação	Restrições
1	OK	Fecha a interface/Cria novo projeto.	Habilitado quando todos os dados obrigatórios tiverem sido digitados.
2	Cancelar	Fecha a interface.	Sempre habilitado.

3.2.1.3.3 Interface de usuário Tela Caixa de Entrada

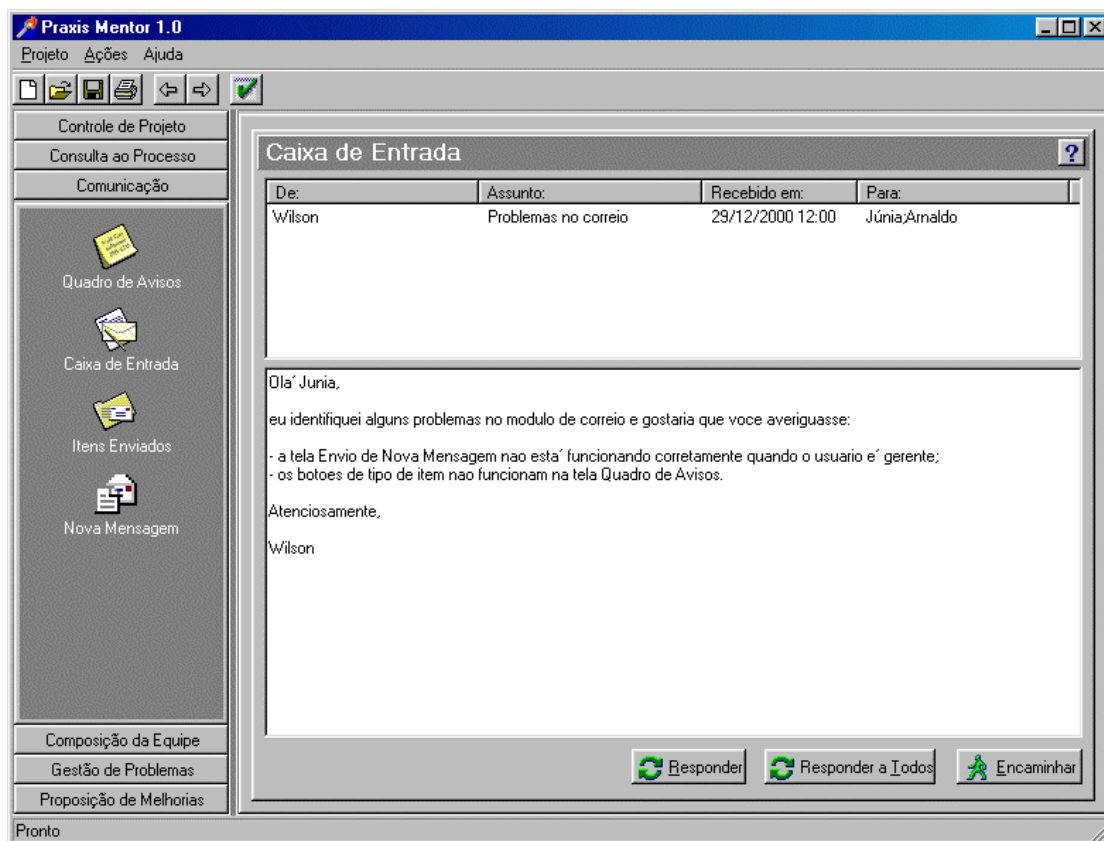



Diagrama de estados

Não aplicável.


Relacionamentos com outras interfaces

Os botões Responder, Responder a Todos e Encaminhar acionam a Tela Comunicação - Nova Mensagem.
O botão  aciona a interface Tela Consulta ao Processo.

Campos

Número	Nome	Valores válidos	Formato	Tipo	Restrições
1	De:	Não vazio.	Até 50 caracteres	Texto	Preenchido pelo Praxis Mentor / não alterável.
2	Assunto:	-	Até 50 caracteres	Texto	Preenchido pelo Praxis Mentor / não alterável.
3	Recebido em:	Não vazio.	dd/mm/aaaa hh:mm	Data	Preenchido pelo Praxis Mentor / não alterável.
4	Para	Não vazio	Até 255 caracteres	Texto	Preenchido pelo Praxis Mentor / não alterável.
5	Corpo da Mensagem	-	Até 255 caracteres	Texto	Preenchido pelo Praxis Mentor / não alterável.

Comandos

Número	Nome	Ação	Restrições
1	Responder	Aciona a Tela Nova Mensagem com o campo "Para:" preenchido com o valor do campo "De:" dessa tela.	Sempre habilitado.
2	Responder a todos	Aciona a Tela Nova Mensagem com o campo "Para:" preenchido com o valor dos campos "De:" e "Para" dessa tela.	Sempre habilitado.
3	Encaminhar	Aciona a Tela Nova Mensagem com o campo "Para:" em branco.	Sempre habilitado.
4	Excluir (tecla delete)	Remove a mensagem selecionada.	Habilitado quando houver pelo menos uma mensagem. Exige confirmação.
5		Aciona o tópico de Caixa de Entrada da Tela Consulta ao Processo.	Sempre habilitado.

Observações

Os outros comandos são de acordo com a descrição da interface Tela Principal.

3.2.1.4 Requisitos adiados

Alguns requisitos são identificados durante a elaboração da Especificação dos Requisitos mas, por algum motivo, são adiados para versões futuras do software. O armazenamento desta informação neste momento é importante para facilitar a Engenharia de Requisitos em novas versões. Os requisitos adiados identificados para o Praxis Mentor são apresentados na tabela a seguir:

Número	Referência ao requisito	Detalhes
1	Sincronismo entre os artefatos de planejamento e os dados cadastrados na ferramenta.	Gestão dos dados gerenciais do projeto (datas, recursos, riscos, etc.) e atualização automática dos artefatos relacionados.
2	Integração com ferramentas de Controle de Configurações, de Modelagem e Editores de Texto.	Integração com ferramentas de Gestão de Configurações, de Modelagem e Editores de Texto de modo que se possa trabalhar com os artefatos do projeto a partir do <u>Praxis Mentor</u> .

Tabela 19 - Requisitos adiados do Praxis Mentor

3.2.1.5 Detalhamento dos requisitos funcionais

Os requisitos funcionais descrevem as funções que o produto deverá realizar em benefício dos usuários. Segundo Paula [Paula01], existem muitas maneiras de se descrever estas funções. No Praxis, cada função é descrita por um caso de uso. A descrição dos fluxos dos casos de uso define os detalhes dos requisitos funcionais.

Nas próximas seções será apresentado o detalhamento dos fluxos dos casos de uso referentes às interfaces descritas anteriormente. É importante esclarecer que nem sempre o relacionamento entre casos de uso e interfaces é de um para um, como neste caso. Um caso de uso pode ser implementado por várias interfaces e vice-versa.

3.2.1.5.1 Caso de uso Criação de Projetos

Precondições

O Praxis Mentor está no Modo Administrador.
O Administrador selecionou o item Novo Projeto.

Fluxo principal

O Administrador informa os dados do novo projeto.
O Administrador informa o gerente do novo projeto.
Se o gerente informado ainda não estiver cadastrado no sistema, o Praxis Mentor insere o gerente no sistema.
O Administrador aciona criação do projeto.
Se algum dado do projeto não tiver sido informado, o Praxis Mentor emite uma mensagem de erro e aborta a operação.
Se não falta nenhum dado:
 O Praxis Mentor insere o gerente na equipe do projeto.
 O Praxis Mentor verifica se a pasta informada para o projeto já existe.
 Se a pasta existir, o Praxis Mentor copia os artefatos para a mesma e modifica os nomes dos artefatos de acordo com nome dado ao projeto.
 Se a pasta não existir, Praxis Mentor cria a mesma e copia os artefatos para ela com os nomes modificados de acordo com o nome dado ao projeto.

3.2.1.5.2 Caso de uso Controle de Mensagens Recebidas

Precondições

O Usuário do Processo selecionou o item de comunicação Caixa de Entrada.

Fluxo principal

O Praxis Mentor exibe a lista das mensagens existentes.
O Usuário do Processo seleciona uma mensagem.
O Praxis Mentor exibe o conteúdo da mensagem selecionada.

Subfluxos

Não aplicável.

Fluxos alternativos

Fluxo alternativo Responder ao Remetente

Precondições	O <u>Usuário do Processo</u> escolhe responder ao remetente.
Passos	O <u>Praxis Mentor</u> exibe a interface de usuário Nova Mensagem com o campo “Para” preenchido com o valor do campo “De” da interface atual. O <u>Usuário do Processo</u> preenche os dados da mensagem e seleciona a operação a ser realizada: “Enviar” ou “Limpar”. Se o <u>Usuário do Processo</u> escolher “Enviar”: O <u>Praxis Mentor</u> envia a mensagem para os destinatários. Se o <u>Usuário do Processo</u> escolher “Limpar”: O <u>Praxis Mentor</u> limpa todos os dados da mensagem.

Fluxo alternativo Responder a Todos

Precondições	O <u>Usuário do Processo</u> escolhe responder uma mensagem a todos da lista de destinatários.
Passos	O <u>Praxis Mentor</u> exibe a interface de usuário Nova Mensagem com o campo “Para” preenchido com o valor dos campos “De” e “Para” da interface atual. O <u>Usuário do Processo</u> preenche os dados da mensagem e seleciona a operação a ser realizada: “Enviar” ou “Limpar”.. Se o <u>Usuário do Processo</u> escolher “Enviar”: O <u>Praxis Mentor</u> envia a mensagem para os destinatários. Se o <u>Usuário do Processo</u> escolher “Limpar”: O <u>Praxis Mentor</u> limpa todos os dados da mensagem.

Fluxo alternativo Encaminhar

Precondições	O <u>Usuário do Processo</u> escolhe encaminhar uma mensagem.
Passos	<p>O <u>Praxis Mentor</u> exibe a interface de usuário Nova Mensagem com o campo “Para” vazio.</p> <p>O <u>Usuário do Processo</u> preenche os dados da mensagem e seleciona a operação a ser realizada: enviar ou limpar.</p> <p>Se o <u>Usuário do Processo</u> escolher “Enviar”:</p> <p style="padding-left: 40px;">O <u>Praxis Mentor</u> envia a mensagem para os destinatários.</p> <p style="padding-left: 40px;">Se nenhum destinatário tiver sido informado, o <u>Praxis Mentor</u> emite uma mensagem de erro e aborta a operação.</p> <p>Se o <u>Usuário do Processo</u> escolher “Limpar”:</p> <p style="padding-left: 40px;">O <u>Praxis Mentor</u> limpa todos os dados da mensagem.</p>

Fluxo alternativo Excluir Mensagem

Precondições	O <u>Usuário do Processo</u> escolhe excluir uma mensagem.
Passos	<p>O <u>Praxis Mentor</u> solicita confirmação da operação.</p> <p>O <u>Usuário do Processo</u> confirma a operação.</p> <p>O <u>Praxis Mentor</u> exclui a mensagem selecionada.</p>

3.2.1.6 Detalhamento dos requisitos não funcionais

Os requisitos não funcionais incluem os requisitos de desempenho e outros atributos de qualidade do produto. Também devem ser incluídos nesta seção os requisitos lógicos de dados e as restrições ao desenho. Segundo Paula [Paula01], os requisitos não funcionais devem ser enunciados de forma precisa e quantitativa, mesmo que seja difícil formular valores razoáveis no levantamento dos requisitos de uma primeira versão do produto.

Os requisitos não funcionais do Praxis Mentor são mostrados nas seções que se seguem.

3.2.1.6.1 Requisitos de dados persistentes

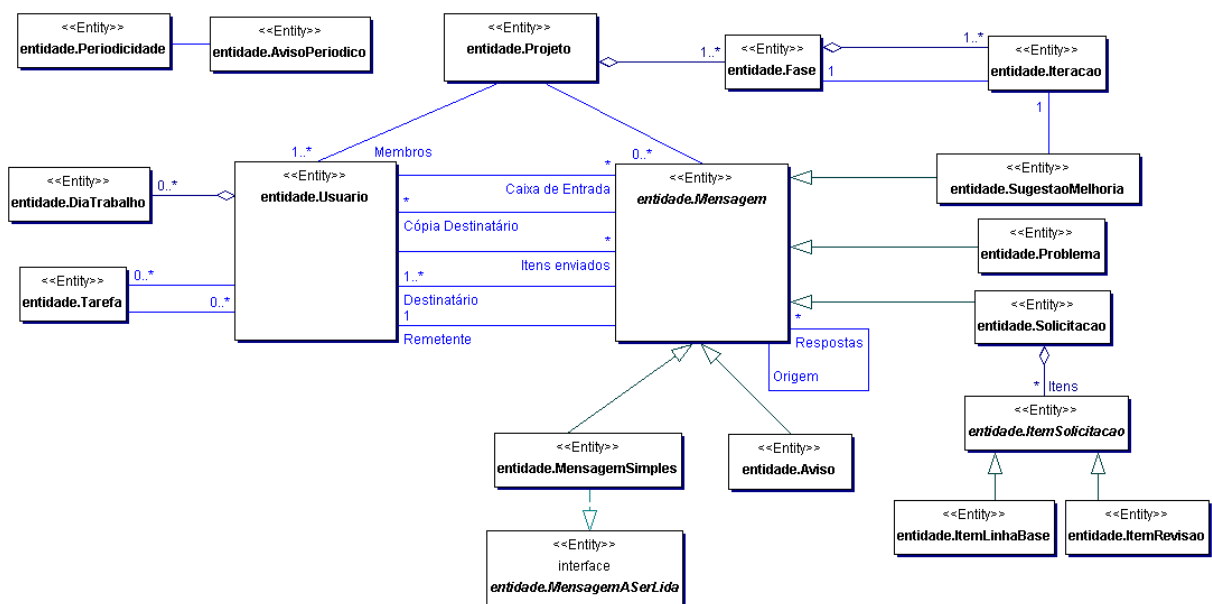


Figura 14 - Diagrama de Classes Persistentes

3.2.1.6.2 Restrições ao desenho

O Praxis Mentor possui uma restrição de desenho de Extensibilidade: o produto deve ser desenvolvido levando-se em consideração que poderá ser estendido para atender a outras partes do processo.

3.2.1.6.3 Atributos da qualidade

O Praxis Mentor possui dois atributos de qualidade que devem ser respeitados: Segurança do Acesso e Apreensibilidade. De acordo com o primeiro, o Praxis Mentor deverá restringir o acesso através de senhas para os usuários. De acordo com o segundo, um usuário proficiente no processo deverá ser capaz de aprender a operar o Praxis Mentor com uma semana de treinamento.

3.2.1.7 Modelo de Análise

Segundo Paula [Paula01], o Modelo de Análise deve conter os detalhes necessários para servir de base para o desenho do produto, mas deve-se evitar a inclusão de detalhes que pertençam ao domínio da implementação e não do problema. Quando se usa um Modelo de Análise orientado por objetos, os requisitos funcionais são tipicamente descritos e verificados através dos seguintes recursos de notação:

- os casos de uso descrevem o comportamento esperado do produto como um todo. Os diagramas de casos de uso descrevem os relacionamentos dos casos de uso entre si e com os atores, enquanto os fluxos descrevem os detalhes de cada caso de uso;
- as classes representam os conceitos do mundo da aplicação que sejam relevantes para a descrição precisa dos requisitos. Os diagramas de classes mostram os relacionamentos entre estas e as especificações das classes descrevem os respectivos detalhes;
- as realizações dos casos de uso mostram como objetos das classes descritas colaboram entre si para realizar os principais roteiros que podem ser percorridos dentro de cada caso de uso.

3.2.1.7.1 Diagramas de classes

Os diagramas de classes foram divididos por assunto da seguinte forma: Aplicação, Comunicação, Consulta ao Processo, Controle de Projetos e Planejamento de Projetos. Esta divisão representa todos os aspectos do processo cobertos pelo Praxis Mentor.

Uma breve descrição sobre cada diagrama é apresentada a seguir.

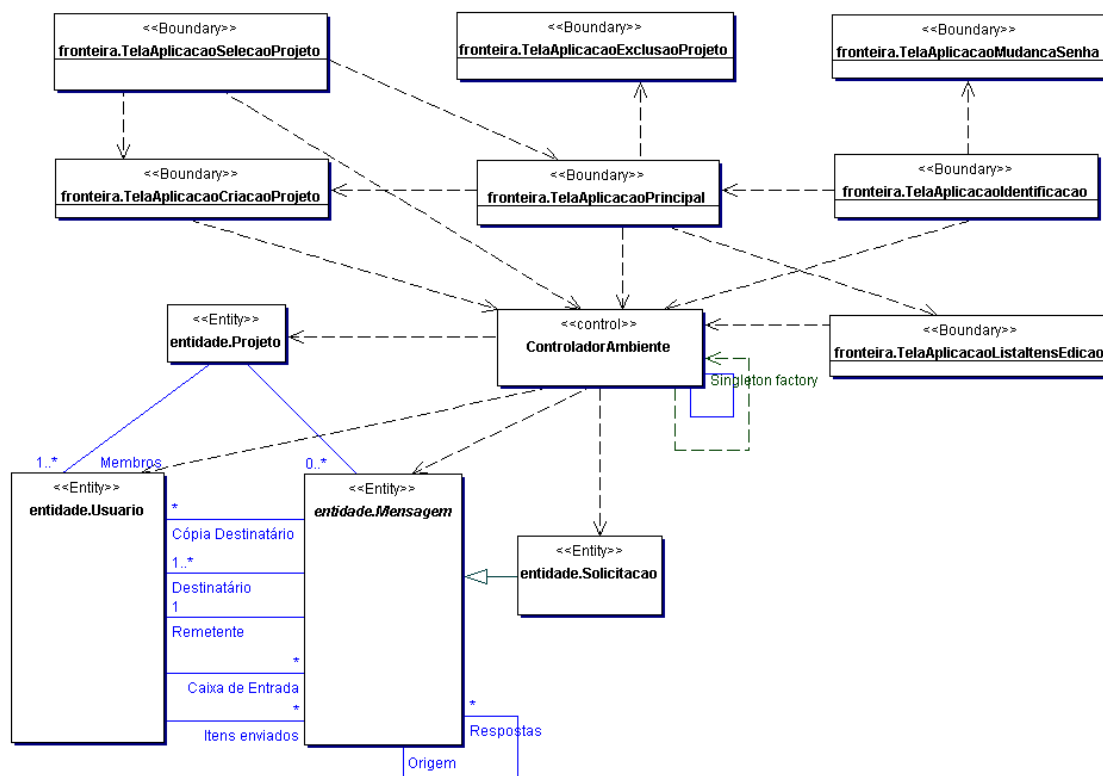


Figura 15 - Diagrama de Classes Aplicação

O diagrama da Figura 15 contempla as classes que representam as telas genéricas diretamente ligadas à tela principal da Aplicação. Ele contém algumas classes de fronteira (telas) e uma classe de controle que é responsável por gerenciar todo o comportamento e troca de informações das classes de fronteira com as classes de entidade que armazenam os dados. Encontram-se neste grupo as classes que representam as telas de **Identificação**, **Mudança de Senha**, **Seleção**, **Criação** e **Exclusão de Projetos**, a classe de controle **ControladorAmbiente** e as classes de entidade **Mensagem**, **Solicitação**, **Projeto** e **Usuário**.

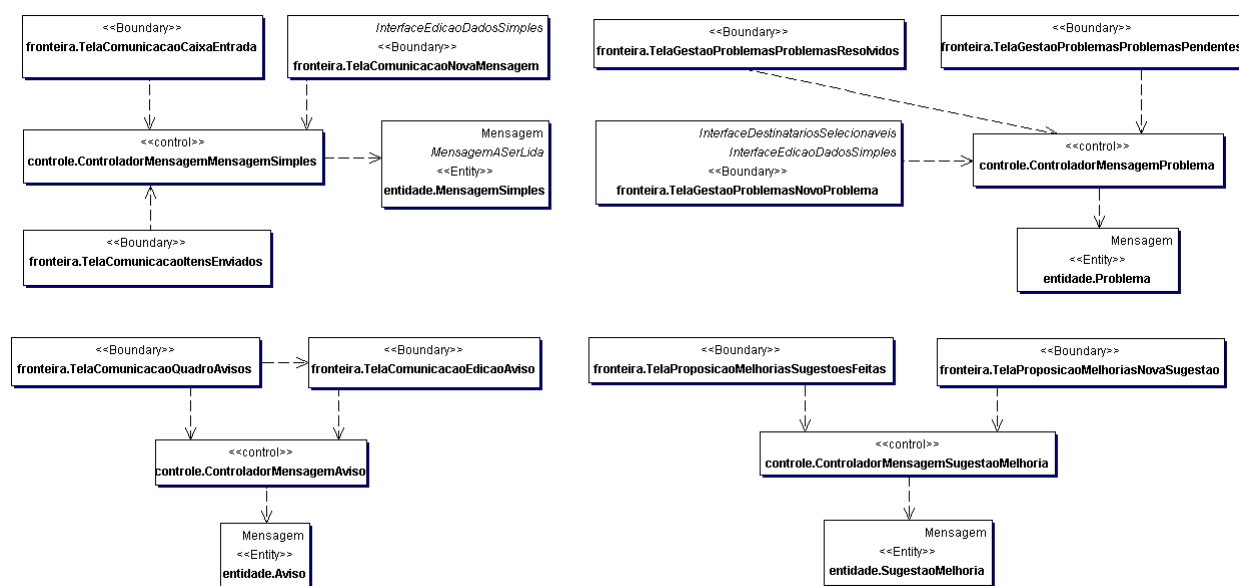


Figura 16 - Diagrama de Classes Comunicação

O diagrama da Figura 16 apresenta as classes ligadas à Comunicação da Equipe. Tais classes compreendem aquelas que tratam trocas de mensagens, manutenção de quadro de avisos, gestão de problemas e proposição de melhorias para o processo. Este diagrama contém as classes de fronteira, controle e entidade que representam os conceitos citados.

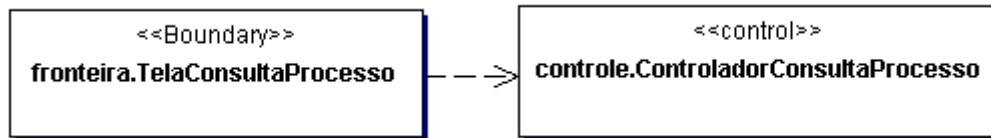


Figura 17 - Diagrama de Classes Consulta ao Processo

O diagrama da Figura 17 contempla as classes relacionadas ao tópico de Consulta ao Processo. Ele contém apenas as classes de fronteira e controle envolvidas com este tópico. A classe de controle extrai as informações a serem exibidas pela classe de fronteira a partir de arquivos HTML, que formam um dos componentes usados pelo Praxis Mentor.

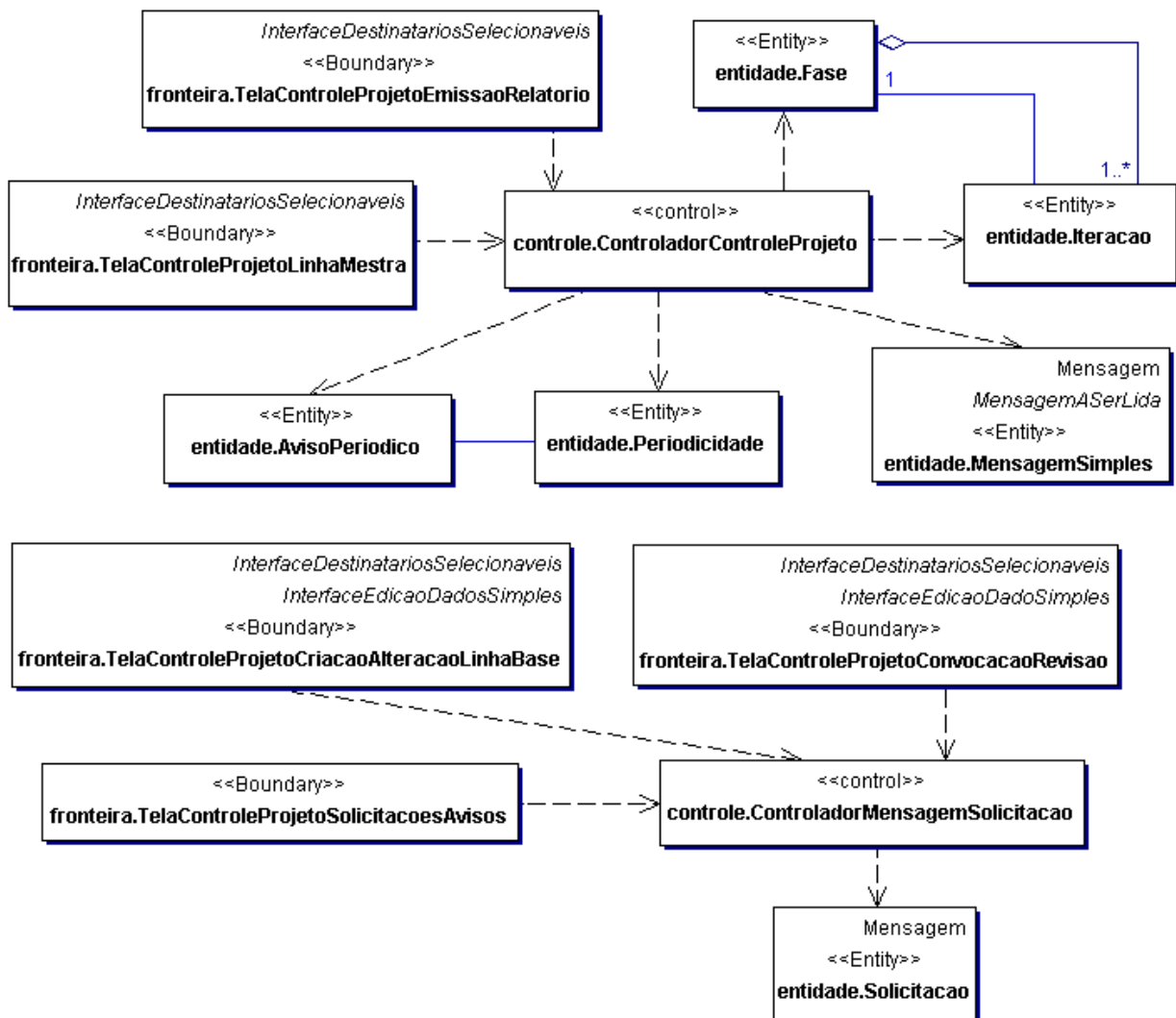


Figura 18 - Diagrama de Classes Controle de Projetos

O diagrama da Figura 18 apresenta as classes relacionadas ao Controle de Projetos. Estão

aqui presentes as classes que representam atividades como o replanejamento, a emissão de relatórios, a convocação de revisões, a criação e alteração de Linha de Base e as solicitações e avisos emitidos. Todas as classes de fronteira, controle e entidade relacionadas ao Controle de Projetos se encontram neste diagrama.

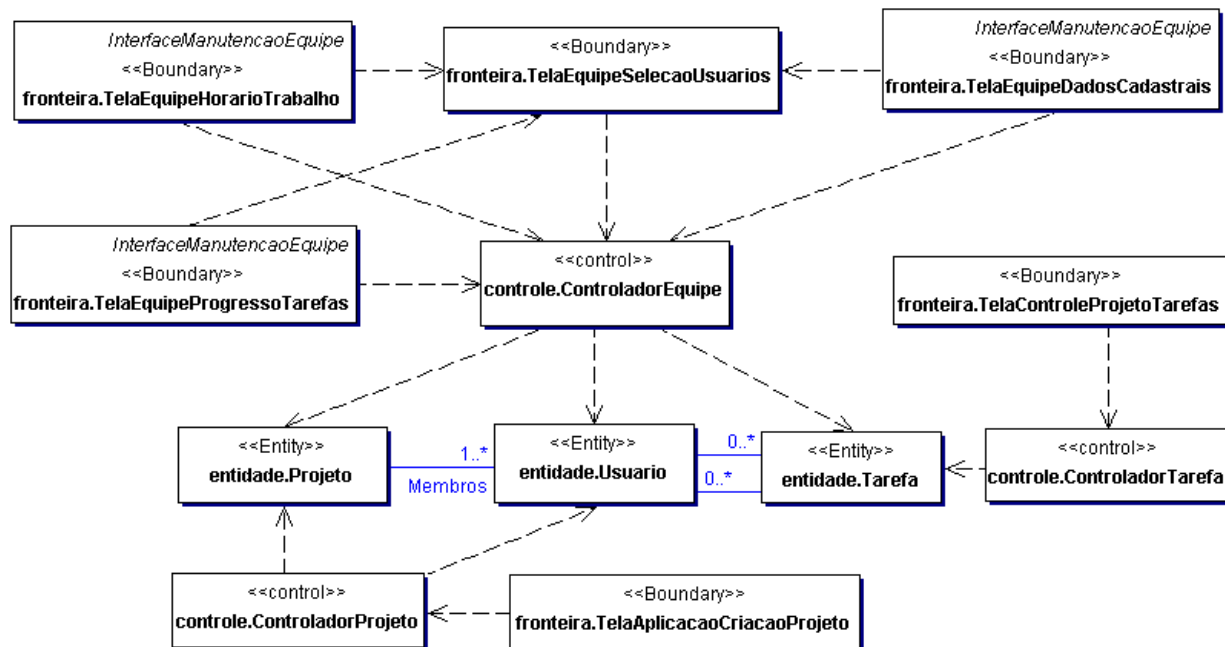


Figura 19 - Diagrama de Classes Planejamento de Projetos

O diagrama da Figura 19 contempla as classes relacionadas ao Planejamento de Projetos. As classes aqui apresentadas estão envolvidas com a composição da equipe e com a criação de projetos. As atividades de composição da equipe compreendem informações sobre o horário de trabalho, o progresso das tarefas da equipe, os dados cadastrais e a distribuição das tarefas pelos membros da equipe. Já a criação de um projeto envolve a manipulação dos artefatos do processo e a definição de um gerente para o mesmo. Este diagrama contém, assim como os anteriores, todas as classes de fronteira, controle e entidade relacionadas aos conceitos citados.

3.2.1.7.2 Realizações dos casos de uso

Como dito anteriormente, as realizações dos casos de uso mostram como os objetos das classes descritas colaboram entre si para realizar os principais roteiros que podem ser percorridos dentro de cada caso de uso. Nas figuras a seguir são apresentados roteiros das realizações dos casos de uso **Controle de Mensagens Recebidas** e **Criação de Projetos** que tiveram seus fluxos detalhados na seção 3.2.1.5 (Detalhamento dos requisitos funcionais). Para o caso de uso Controle de Mensagens Recebidas é apresentado o **Fluxo Alternativo Responder ao Remetente** e para o caso de uso Criação de Projetos é apresentado o **Fluxo Principal**.

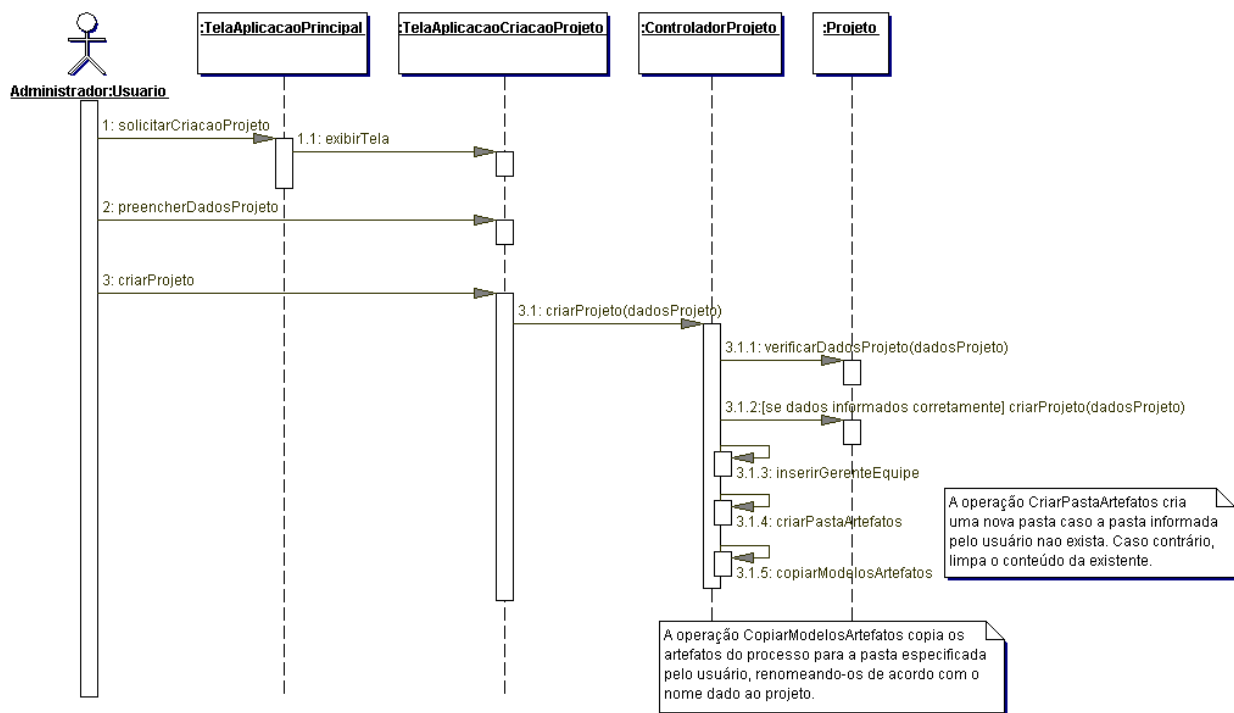


Figura 20 - Realizações do Caso de Uso Criação de Projetos – Fluxo Principal

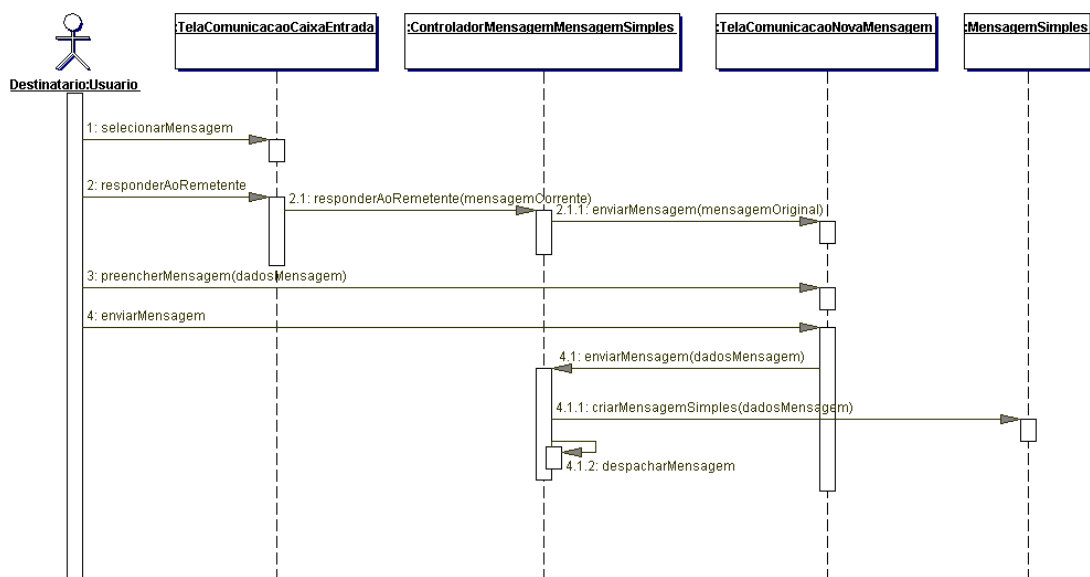


Figura 21 - Realizações do Caso de Uso Controle de Mensagens Recebidas – Fluxo Alternativo Responder ao Remetente

3.2.1.8 Classificação dos requisitos

Nesta atividade são listados os requisitos que farão parte do Cadastro dos Requisitos do Software (CRSw). No Praxis, cada interface, caso de uso, classe e requisito não funcional é considerado como um requisito diferente e individual.

A Tabela 20 mostra o cadastro inicial dos requisitos do Praxis Mentor. A coluna **Importância** se refere à definição da priorização dos requisitos. A coluna **Complexidade** diz respeito à estimativa do esforço e dos riscos de implementação em comparação com outros

requisitos do projeto e aceita os valores alta, baixa ou média. Finalmente, a coluna **Estabilidade** define a estimativa da probabilidade de que o requisito venha a ser alterado no decorrer do projeto, com base na experiência de projetos correlatos, podendo também ser alta, baixa ou média.

Para exemplificar a classificação dada, pode-se observar o requisito 1. Ele foi classificado como **Desejável** porque se trata de uma tela de apoio que serve apenas para facilitar a escolha dos destinatários de uma mensagem. Isto significa que esta tela não deveria ser implementada se não houvesse tempo hábil para tal. Sua complexidade foi considerada **Média** porque se trata de uma tela com poucos componentes e facilmente alterável. Sua estabilidade também foi considerada **Média** porque o usuário pode pedir alterações na mesma devido a problemas de usabilidade, por exemplo.

Número	Nome do requisito	Tipo	Importância	Complexidade	Estabilidade
1	Interface de Usuário Tela Destinatários	Interface	Desejável	Média	Média
2	Interface de Usuário Tela Comunicação – Caixa de Entrada	Interface	Essencial	Baixa	Alta
3	Caso de uso Controle de Mensagens Recebidas	Caso de uso	Essencial	Baixa	Alta
4	Classe de Controle ControladorProjeto	Classe	Essencial	Baixa	Média
5	Classe de Fronteira TelaAplicacaoCriacaoProjeto	Classe	Essencial	Média	Média
6	Atributo da qualidade Segurança do Acesso	Não funcional	Desejável	Média	Média

Tabela 20 - Amostra do cadastro inicial dos requisitos do Praxis Mentor

3.2.1.9 Decisões de especificação

Durante a especificação dos requisitos do Praxis Mentor, foram tomadas algumas decisões que normalmente só são tomadas na fase de Construção. Tais decisões foram antecipadas principalmente por restrições de tempo do projeto. É importante que estas decisões fiquem documentadas na Especificação dos Requisitos para que fique claro o motivo da antecipação das mesmas.

As tabelas a seguir apresentam as decisões que foram antecipadas para o Praxis Mentor:

O ambiente de desenvolvimento a ser utilizado na fase de construção já foi definido e será o Java. Esta decisão foi antecipada como uma forma de adiantar algumas das atividades do desenho em função do pouco tempo disponível para o término do projeto.

A linguagem Java foi escolhida por ser atual, de grande difusão no mercado e por fornecer um excelente suporte a tecnologias como Orientação por Objetos e *Patterns*.

Tabela 21 - Decisão sobre o ambiente de desenvolvimento do Praxis Mentor

A nomenclatura utilizada no Modelo de Análise deste documento está voltada para a implementação na linguagem Java. Normalmente a definição dessa nomenclatura é uma decisão de desenho, mas foi aqui antecipada em virtude do pouco tempo disponível para a conclusão do projeto e da definição prévia do ambiente de desenvolvimento.

Tabela 22 - Decisão sobre a nomenclatura utilizada no Modelo de Análise do Praxis Mentor

3.2.2 Revisão da Especificação dos Requisitos do Software

Conforme mencionado anteriormente, o fechamento da fase de Elaboração inclui três tipos de procedimentos de controle: uma revisão técnica formal, uma auditoria da qualidade e uma revisão gerencial da equipe responsável pela Elaboração.

Para o fechamento desta fase do projeto Praxis Mentor, os dois primeiros procedimentos de controle foram realizados e o terceiro não foi necessário pois a Equipe do projeto é composta por uma única pessoa.

A auditoria da qualidade foi realizada pelo professor orientador deste projeto, Wilson de Pádua Paula Filho, e resultou em algumas mudanças em relação à conformidade com os padrões do Praxis.

A revisão técnica formal não foi realizada exatamente como é prevista no Praxis. De acordo com o processo, um grupo de revisores independentes (que não pertencem à equipe responsável pela Elaboração) deveria realizar a verificação da qualidade técnica da Especificação e produzir um relatório com os resultados. Por falta de tempo e pessoal adequado para a realização de uma revisão como esta, optou-se apenas por uma revisão informal feita por uma pessoa com conhecimento do processo que produzisse um documento contendo críticas e sugestões de melhorias para o material. Esta revisão informal foi feita pela ex-aluna de graduação da UFMG e atual aluna de mestrado da UNICAMP – SP, Gisele Rodrigues de Mesquita Ferreira.

As sugestões fornecidas por Gisele foram analisadas e algumas foram implementadas. Outras foram até consideradas interessantes, mas demandariam um tempo maior do que o disponível para melhoria do material e foram postergadas para trabalhos futuros. Um documento contendo as sugestões e análise das mesmas foi produzido e arquivado junto com a ERSw. A Tabela 23 apresenta exemplos de sugestões aceitas, rejeitadas e postergadas com a respectiva justificativa fornecida:


Número de ordem	Sugestão	Situação	Justificativa
1	Seção 1.2.3 Limites do Produto: “O Praxis Mentor 1.0 não fará integração com alguma ferramenta de suporte ao processo como Rose, editores de texto...”. Isto só está falado na seção 2.6 e está causando suspense.	Implementada	Será colocado como limite da versão 1.0.
2	Tela Quadro de Avisos – Edição: falta um botão para fechar esta janela ou cancelar a edição.	Rejeitada	A idéia é ser uma tela simples, similar às anotações do <i>Outlook</i> . A tela é fechada através do botão  .
3	Tanto Dados Pessoais quanto Horário de Trabalho e Tarefas permitem a criação e remoção de membros da equipe. Acho que isto deveria ser mudado. Deveria existir uma tela que mostrasse uma lista dos membros da equipe onde as opções de novo e excluir estariam disponíveis. Além disto, para um membro selecionado na lista, deveria haver botões que levassem aos seus dados pessoais, horário de trabalho e tarefas.	Postergada	É uma boa sugestão, mas a forma atual não é pior e, por falta de tempo, será melhor deixar como está. No futuro esta opção poderá ser reavaliada.

Tabela 23 - Resultado da revisão técnica informal do Praxis Mentor

3.2.3 Plano de Desenvolvimento do Software

Segundo Paula [Paula01], um Plano de Desenvolvimento do Software (PDSw) deve fornecer:

- os elementos necessários para a decisão de levar adiante as fases de Construção e Transição de um projeto de software, nos termos propostos pela Especificação dos Requisitos do Software;
- a base para o acompanhamento e controle do projeto, até a colocação de seus resultados em operação.

Neste trabalho serão apresentados dois aspectos relativos aos tópicos citados: o dimensionamento do projeto e sua análise de riscos. A análise de riscos permite a minimização dos problemas oriundos da concretização de algum risco associado ao projeto. Já o dimensionamento é conseguido através da aplicação de procedimentos de planejamento de projetos baseados nas práticas mais comuns descritas na literatura. No caso deste trabalho o procedimento escolhido foi uma técnica sugerida pelo próprio processo: a Análise de Pontos de Função. Uma breve definição desta técnica será mostrada antes da apresentação do planejamento do Praxis Mentor.

3.2.3.1 Análise de Pontos de Função

A Análise de Pontos de Função (APF) é uma técnica para mensuração de sistemas desenvolvida inicialmente na IBM em 1974 e aprimorada nos anos seguintes através da experiência de suas aplicações práticas. Um Manual de Contagem de Pontos de Função foi publicado no final dos anos 80 pelo Grupo Internacional de Usuários de Pontos de Função (IFPUG). Atualmente, este manual se encontra na versão 4.0 e é utilizado por empresas associadas ao grupo em diversos países. Maiores informações sobre a técnica e o manual podem ser obtidas em [Ambler97].

A grande vantagem desta técnica é que ela se baseia na teoria de que as funções de uma aplicação são a melhor medida para o tamanho da mesma. A APF dimensiona o sistema de acordo com a perspectiva do usuário, quantificando a funcionalidade que o sistema proporciona a ele. Este fato torna a técnica de avaliação independente da linguagem de programação, do processo de desenvolvimento ou da tecnologia utilizada e faz com que ela possa ser aplicada em várias fases do ciclo de vida de desenvolvimento.

Uma vez calculados os pontos de função, eles podem ser utilizados com diversas finalidades:

- medida da produtividade da equipe;
- cálculo do esforço do projeto e do tempo gasto;
- cálculo da taxa de manutenção do software;
- comparação de produtividade entre ambientes de desenvolvimento.

3.2.3.2 Dimensionamento do Praxis Mentor

Como citado anteriormente, a técnica de Análise de Pontos de Função foi utilizada na elaboração do Plano de Desenvolvimento do Software do Praxis Mentor para a obtenção do dimensionamento do sistema e geração de estimativas de esforço e prazo.

Ao final do projeto outras informações importantes também poderão ser extraídas a partir do resultado da análise feita. Dados sobre a produtividade da equipe e uma avaliação da precisão das estimativas feitas são bons exemplos deste tipo de informação. Outro fator importante é que esses dados formam uma base histórica de desenvolvimento que poderá ser usada em projetos futuros para aumentar a precisão das estimativas.

O resultado da Análise de Pontos de Função do Praxis Mentor gerou um total de 341,22 pontos de função. Para se ter uma idéia rápida sobre o dimensionamento do sistema, pode-se derivar o número de linhas de código correspondente ao número de pontos de função através de tabelas publicadas na literatura [Jones94]. Para a linguagem Java, que foi utilizada neste projeto, esse número é 50. O tamanho do projeto em linhas de código é, portanto, 17.061. Se for considerada uma produtividade de 2500 linhas de código por mês, o que tem se mostrado verdadeiro durante o desenvolvimento deste projeto, o tempo total para o desenvolvimento seria de aproximadamente 6,8 meses.

Para a obtenção de estimativas mais precisas pode-se converter os pontos de função para linhas de código e utilizar um macro modelo de estimativa como o COCOMO (*Constructive Cost Model*) ou usar regressão linear de dados obtidos numa base histórica de projetos anteriores. Tais procedimentos são descritos em [Boehm00] e [Humphrey95], respectivamente.

Como no caso do Praxis Mentor não existia uma base histórica de projetos anteriores, os cálculos de estimativas foram feitos utilizando o macro modelo COCOMO e alguns conhecimentos adquiridos em projetos anteriores, mas que ainda não haviam formado uma base histórica.

Os cálculos realizados são apresentados nas tabelas a seguir. O cálculo detalhado dos pontos de função foi omitido por ser extenso e fugir ao escopo deste trabalho. Sendo assim, os cálculos mostrados a seguir começam a partir do número total de pontos de função encontrados: 341,22 que equivalem a 17.061 linhas de código, como explicado anteriormente.

Regras do COCOMO	
Linhas de código	17061,0
Esforço: $PM = 2,4 * (KLC)^{1,05}$	18,0

Tabela 24 - Regras do COCOMO aplicadas ao Praxis Mentor

Na Tabela 24, KLC é igual ao número de linhas de código dividido pela produtividade da equipe (LC/2500). Como já foi dito anteriormente, a produtividade considerada neste projeto foi de 2500 linhas de código por mês. Esta produtividade foi inferida de experiências em projetos anteriores que utilizavam a linguagem Java e comprovada pela implementação dos primeiros casos de uso deste projeto.

Poderia ter sido usada a tabela de McConnel, apresentada em [McConnell96], para o cálculo desta produtividade e, conseqüentemente, do esforço. Entretanto, os resultados conseguidos pelo

uso da Regra de COCOMO se apresentaram mais realistas. De acordo com a tabela de McConnell, a produtividade de um projeto com 17.000 linhas de código é de aproximadamente 600 LC/pessoa-mês e o esforço, calculado como LC/produtividade, é de 28,4 pessoas/mês. A análise dos casos de uso já implementados mostra que o cálculo que utiliza a Regra de COCOMO está mais próximo da realidade e, por isso, ele foi utilizado para a obtenção das demais estimativas.

Tabela de McConnell	
Produtividade	600
Esforço	28,4

Tabela 25 - Tabela de McConnell aplicada ao Praxis Mentor

O esforço calculado na Tabela 24 é dado em pessoas/mês. Considerando-se que uma pessoa trabalhe 160 horas em um mês, pode-se multiplicar o esforço encontrado e obter-se o número de horas a serem trabalhadas: 2885. Baseando-se, finalmente, na experiência do uso do processo Praxis ou em dados históricos, se existirem, pode-se gerar a distribuição destas horas pelas fases do projeto e se obter a distribuição do esforço de desenvolvimento por fases.

Pacote de trabalho	Sigla	% esforço desenvolvimento	Horas de desenvolvedores
Desenho inicial	DI	23,0	663
Liberação 1	L1	33,0	952
Liberação 2	L2	33,3	961
Testes alfa	TA	10,7	309
Total		100,0	2885

Tabela 26 - Distribuição do esforço de desenvolvimento do Praxis Mentor por fases

Em projetos que contenham uma equipe de maior número de pessoas, é interessante também o cálculo da distribuição do esforço por recursos. No caso do Praxis, a equipe é composta de apenas uma pessoa e este cálculo torna-se desnecessário.

Outra estimativa interessante a ser obtida é a de orçamento. Normalmente é preferível não mostrar gastos financeiros reais no PDSw, usando-se, no lugar disto, a mesma unidade de esforço utilizada para computar o **valor adquirido**⁵ no projeto [Humphrey95]. Por exemplo, o orçamento pode ser expresso em horas médias de pessoal do fornecedor. Ele deve incluir o cálculo da evolução prevista do valor adquirido e o esforço necessário após o fim do desenvolvimento (Testes Beta e Operação Piloto). Tal esforço é obtido a partir da experiência de projetos anteriores, ou simplesmente estimado, no caso de não existir tal experiência.

⁵ No cálculo de valor adquirido, o valor de cada atividade é definido como o quociente entre o respectivo esforço e o esforço total do projeto.

Pacote de trabalho	Horas do fornecedor	Valor da atividade	Horas acumuladas	Valor adquirido
DI	663	0,22	663	0,22
L1	952	0,31	1615	0,53
L2	961	0,32	2576	0,85
TA	309	0,10	2885	0,95
TB	88	0,03	2973	0,98
OP	60	0,02	3033	1,00
Total	3033	1,00	3033	1,00

Tabela 27 - Orçamento do Praxis Mentor

Finalmente pode-se derivar o cronograma do projeto a partir do cálculo de pontos de função e conseqüente cálculo de Linhas de Código. Este cronograma é apresentado através de uma tabela e de um diagrama de Gantt.

Atividade	Início	Fim	Duração (dias)
Desenvolvimento total	07/08/00	18/01/02	379,14
Construção	07/08/00	24/12/01	360,64
Desenho Inicial	07/08/00	29/11/00	82,88
Liberação # 1	30/11/00	15/05/01	119,00
Liberação # 2	16/05/01	31/10/01	120,13
Testes Alfa	01/11/01	24/12/01	38,63
Transição	26/12/01	18/01/02	18,50
Testes Beta	26/12/01	08/01/02	11,00
Operação Piloto	09/01/02	18/01/02	7,50

Tabela 28 - Cronograma tabular do Praxis Mentor

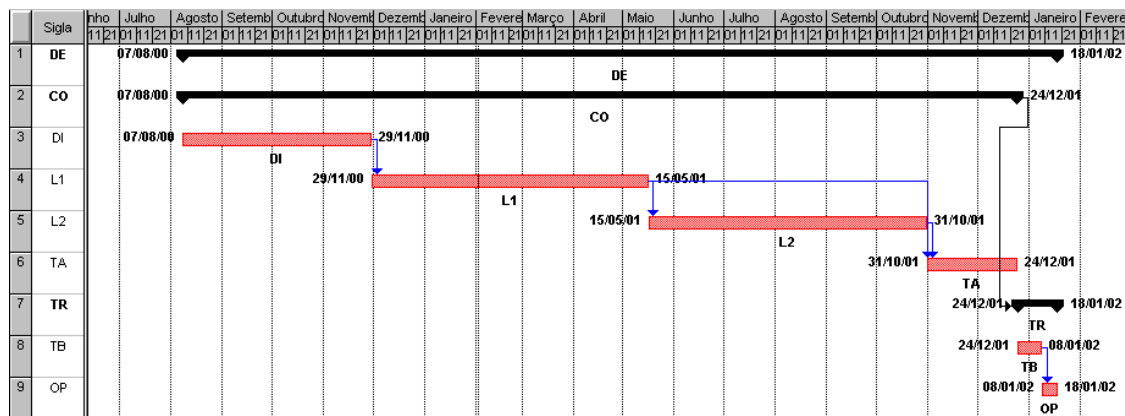


Figura 22 - Diagrama de Gantt do Praxis Mentor

Como pode ser visto através do cronograma do Praxis Mentor, o prazo para término do projeto é muito maior (quase um ano a mais) do que o prazo final deste trabalho. A estratégia adotada para se tentar cumprir a maior parte do cronograma dentro do tempo disponível foi elaborar o desenho de todos os casos de uso da ferramenta e assumir que só seriam

implementados aqueles casos de uso que estivessem completamente fechados até o fim do prazo deste trabalho. A implementação dos demais casos de uso ficaria para trabalhos futuros de fechamento e melhoria da ferramenta. Outra estratégia foi eliminar os procedimentos padronizados de testes e as atividades da fase de Transição, que também foram postergados para trabalhos futuros.

3.2.3.3 Análise de riscos do Praxis Mentor

Para que os problemas decorrentes da concretização de algum risco associado ao projeto sejam minimizados, é interessante que os riscos sejam levantados e documentados o quanto antes.

Segundo Paula [Paula01], deve-se descrever no PDSw os principais riscos associados ao projeto, excluindo-se hipóteses catastróficas e muito pouco prováveis. Para cada risco identificado deve-se apresentar sua gravidade, probabilidade de ocorrência, impacto previsto no projeto e respectiva contramedida (curativa ou preventiva). Além disto, pode-se indicar os procedimentos de monitoração e análise dos riscos ao longo do projeto.

Os riscos identificados para o Praxis Mentor e os procedimentos definidos para sua análise e monitoração são apresentados nas tabelas a seguir.

Número	Risco	Gravidade	Probabilidade de Ocorrência	Impacto previsto	Contramedidas previstas
1	Defeitos na ERSw	Média	Média	Vários dias de atraso por alteração de requisitos	Incluir na primeira liberação executável os requisitos mais complexos
2	Falta de material para criar a parte de Consulta ao Processo.	Alta	Baixa	Impossibilidade de implementar a interface de Consulta ao Processo	Cobrar providência do cliente

Tabela 29 - Principais riscos ao projeto Praxis Mentor

Acompanhamento dos riscos dentro dos Relatórios Mensais do Projeto Praxis Mentor 1.0. A cada mês, a tabela de riscos do projeto será atualizada, levando-se em conta a evolução do projeto e do ambiente externo. Será também atualizada uma tabela de riscos concretizados, indicando-se a respectiva data de ocorrência e o impacto causado no projeto.

Tabela 30 - Procedimentos de análise dos riscos do Praxis Mentor

3.2.4 Plano da Qualidade do Software

Segundo Paula [Paula01], um Plano da Qualidade do Software deve fornecer a base para o acompanhamento e controle do projeto por parte do Grupo da Garantia da Qualidade do Software.

O preenchimento do Plano da Qualidade do Software pode ser facilitado pelo uso das tabelas contidas em um gabarito (*template*), que faz parte da documentação auxiliar do processo. Em algumas seções, o gabarito é preenchido com dados relativos ao Praxis; em cada projeto, só é necessário completar dados específicos do projeto ou variações em relação ao Praxis.

Neste trabalho serão apresentadas apenas a seções que contenham dados específicos do Praxis Mentor ou que tenham sofrido variações em relação ao processo Praxis.

3.2.4.1 Responsabilidades no projeto

É importante descrever as responsabilidades das pessoas que são designadas para cada tarefa da Garantia da Qualidade. Neste projeto a equipe é muito reduzida e basicamente a estratégia é de que o orientador do projeto exerça o papel de Gestor da Qualidade enquanto que os demais papéis sejam exercidos pela própria autora do trabalho.

Número de ordem	Função relativa a este projeto	Responsável
1	Gerente do Projeto	Júnia Gaudereto Carvalho
2	Gestor da Qualidade	Wilson de Pádua Paula Filho
3	Gestor de Requisitos	Júnia Gaudereto Carvalho
4	Gestor de Documentos	Júnia Gaudereto Carvalho
5	Contato no GGQSw	Júnia Gaudereto Carvalho
6	Contato no GGCSw	Júnia Gaudereto Carvalho

Tabela 31 - Responsabilidades da equipe do Praxis Mentor

3.2.4.2 Ferramentas de Garantia da Qualidade

Os artefatos gerados durante o projeto, tanto técnicos quanto gerenciais, são de grande importância e complexidade. A elaboração destes artefatos com qualidade e precisão só poderia ser possível com o uso de ferramentas de apoio desenvolvidas para o fim específico.

O artefato técnico Especificação dos Requisitos do Software, por exemplo, contempla um Modelo de Análise do sistema que deve ser construído utilizando a UML. Para a elaboração desse modelo, é necessário o uso de uma ferramenta CASE que ofereça suporte à UML e facilite o trabalho de desenho dos diagramas.

É muito importante que seja feita uma escolha criteriosa das ferramentas a serem utilizadas. Uma vez identificadas, as ferramentas devem ser documentadas. Deve-se armazenar o nome do fabricante e o objetivo com o qual cada ferramenta é utilizada no projeto.

Número de ordem	Objetivo	Nome	Fabricante
1	Análise de rastreabilidade	Together 4.1	TogetherSoft
2	Verificação do Modelo de Análise	Together 4.1	TogetherSoft
3	Verificação do Modelo de Desenho	Together 4.1	TogetherSoft
4	Gestão de Configurações	Visual Source Safe 5.0	Microsoft
5	Verificação do Código	JBuilder 4 Foundation	Borland
6	Depuração do Código	JBuilder 4 Foundation	Borland
7	Automação dos Testes	Jtest 3.2	ParaSoft Corporation

Tabela 32 - Ferramentas de Garantia da Qualidade do Praxis Mentor

Nas próximas seções será fornecida uma breve descrição das ferramentas utilizadas neste projeto, explicitando-se como cada uma delas colaborou para o desenvolvimento do mesmo.

3.2.4.2.1 Together 4.1

O Together é uma ferramenta CASE que oferece suporte à UML e que é muito útil nas tarefas de análise e desenho de sistemas. Além disso, apesar de não ter como objetivo principal ser uma ferramenta para desenvolvimento de código, possibilita a execução desta tarefa com facilidades como preenchimento automático de código e compilação e execução integrada ao ambiente.

O principal destaque do Together é a garantia de total coerência entre os modelos visuais e o código fonte associado, o que caracteriza seu lema “*Always in Sync*”. Durante as fases de desenho e implementação, esse recurso é de extrema utilidade, evitando os infundáveis ciclos de engenharia direta (geração de código a partir de modelos) e engenharia reversa (abstração de modelos a partir de código). Esses passos, além de demorados, costumam provocar grandes perdas de informação.

Finalmente, o suporte oferecido pelo Together aos principais diagramas da UML, padrões de desenho (*design patterns*) e características específicas da linguagem Java fazem dele uma ferramenta valiosa para os desenvolvedores de software.

O uso desta ferramenta no projeto Praxis Mentor foi fundamental para garantir a qualidade da modelagem e do código gerado a partir dela e para reduzir o tempo de desenvolvimento que poderia ter sido consideravelmente maior levando-se em conta as características da linguagem Java [Eckel00]. Maiores informações sobre esta ferramenta podem ser encontradas em [Together00].

3.2.4.2.2 Visual SourceSafe 5.0

O Microsoft Visual SourceSafe 5.0 é uma ferramenta completa de Gestão de Configurações. Ela possui todos os recursos apontados por Paula, em [Paula01], como necessários para o gerenciamento de versões. Tais recursos incluem:

- identificação única para os Itens de Configuração de Software;
- suporte a formas controladas de extração (*check-out*) e inserção (*check-in*) de qualquer versão de um Item de Configuração;
- levantamento e visualização das diferenças entre os itens das Linhas de Base e as versões em desenvolvimento;
- fornecimento de recursos para manutenção da integridade e segurança das Bibliotecas de Gestão de Configurações de Software;
- emissão de relatórios de *status* das Bibliotecas de Gestão de Configurações de Software.

Esta ferramenta foi utilizada neste projeto e se apresentou muito satisfatória na tarefa de gestão de configurações. Ela permitiu o controle das alterações dos artefatos do projeto e ajudou a garantir a segurança e a integridade dos mesmos. Maiores informações sobre a ferramenta podem ser encontradas em [SourceSafe96].

3.2.4.2.3 JBuilder 4 Foundation

O JBuilder é um ambiente de desenvolvimento em Java desenvolvido pela Borland. A sua versão 4 Foundation é totalmente feita em Java e distribuída como *freeware*.

Como principais destaques desta ferramenta, pode-se citar:

- **ambiente visual para desenvolvimento de interface:** permite ao desenvolvedor criar, através de manipulação direta, objetos visuais como botões, painéis, caixas e listas de seleção, sem a necessidade de escrever linhas de código (exigência comum em ferramentas de desenvolvimento sem ambiente visual para implementação de interface);
- **integração com sistemas de ajuda do Java:** durante a implementação do código fonte, é possível acessar os sistemas de ajuda do Java para um contexto específico como uma classe ou um método sobre o qual se queira maiores informações;
- **recursos auxiliares ao desenvolvimento de código:** preenchimento automático de código a partir de prefixo e coloração de palavras chaves e elementos básicos de linguagem são alguns dos recursos disponíveis nessa ferramenta, facilitando sobremaneira o desenvolvimento de código.

O uso desta ferramenta no desenvolvimento do Praxis Mentor foi de extrema importância para reduzir o tempo de construção das interfaces. Ela se apresentou muito adequada para o tipo de desenvolvimento requerido. Maiores informações sobre o JBuilder 4 Foundation podem ser encontradas em [JBuilder00].

3.2.4.2.4 Jtest 3.2

O Jtest é uma ferramenta de automação de testes que trabalha tanto com análise estática quanto com análise dinâmica de código.

Em termos de análise estática, ele contém um conjunto de cerca de 180 regras que podem ser aplicadas ao código fonte de modo a identificar práticas ruins ou mesmo erros em classes sem a necessidade de execução das mesmas. Essas regras são separadas em cinco níveis de severidade e podem ser habilitadas ou desabilitadas uma a uma para a realização de testes específicos.

Em termos de análise dinâmica, o Jtest instancia objetos e executa métodos para classes conforme o desejo do usuário, identificando quebras de contrato como a ocorrência de exceções não especificadas durante a chamada de métodos com valores inválidos. Inúmeros tipos de testes como caixa preta, caixa branca ou condições de contorno podem ser aplicados ao código fonte de acordo com a necessidade dos usuários. Também é possível realizar testes de regressão com o Jtest.

Esta ferramenta foi pouco utilizada neste trabalho devido ao fato de ele ter sido encerrado na iteração de Testes Alfa da fase de Construção. Mas poderá ser muito utilizada na continuação deste projeto, pois já se apresentou muito satisfatória mesmo para os poucos testes realizados. Maiores informações sobre o Jtest poderão ser encontradas em [Jtest00].

3.2.4.3 Política de controle de mídia

É também muito importante para o projeto a definição de procedimentos a serem seguidos para garantir a qualidade e integridade dos arquivos do mesmo. Devem ser especificadas a localização física da base de dados e a frequência e mídia de backup.

Todas as linhas de base serão guardadas no projeto Praxis Mentor 1.0 da Biblioteca de Gestão de Configurações, localizada na pasta Praxis Mentor 1.0 da máquina *Júnia* no escritório de Júnia Gaudereto Carvalho. Esta base de dados será copiada em *zip disk*, com periodicidade diária, e CD-ROM, com periodicidade mensal.

Tabela 33 - Política de controle de mídia do Praxis Mentor

Capítulo 4

Construção

4.1 Definição

A fase de Construção é aquela em que é desenvolvida uma liberação completamente operacional do produto atendendo aos requisitos especificados. Ela parte de uma iteração de Desenho Inicial, na qual é realizado o desenho do produto em um nível mais alto de abstração, de forma a permitir a divisão das funções e dos componentes do produto ao longo das iterações seguintes. Tais iterações constituem as Liberações, ou seja, implementações parciais das funções do produto até que se atinja o produto final. A integração final do produto é testada na iteração Testes Alfa. É importante notar que todas as liberações, exceto a última, são liberações parciais por não contemplarem todos os requisitos especificados.

4.1.1 Desenho Inicial

O Desenho Inicial constitui o principal passo da fase de Construção. Segundo Paula [Paula01], um desenho bem feito pode garantir os seguintes benefícios:

- produção de uma arquitetura robusta, estável e flexível;
- produção de interfaces de usuários que tornem o produto fácil de usar;
- escolha de soluções tecnológicas mais adequadas para satisfazer os requisitos de forma rápida, barata e confiável;
- identificação de componentes comerciais ou de projetos anteriores que possam ser reutilizados, reduzindo o esforço de desenvolvimento;
- decisão para as questões técnicas importantes para a implementação, deixando para a confecção das liberações apenas a realização de detalhes;
- divisão adequada do produto em componentes cuja implementação possa ser dividida de forma eficaz entre os membros da equipe de desenvolvedores, diminuindo os prazos de implementação.

A atividade inicial do desenho é o desenho da estrutura arquitetônica principal do produto, que consiste na divisão do mesmo em camadas, pacotes lógicos e subsistemas. Esta atividade envolve decisões técnicas estratégicas como a identificação de tecnologias adequadas para a implementação de subsistemas e a localização de componentes externos reutilizáveis.

Uma segunda atividade é o desenho das interfaces de usuário que deve atender aos requisitos de usabilidade. Dependendo do nível de importância destes requisitos torna-se necessária a realização de testes de usabilidade. Uma vez desenhadas as interfaces, os casos de uso devem ser detalhados em termos dos elementos das mesmas. As realizações dos casos de uso devem ser refeitas contemplando as classes de desenho. Estes casos de uso detalhados fornecem os elementos necessários para planejar e desenhar os testes de aceitação.

Outra atividade pertinente é o desenho dos dados persistentes, que identifica as soluções para o acoplamento entre o modelo interno de desenho, orientado por objetos, e a arquitetura de sistemas externos de armazenamento como, por exemplo, banco de dados relacionais.

Os resultados das atividades descritas anteriormente são sintetizados no artefato Descrição do Desenho do Software (DDSw).

Algumas atividades relacionadas aos testes de aceitação do produto também pertencem a esta iteração. O planejamento e a especificação destes testes, que resultam no artefato Descrição dos Testes do Software (DTSw), são algumas delas. Outra atividade relacionada aos testes de aceitação, mas que é opcional nesta iteração, é a geração dos scripts de automação dos testes. Tais scripts formam o artefato Bateria de Testes de Regressão do Software (BTRSsw).

O fim da iteração de Desenho Inicial se dá quando o produto tiver sido repartido em um conjunto de subsistemas com interfaces bem definidas entre si. A partir daí pode-se planejar as liberações, identificando o conjunto de casos de uso que cada uma implementará.

Os procedimentos de controle que devem ser realizados ao final desta iteração são: aprovação do DDSw em revisão técnica, aprovação do desenho das interfaces de usuário pelos usuários chaves, aprovação dos artefatos em auditoria da qualidade e aprovação dos artefatos em revisão gerencial, na qual é feito o balanço da iteração.

4.1.2 Liberações

Cada liberação, como explicado anteriormente, representa um subconjunto de funções do produto que será implementado e avaliado pelos usuários.

O desenho detalhado de uma liberação resolve aspectos como nome de todas as unidades a serem implementadas, assinatura e visibilidade das operações, implementação dos relacionamentos, questões de escopo, tratamentos de erros e exceções, algoritmos e estruturas de dados. Os testes de integração devem ser planejados e desenhados, geralmente utilizando subconjuntos dos testes de aceitação.

Durante a codificação, o desenho detalhado é convertido em código fonte e posteriormente executável pelos compiladores das linguagens de implementação. O código gerado é dividido em dois artefatos: Código Fonte do Software (CFSw) e Código Executável do Software (CESw). O código fonte deve ser inspecionado para a verificação de sua conformidade com o desenho detalhado e com o padrão de codificação. Após a inspeção, o novo código é compilado e os testes

de unidade são executados.

No processo de integração da liberação o código novo é ligado aos componentes produzidos nas liberações anteriores e aos componentes externos necessários. Após este momento os testes de integração são executados e a liberação é submetida à avaliação dos usuários. Os resultados dos testes de unidade e de integração realizados nesta iteração são consolidados no artefato Relatórios dos Testes do Software (RTSw).

Os procedimentos de controle para cada liberação são: inspeção do desenho detalhado e do código das novas unidades da liberação, aprovação da liberação pelos usuários chaves, aprovação da conformidade com o processo e dos resultados dos testes em auditoria da qualidade e aprovação em revisão gerencial na qual é feito o balanço da iteração.

4.1.3 Testes Alfa

A fase de Construção é encerrada com os Testes Alfa. Estes testes são elaborados no ambiente do fornecedor, mas é recomendável que sejam realizados por uma equipe especializada e independente da equipe de desenvolvimento. Durante os testes são identificados problemas que podem ser correspondentes a defeitos de desenho ou requisitos, por exemplo, e as correções necessárias devem ser feitas não apenas no código, mas em todos os documentos afetados. O Cadastro de Requisitos (CRSw), se adequadamente mantido, permitirá a identificação de todos os documentos afetados. O resultado destes testes é consolidado no artefato Relatórios dos Testes do Software (RTSw).

Nesta iteração também deve ser elaborado o material de suporte à fase de Transição. O principal item deste material é o Manual do Usuário do Software (MUSw), no qual os procedimentos dos usuários podem ser descritos a partir dos casos de uso detalhados e outros aspectos do desenho das interfaces de usuário.

No encerramento desta iteração, o único procedimento de controle previsto é uma auditoria da qualidade. Esta auditoria verificará os resultados dos testes de aceitação documentados nos relatórios de testes. Após a aprovação nesta auditoria o produto é entregue ao cliente para implantação nas instalações da fase de Transição.

4.2 Construção do Praxis Mentor

Durante a fase de Construção do Praxis Mentor foram produzidos os seguintes artefatos: Modelo de Desenho do Software (MDSw), Descrição do Desenho do Software (DDSw), Códigos Fontes do Software (CFSw), Códigos Executáveis do Software (CESw) e Manual do Usuário do Software (MUSw). Os artefatos referentes aos testes (BTRSw, DTSw e RTSw) não chegaram a ser confeccionados, visto que havia sido definido que este projeto seria paralisado na iteração de Testes Alfa e apenas o MUSw seria produzido nesta iteração. A decisão de encerrar o projeto neste ponto se deve ao esgotamento do prazo deste trabalho. Este esgotamento de prazo não significa que o projeto não tenha sido capaz de cumprir seus prazos. Na realidade, o prazo real do projeto Praxis Mentor seria daqui a um ano, conforme mostrado no capítulo anterior. Mas o presente trabalho de mestrado que contempla o projeto Praxis Mentor está chegando ao fim. As atividades referentes à iteração Testes Alfa e à fase de Transição foram, desta forma, postergadas

para projetos futuros de finalização e melhoria do Praxis Mentor.

Para ilustrar as decisões tomadas nesta fase, serão apresentados alguns trechos representativos dos artefatos MDSw, DDSw, CFSw e MUSw. Assim como no capítulo anterior, o conteúdo dos artefatos a serem apresentados não será mostrado na íntegra, pelo fato de eles serem muito extensos e detalhados.

4.2.1 Descrição do Desenho do Software

Durante a iteração de Desenho Inicial, foi concebida a arquitetura do Praxis Mentor e foram definidos seus componentes internos e externos mais importantes. Tal concepção foi realizada em um nível de detalhe suficiente para decidir as principais questões de arquitetura e tecnologia e permitir o planejamento detalhado das atividades de implementação. As decisões mais importantes em relação à estrutura do Praxis Mentor se encontram na Descrição do Desenho do Software.

A DDSw resulta do fluxo de Desenho e estabelece a estrutura com que o produto deverá ser implementado para satisfazer aos requisitos. Ela contempla o desenho das interfaces de usuário, o desenho interno e o desenho das liberações. O desenho das interfaces de usuário descreve a visão que os usuários terão do produto e a maneira com que eles irão interagir com ele. Já o desenho interno descreve as partes lógicas e físicas do modelo e a maneira como elas interagem entre si e com sistemas externos. Finalmente, o desenho das liberações descreve como o produto foi dividido em porções de funcionalidades a serem implementadas.

As seções mais representativas da DDSw do Praxis Mentor são apresentadas a seguir.

4.2.1.1 Desenho das interfaces de usuário

O desenho das interfaces de usuário abrange tanto o desenho externo (parte gráfica e funcional) quanto o desenho interno das classes de fronteira correspondentes. Diretrizes específicas para o desenho interno e externo de interfaces podem ser encontradas em [Paula01].

Uma preocupação que se deve ter ao desenhar interfaces é com relação à facilidade de uso. Esta facilidade deve estar diretamente ligada às pessoas que efetivamente irão utilizar o produto. Sendo assim, é interessante se ter uma descrição do perfil dos usuários do produto para que se possa definir mais precisamente e adequadamente as características das interfaces.

A Tabela 34 apresenta as características dos usuários do Praxis Mentor. Estas características foram levadas em consideração no desenho das interfaces da ferramenta. A característica da frequência de uso, por exemplo, levou à decisão da escolha de ícones, uma vez que a interface deveria ser facilmente entendida por um usuário de uso não freqüente. Levou também à proposta de uma barra de atalhos para atender aos usuários que fariam um uso mais freqüente da ferramenta. Já o conhecimento da área da aplicação, ou seja, o conhecimento da área de informática, induziu à utilização de uma interface no padrão do Windows, uma vez que a maioria dos usuários possui um completo domínio deste ambiente.

Não houve tempo hábil para a realização de testes de usabilidade com os futuros usuários do Praxis Mentor, o que teria sido interessante para a melhor definição das características dos mesmos. Neste caso, os testes de usabilidade poderão ser realizados com a versão final do

produto e os resultados poderão ser utilizados para a melhoria da interface em versões futuras do mesmo.

Número de ordem	Atores	Permissão de acesso	Frequência de uso	Nível educacional	Proficiência na aplicação	Proficiência em Informática
1	Usuário do Processo	Consulta ao Processo, Comunicação Interna da Equipe, Composição da Equipe, Alteração de Linha de Base, Gestão de Problemas e Proposição de Melhorias do Processo.	Diária	2º Grau, no mínimo	Completa	Processo Aplicação Windows 95
2	Administrador	Irrestrita.	Mensal	3º Grau	Completa	Processo Aplicação Windows 95
3	Gerente de Projeto	Restrito apenas à criação e exclusão de Projetos.	Diária	3º Grau	Completa	Processo Aplicação Windows 95
4	Desenvolvedor	Todos os do Usuário do Processo	Diária	2º grau	Completa	Processo Aplicação Windows 95
5	Membro do GGQSw	Todos os do Usuário do Processo	Diária	3º Grau	Completa	Processo Aplicação Windows 95
6	Membro do GGCSw	Todos os do Usuário do Processo mais: Criação ou Alteração de Linha de Base.	Semanal	3º Grau	Completa	Processo Aplicação Windows 95
7	Membro da CCCSw	Todos os do Usuário do Processo mais: Criação ou Alteração de Linha de Base.	Semanal	3º Grau	Completa	Processo Aplicação Windows 95

Tabela 34 - Características dos usuários do Praxis Mentor

4.2.1.1.1 Estrutura estática do produto

A estrutura estática do produto apresenta as relações estáticas entre as classes que implementam as interfaces de usuário. Ela pode descrever, por exemplo, como interfaces “mães” contêm interfaces “filhas”, ou a estrutura de pastas de interfaces implementadas como páginas da

Web. A estrutura estática do Praxis Mentor é apresentada na Tabela 35.

A *Tela Principal* é um quadro SDI (*single document interface*). Dentro dela, pastas e componentes multi-páginas simulam o funcionamento de um bloco indexado por abas de capítulos e folhas. Estas pastas e páginas compreendem a maior parte da funcionalidade da ferramenta.

Finalmente, existem interfaces que são implementadas como formulários independentes da *Tela Principal*:

- *Tela de Apresentação*, que é um formulário do tipo *Splash Screen*.
- *Tela Identificação* e *Tela Mudança de Senha*.
- *Tela Seleção de Projeto/Novo Projeto*, que é exibida após a *Tela Identificação* e antes da *Tela Principal*.
- *Tela Propriedades de Projeto / Novo Projeto*, *Tela Exclusão de Projeto*, *Tela Lista de Itens em Edição* e *Tela Seleção de Destinatários*.

Tabela 35 - Estrutura estática do Praxis Mentor

4.2.1.1.2 Estrutura dinâmica do produto

A estrutura dinâmica do produto mostra como é possível navegar (transferir controle) entre os objetos da interface de usuário. A Figura 23 apresenta a estrutura dinâmica do Praxis Mentor.

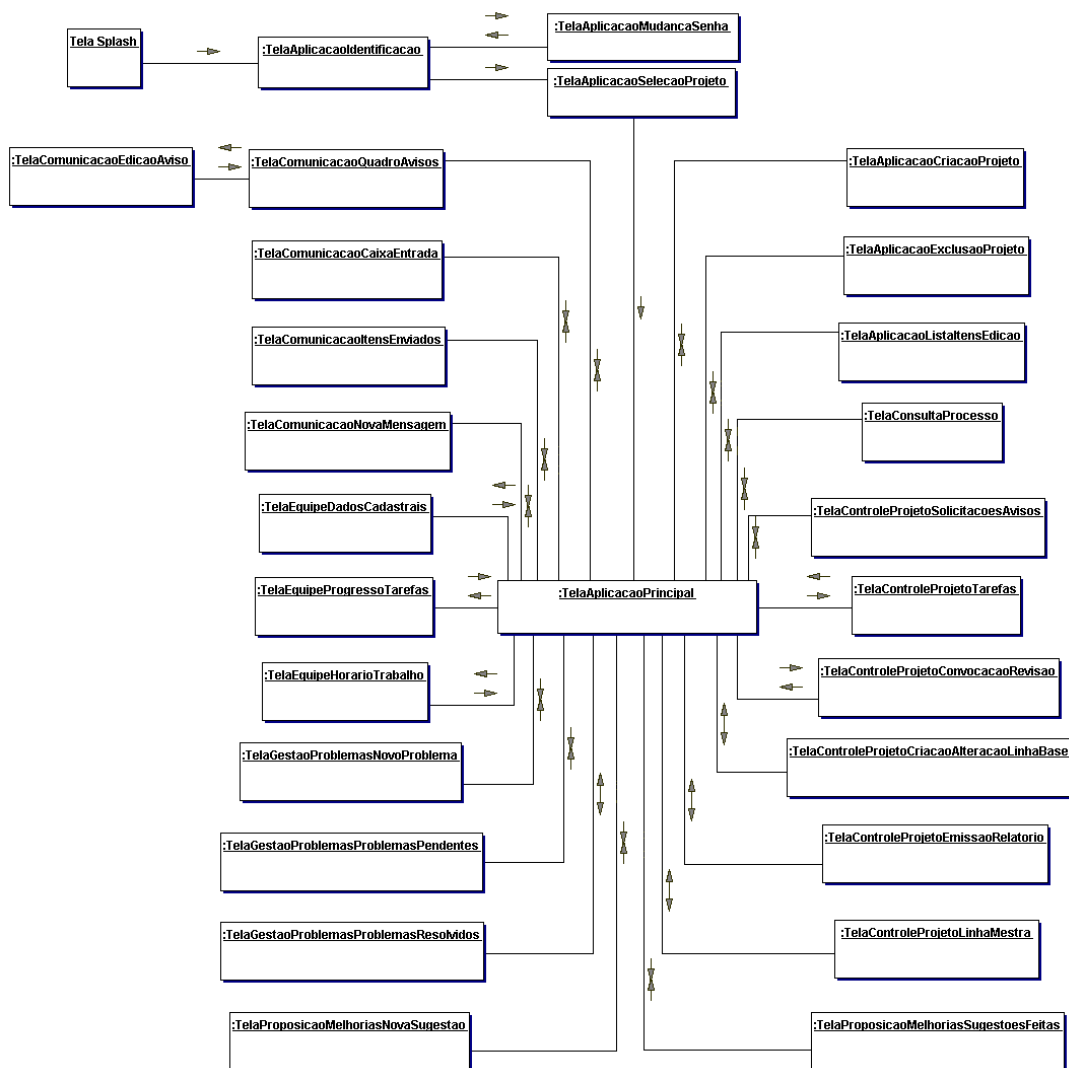


Figura 23 - Estrutura Dinâmica do Praxis Mentor

4.2.1.1.3 Funções do produto

A lista de funções do produto apresenta os casos de uso implementados através deste desenho. No caso do Praxis Mentor esta lista foi herdada da Especificação dos Requisitos.

Número de ordem	Caso de uso	Descrição
1	Consulta ao Processo	Processamento de exibição de todo o conteúdo do processo de maneira hierarquizada.
2	Comunicação Interna da Equipe	Processamento de inclusão, exclusão e alteração de mensagens para outros membros da equipe do projeto. Controle de visibilidade dessas mensagens de acordo com o papel do usuário.
3	Proposição de Melhorias do Processo	Processamento de inclusão, exclusão e alteração de melhorias para o processo.
4	Gestão de Problemas	Processamento de inclusão, exclusão e alteração de mensagens de tratamento de problemas para o gerente do projeto. Controle de visibilidade dessas mensagens de acordo com o papel do usuário.
5	Gestão do Planejamento de Projetos	Processamento de inclusão, exclusão e alteração de projetos e de membros da equipe dos projetos.
6	Controle de Projetos	Controle do fluxo das atividades relacionadas ao controle de projetos como: Replanejamento, Emissão de Relatórios de Acompanhamento e Finais, Convocação de Revisões e Criação/Alteração de Linha de Base.

Tabela 36 - Funções do Praxis Mentor

4.2.1.1.4 Convenções gerais utilizadas

É interessante a adoção de convenções de desenho que possam ser aplicadas a toda a interface com o usuário. Constituem exemplos de tais convenções as regras para exibição de mensagens, os mecanismos de navegação e as formas de diferenciação entre campos obrigatórios e opcionais. As convenções adotadas para o Praxis Mentor estão listadas na Tabela 37.

Número de ordem	Tipo de convenção	Descrição da convenção
1	Exibição de mensagens de erro	As mensagens de erro serão exibidas em caixas de mensagem.
2	Exibição de mensagens de sucesso	As mensagens de sucesso serão exibidas na barra de status.
3	Mecanismo de navegação	A navegação entre campos de um formulário é feita através da tecla TAB. A navegação se dará somente entre os campos que estiverem habilitados no momento.
4	Gráfica	Os campos que não estiverem disponíveis estarão na cor cinza e desabilitados, não sendo possível navegar até eles.

Tabela 37 - Convenções gerais do Praxis Mentor

4.2.1.1.5 Componentes das interfaces de usuário

Devem ser apresentados todos os componentes utilizados nas interfaces de usuário. A lista aqui apresentada deve ser consistente com as classes e os objetos representativos destes componentes apresentados na seção 4.2.1.2 (Desenho interno).

A descrição dos componentes de interface é semelhante à descrição dos requisitos de interface de usuário apresentados no capítulo anterior. A diferença é que a informação aqui apresentada corresponde ao que realmente será implementado.

A descrição detalhada dos componentes de algumas interfaces de usuário do Praxis Mentor será mostrada a seguir. As interfaces aqui apresentadas são as mesmas especificadas no capítulo anterior. A idéia é mostrar as diferenças entre a especificação de interface feita na ERSw e a feita no documento de desenho, DDSw.

Interface de usuário Tela Principal

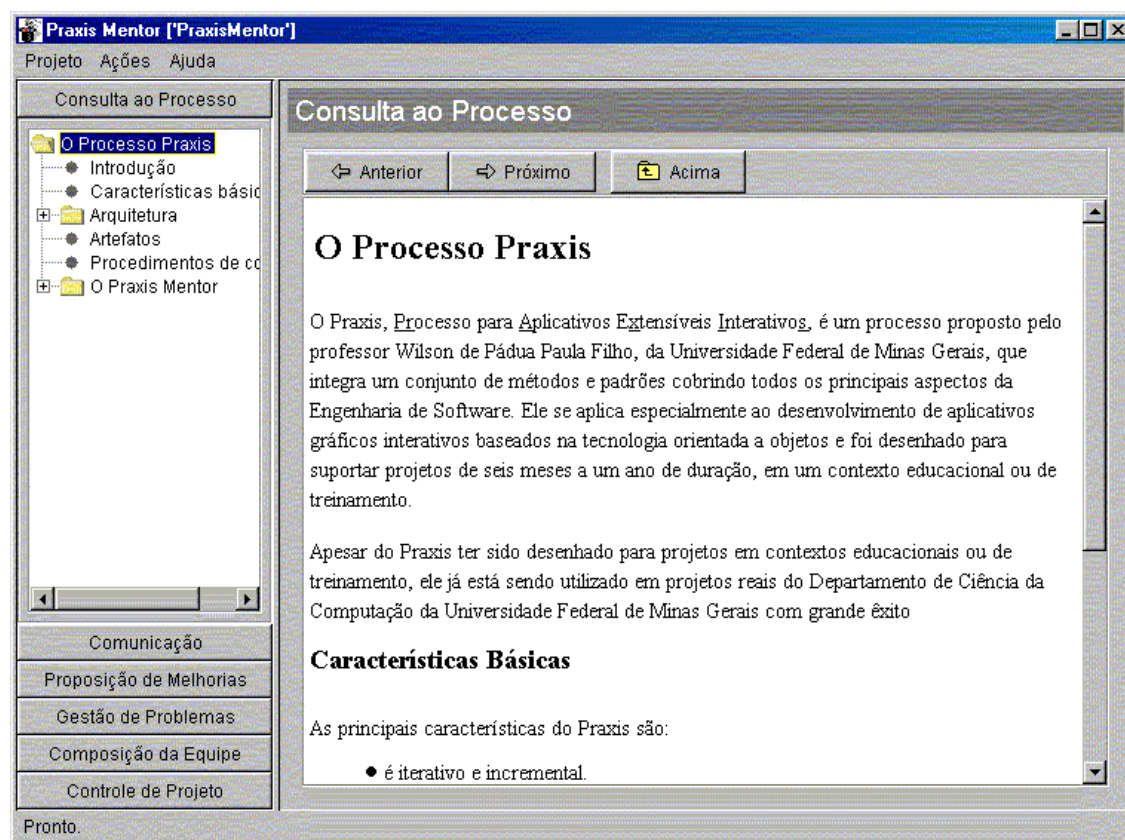
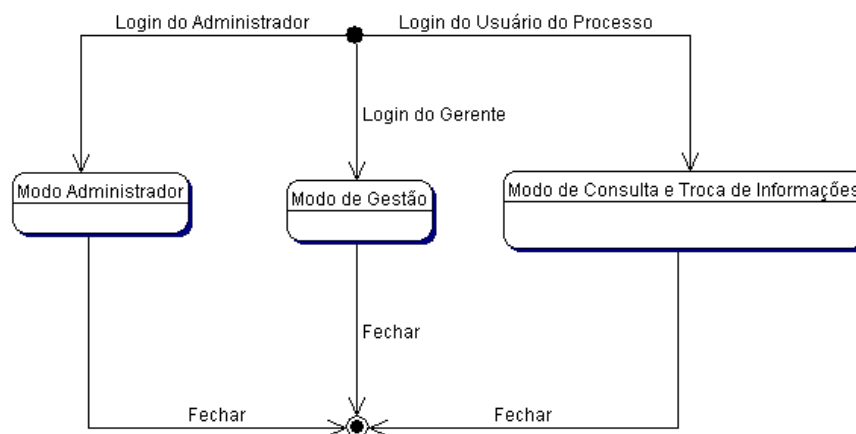


Diagrama de estados



Relacionamentos com outras interfaces

A partir dessa tela todas as outras podem ser acionadas, com exceção das telas *Tela Identificação* e *Tela Mudança de Senha*. Ela contém um componente de barra de atalho estilo navegador, que permite acesso hierárquico às funções do sistema.

Campos

Não aplicável.

Lista de comandos

Número	Grupo	Item	Ação
1	Projeto	Novo	Habilita e exibe a Tela Propriedades de Projeto / Novo Projeto.
		Abrir	Habilita e exibe a Tela Seleção de Projetos.
		Fechar	Fecha o projeto.
		Excluir	Exclui um projeto a ser escolhido.
		Exportar	Exporta os dados do projeto para formato HTML.
		Imprimir	Imprime dados do projeto.
		Propriedades	Habilita e exibe a Tela Propriedades de Projeto / Novo Projeto.
		Sair	Sai do Praxis Mentor.
2	Ações	Consulta ao Processo	Habilita e exibe a interface com os tópicos de Consulta ao Processo.
		Comunicação	Habilita e exibe a interface com os tópicos de Comunicação.
		Comunicação - Quadro de Avisos	Habilita e exibe a Tela Comunicação - Quadro de Avisos.
		Comunicação - Caixa de Entrada	Habilita e exibe a Tela Comunicação – Caixa de Entrada.
		Comunicação - Itens Enviados	Habilita e exibe a Tela Comunicação – Itens Enviados.
		Comunicação - Nova Mensagem	Habilita e exibe a Tela Comunicação – Nova Mensagem.
		Proposição de Melhorias	Habilita e exibe a interface com os tópicos de Proposição de Melhorias.
		Proposição de Melhorias - Sugestões Feitas	Habilita e exibe a Tela Proposição de Melhorias – Sugestões Feitas.
		Proposição de Melhorias - Nova Sugestão	Habilita e exibe a Tela Proposição de Melhorias – Nova Sugestão.
		Gestão de Problemas	Habilita e exibe a interface com os tópicos de Gestão de Problemas.
		Gestão de Problemas - Problemas Pendentes	Habilita e exibe a Tela Gestão de Problemas – Problemas Pendentes.
		Gestão de Problemas - Problemas Resolvidos	Habilita e exibe a Tela Gestão de Problemas – Problemas Resolvidos.
		Gestão de Problemas - Novo Problema	Habilita e exibe a Tela Gestão de Problemas – Novo Problema.

		Composição da Equipe	Habilita e exibe a interface com os tópicos de Composição da Equipe.
		Composição da Equipe - Dados Cadastrais	Habilita e exibe a Tela Comunicação da Equipe – Dados Cadastrais.
		Composição da Equipe - Horário de Trabalho	Habilita e exibe a Tela Comunicação da Equipe – Horário de Trabalho.
		Composição da Equipe - Progresso de Tarefas	Habilita e exibe a Tela Comunicação da Equipe – Progresso de Tarefas.
		Controle de Projeto	Habilita e exibe a interface com os tópicos de Controle de Projetos.
		Controle de Projeto - Linha Mestra	Habilita e exibe a Tela Controle de Projeto – Linha Mestra.
		Controle de Projeto – Tarefas	Habilita e exibe a Tela Controle de Projeto – Atribuição de Tarefas.
		Controle de Projeto - Solicitações e Avisos	Habilita e exibe a Tela Controle de Projeto – Solicitações e Avisos.
		Controle de Projeto - Solicitar Revisão	Habilita e exibe a Tela Controle de Projeto – Solicitar Revisão.
		Controle de Projeto - Solicitar Criação ou Alteração de Linha de Base	Habilita e exibe a Tela Controle de Projeto – Solicitar Criação ou Alteração de Linha de Base.
		Controle de Projeto - Emitir Relatório	Habilita e exibe a Tela Controle de Projeto – Emitir Relatório.
		Anterior	Posiciona no tópico anterior de consulta ao processo.
		Próximo	Posiciona no próximo tópico de consulta ao processo.
		Acima	Posiciona no tópico de consulta acima em relação ao tópico corrente.
		Verificar Itens em Edição	Exibe a lista de mensagens iniciadas e não enviadas pelo usuário.
3	Ajuda	Sobre	Habilita e exibe a interface Sobre o Praxis Mentor.

Matriz de habilitação

Número	Grupo	Item	Estado		
			Modo Administrador	Modo de Gestão	Modo de Consulta e Troca de Informações
1	Projeto	Novo	Habilitado	Desabilitado	Desabilitado
		Abrir	Habilitado	Habilitado	Habilitado
		Fechar	Habilitado	Habilitado	Habilitado
		Excluir	Habilitado	Desabilitado	Desabilitado
		Exportar	Habilitado	Habilitado	Habilitado
		Imprimir	Habilitado	Habilitado	Habilitado
		Propriedades	Habilitado	Desabilitado	Desabilitado
		Sair	Habilitado	Habilitado	Habilitado

2	Ações	Consulta ao Processo	Habilitado	Habilitado	Habilitado
		Comunicação	Habilitado	Habilitado	Habilitado
		Comunicação – Quadro de Avisos	Habilitado	Habilitado	Habilitado
		Comunicação -Caixa de Entrada	Habilitado	Habilitado	Habilitado
		Comunicação -Itens Enviados	Habilitado	Habilitado	Habilitado
		Comunicação -Nova Mensagem	Habilitado	Habilitado	Habilitado
		Proposição de Melhorias	Habilitado	Habilitado	Habilitado
		Proposição de Melhorias - Sugestões Feitas	Habilitado	Habilitado	Habilitado
		Proposição de Melhorias - Nova Sugestão	Habilitado	Habilitado	Habilitado
		Gestão de Problemas	Habilitado	Habilitado	Habilitado
		Gestão de Problemas - Problemas Pendentes	Habilitado	Habilitado	Habilitado
		Gestão de Problemas - Problemas Resolvidos	Habilitado	Habilitado	Habilitado
		Gestão de Problemas - Novo Problema	Habilitado	Habilitado	Habilitado
		Composição da Equipe	Habilitado	Habilitado	Habilitado
		Composição da Equipe - Dados Cadastrais	Habilitado	Habilitado	Desabilitado
		Composição da Equipe - Horário de Trabalho	Habilitado	Habilitado	Desabilitado
		Composição da Equipe – Progresso de Tarefas	Habilitado	Habilitado	Desabilitado
		Controle de Projeto	Habilitado	Habilitado	Habilitado
		Controle de Projeto - Linha Mestra	Habilitado	Habilitado	Desabilitado
		Controle de Projeto - Tarefas	Habilitado	Habilitado	Desabilitado
		Controle de Projeto - Solicitações e Avisos	Habilitado	Habilitado	Habilitado
Controle de Projeto - Solicitar Revisão	Habilitado	Habilitado	Habilitado		

		Controle de Projeto - Solicitar Criação ou Alteração de Linha de Base	Habilitado	Habilitado	Desabilitado
		Controle de Projeto - Emitir Relatório	Habilitado	Habilitado	Desabilitado
		Anterior	Habilitado	Habilitado	Habilitado
		Próximo	Habilitado	Habilitado	Habilitado
		Acima	Habilitado	Habilitado	Habilitado
		Verificar Itens em Edição	Habilitado	Habilitado	Habilitado
3	Ajuda	Sobre	Habilitado	Habilitado	Habilitado

Interface de usuário Tela Propriedades de Projeto / Novo Projeto

Propriedades do projeto [Criação de novo projeto]

Sigla: PMTI Nome: Integrador Praxis Mentor - Together

Descrição: Integra as ferramentas Praxis Mentor e Together

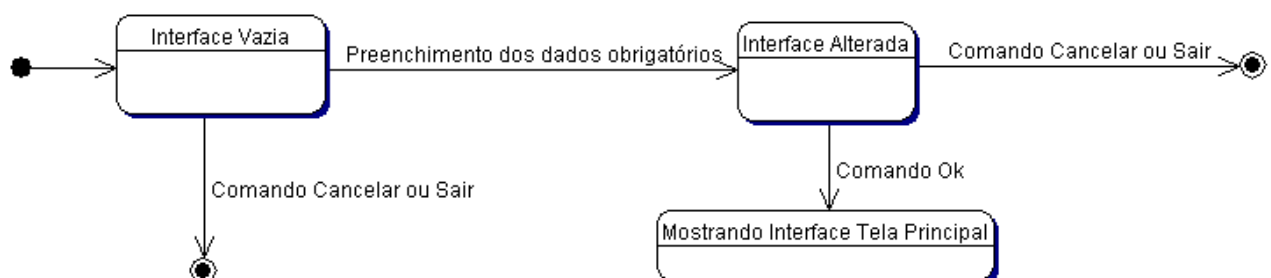
Início: 01/06/2001 Fim: 31/10/2001

Gerente: Júnia Gaudereto Carvalho (junia)

Pasta do projeto: c:\projetos\pmti

OK Cancelar

Diagrama de estados



Relacionamentos com outras interfaces

Esta interface é ativada a partir da *Tela Principal* ou da *Tela Seleção de Projetos*.
O botão “OK” aciona a interface *Tela Principal* com o projeto aberto.
O botão “Cancelar” retorna à interface que chamou essa tela.

Campos

Número	Nome	Valores válidos	Formato	Tipo	Restrições
1	Sigla	Não vazio.	Até 15 caracteres.	Texto	Obrigatório / alterável.
2	Nome	Não vazio.	Até 50 caracteres.	Texto	Obrigatório / alterável.
3	Descrição	-	Até 255 caracteres.	Texto	Opcional / alterável.
4	Início	-	dd/mm/aaaa	Data	Obrigatório / alterável.
5	Fim	Maior que início.	dd/mm/aaaa	Data	Obrigatório / alterável.
6	Gerente (Nome + Usuário)	Não vazio.	Até 65 caracteres.	Texto	Obrigatório / alterável.
7	Pasta do projeto	Não vazio.	Até 255 caracteres.	Texto	Obrigatório / alterável.

Lista de comandos

Número	Nome	Estilo	Ação
1	OK	Botão Java	Fecha a interface/Cria novo projeto.
2	Cancelar	Botão Java	Fecha a interface.

Matriz de habilitação

Número	Comando	Estado		
		Interface Vazia	Interface Alterada	Mostrando Interface Tela Principal
1	OK	Desabilitado	Habilitado	Desabilitado
2	Cancelar	Habilitado	Habilitado	Desabilitado

Interface de usuário Tela Comunicação – Caixa de Entrada

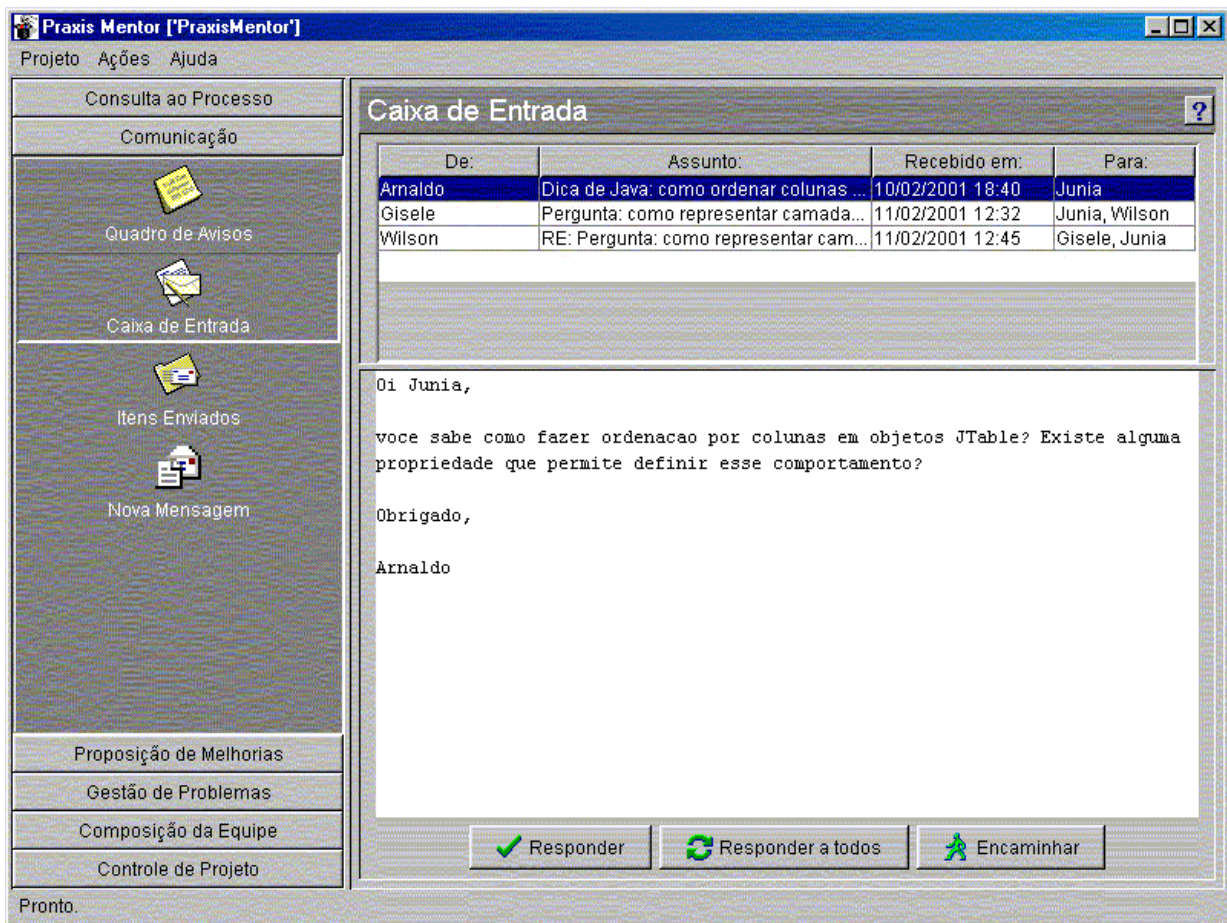
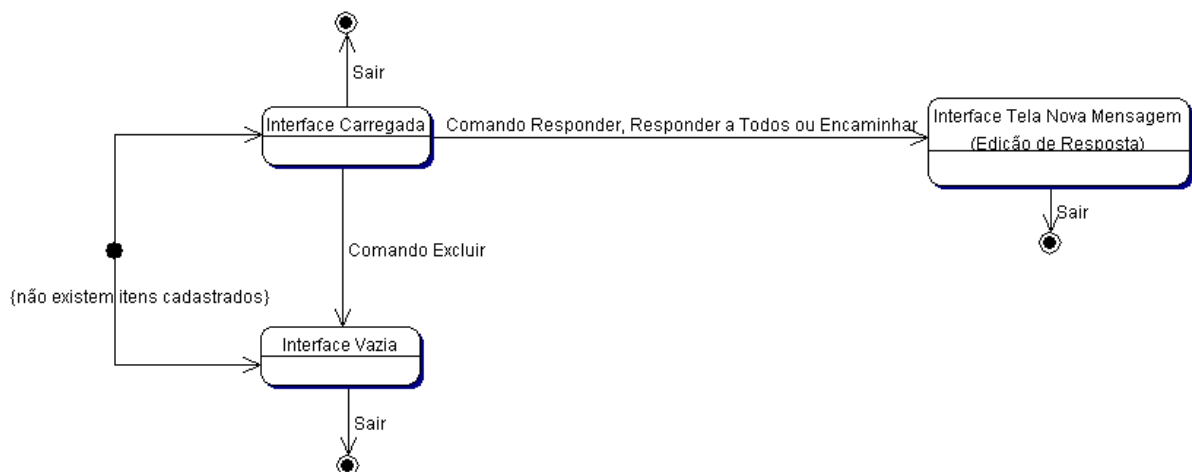



Diagrama de estados




Relacionamentos com outras interfaces

Esta interface é ativada a partir da *Tela Principal*.
Os botões Responder, Responder a Todos e Encaminhar acionam a *Tela Comunicação - Nova Mensagem*.
O botão  aciona a interface *Tela Consulta ao Processo*.


Campos

Número	Nome	Valores válidos	Formato	Tipo	Restrições
1	De:	Não vazio.	Até 50 caracteres	Texto	Preenchido pelo <u>Praxis Mentor</u> / não alterável.
2	Assunto:	-	Até 50 caracteres	Texto	Preenchido pelo <u>Praxis Mentor</u> / não alterável.
3	Recebido em:	Não vazio.	dd/mm/aaaa hh:mm	Data	Preenchido pelo <u>Praxis Mentor</u> / não alterável.
4	Para:	Não vazio	Até 255 caracteres	Texto	Preenchido pelo <u>Praxis Mentor</u> / não alterável.
5	Corpo da Mensagem	-	Até 255 caracteres.	Texto	Preenchido pelo <u>Praxis Mentor</u> / não alterável.

Lista de comandos

Número	Nome	Estilo	Ação
1	Responder	Botão Java	Aciona a Tela Nova Mensagem com o campo “Para:” preenchido com o valor do campo “De:” dessa tela.
2	Responder a todos	Botão Java	Aciona a Tela Nova Mensagem com o campo “Para:” preenchido com o valor dos campos “De:” e “Para” dessa tela.
3	Encaminhar	Botão Java	Aciona a Tela Nova Mensagem com o campo “Para:” em branco.
4	Excluir	Tecla delete	Remove a mensagem selecionada.
5		Botão Java	Aciona o tópico de Caixa de Entrada da Tela Consulta ao Processo.

Matriz de habilitação

Número	Comando	Estado		
		Interface Vazia	Interface Carregada	Interface Tela Nova Mensagem
1	Responder	Desabilitado	Habilitado	-
2	Responder a todos	Desabilitado	Habilitado	-
3	Encaminhar	Desabilitado	Habilitado	-
4	Excluir	Desabilitado	Requer confirmação	-
5		Habilitado	Habilitado	-

Observações

Os outros comandos são de acordo com a descrição na interface *Tela Principal*.

4.2.1.1.6 Funções do produto

No desenho do software, deve ser feita uma descrição detalhada das funções do produto, apresentando os casos de uso em termos das interfaces definitivas. Estes casos de uso podem ser herdados da Especificação dos Requisitos do Software (ERSw), mas devem ser descritos em um nível de detalhe muito maior. Segundo Paula [Paula01], os seguintes elementos devem ser descritos:

- mecanismos de acesso;
- fluxo principal, subfluxos e fluxos alternativos;
- diagramas de estado;
- condições de exceção;
- mensagens.

A descrição detalhada de alguns casos de uso do Praxis Mentor é apresentada a seguir. Novamente foram escolhidos os mesmos casos de uso apresentados no capítulo anterior, a fim de destacar as diferenças entre a descrição feita na ERSw e a descrição de desenho.

Caso de uso Criação de Projetos

Mecanismos de acesso

O acesso ao caso de uso Criação de Projetos é sempre feito através da Tela Propriedades de Projeto / Novo Projeto. Esta é mostrada e ativada quando o item Novo do cardápio Projeto ou o botão “Novo” da Tela Principal é acionado. Inicialmente, a interface está no estado "Interface Vazia".

Fluxo principal

O Praxis Mentor instancia e exibe a Tela Propriedades de Projeto / Novo Projeto.

O Usuário do Processo registra os dados do projeto.

O Praxis Mentor coloca a interface no estado “Interface Alterada”.

O Usuário do Processo pode acionar um dos botões: “OK” ou “Cancelar”.

Se o Usuário do Processo acionar o botão “OK”:

- O Praxis Mentor verifica se todos os campos estão preenchidos.
- Se não estiverem, o Praxis Mentor emite a mensagem MCP-01 e aborta a operação.
- Se estiverem:
 - O Praxis Mentor inclui na tabela Projeto do banco de dados os seguintes dados:
 - Campo codigoProjeto: código gerado automaticamente para o projeto;
 - Campo descricao: campo Descrição da interface;
 - Campo fim: campo Término da interface;
 - Campo inicio: campo Início da interface;
 - Campo nome: campo Nome da interface;
 - Campo sigla: campo Sigla da interface;
 - O Praxis Mentor verifica se o usuário informado como gerente do projeto já está cadastrado no banco de dados, pesquisando pelo código do mesmo na tabela Usuario.
 - Se o usuário não estiver cadastrado, o Praxis Mentor gera um código automático para o usuário e o insere na tabela Usuario.
 - O Praxis Mentor acrescenta o usuário à Equipe do projeto inserindo os seguintes dados na tabela ProjetoUsuario:
 - Campo codigoProjeto: código do projeto;
 - Campo codigoUsuario: código do usuário;
 - Campo papel: “1” (Gerente de Projeto);
 - O Praxis Mentor verifica se a pasta informada para o projeto já existe.
 - Se a pasta existir, Praxis Mentor copia modelos dos artefatos para a mesma e modifica os nomes dos artefatos de acordo com nome dado ao projeto.
 - Se a pasta não existir, Praxis Mentor cria a mesma e copia os modelos dos artefatos para esse diretório com os nomes modificados de acordo com o nome dado ao projeto.
 - Se ocorrer algum problema ao gravar no banco de dados, o Praxis Mentor executa a exceção ECP-01 e aborta a operação.
 - O Praxis Mentor fecha a interface atual, instancia e exibe a *Tela Principal*.

Se o Usuário do Processo acionar o botão “Cancelar”, o Praxis Mentor emite a mensagem MCP-02 solicitando a confirmação do cancelamento.

- Se o Usuário do Processo confirmar, o Praxis Mentor aborta a operação e fecha a interface.
- Caso contrário, o Praxis Mentor ignora o cancelamento.

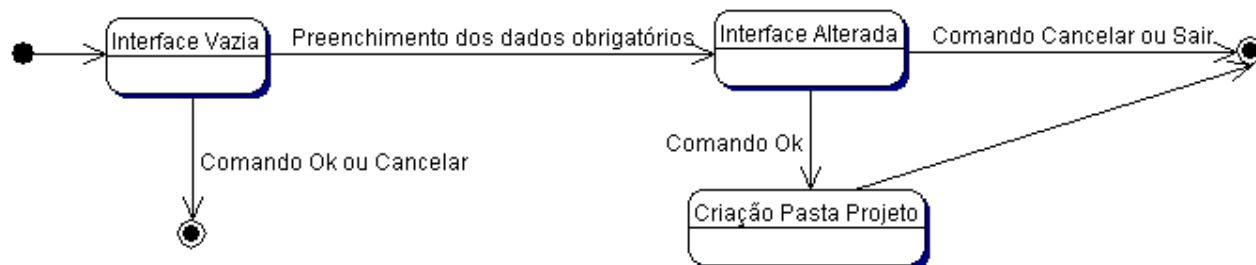
Subfluxos

Não aplicável.

Fluxos alternativos

Não aplicável.

Diagrama de estado



Condições de exceção

Número de ordem	Identificação da exceção	Descrição da exceção	Ação
1	ECP-01	Foi solicitada a criação de um projeto, mas ocorreu um erro no momento de gravar no banco de dados.	Emitir mensagem MCP-03.

Mensagens

Número de ordem	Identificação da mensagem	Classe	Texto da mensagem
1	MCP-01	Informativa	Todos os campos são obrigatórios e devem ser preenchidos para que o projeto seja criado.
2	MCP -02	Requer confirmação	Confirma o cancelamento da criação de novo projeto?
3	MCP -03	Informativa	Problema ao acessar o banco de dados. Entre em contato com o suporte, por favor.

Caso de uso Controle de Mensagens Recebidas

Mecanismos de acesso

O acesso ao caso de uso Controle de Mensagens Recebidas é sempre feito através da *Tela Comunicação – Caixa de Entrada*. Esta é mostrada e ativada quando o item Caixa de Entrada do cardápio Ações\Comunicação ou do cardápio lateral da *Tela Principal* é acionado. Inicialmente, as mensagens que foram recebidas anteriormente são exibidas e a interface fica no estado “Interface Carregada”; se não existir nenhuma mensagem recebida, a interface fica no estado “Interface Vazia”.

Fluxo principal

O Praxis Mentor instancia e exibe a *Tela Comunicação – Caixa de Entrada*.
O Praxis Mentor exibe a lista de mensagens existentes.
Se houver mensagens:
 O Praxis Mentor posiciona na primeira e exibe seu conteúdo.
 O Usuário do Processo pode selecionar uma mensagem e acionar o botão “Responder ao Remetente”, “Responder a Todos” ou “Encaminhar”. Além disso, pode escolher “Excluir” (tecla *delete*).

Subfluxos

Subfluxo Escolher Destinatários

O Praxis Mentor instancia e exibe a interface de usuário *Tela Destinatários*.
Se houver membros cadastrados:
 O Praxis Mentor exibe a lista dos membros cadastrados.
 O Usuário do Processo seleciona um membro na lista.
 O Usuário do Processo pode acionar os botões “Para->”, “CC->”, “OK” ou “Cancelar”.
 Se o Usuário do Processo acionar o botão “Para->”, o Praxis Mentor acrescenta o membro da equipe selecionado à lista de destinatários. Se não houver nenhum membro selecionado, nada acontece.
 Se o Usuário do Processo acionar o botão “CC->”, o Praxis Mentor acrescenta o membro da equipe selecionado à lista de destinatários adicionais. Se não houver nenhum membro selecionado, nada acontece.
 Se o Usuário do Processo acionar o botão “OK”, o Praxis Mentor acrescenta a lista de destinatários ao campo “Para” da mensagem e a lista de destinatários adicionais ao campo “CC” da mensagem.
 Se o Usuário do Processo acionar o botão “Cancelar”, o Praxis Mentor fecha a interface e ignora os membros selecionados.
Se não houver membros cadastrados:
 O Usuário do Processo só poderá acionar o botão “Cancelar”.
 Se o Usuário do Processo acionar o botão “Cancelar”, o Praxis Mentor fecha a interface.

Fluxos alternativos

Fluxo alternativo Responder ao Remetente

Precondições	<p>O <u>Usuário do Processo</u> seleciona uma mensagem.</p> <p>O <u>Usuário do Processo</u> aciona o botão “Responder ao Remetente”.</p>
Passos	<p>O <u>Praxis Mentor</u> instancia e exibe a interface de usuário <i>Tela Comunicação - Nova Mensagem</i>.</p> <p>O <u>Praxis Mentor</u> preenche o campo “Para” dessa interface com o valor do campo “De” da mensagem original e o campo “Assunto” com o valor do campo “Assunto” da mensagem original antecedido pelo prefixo “RE:” .</p> <p>Se o <u>Usuário do Processo</u> acionar os botões “Para...” ou “CC...”, o <u>Praxis Mentor</u> executa o subfluxo Escolher Destinatários.</p> <p>O <u>Usuário do Processo</u> preenche o texto da mensagem.</p> <p>O <u>Usuário do Processo</u> seleciona a operação a ser realizada: Enviar ou Limpar.</p> <p>Se o <u>Usuário do Processo</u> escolher “Enviar”:</p> <p style="padding-left: 20px;">O <u>Praxis Mentor</u> verifica se o campo “Para:” está preenchido.</p> <p style="padding-left: 20px;">Se não estiver, o <u>Praxis Mentor</u> emite a mensagem MMR-01 e aborta a operação.</p> <p style="padding-left: 20px;">Se estiver:</p> <p style="padding-left: 40px;">O <u>Praxis Mentor</u> verifica se todos os destinatários da mensagem estão cadastrados no sistema.</p> <p style="padding-left: 40px;">Se algum destinatário não estiver cadastrado, o <u>Praxis Mentor</u> executa a exceção EMR-01 e aborta o envio.</p> <p style="padding-left: 40px;">O <u>Praxis Mentor</u> inclui na tabela Mensagem do banco de dados os seguintes dados:</p> <ul style="list-style-type: none"> - Campo codigoMensagem: código gerado automaticamente para a mensagem; - Campo codigoProjeto: código do projeto aberto; - Campo corpo: texto da mensagem; - Campo enviadoEm: data atual do sistema; - Campo enviado: sim; - Campo assunto: valor do campo “Assunto” da interface, antecedido por “RE”; - Campo recebidoEm : data atual do sistema; - Campo remetente: código referente ao <u>Usuário do Processo</u> na tabela Usuario; <p style="padding-left: 40px;">O <u>Praxis Mentor</u> inclui na tabela MensagemSimples do banco de dados os seguintes dados:</p> <ul style="list-style-type: none"> - Campo codigoMensagem: mesmo código gerado automaticamente para o aviso e armazenado no campo codigoMensagem da tabela Mensagem; - Campo codigoProjeto: código do projeto aberto; <p style="padding-left: 40px;">Para cada destinatário da mensagem:</p> <p style="padding-left: 60px;">O <u>Praxis Mentor</u> inclui na tabela MensagemDestinatario do banco de dados os seguintes dados:</p> <ul style="list-style-type: none"> - Campo codigoMensagem: mesmo código gerado automaticamente para o aviso e armazenado no campo codigoMensagem da tabela Mensagem; - Campo codigoProjeto: código do projeto aberto; - Campo codigoUsuario: código referente ao Usuário do Processo na tabela Usuario; <p style="padding-left: 40px;">Para cada destinatário de cópia da mensagem, se existirem:</p> <p style="padding-left: 60px;">O <u>Praxis Mentor</u> inclui na tabela MensagemDestinatarioCopia do banco de dados os seguintes dados:</p> <ul style="list-style-type: none"> - Campo codigoMensagem: mesmo código gerado automaticamente para o aviso e armazenado no campo codigoMensagem da tabela Mensagem;

	<ul style="list-style-type: none"> - Campo codigoProjeto: código do projeto aberto; - Campo codigoUsuario: código referente ao Usuário do Processo na tabela Usuario; <p>Se ocorrer algum problema ao gravar no banco de dados, o <u>Praxis Mentor</u> executa a exceção EMR-02 e aborta a operação.</p> <p>O <u>Praxis Mentor</u> acrescenta a nova mensagem na lista de Itens Enviados para que ela possa aparecer na <i>Tela Comunicação – Itens Enviados</i>.</p> <p>O <u>Praxis Mentor</u> coloca a interface no estado "Atualizada".</p> <p>Se o <u>Usuário do Processo</u> escolher “Limpar”: O <u>Praxis Mentor</u> limpa todos os dados da mensagem.</p>
--	---

Fluxo alternativo Responder a Todos

Precondições	<p>O <u>Usuário do Processo</u> seleciona uma mensagem.</p> <p>O <u>Usuário do Processo</u> aciona o botão “Responder a Todos”.</p>
Passos	Idem a Responder ao Remetente, com a diferença de que o campo “Para” é preenchido com todos os destinatários da mensagem original (incluindo campo “CC”).

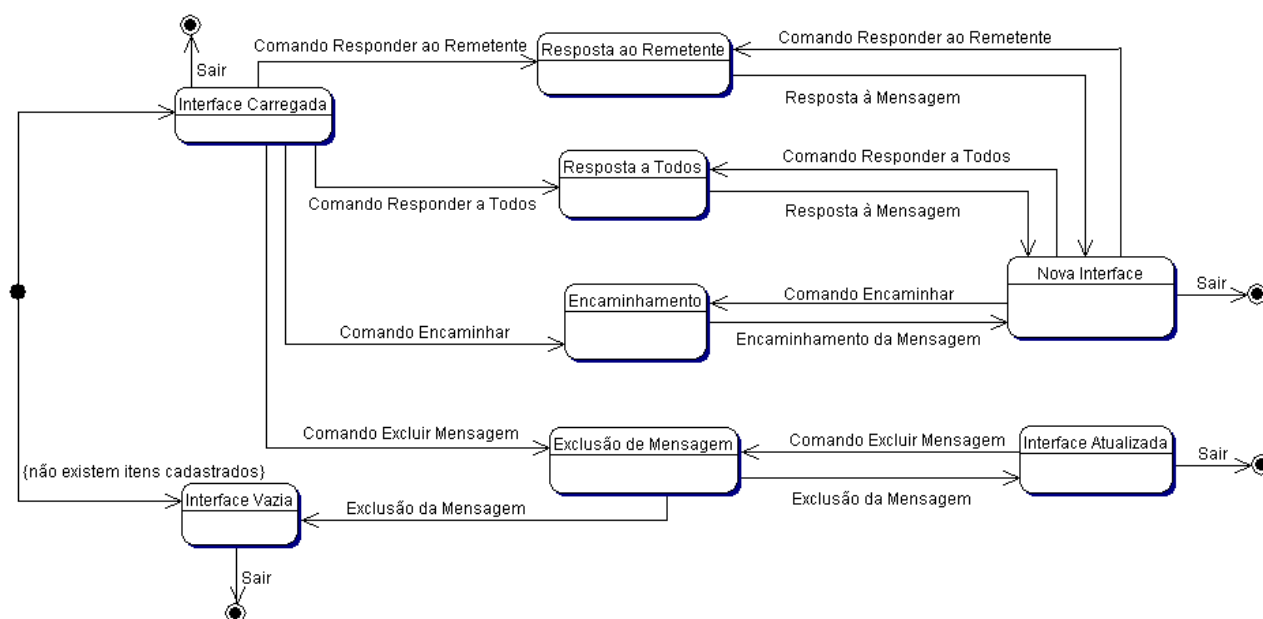
Fluxo alternativo Encaminhar

Precondições	<p>O <u>Usuário do Processo</u> seleciona uma mensagem.</p> <p>O <u>Usuário do Processo</u> aciona o botão “Encaminhar”.</p>
Passos	Idem a Responder ao Remetente, com a diferença de que o campo “Para” não é preenchido e o campo “Assunto” é preenchido com o valor do campo “Assunto” da mensagem original antecedido pela sigla “EN:”

Fluxo alternativo Excluir Mensagem

Precondições	<p>O <u>Usuário do Processo</u> seleciona uma mensagem.</p> <p>O <u>Usuário do Processo</u> escolhe “Excluir” (tecla <i>delete</i>).</p>
Passos	<p>O <u>Praxis Mentor</u> emite a mensagem MMR-02, solicitando confirmação da exclusão da mensagem selecionada.</p> <p>Se a exclusão for confirmada:</p> <p>O <u>Praxis Mentor</u> altera o campo visível para falso na entrada formada pelo código da mensagem, do projeto e do usuário na tabela MensagemDestinatario ou MensagemDestinatarioCopia do banco de dados do <u>Praxis Mentor</u>.</p> <p>Se o <u>Usuário do Processo</u> que estiver excluindo a mensagem for o próprio remetente, o <u>Praxis Mentor</u> altera o campo remetenteVisível para falso na entrada formada pelo código da mensagem e do projeto.</p> <p>Caso não exista mais nenhum destinatário para a mensagem com o valor do campo visível igual a verdadeiro e o campo remetenteVisível também estiver igual a falso, o <u>Praxis Mentor</u> exclui das tabelas MensagemDestinatario, MensagemDestinatarioCopia, Mensagem e MensagemSimples do banco de dados a entrada formada pelo código da mensagem e do projeto.</p> <p>O <u>Praxis Mentor</u> atualiza a interface para refletir a exclusão da respectiva mensagem.</p> <p>O <u>Praxis Mentor</u> coloca a interface no estado "Interface Vazia" se não restar nenhuma mensagem e no estado "Interface Atualizada", caso contrário.</p>

Diagrama de estado



Condições de exceção

Número de ordem	Identificação da exceção	Descrição da exceção	Ação
1	EMR-01	Foi informado algum usuário inexistente como destinatário da mensagem.	Emitir mensagem MMR-03.
2	EMR-02	Foi solicitado o envio de uma mensagem, mas ocorreu um erro no momento de gravar no banco de dados.	Emitir mensagem MMR-04.

Mensagens

Número de ordem	Identificação da mensagem	Classe	Texto da mensagem
1	MMR-01	Informativa	O campo “Para” é obrigatório e deve ser preenchido para que a mensagem seja enviada.
2	MMR-02	Requer confirmação	Confirma a exclusão desta mensagem?
3	MMR-03	Informativa	Um ou mais destinatários dessa mensagem não existem. Por favor, escolha os usuários cadastrados através dos comandos “Para...” ou “CC...”.
4	MMR-04	Informativa	Problema ao acessar o banco de dados. Entre em contato com o suporte, por favor.

4.2.1.2 Desenho interno

Segundo Paula [Paula01], a arquitetura de um produto expressa a sua divisão em subsistemas e outros componentes de nível mais baixo, as interfaces entre os componentes e as interações através das quais eles realizam as funções do produto, atendendo aos requisitos não funcionais.

As principais decisões de arquitetura do Praxis Mentor são apresentadas nas seções que se seguem.

4.2.1.2.1 Estratégias de arquitetura

A Tabela 38 apresenta as estratégias de arquitetura referentes à interface de usuário, banco de dados e ambiente de desenvolvimento.

A interface de usuário seguirá o padrão Windows 9x para facilitar o aprendizado da utilização do produto por parte dos usuários.
O sistema de gerência de banco de dados utilizado será o Microsoft Access, devido à sua simplicidade de uso, difusão no mercado e compatibilidade com o paradigma relacional.
O ambiente de desenvolvimento será o Java, pela difusão no mercado e excelente suporte para a tecnologia orientada por objetos.

Tabela 38 - Estratégias de arquitetura do Praxis Mentor

As subseções a seguir apresentam estratégias de arquitetura referentes aos componentes do Praxis Mentor e às decisões de implementação consideradas relevantes para o desenvolvimento do projeto.

Camada de persistência

No desenvolvimento do Praxis Mentor o acesso ao banco de dados é um ponto fundamental. A ferramenta contém mais de trinta telas e praticamente todas elas tratam dados que são armazenados entre execuções consecutivas, enquanto poucas são as tarefas de processamento na ferramenta. Com isso, há a necessidade de se prover mecanismos eficientes e de fácil acesso para prover persistência aos objetos tratados.

Optou-se então pela implementação de uma camada de persistência para a aplicação. Segundo Ambler [Ambler99], usando camadas de persistência, é possível esconder detalhes sobre o acesso ao banco de dados em classes especializadas, tornando mais simples e limpo o código de classes de tratamento de interação com o usuário. Em casos mais extremos, é possível até mesmo tornar a aplicação praticamente independente de alterações no banco de dados, o que representa, é claro, um custo em performance. Algumas das idéias usadas na implementação da camada de persistência do Praxis Mentor foram extraídas de [Travis01].

Devido a questões como tamanho, nível de robustez requerido e complexidade da ferramenta, decidiu-se por implementar uma camada de persistência baseada em um controlador simples, algumas classes para leitura e gravação de associações de objetos e algum código para manipulação de objetos a partir de conjuntos de pares “propriedade/valor”. O funcionamento básico da camada de persistência implementada para o Praxis Mentor é apresentado a seguir:

1. Para a instanciação de objetos a partir de registros do banco de dados, todas as classes de dados implementam um construtor com o parâmetro “propriedades”, que é um objeto *java.util.Hashtable*. Esse construtor é responsável por definir os valores das propriedades básicas do objeto a partir dos pares “dado/valor” contidos no parâmetro *Hashtable*;
2. Para a associação de objetos, foram desenvolvidas classes especiais que recebem conjuntos de campos a serem ligados e fazem a leitura de mais de uma tabela simultaneamente, de modo a fazer a ligação entre os objetos. Com o uso dessas classes, é possível implementar a associação de pares de objetos em uma classe controladora com pouco código e nenhuma lógica específica de manipulação de banco de dados;
3. Para gravação de objetos no banco de dados, todas as classes de dados implementam um método “obterPropriedades”, que obtém a lista de pares “dado/valor” com as propriedades básicas de cada objeto. Esse método é usado por rotinas que iteram por listas de objetos e armazenam seus dados em registros no banco de dados;
4. Para a gravação de associações de objetos, também foram desenvolvidas classes especiais que recebem pares de objetos e listas de campos nos quais identificadores desses objetos são armazenados. Com isso, é possível implementar funções para persistência de associações de diversos pares específicos de classes em uma classe controladora de forma simples, utilizando pouco código e praticamente nenhuma lógica de manipulação de banco de dados (de modo bem similar às classes para associação de objetos).

Um ponto importante no desenvolvimento da camada de persistência foi o uso de reflexão. Através desse mecanismo provido pela linguagem Java, pode-se acessar membros de classes dinamicamente, o que é extremamente útil no desenvolvimento de rotinas genéricas de leitura e gravação de objetos em banco de dados.

Através de reflexão, um código genérico pode usar uma referência a uma classe específica para:

- obter referências a métodos, construtores e atributos da classe;
- instanciar objetos da classe dinamicamente sem necessidade de declaração estática de tipo;
- acionar métodos tanto em objetos quanto na classe (métodos estáticos) se utilizando, por exemplo, de cadeias de caracteres com nomes dos métodos.

Pelas características citadas, o uso de reflexão foi fundamental na redução do volume de código necessário para a implementação das rotinas de acesso a banco de dados. Maiores informações sobre este recurso podem ser encontradas em [Eckel00].

Implementação em três camadas: fronteira, entidade e controle

Inúmeros são os benefícios obtidos com o desenvolvimento em múltiplas camadas. Primeiramente, ganha-se em flexibilidade na escolha de plataformas para interface com o usuário (Windows ou Internet, por exemplo) e armazenamento de dados (bancos de dados ou sistemas de

arquivos, por exemplo). Além disso, aumenta-se a coesão nos módulos desenvolvidos e diminui-se o acoplamento, que é feito através de um contrato muito bem estabelecido entre as camadas. Com essa separação bem definida, a manutenção de código é centralizada e facilitada. Finalmente, o código fica sensivelmente mais reutilizável, visto que as camadas tendem a oferecer serviços mais genéricos.

No caso específico do Praxis Mentor, optou-se pelo desenvolvimento em três camadas, o modelo mais comum na atualidade. Nele, a primeira camada, a de fronteira, cuida de toda a interação com o usuário, disponibilizando ações para ele, lendo, mostrando e permitindo a alteração de dados do domínio da aplicação. A segunda camada, denominada camada de controle, cuida da implementação dos casos de uso que se aplicam aos dados, incluindo questões como persistência de objetos. Finalmente, a terceira camada, denominada camada de entidade, cuida da representação correta e eficiente dos dados em memória. Através dessa representação, pode-se navegar pelos objetos, manipulando suas propriedades e obtendo referências a outros objetos associados.

A camada de fronteira controla a interação com o usuário e contém classes que usam as bibliotecas gráficas AWT e Swing, presentes nas versões atuais do Java. Essas bibliotecas disponibilizam os componentes gráficos mais comuns existentes em ambientes de janelas como o Windows. Dentre esses componentes gráficos, foram utilizados botões de comando, botões de seleção, botões de opção, caixas de texto, painéis, tabelas, listas de seleção e caixas de seleção combinada com edição, dentre outros. Maiores informações sobre as bibliotecas podem ser encontradas em [Coad+99] e [Fowler00].

A camada de controle fornece serviços para a camada de fronteira. Na verdade, esses serviços representam a realização dos casos de uso definidos pelo desenhista do sistema. Lógicas como cálculos de fatores, iterações sobre listas e até mesmo armazenamento de objetos em arquivos ou bancos de dados são implementadas nesses serviços.

Para a realização de casos de uso, a camada de controle usa os objetos e os serviços disponibilizados pela camada de entidade. Esta, que é a última camada do modelo de três camadas, contém a representação em memória dos objetos tratados pela aplicação e seus relacionamentos. Esses objetos oferecem serviços básicos diretamente ligados ao seu estado e à sua dinâmica.

Desenvolvimento de componentes e classes utilitárias especiais

De modo a evitar replicação de código e fornecer alguns serviços avançados de implementação complexa ou específica a algum contexto (como manipulação de estruturas de dados ou cadeias de caracteres), foram desenvolvidos alguns componentes e classes utilitárias auxiliares.

Em termos de interação com o usuário, foram desenvolvidos componentes para:

- acesso rápido a funções no sistema (uma barra de atalhos estilo *Microsoft Outlook*);
- visualização e edição de dados em tabelas com recursos avançados como ordenação por colunas e edição in-loco com controles visuais como caixa de seleção ou botão de seleção;

- sistema de consulta hipertexto baseado em arquivos HTML;
- listas com itens que mostram ícones e permitem seleção através de caixas de seleção (*checkboxes*).

Em termos de serviços auxiliares à implementação, foram desenvolvidas classes utilitárias para:

- tratamento de vetores e matrizes de tipos básicos, com funções para redimensionamento e concatenação de vetores/matrizes;
- tratamento de cadeias de caracteres (*strings*), com funções para uso de máscaras, substituição de padrões de caracteres ou replicação;
- manipulação de estruturas de dados avançadas disponibilizadas pelo Java, tais como *Hashtable* e *Vector*;
- manipulação de listas de registros em tabelas em banco de dados.

Utilização de padrões (*patterns*)

Segundo Grand [Grand98], padrões de software⁶ (do inglês *software patterns*) são soluções reusáveis para problemas recorrentes que acontecem durante o desenvolvimento de software. Alguns padrões merecem destaque e serão descritos rapidamente a seguir:

- **Marker Interface**: padrão que usa interfaces sem membros para indicar atributos semânticos da classe, que podem ser testados através do uso do operador *instanceof* da linguagem Java. Este padrão foi proposto inicialmente em [Grand98], onde se pode encontrar também um modelo para implementação;
- **Singleton**: padrão que assegura que apenas uma instância de uma classe existirá em todo o sistema e provê apenas um ponto de acesso a ela. Internamente isto é feito através do uso de métodos e blocos de inicialização estáticos. Este padrão foi proposto inicialmente em [Gamma+95] e um exemplo de implementação em Java pode ser encontrado em [Grand98];
- **Adapter**: padrão que usa a implementação vazia de uma interface de modo a prover a funcionalidade prometida por ela sem ter que definir qual classe é usada para implementação. Este padrão foi proposto inicialmente em [Gamma+95] e um exemplo de implementação em Java pode ser encontrado em [Grand98];
- **Iterator**: padrão que provê uma forma de acessar elementos de um objeto agregado seqüencialmente sem expor a sua representação interna. Este padrão foi proposto inicialmente em [Gamma+95] e um exemplo de implementação em Java pode ser encontrado em [Grand98];

⁶ Tendo em vista o fato de esse documento tratar única e exclusivamente do desenvolvimento de um produto de software, o Praxis Mentor, padrões de software serão tratados daqui por diante simplesmente por padrões.

- **Builder**: padrão que separa a construção de objetos complexos da sua representação, de modo a facilitar esse processo e permitir a utilização de seqüências semelhantes para a geração de objetos diferentes. Este padrão foi proposto inicialmente em [Gamma+95] e um exemplo de implementação em Java pode ser encontrado em [Grand98];
- **Lazy Instantiation**: padrão que assegura que um determinado membro ou instância de uma classe só será instanciado se necessário;
- **Composite**: padrão que permite compor objetos com listas de outros objetos, formando estruturas “todo-partes”. Este padrão permite tratar essas listas e os objetos individuais uniformemente. Ele foi proposto inicialmente em [Gamma+95] e um exemplo de implementação em Java pode ser encontrado em [Grand98];
- **Observer**: padrão que permite notificar um conjunto de objetos quanto a alterações no estado de um outro objeto. Este padrão foi proposto inicialmente em [Gamma+95] e um exemplo de implementação em Java pode ser encontrado em [Grand98].

Aplicações de padrões na arquitetura do Praxis Mentor

Durante o desenvolvimento do Praxis Mentor, inúmeros padrões foram usados. Dentre eles, os padrões *Singleton* e *Lazy Instantiation* merecem destaque especial.

O padrão *Singleton* foi usado na implementação de todas as classes controladoras. Esse tipo de classe provê serviços mas não precisa armazenar estado interno específico. Com isso, pode-se, e na verdade deve-se, ter apenas uma instância dessas classes ativa em todo o sistema, o que traz benefícios como economia de memória e facilidade de acesso, visto que as classes usuárias dos seus serviços não precisam armazenar referências para os controladores.

O padrão *Lazy Instantiation*, por sua vez, é usado em um outro ponto muito importante da ferramenta: a carga de telas. Tendo em vista o fato de que o usuário raramente acessa todas as telas durante uma execução do aplicativo, optou-se por usar a instanciação sob demanda para elas, ou seja, as telas só são carregadas quando o usuário as seleciona. Isso penaliza um pouco a primeira ativação de cada tela (momento no qual a tela é carregada) mas diminui sensivelmente o tempo de carga do aplicativo.

Além destes, podemos destacar a utilização dos seguintes padrões:

- **Adapter**: na definição de tratamento para eventos em componentes visuais, visto que muitas vezes o tratador a ser implementado contém inúmeros métodos para os quais nenhum tratamento específico precisa ser provido;
- **Iterator**: em classes que contêm múltiplas referências para objetos de outras classes (Mensagem => Usuários em papel de destinatários), esse padrão é comumente usado para facilitar a iteração sobre os itens múltiplos (destinatários, no exemplo dado).

4.2.1.2.2 Visão lógica

A visão lógica do modelo de desenho contempla os diagramas de pacotes lógicos, de classes e de interação constantes do Modelo de Desenho. Tais diagramas definem o agrupamento de classes correlatas, as classes em si e a colaboração entre elas para a realização de um caso de uso, respectivamente.

Pacotes lógicos de desenho

Segundo Paula [Paula01], os subsistemas e outros componentes de um produto podem ser representados por pacotes lógicos de desenho. Estes pacotes lógicos são grupo de classes e outros elementos de modelagem, que apresentam fortes relacionamentos entre si (alta coesão interna) e poucos relacionamentos com elementos de outros pacotes lógicos (baixo acoplamento externo).

O diagrama da Figura 24 mostra o pacote lógico principal do Praxis Mentor. Ele apresenta um tipo de organização muito utilizada que é a organização de pacotes lógicos em camadas. Esta organização é baseada na arquitetura já explicada em que uma camada pode usar os serviços das camadas inferiores, mas desconhece as que estão acima dela. O texto das notas fornece a explicação sobre a divisão.

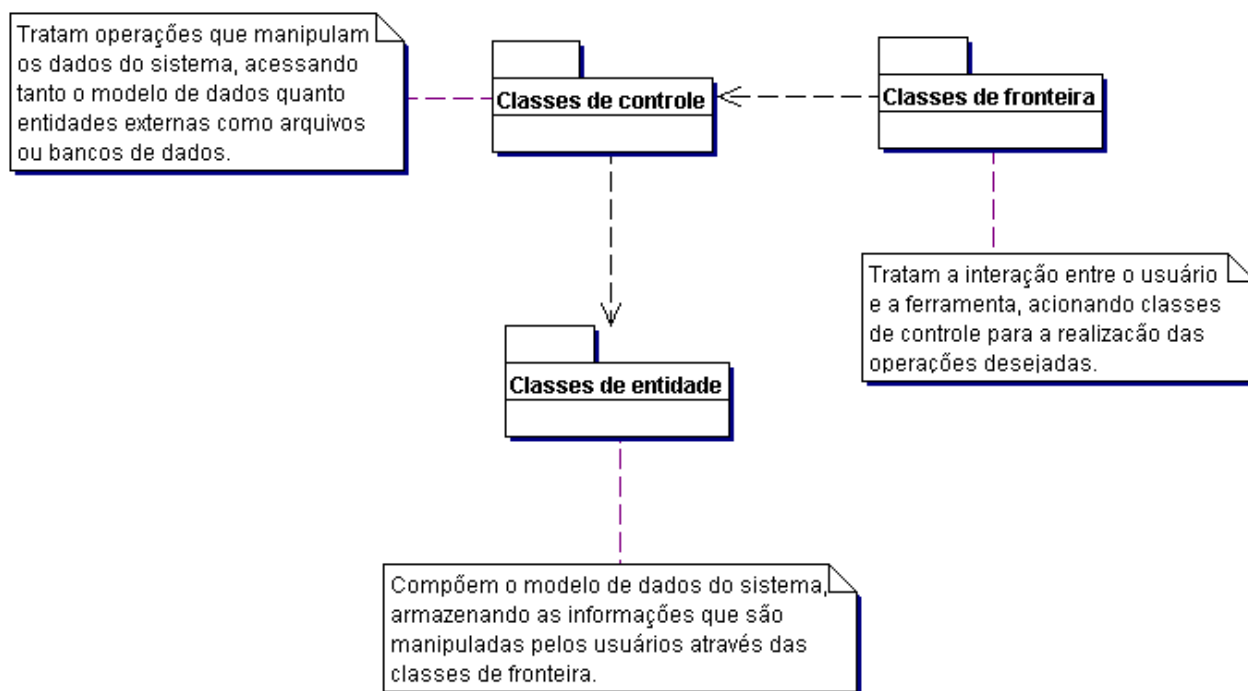


Figura 24 - Diagrama de Pacotes Lógicos do Praxis Mentor

Diagramas de classes

Segundo Paula [Paula01], os diagramas de classes de desenho devem ajudar a entender as principais decisões de desenho arquitetônico. Eles devem ser bem mais detalhados que os diagramas do Modelo de Análise e podem partir ou agrupar as classes de análise se razões tecnológicas ou requisitos não funcionais prevalecerem sobre a visão basicamente funcional do

Modelo de Análise.

Os principais diagramas de classes do Praxis Mentor são apresentados nas figuras a seguir. Os dois primeiros diagramas são relacionados aos dados persistentes. Os três próximos diagramas mostram as interfaces de usuário acessadas pelos principais atores e os diagramas restantes apresentam as classes agrupadas pelas funcionalidades oferecidas pela ferramenta.

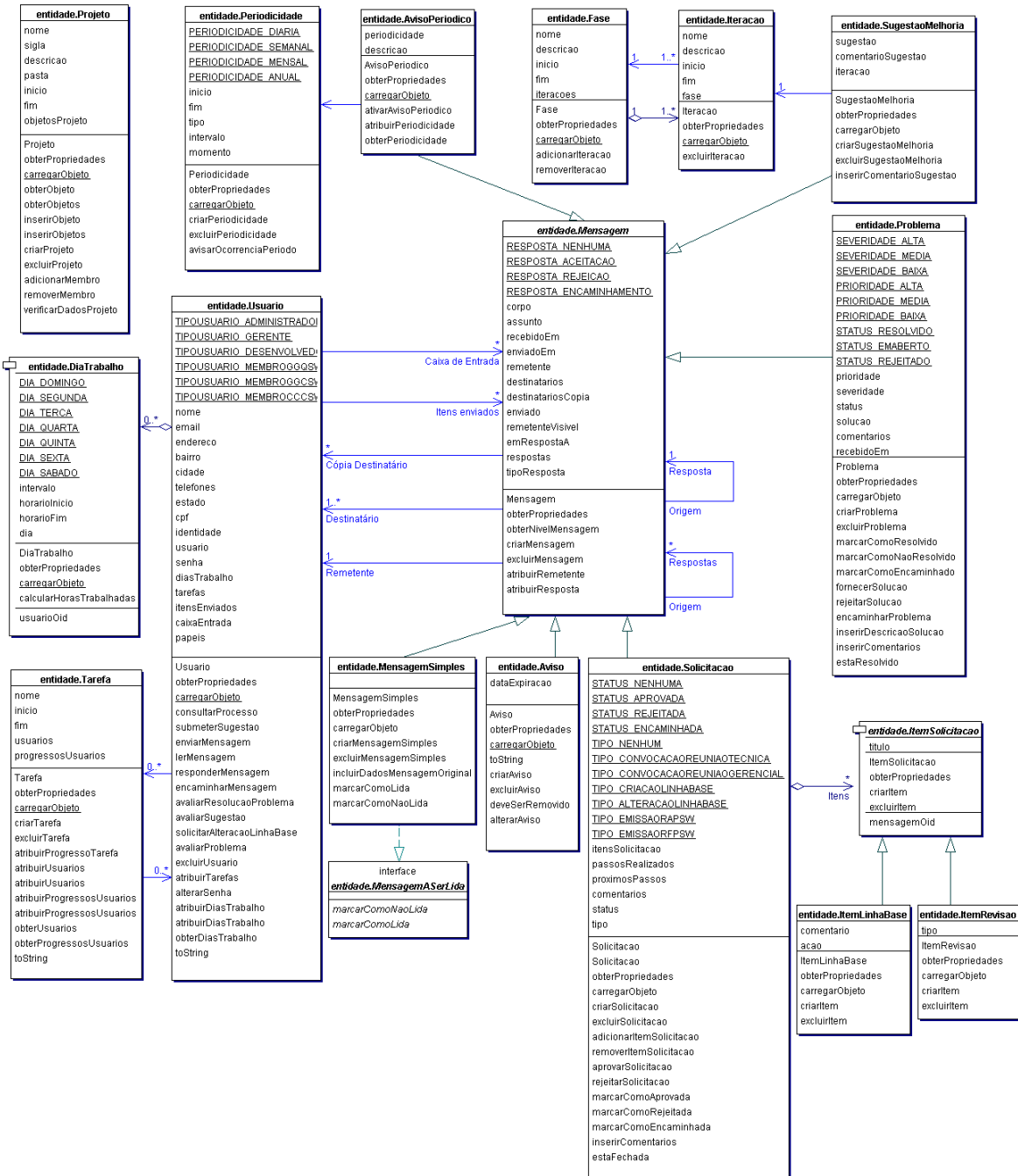


Figura 25 - Diagrama de Classes Persistentes do Praxis Mentor

O diagrama da Figura 25 apresenta as classes persistentes do Praxis Mentor. Os objetos persistentes são aqueles que continuam a existir após a execução dos programas que os criaram ou atualizaram. Tais objetos podem ser armazenados em arquivos ou bancos de dados de diversas tecnologias. Os sistemas de gerência de bancos de dados orientados por objetos são a solução

mais direta para o armazenamento de objetos persistentes, mas esta tecnologia é pouco difundida. A tecnologia dominante na área de bancos de dados ainda é a relacional.

Uma solução que tem sido muito adotada nesta questão do armazenamento de objetos persistentes é a aplicação de regras de transposição do modelo orientado por objetos para o modelo relacional. Esta questão no Praxis Mentor não foi resolvida de maneira diferente. Algumas das regras utilizadas para a transposição podem ser encontradas em [Blaaha+98]. O diagrama da Figura 26 apresenta o modelo relacional de desenho do banco de dados do Praxis Mentor.

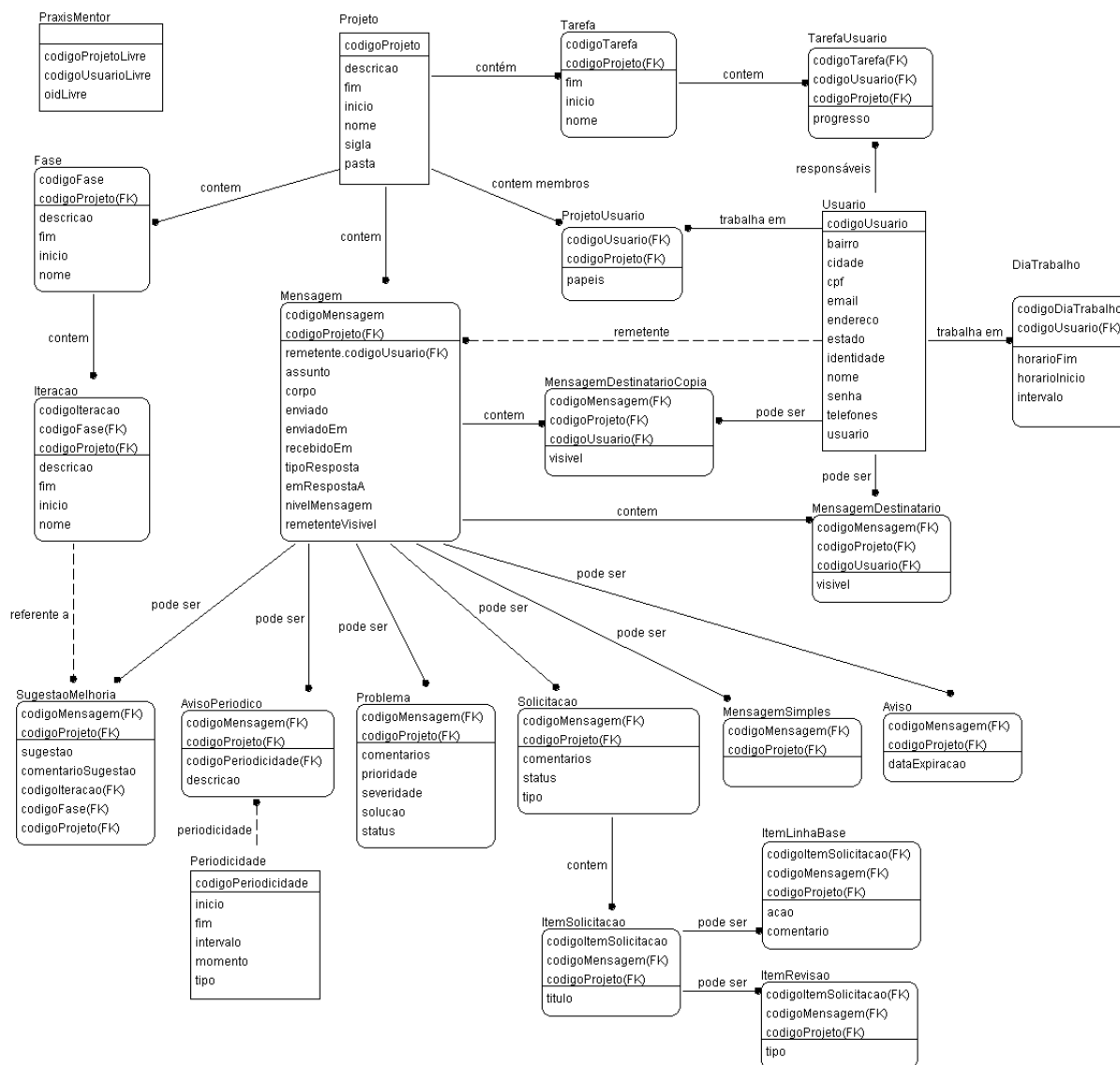


Figura 26 - Diagrama de Desenho do Banco de Dados do Praxis Mentor

O diagrama da Figura 26 foi feito utilizando a notação IDEF1x, que é uma das mais usadas para a modelagem de dados e foi escolhida para tal tarefa pela ferramenta CASE Together utilizada neste trabalho. Informações sobre esta notação podem ser encontradas em [IDEF1x00].

Este diagrama poderia ter sido feito utilizando a notação UML. Existem extensões da UML para a modelagem de dados que introduzem estereótipos de bancos de dados relacionais. Um exemplo é o *UML Data Modeling Profile*, implementado na ferramenta *Rational Rose Data*

Modelar da empresa Rational Software Corporation. Maiores informações sobre o assunto podem ser encontradas em [UMLDM00].

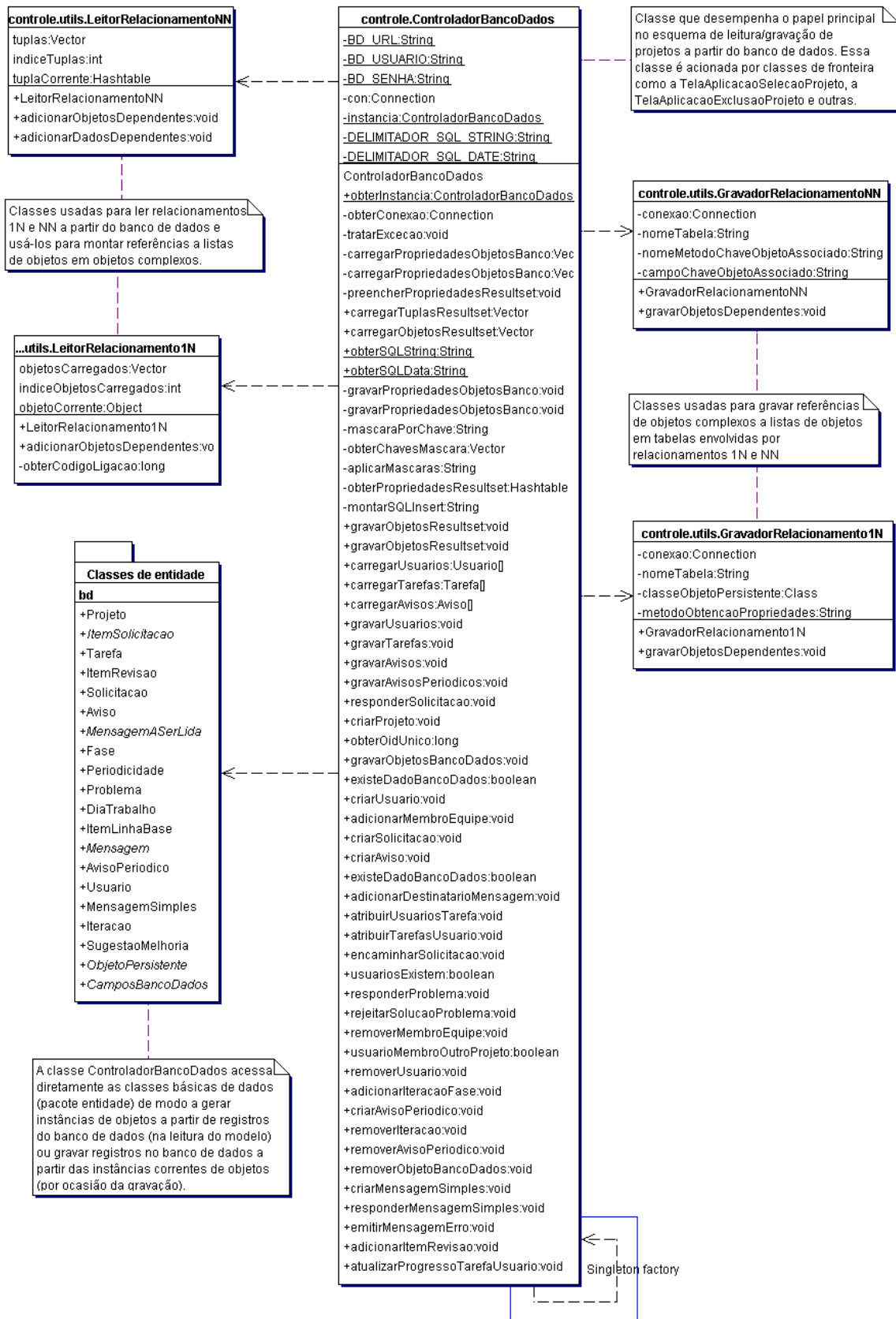


Figura 27 - Diagrama de Classes da Camada de Persistência do Praxis Mentor

Como já dito anteriormente, foi implementada uma camada de persistência para facilitar o acesso aos dados persistentes do Praxis Mentor. O diagrama da Figura 27 apresenta as classes que compõem esta camada. As notas contidas no diagrama fornecem as informações para o entendimento do mesmo.

Após o início da implementação da camada de persistência, notou-se que seu desenho estava pouco robusto, principalmente devido à presença, na classe **ControladorBancoDados**, de métodos relativos às classes persistentes. A solução para este problema poderia ser a definição destes métodos em classes herdeiras específicas e a permanência apenas dos métodos genéricos na superclasse. Como este problema só foi identificado após o início da implementação e causaria muito retrabalho, optou-se por apenas documentá-lo como sugestão de melhoria de desenho para próximas versões da camada de persistência.

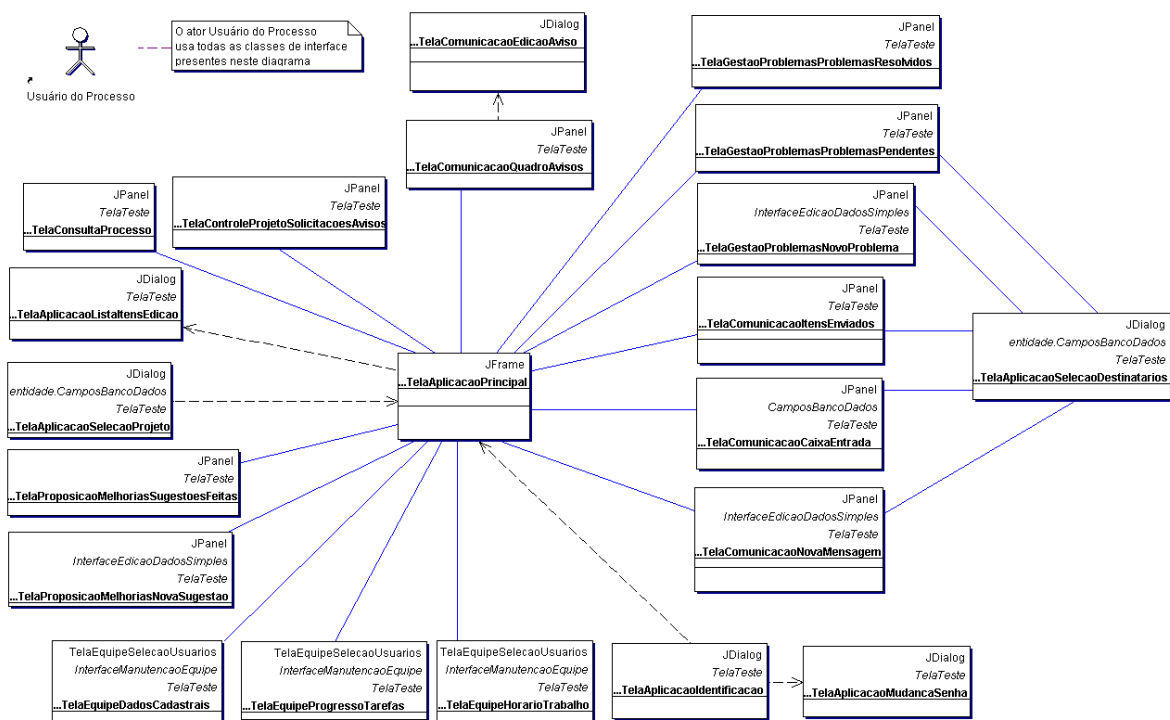


Figura 28 - Diagrama de Interfaces do Usuário do Processo

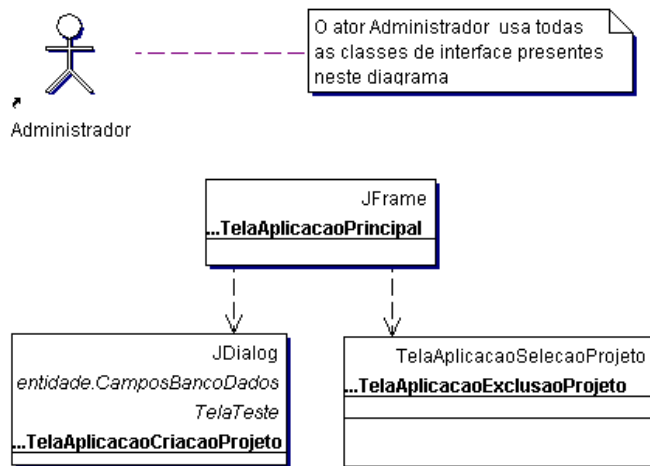


Figura 29 - Diagrama de Interfaces do Administrador

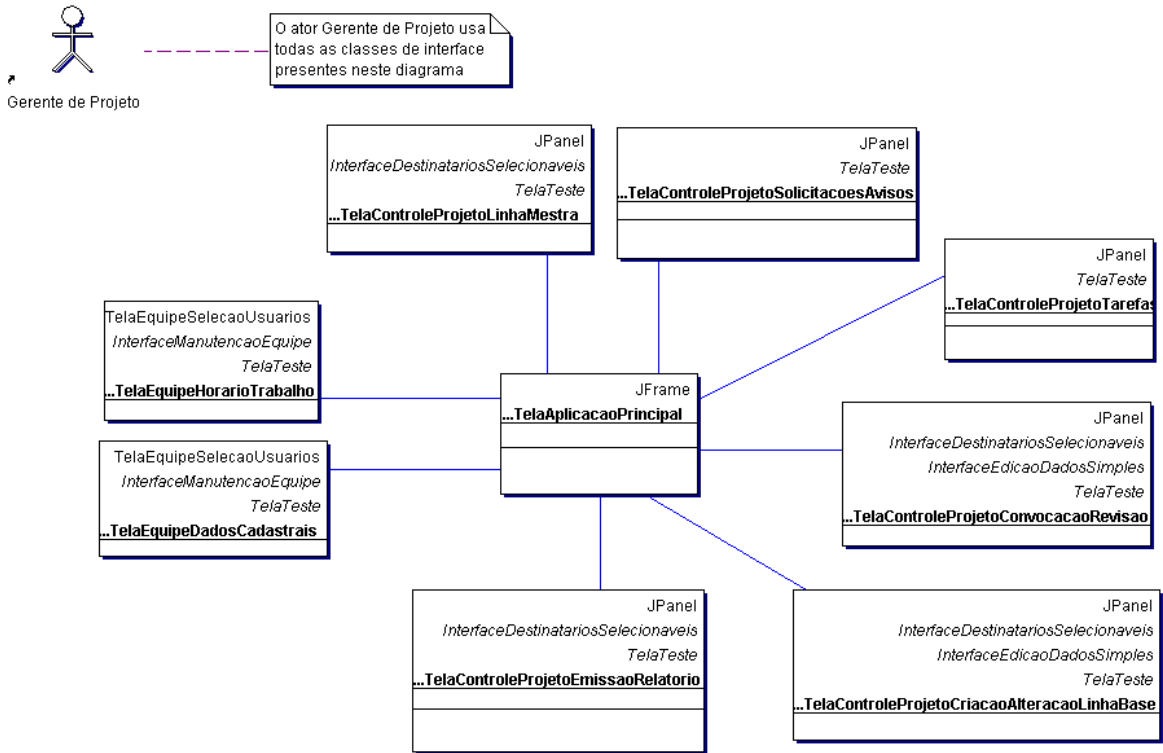


Figura 30 - Diagrama de Interfaces do Gerente de Projeto

Os três diagramas anteriores apresentam as interfaces acessadas pelos atores *Usuário do Processo*, *Administrador* e *Gerente de Projeto*. A interação dos atores *Desenvolvedor*, *Membro do GGQSw*, *Membro do GGCSw* e *Membro da CCCSw* com a ferramenta é muito semelhante à do ator *Usuário do Processo*, com diferença apenas no foco que cada usuário dá ao conteúdo tratado. Por esse motivo, não há necessidade de se detalhar os diagramas para esses atores.

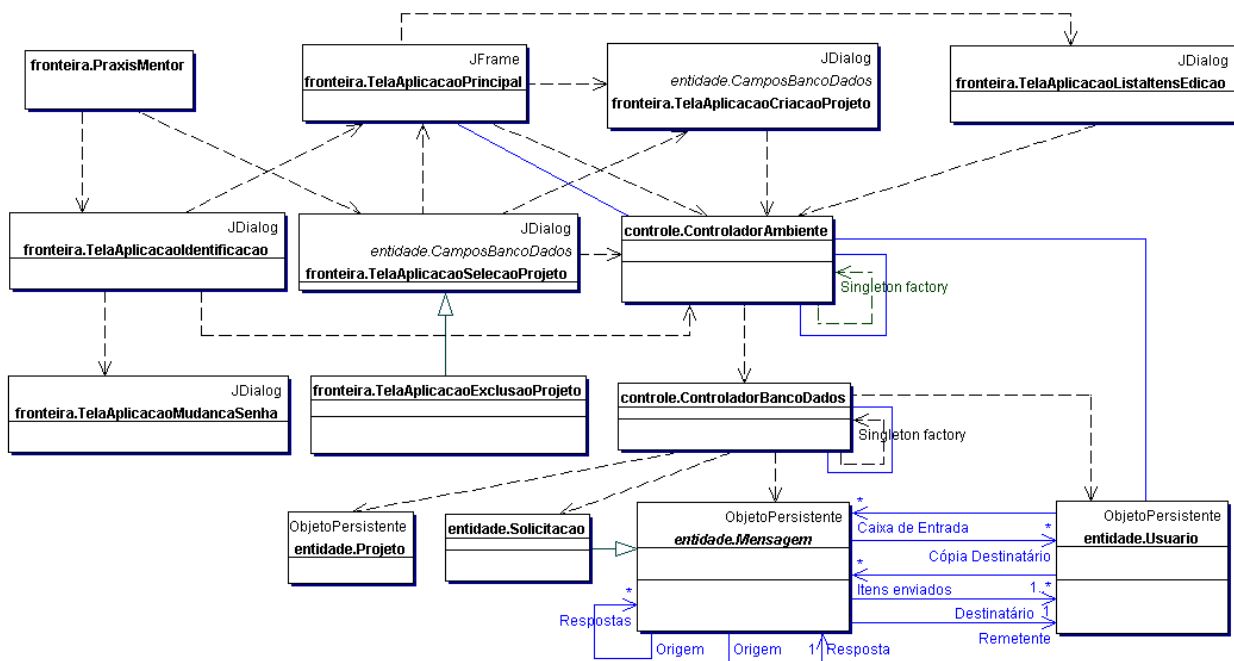


Figura 31 - Diagrama de Classes Aplicação

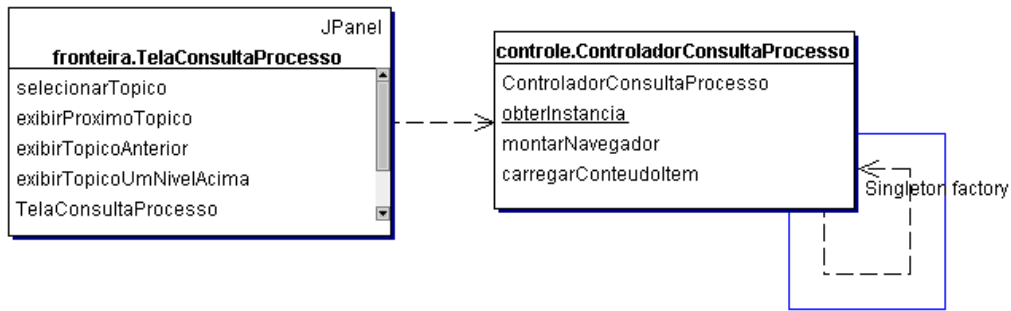


Figura 32 - Diagrama de Classes Consulta ao Processo

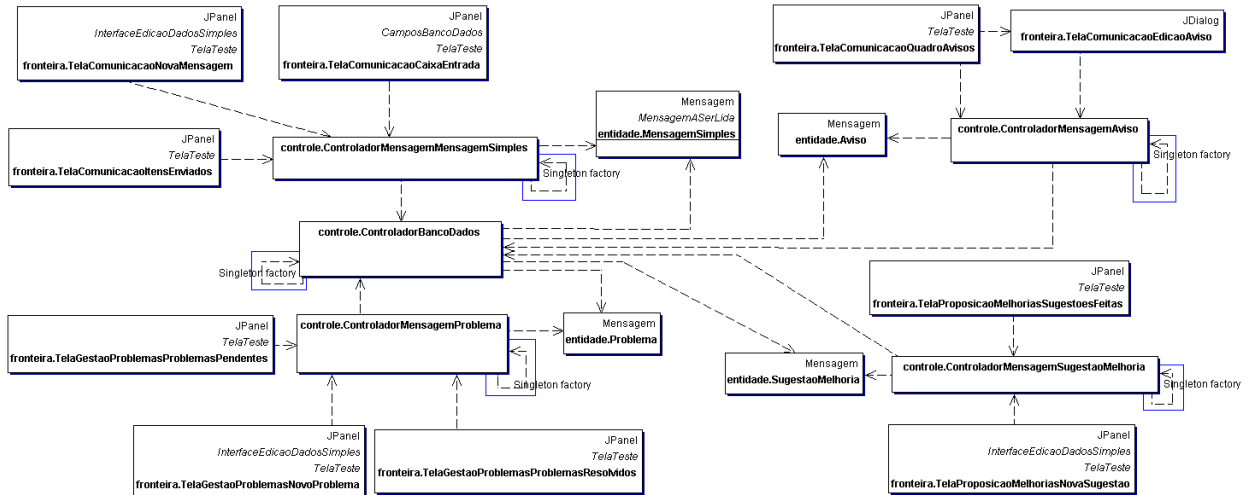


Figura 33 - Diagrama de Classes Comunicação

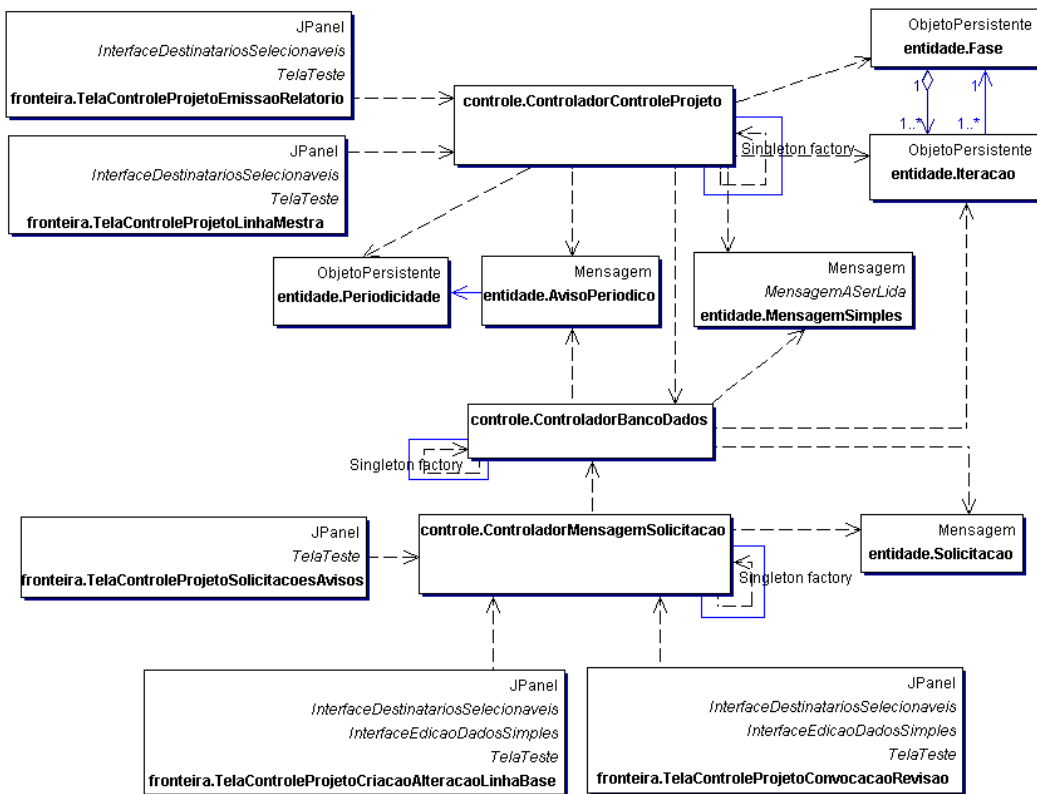


Figura 34 - Diagrama de Classes Controle de Projetos

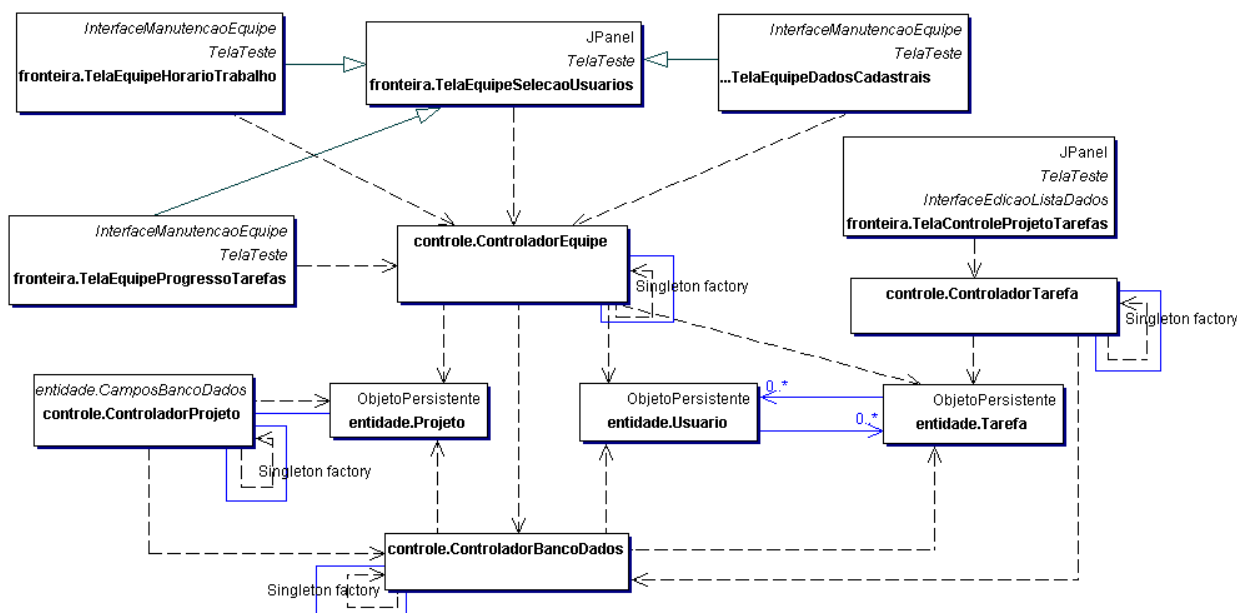


Figura 35 - Diagrama de Classes Planejamento de Projetos

Os últimos cinco diagramas apresentam as classes de desenho agrupadas pelos casos de uso primários da ferramenta que representam em alto nível todas as funcionalidades oferecidas pelo Praxis Mentor. Todos os diagramas apresentam as classes de fronteira, controle e entidade que colaboram para a implementação da funcionalidade em questão, da mesma forma mostrada no Modelo de Análise.

Diagramas de interação

O modelo de desenho deve mostrar a realização dos casos de uso através dos diagramas de seqüência e colaboração. Os diagramas de seqüência são utilizados para se enfatizar o ordenamento temporal das mensagens e os diagramas de colaboração para enfatizar os mecanismos através dos quais as classes envolvidas colaboram para a realização do caso de uso. Detalhes importantes do processamento interno das classes devem ser mostrados como anotações.

São apresentados a seguir os diagramas de seqüência e colaboração para alguns casos de uso do Praxis Mentor. Mais uma vez, foram escolhidas as mesmas realizações de casos de uso já mostradas no capítulo anterior, a título de comparação entre o modelo de Análise e o de Desenho.

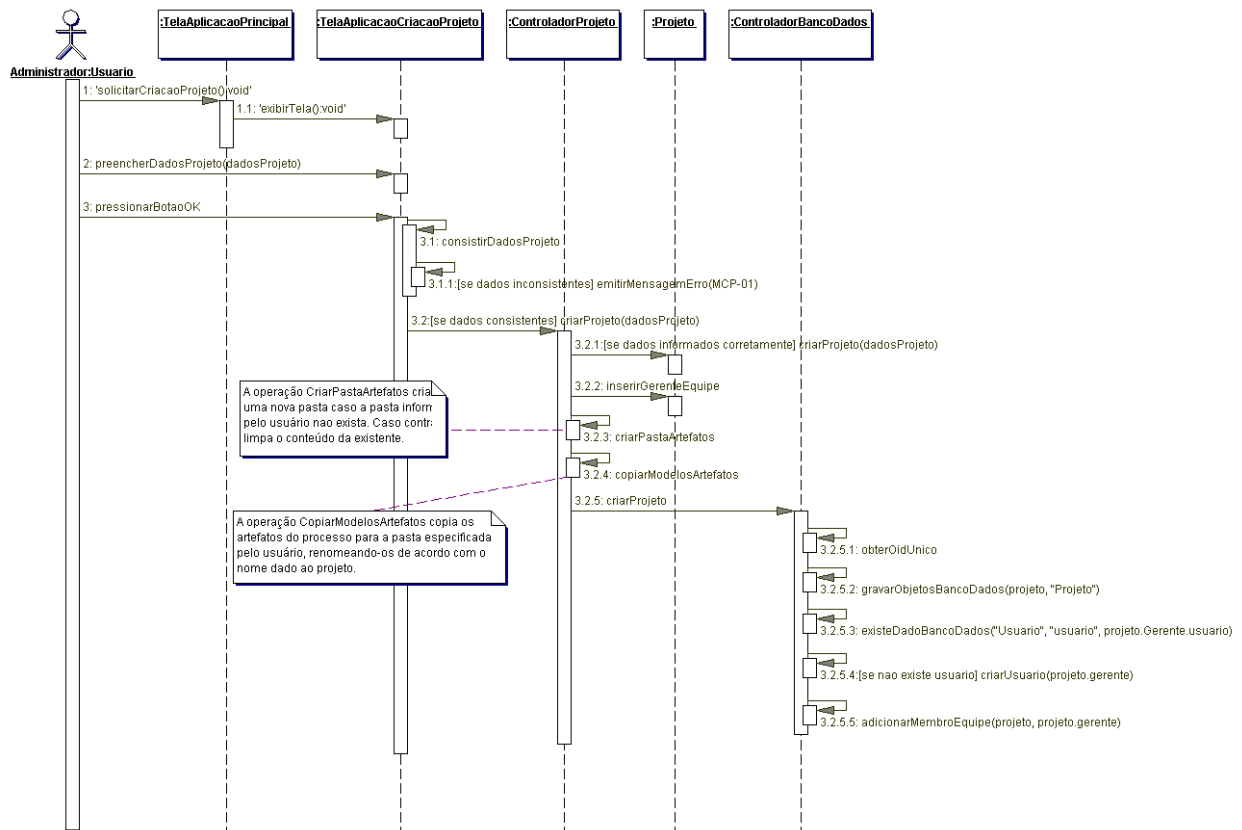


Figura 36 - Diagrama de Seqüência Criação de Projetos – Roteiro Principal

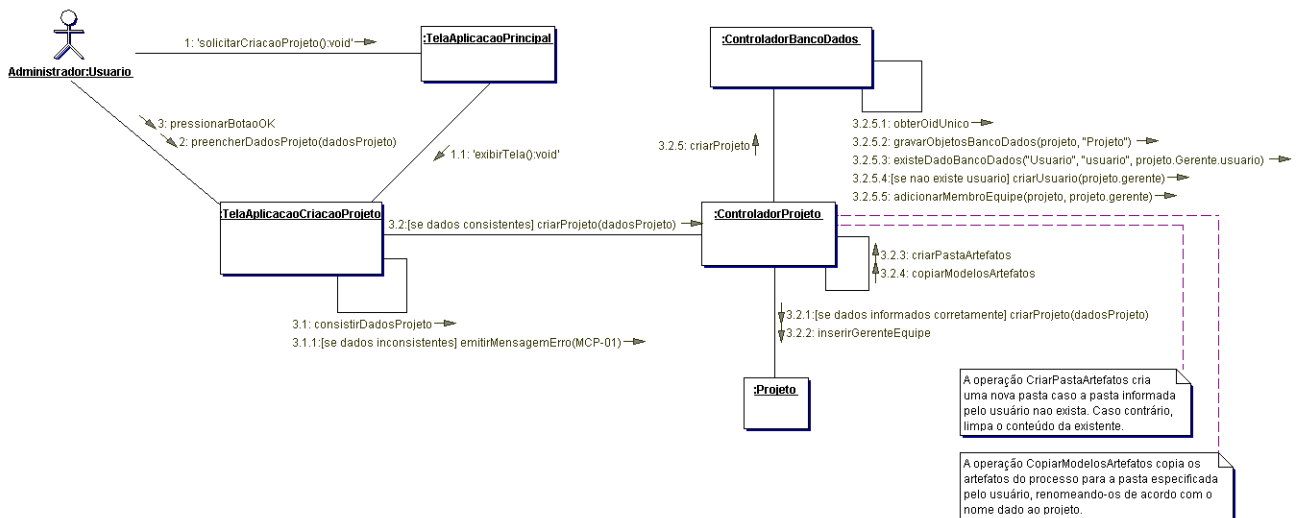


Figura 37 - Diagrama de Colaboração Criação de Projetos – Roteiro Principal

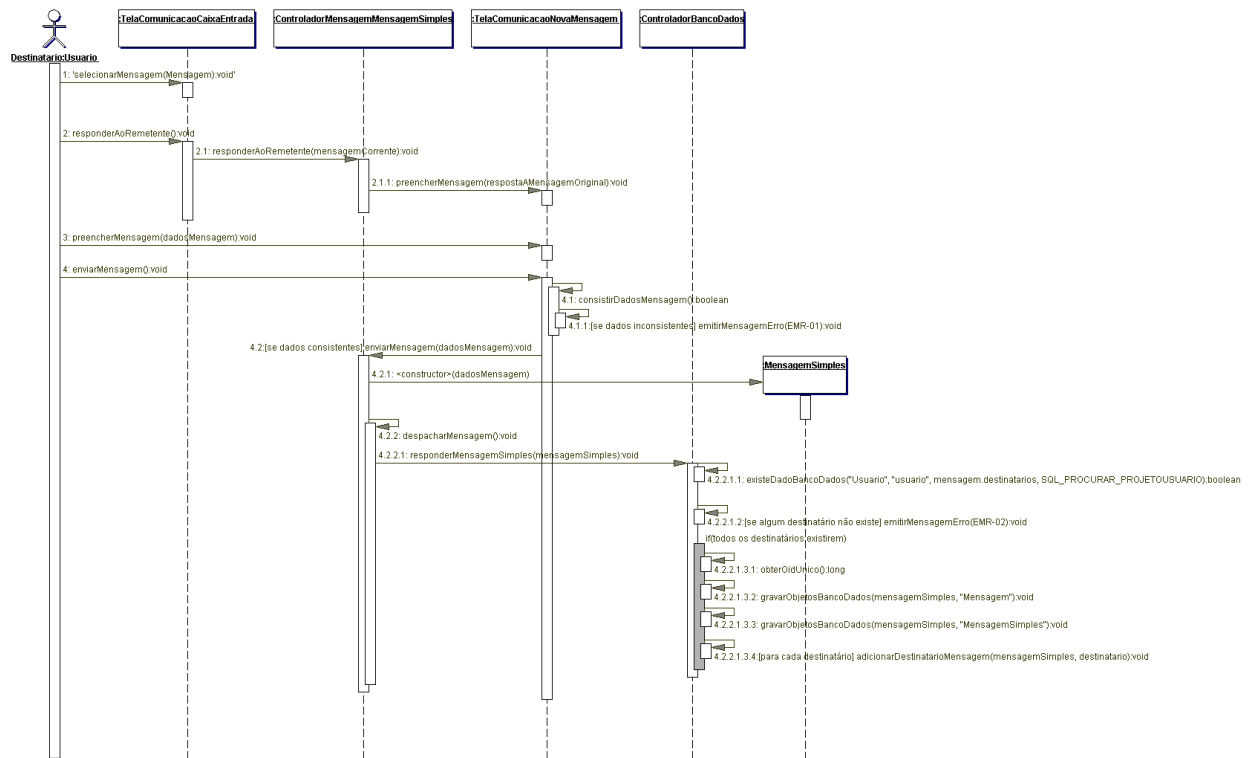


Figura 38 - Diagrama de Seqüência Controle de Mensagens Recebidas – Roteiro Responder ao Remetente

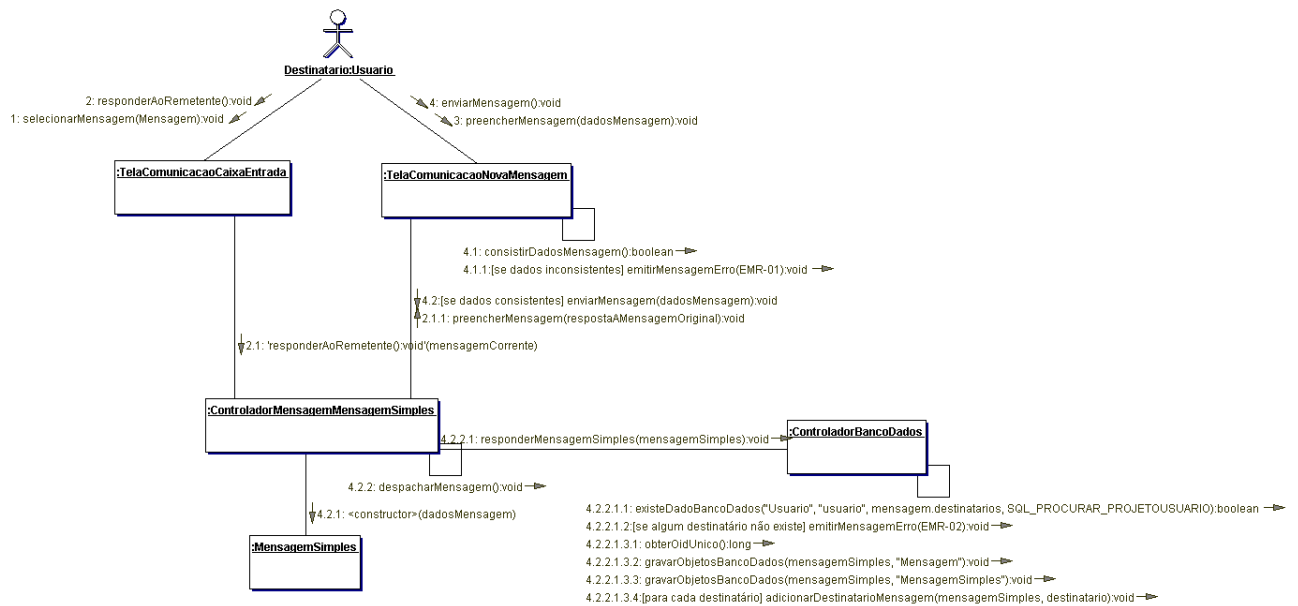


Figura 39 - Diagrama de Colaboração Controle de Mensagens Recebidas – Roteiro Responder ao Remetente

4.2.1.2.3 Visão física

A visão física do modelo de desenho contempla os diagramas de componentes físicos e os diagramas de desdobramento constantes do Modelo de Desenho. O primeiro deles apresenta os elementos estáticos do produto, como arquivos de código fonte, código executável e de

componentes reutilizados. Já o segundo apresenta a configuração física de processadores, dispositivos e conexões onde o produto será implantado.

Os diagramas de componentes físicos e de desdobramento do Praxis Mentor são apresentados nas figuras a seguir. As notas presentes nos diagramas contêm as explicações necessárias para o entendimento dos mesmos.

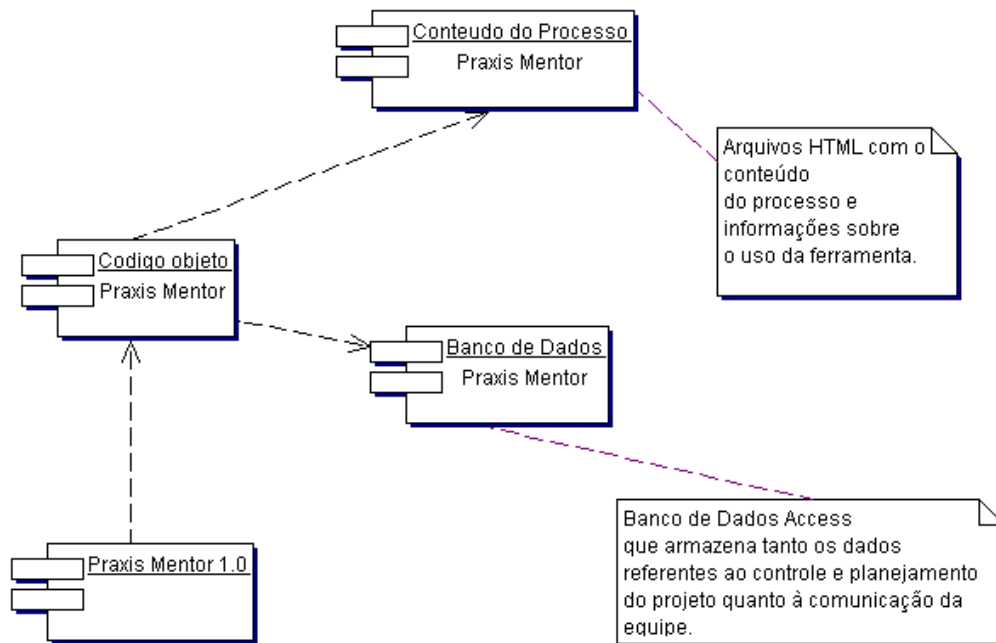


Figura 40 - Diagrama de Componentes Físicos do Praxis Mentor

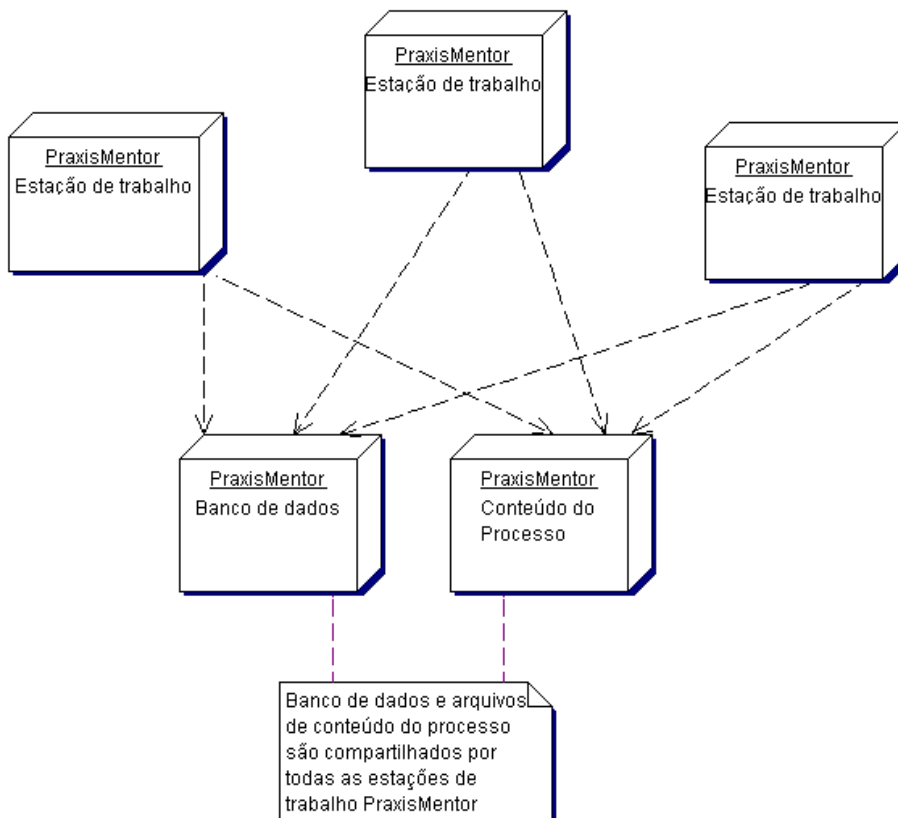


Figura 41 - Diagrama de Desdobramento do Praxis Mentor

4.2.1.3 Desenho das liberações

Segundo Paula [Paula01], o desenho deve usar uma abordagem progressiva e iterativa, através da construção de versões operacionais parciais, que sirvam como pontos de controle e permitam a avaliação por parte dos usuários. Estas versões parciais são as liberações.

A ordem das liberações deve procurar reduzir o risco do desenvolvimento colocando à prova nas primeiras liberações as principais áreas de risco como: os requisitos funcionais mais importantes, as interfaces gráficas de usuário, novos equipamentos ou sistemas com os quais a equipe de desenvolvimento não esteja familiarizada.

O planejamento do restante da fase de Construção deve ser feito com base nas liberações. A ordem das liberações deve ser a ordem de integração do produto. O plano de liberações deve ser consistente com os Planos de Testes do Software e com o Plano de Desenvolvimento do Software. Na Descrição do Desenho do Software (DDSw), o plano é refletido através da lista de liberações.

A lista de liberações do Praxis Mentor é apresentada na Tabela 39.

Número de ordem	Nome da liberação executável	Objetivos
1	Gestão de Projetos	Implementar os casos de uso relacionados com Consulta ao Processo, Controle e Planejamento de Projetos, permitindo o povoamento do banco de dados do <u>Praxis Mentor</u> .
2	Comunicação da Equipe	Implementar os casos de uso relacionados com a Comunicação (Gestão de Problemas, Proposição de Melhorias e Comunicação Interna da Equipe), completando as funções do <u>Praxis Mentor</u> .

Tabela 39 - Lista das liberações do Praxis Mentor

4.2.1.3.1 Especificação das liberações

As liberações geralmente são formadas por uma combinação de código novo, código já escrito que foi revisado e testado em liberações anteriores, componentes de teste tais como cotos (*stubs*) e controladores (*drivers*) e implementações simplificadas de algumas classes mais complexas, que serão detalhadas em liberações posteriores.

É aconselhável que as liberações sejam divididas de forma que cada uma implemente classes e casos de uso completos. Em casos especiais pode ser admissível que uma liberação implemente apenas algumas operações de uma classe ou roteiros de caso de uso, como nos casos em que isto é indispensável para tratar riscos maiores. Por exemplo, pode ser interessante submeter à avaliação dos usuários a implementação parcial de algum caso de uso importante antes de realizar a implementação completa.

A especificação de cada liberação do Praxis Mentor é apresentada nas tabelas a seguir. Esta especificação foi feita levando-se em conta as diretrizes expostas anteriormente. A primeira liberação implementa a parte de Gestão de Projetos e a segunda implementa a parte de Comunicação da Equipe. A parte de Comunicação foi deixada para a última liberação porque é a menos complexa e pode até ser substituída pelo uso de ferramentas de e-mail existentes. A parte de Gestão de Projetos possui as funções mais específicas e importantes do projeto e, por isso, foi tratada na primeira liberação.

Item	Descrição	
Classes a serem implementadas	Controle	ControladorAmbiente, ControladorConsultaProcesso, ControladorControleProjeto, ControladorEquipe, ControladorProjeto ControladorTarefa
	Entidades	DiaTrabalho, Fase, ItemLinhaBase, ItemRevisao, ItemSolicitacao, Iteracao, Mensagem, AvisoPeriodico , Periodicidade, Projeto, Solicitação, Tarefa, Usuario
	Fronteiras	TelaAplicacaoCriacaoProjeto , TelaAplicacaoExclusaoProjeto, TelaAplicacaoIdentificacao, TelaAplicacaoItensEdicao, TelaAplicacaoMudancaSenha, TelaAplicacaoPrincipal, TelaAplicacaoSelecaoProjeto, TelaConsultaProcesso, TelaControleProjetoCriacaoAlteracaoLinhaBase, TelaControleProjetoTarefas, TelaControleProjetoConvocacaoRevisao, TelaControleProjetoEmissaoRelatorio, TelaControleProjetoLinhaMestra, TelaControleProjetoSolicitacoesAvisos, TelaEquipeDadosCadastrais, TelaEquipeHorarioTrabalho, TelaEquipeProgressoTarefas
Classes a serem alteradas	Nenhuma.	
Casos de uso a serem implementados	Consulta ao Processo Gestão do Planejamento de Projetos Criação de Projetos Gestão da Composição da Equipe Gestão de Dados Cadastrais Definição de Horário de Trabalho Atribuição de Progresso de Tarefas Controle de Projetos Gestão de Tarefas Replanejamento Gestão de Solicitações e Avisos Convocação de Revisão Gestão de Linha de Base Emissão de Relatório	
Artefatos de teste	-	
Componentes reutilizados	Biblioteca com utilitários para acesso a banco de dados, tratamento de <i>strings</i> e números e componentes gerais de interface como listas e caixas de seleção.	

Tabela 40 - Liberação 1 do Praxis Mentor – Gestão de Projetos

Item	Descrição	
Classes a serem implementadas	Controle	ControladorMensagemAviso, ControladorMensagemMensagemSimples, ControladorMensagemProblema, ControladorMensagemSolicitacao, ControladorMensagemSugestaoMelhoria
	Entidades	Aviso , Mensagem, MensagemASerLida, MensagemSimples, Problema, SugestaoMelhoria
	Fronteiras	TelaComunicacaoCaixaEntrada, TelaComunicacaoEdicaoAviso TelaComunicacaoItensEnviados, TelaComunicacaoNovaMensagem TelaComunicacaoQuadroAvisos, TelaProposicaoMelhoriasNovaSugestao, TelaProposicaoMelhoriasSugestoesFeitas, TelaGestaoProblemasNovoProblema, TelaGestaoProblemasProblemasPendentes, TelaGestaoProblemasProblemasResolvidos
Classes a serem alteradas	ControladorAmbiente, ControladorProjeto, TelaAplicacaoPrincipal, TelaConsultaProcesso	
Casos de uso a serem implementados	Comunicação Interna da Equipe Manutenção de Quadro de Avisos Controle de Mensagens Recebidas Controle de Itens Enviados Cadastramento de Mensagem Gestão de Problemas Controle de Problemas Pendentes Controle de Problemas Resolvidos Cadastramento de Problema Proposição de Melhorias do Processo Controle de Sugestões Feitas Cadastramento de Sugestão	
Artefatos de teste	-	
Componentes reutilizados	Biblioteca com utilitários para acesso a banco de dados, tratamento de <i>strings</i> e números e componentes gerais de interface como listas e caixas de seleção.	

Tabela 41 - Liberação 2 do Praxis Mentor – Comunicação da Equipe

4.2.2 Códigos Fontes e Executáveis do Software

Durante as iterações de Liberações foram gerados os códigos fonte e executável do Praxis Mentor de acordo com a arquitetura e com os componentes internos e externos definidos. O planejamento de cada liberação, assim como o conjunto das classes envolvidas, foi mostrado na seção anterior.

A tabela a seguir apresenta uma porção do código fonte relativo à implementação da classe **ControladorBancoDados**.

```
package controle;

import entidade.*;
import util.net.URL;
import util.sql.*;
import util.util.*;
import util.lang.reflect.*;

/**
ControladorBancoDados - Classe que controla a carga e gravação do modelo de dados em
banco de dados

```

```

@author Júnia Gaudereto Carvalho
@version 1.0
*/
public class ControladorBancoDados {

    /** URL JDBC na qual o banco de dados se encontra */
    private final static String BD_URL = "jdbc:odbc:PRAXISMENTOR";

    /** Nome de usuário para acesso ao banco de dados */
    private final static String BD_USUARIO = "admin";

    /** Senha para acesso ao banco de dados */
    private final static String BD_SENHA = "";

    /** Objeto Connection privado usado para acesso a banco de dados */
    private Connection con = null;

    /** Instancia com para singleton */
    private static ControladorBancoDados instancia = null;

    /// METODOS PARA SINGLETON

    /** Construtor básico para a classe */
    ControladorBancoDados()
    {
        try
        {
            obterConexao();
        } catch (Exception e)
        {
            e.printStackTrace();
        }
    }

    /**
    Obtém instancia única de ControladorBancoDados
    @return Objeto ControladorBancoDados único por JVM
    */
    public static ControladorBancoDados obterInstancia()
    {
        if (instancia == null)
        {
            instancia = new ControladorBancoDados();
        }
        return instancia;
    }

    /// METODOS PRIVADOS

    /**
    Função que acessa o objeto conexão
    @return Retorna um objeto conexão para acesso ao banco de dados
    */
    private Connection obterConexao() throws Exception
    {
        if (con == null)
        {
            // inicializa o motor do JDBC
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

            // Tenta conexão ao driver Jdbc criado para o Praxis Mentor.
            con = DriverManager.getConnection(BD_URL, BD_USUARIO, BD_SENHA);
        }
        return con;
    }
}

```

Tabela 42 - Código fonte da classe ControladorBancoDados do Praxis Mentor

4.2.3 Manual do Usuário do Software

Segundo Paula [Paula01], todas as atividades do fluxo de implementação do Praxis são aplicáveis à documentação de usuário. Ela tem que ser desenhada, revisada e codificada.

Segundo o mesmo autor, os documentos de usuário podem ter diferentes modos de uso:

- **modo instrucional**: os documentos deste modo destinam-se a ajudar o usuário a aprender o que lhe for necessário a respeito de um produto. Este tipo de documento pode ser **orientado para informação**, quando tem por objetivo oferecer ao leitor informação necessária para entendimento do produto e de suas funções, ou **orientado para tarefa**, quando tem por objetivo mostrar ao leitor como realizar uma tarefa ou atingir um objetivo;
- **modo referencial**: os documentos de modo referencial destinam-se a armazenar e organizar informação necessária sobre um produto, facilitando a pesquisa por palavras-chaves. São exemplos de documentos referenciais típicos: manual de comandos, manual de mensagens de erro, manual de interface de programação e cartela de referência rápida;
- **modo misto**: um manual pode combinar os modos instrucional e referencial e as orientações para informação ou para tarefa, desde que cada um deles esteja em uma seção separada do corpo do documento, com um título que identifique claramente o tipo de material.

O Manual do Usuário do Software do Praxis Mentor foi elaborado de acordo com os padrões do Praxis para documentação de usuário. Ele pode ser classificado como um manual de modo referencial. As seções mais representativas deste manual são apresentadas a seguir.

4.2.3.1 Introdução

Segundo Paula [Paula01], se houver seções diferentes para determinados públicos, deve-se descrever quais e indicar o nível de experiência e treinamento prévio esperado para cada público.

Este manual se destina aos usuários do processo Praxis e da ferramenta Praxis Mentor 1.0. Ele contém algumas instruções que são específicas para os usuários com papéis de Administrador e Gerente de Projeto. Tais instruções se encontram destacadas no texto e não precisam ser lidas pelos demais usuários do processo.

Tabela 43 - Público alvo do manual do Praxis Mentor

Deve-se descrever também a versão do produto que está sendo documentada e do ambiente de hardware e de software no qual o produto será executado.

Este manual se refere à versão 1.0 do produto Praxis Mentor.

A configuração mínima para se utilizar o sistema é a seguinte:

- Sistema operacional Windows 95 ou posterior.
- Computador com microprocessador Pentium 200 MHz ou superior.
- 64 MB de memória RAM (mínimo).
- 100 MB de espaço livre em disco rígido.
- Unidade de disco CDROM.
- Adaptador gráfico Super VGA com resolução mínima de 800x600 pixels e 256 cores.
- Mouse.
- Impressora (opcional).

Tabela 44 - Aplicabilidade do Praxis Mentor

É igualmente importante descrever a missão do produto, indicando as aplicações esperadas.

O Praxis Mentor 1.0 visa oferecer apoio informatizado à utilização do processo Praxis em projetos de desenvolvimento de software.

Tabela 45 - Missão do Praxis Mentor

Devem ser apresentadas as convenções de símbolos e estilo utilizadas no documento, assim como as convenções usadas para descrever a sintaxe dos comandos do produto.

As convenções utilizadas na elaboração deste manual foram:

- Os parágrafos com a indicação **ATENÇÃO** descrevem aspectos que devem ser observados no preenchimento dos dados, conforme exemplificado a seguir:



ATENÇÃO

Devem ser observadas as seguintes definições gerais para preenchimento dos campos dos formulários:

- Endereço: Preencher com rua, número, bairro.
- Telefone: Número do telefone com 08 dígitos e código de área – DDD – com 02 dígitos.
- CEP: Código de Endereçamento Postal.

- Os parágrafos com as indicações **ADMINISTRADORES** e **GERENTES** indicam tarefas pertinentes a esses dois papéis específicos e devem ser ignorados pelos demais usuários da ferramenta.

Tabela 46 - Convenções utilizadas no manual do Praxis Mentor

4.2.3.2 Conceitos básicos do produto

É interessante se definir também os conceitos básicos do produto para que o usuário possa ter uma visão geral do mesmo.

Controle de Acesso: o Praxis Mentor possui o conceito de “login”. Cada usuário que desejar utilizar a ferramenta deverá fornecer valores de usuário e senha já cadastrados. Este cadastro é de responsabilidade do Administrador que possui acesso ilimitado à ferramenta. Na primeira vez que um usuário utilizar o Praxis Mentor, a senha terá o mesmo valor do campo usuário informado pelo Administrador e o Praxis Mentor exigirá que a troca de senha seja efetuada. Neste momento, cada usuário deverá informar uma senha secreta e particular e deverá memorizá-la para ser utilizada nas próximas vezes em que ele usar o Praxis Mentor.

O Praxis Mentor 1.0 fornece recursos baseados nos seguintes conceitos:

- **consulta ao processo:** exibição de textos em formato HTML que contemplam o conteúdo do processo e dicas de uso da ferramenta;
- **comunicação da equipe:** trocas de mensagens, manutenção de quadro de avisos, gestão de problemas e proposição de melhorias para o processo.
- **controle de projetos:** criação de uma linha mestra com todos os marcos do projeto, gestão de tarefas e atribuição das mesmas aos membros da equipe, gestão de solicitações e avisos, solicitação de revisões, solicitação de criação e alteração de linha de base e aviso de emissão de relatórios.
- **planejamento de projetos:** composição da equipe e criação de projetos. Através do recurso de composição da equipe é possível adicionar um novo membro à equipe do projeto, informar seu horário de trabalho e alterar o progresso das tarefas sob sua responsabilidade. Já o recurso de criação de projetos permite a geração de uma estrutura de pastas para armazenar os artefatos gerados pelo processo. No momento da criação do projeto, todos os gabaritos (*templates*) dos artefatos do projeto são copiados para a pasta informada pelo usuário renomeados de acordo com o nome dado ao projeto. A definição do gerente do projeto também é feita neste momento.

Tabela 47 - Conceitos básicos do Praxis Mentor

4.2.3.3 Funções do produto

Finalmente, deve-se descrever a maneira de utilizar as funções do produto. Cada função normalmente corresponde a um caso de uso do produto ou a um roteiro que percorra variantes importantes do fluxo de um caso de uso.

A descrição de algumas funções do Praxis Mentor é apresentada a seguir. Novamente foram escolhidas aquelas funções que correspondem aos casos de uso já apresentados como exemplo.

4.2.3.3.1 Criação de Projetos

Objetivo

O objetivo desta seção é descrever a maneira de usar a função Criação de Projetos do Praxis Mentor.

O acesso à função Criação de Projetos é sempre feito através da *Tela Propriedades de Projeto / Novo Projeto*. Esta é mostrada e ativada quando o item Novo do cardápio Projeto da *Tela Principal* é acionado.

Materiais

Não aplicável.

Preparação

Não aplicável.

Entradas

Os dados a serem informados para a criação de um projeto são exibidos na tela apresentada na figura a seguir. São eles:

- sigla;
- nome;
- descrição;
- início;
- fim;
- gerente;
- pasta do projeto.

A imagem mostra uma janela de diálogo intitulada "Propriedades do projeto [Criação de novo projeto]". A janela contém os seguintes campos e controles:

- Sigla:** Campo de texto com o valor "PMTI".
- Nome:** Campo de texto com o valor "Integrador Praxis Mentor - Together".
- Descrição:** Área de texto com o conteúdo "Integra as ferramentas Praxis Mentor e Together".
- Início:** Campo de data com o valor "01/06/2001".
- Fim:** Campo de data com o valor "31/10/2001".
- Gerente:** Campo de lista suspensa com o nome selecionado "Júnia Gaudereto Carvalho (junia)".
- Pasta do projeto:** Campo de texto com o caminho "c:\projetos\pmti".
- Botões "OK" (com ícone de checkmark verde) e "Cancelar" (com ícone de X vermelho) na base da janela.

Precauções

Antes de cancelar a criação de um novo projeto o usuário deverá verificar se deseja mesmo cancelar a operação. Uma vez confirmado o cancelamento, todos os dados já informados serão perdidos. A seguinte mensagem de confirmação será exibida: “Confirma o cancelamento da criação de novo projeto?”.

Chamada

Não aplicável.

Suspensão de operações

Não aplicável.

Encerramento de operações

Esta função é encerrada através dos botões “Ok” ou “Cancelar”. Se o usuário acionar o botão “Ok”, o projeto será criado. Se o usuário acionar o botão “Cancelar”, a criação do projeto será abortada e todos os dados informados serão perdidos.

Saídas

Ao término da criação do projeto, o Praxis Mentor terá copiado para a pasta informada pelo usuário todos os modelos de artefatos do Praxis renomeados de acordo com o nome dado ao projeto. Esta pasta armazenará todo o conteúdo relativo ao projeto.

Condições de erro

Se o usuário tentar criar um projeto sem preencher algum campo, a seguinte mensagem será exibida: “Todos os campos são obrigatórios e devem ser preenchidos para que o projeto seja criado.”.

Se for solicitada a criação de um projeto mas ocorrer um erro no momento de gravar no banco de dados, será levantada uma exceção que apresentará a mensagem: “Problema ao acessar o banco de dados. Entre em contato com o suporte, por favor.”

Informações relacionadas

ADMINISTRADORES e GERENTES

Apenas os usuários que tiverem o papel de Administrador terão acesso a esta função. Os usuários que tiverem o papel de Gerente de Projeto poderão apenas alterar os dados *descrição*, *data de início* e *data de fim* de um projeto.

4.2.3.3.2 Controle de Mensagens Recebidas

Objetivo

O objetivo desta seção é descrever a maneira de usar a função Controle de Mensagens Recebidas do Praxis Mentor.

O acesso à função Controle de Mensagens Recebidas é sempre feito através da Tela Comunicação – Caixa de Entrada. Esta tela é mostrada e ativada quando o item Caixa de Entrada do cardápio Ações\Comunicação ou do cardápio lateral da Tela Principal é acionado. Inicialmente, as mensagens que foram recebidas anteriormente são exibidas.

Materiais

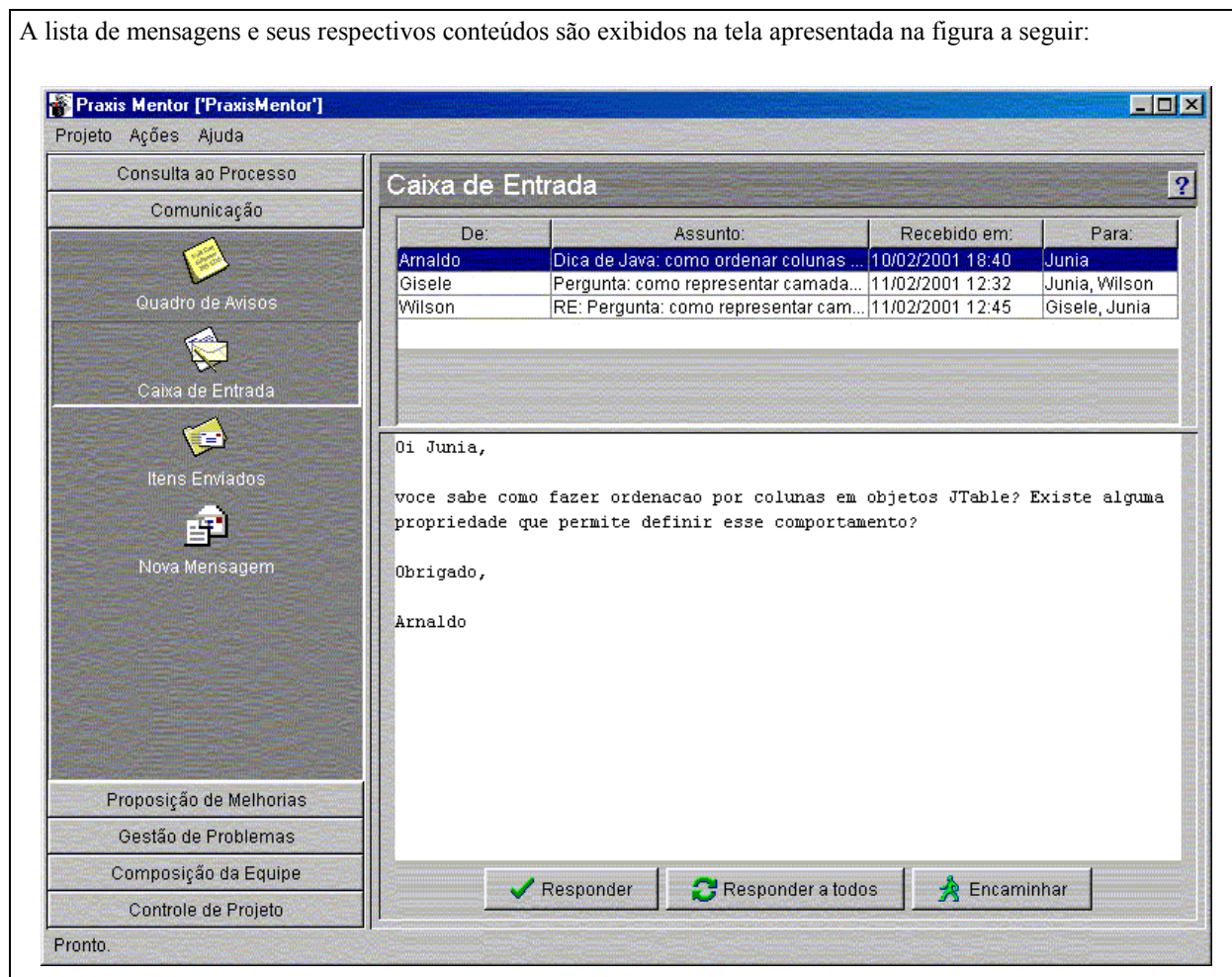
Não aplicável.

Preparação

Não aplicável.

Entradas

A lista de mensagens e seus respectivos conteúdos são exibidos na tela apresentada na figura a seguir:



Precauções

Antes de excluir uma mensagem o usuário deverá verificar se deseja mesmo excluí-la e se está selecionando a mensagem correta. Uma vez confirmada a operação, ela não poderá ser cancelada e a mensagem será definitivamente excluída.

A seguinte mensagem de confirmação será exibida: “Confirma a exclusão desta mensagem?”.

Chamada

O usuário poderá selecionar uma mensagem e acionar um dos seguintes botões: “Responder”, “Responder a Todos” ou “Encaminhar”. Todos esses comandos acionarão a função **Cadastramento de Mensagem**.

O usuário poderá também selecionar uma mensagem e excluí-la através da tecla *delete* ou do botão “Excluir”.

Suspensão de operações

Se o usuário acionar um dos botões que leva à função **Cadastramento de Mensagem** e não acionar o botão “Enviar”, a operação ficará suspensa. No momento em que o usuário for encerrar a execução do Praxis Mentor, será exibida uma lista com os itens que ficaram em edição para que o usuário possa finalizá-los. Se o usuário preferir finalizá-los em outro momento, eles permanecerão armazenados nesta lista.

Encerramento de operações

Esta função é finalizada quando o usuário aciona um dos seguintes botões: “Responder”, “Responder a Todos” ou “Encaminhar” e depois aciona o botão “Enviar” da função **Cadastramento de Mensagem**.

Se o usuário tiver selecionado esta função apenas para consulta de mensagens, a função será finalizada quando ele selecionar qualquer outra função do Praxis Mentor.

Saídas

Não aplicável.

Condições de erro

Se o usuário informar algum usuário inexistente como destinatário da mensagem, será levantada uma exceção que apresentará a mensagem: “Um ou mais destinatários dessa mensagem não existem. Por favor, escolha os usuários cadastrados através dos comandos “Para...” ou “CC...”.”

Se o usuário tentar encaminhar uma mensagem sem preencher o campo “Para”, a seguinte mensagem será exibida: ““Para” é obrigatório e deve ser preenchido para que a mensagem seja enviada.”.

Se for solicitado o envio de uma mensagem mas ocorrer um erro no momento de gravar no banco de dados, será levantada uma exceção que apresentará a mensagem: “Problema ao acessar o banco de dados. Entre em contato com o suporte, por favor.”

Capítulo 5

Transição

5.1 Definição

A fase de Transição se inicia após o término da Construção e encerramento dos Testes Alfa e é composta de duas iterações: Testes Beta e Operação Piloto. Na primeira iteração os testes de aceitação são repetidos no ambiente dos usuários. Já na segunda, é realizada a operação experimental do produto em uma instalação piloto do cliente, com a resolução de eventuais problemas através de processo de manutenção.

Durante a iteração Testes Beta, os problemas relacionados com o funcionamento em um ambiente real são identificados e resolvidos. A documentação do usuário também é testada. O principal resultado desta iteração é o artefato Relatórios dos Testes do Software (RTSw), que contém os resultados dos testes realizados. A aprovação desta iteração inclui o controle através de auditoria da qualidade e revisão gerencial e a aprovação do cliente. Esta aprovação significa a autorização do cliente para o início da Operação Piloto.

Segundo Paula [Paula01], a Operação Piloto representa um estado de “liberdade vigiada” do produto. Os problemas encontrados pelos usuários passam a ser tratados pelo processo de manutenção do produto, possibilitando a avaliação da qualidade dos serviços de manutenção e suporte ao usuário. Os defeitos encontrados permitem uma primeira avaliação do produto final.

Durante a Operação Piloto é feita uma análise do projeto como um todo, em especial dos problemas de processo encontrados. O resultado desta avaliação e da iteração é um Relatório Final do Projeto (RFPSw), onde são armazenadas as métricas importantes do projeto e as lições que proporcionarão a melhoria do processo em projetos futuros. Os critérios de aprovação desta iteração são uma auditoria da qualidade, uma revisão gerencial e a aceitação final do produto pelo cliente.

5.2 Transição do Praxis Mentor

A fase de Transição do Praxis Mentor não chegou a ser iniciada. Conforme mencionado anteriormente, o adiamento desta fase para trabalhos futuros fez-se necessário devido às restrições de tempo existentes para o projeto.

Capítulo 6

Conclusão e Trabalhos Futuros

6.1 Conclusão

O desenvolvimento de software está muito além da simples geração de inúmeras linhas de código. O software construído deve ser correto, o que envolve o uso de um processo de desenvolvimento voltado para os casos de uso do produto. O software deve ser robusto, o que envolve a escolha de uma arquitetura correta. O software deve ser extensível, o que deve ser feito de modo iterativo e incremental através de diferentes visões do produto.

Através do suporte às características citadas, o uso de um processo de desenvolvimento de software consegue viabilizar a produção de sistemas de alta qualidade que atendam às necessidades dos usuários finais e que sejam bastante previsíveis. Geralmente os processos contemplam as melhores práticas de desenvolvimento de software como forma de atender a um grande intervalo de projetos e organizações. Em termos de gerência, os processos provêm uma abordagem disciplinada de como planejar e controlar projetos numa organização produtora de software.

Para facilitar e viabilizar a aplicação de um processo de desenvolvimento de software em uma organização, são utilizadas ferramentas de apoio que disponibilizam diretrizes e recursos para auxiliar o usuário no uso do processo e garantir uma correta aplicação do mesmo.

O presente trabalho se propôs a conceber e implementar uma ferramenta de apoio a um processo que pudesse atender a todas essas características. O processo em questão é o Praxis, Processo para Aplicativos Extensíveis Interativos, definido em [Paula01], e a ferramenta proposta foi denominada Praxis Mentor.

O Praxis é um processo que integra um conjunto de métodos e padrões cobrindo todos os principais aspectos da Engenharia de Software. Ele se aplica especialmente ao desenvolvimento de aplicativos gráficos interativos baseados na tecnologia orientada por objetos e foi desenhado para suportar projetos de seis meses a um ano de duração em um contexto educacional ou de treinamento. Apesar desta restrição de contexto, ele tem sido utilizado em projetos reais da Universidade Federal de Minas Gerais com grande êxito.

O Praxis Mentor é uma ferramenta CASE que oferece suporte a todas as fases do processo instruindo o desenvolvedor no uso do mesmo, fornecendo acesso fácil a todos os documentos e modelos utilizados em cada fase e facilitando a gestão do processo.

Ele facilita a comunicação da equipe fornecendo funcionalidades para Gestão de Problemas, Proposição de Melhorias para o processo e Comunicação da Equipe através de trocas de mensagens e da manutenção de um Quadro de Avisos a que todos os membros têm acesso. Ele fornece apoio ao Controle do Projeto disponibilizando funcionalidades como Solicitação de Revisões, Solicitação de Criação ou Alteração de Linha de Base, Aviso de Emissão de Relatórios, Gestão e Atribuição de Tarefas e criação de uma Linha Mestra com todos os marcos do projeto. Ele ainda permite a criação de Avisos Periódicos que podem ser configurados para alertar o usuário em determinado momento. Finalmente, o Praxis Mentor oferece suporte ao Planejamento de Projetos auxiliando na tarefa de Composição da Equipe e na Criação de Projetos. Através dos recursos de Composição da Equipe, é possível se criar um membro, definir seu horário de trabalho e informar o progresso das tarefas sob sua responsabilidade. O recurso de Criação de Projetos gera uma estrutura de pastas para o novo projeto, copiando para a pasta definida pelo usuário todos os gabaritos (*templates*) do processo renomeados de acordo com o nome dado ao projeto.

A concepção e implementação do Praxis Mentor foi realizada seguindo o processo Praxis. Este fato foi de grande importância para o sucesso do projeto, uma vez que proporcionou uma experiência real de uso do processo. Esta experiência foi fundamental para o conhecimento das necessidades e dificuldades dos usuários. Algumas necessidades e dificuldades foram levantadas nas fases posteriores à concepção da ferramenta e, por isso, o tratamento delas foi postergado para futuras expansões do Praxis Mentor.

Em relação aos aspectos técnicos do projeto, pode-se destacar a utilização de ferramentas de automação e da linguagem Java como fatores de sucesso para o projeto. A ferramenta CASE *Together* teve uma importância fundamental ao tornar transparente a geração do código relativo aos modelos pertinentes. A linguagem Java, por sua vez, proporcionou uma implementação muito fiel aos modelos criados devido à característica de ser puramente orientada por objetos.

Em relação aos aspectos gerenciais do projeto, destacam-se os procedimentos de cálculos de métricas como fatores de grande importância para o cumprimento dos prazos existentes para o trabalho. Através do uso da técnica de Análise de Pontos de Função, foi possível a realização de estimativas de tamanho e esforço que serviram de diretrizes para as principais decisões de planejamento do projeto. Devido a problemas de prazos identificados na fase de Elaboração, por exemplo, optou-se por postergar para projetos futuros todas as atividades relacionadas à iteração de Testes Alfa da fase de Construção e todas as atividades da fase de Transição. Optou-se, também, por elaborar o desenho de todos os casos de uso, utilizá-los nas liberações e implementar o que fosse possível até o fim do projeto. Outras decisões de cunho técnico, como a implementação de uma camada de persistência, foram tomadas para tentar minimizar o tempo de desenvolvimento e possibilitar a implementação da maior parte dos casos de uso possível. A união de todas essas decisões acabou por permitir a implementação de todos os casos de uso propostos.

O resultado deste trabalho foi a concepção de uma ferramenta que oferece o suporte básico ao uso do processo. Nesta primeira versão foram tratados com maior importância os aspectos gerenciais de controle de projeto e gestão de equipes. As questões relacionadas à parte de comunicação da equipe também tiveram um suporte especial. Ainda para esta parte gerencial muitas melhorias poderão ser feitas na ferramenta, principalmente no que diz respeito à

integração com as demais ferramentas utilizadas para a criação dos artefatos do processo. Esta integração foi propositalmente postergada para uma próxima versão do Praxis Mentor devido às restrições de tempo existentes para este projeto. Mesmo assim, pode-se considerar os resultados do projeto Praxis Mentor 1.0 bastante satisfatórios. Levando-se em consideração que toda a concepção desta primeira versão serviu de base para o aprendizado do próprio processo e que produziu documentos completos de especificação e desenho que poderão ser utilizados para a continuação deste trabalho, pode-se perceber o valor agregado a este projeto. Ele foi responsável pela concepção das primeiras idéias para a ferramenta e também proporcionou uma avaliação do processo que permite um levantamento de requisitos mais apurado para uma próxima versão da ferramenta.

A análise que se pode fazer do processo Praxis é que ele atende de forma bastante satisfatória a todas as necessidades de cada fase do ciclo de vida de desenvolvimento. Entretanto, existe uma dificuldade de se produzir e gerenciar todos os artefatos gerados pelo processo. O que se pôde perceber neste trabalho é que alguns dos artefatos acabam por ser subutilizados ou até mesmo não produzidos. Este fato torna ainda mais clara a necessidade de se utilizar ferramentas de apoio ao uso do processo.

O uso de ferramentas técnicas como o *Together*, por exemplo, é fundamental para a garantia da qualidade da modelagem, do código gerado a partir desta modelagem e da manutenção da consistência entre o código e o modelo. Ferramentas de gestão de configurações, como o *Microsoft Visual Source Safe*, também são de grande importância para o controle e integridade dos artefatos produzidos. Ferramentas adequadas para codificação e testes também são importantes para garantir a qualidade e a rapidez do desenvolvimento. Em relação à parte gerencial, uma ferramenta de controle de projetos como o *Microsoft Project* pode ser muito útil para auxiliar na tarefa de planejamento do projeto. E, finalmente, uma ferramenta que forneça apoio ao processo e que ofereça as diretrizes para o uso das demais ferramentas pode ser de fundamental importância para a garantia do sucesso no uso do processo e para o melhor aproveitamento de todo o suporte que o uso de um processo pode oferecer.

E isto é o que o Praxis Mentor oferece aos seus usuários. O caminho a seguir a partir deste momento é a realização de um acompanhamento no uso da ferramenta e a captação das necessidades que irão surgir para a implementação das melhorias necessárias. E, além disso, o oferecimento de melhores condições para que os usuários do Praxis explorem as facilidades de planejamento e desenvolvimento oferecidas pelo processo.

Algumas melhorias já levantadas para o Praxis Mentor são apresentadas na próxima seção.

6.2 Trabalhos Futuros

Algumas melhorias para o Praxis Mentor foram identificadas ao longo do projeto, na medida em que se ia conhecendo melhor o processo Praxis e percebendo as dificuldades enfrentadas pelos usuários ao utilizá-lo. Outras melhorias consistem em requisitos adiados propositalmente para próximas versões ou requisitos que eram opcionais e não puderam ser implementados. Estas melhorias são listadas a seguir agrupadas pelas categorias citadas.

6.2.1 Requisitos adiados

- **consistência entre os dados do projeto (tabelas como as do Plano de Desenvolvimento do Software, por exemplo) e os artefatos do processo:** esta consistência pode ser garantida através do cadastro dos dados de planejamento do projeto como datas, recursos, riscos, etc na ferramenta e geração automática de parte dos artefatos ou da importação desses dados a partir dos artefatos do processo;
- **integração com uma ferramenta de Gestão de Configurações (como o *Microsoft Visual SourceSafe*, por exemplo), com uma ferramenta de Modelagem (como o *Together* ou *Rational Rose*) ou com uma ferramenta de Edição de Textos (como o *Microsoft Word*):** a integração com estas ferramentas pode possibilitar a edição dos artefatos do projeto a partir do Praxis Mentor.
- **incorporação de atividades como backup e recuperação das bases de dados do sistema:** o apoio a estas atividades pode proporcionar um controle melhor dos dados do projeto, minimizando os riscos de perda de informações.
- **fase de Transição:** a realização das atividades relativas a esta fase pode ser muito importante para a garantia da correção e qualidade da ferramenta e para uma correta implantação da mesma.
- **realização de testes de usabilidade:** a realização de testes de usabilidade com os usuários finais poderá servir de base para a melhoria da interface em versões futuras do produto.

6.2.2 Requisitos opcionais da versão 1.0 não implementados

- **implementação de recursos que facilitem o cálculo de métricas do processo:** tais recursos podem ser ligados ao cálculo de pontos de função, à coleta de horas trabalhadas de cada membro da equipe por tarefa ou à geração de relatórios que abordem todas essas informações. Na presente versão do Praxis Mentor, o que existe neste sentido é o acompanhamento do progresso das tarefas alocadas a um determinado membro da equipe;
- **auxílio na personalização do processo:** em alguns casos pode ser necessário personalizar o Praxis para incorporá-lo à prática de uma organização produtora de software. O Praxis Mentor poderia fornecer recursos tais como a criação de novas fases, fluxos e papéis para tornar esse processo mais ágil. Atualmente a ferramenta possibilita um pequeno grau de personalização, com recursos como a criação de novas iterações para o processo e a importação de conteúdo para consulta ao processo. O último recurso permite, por exemplo, que as novas diretrizes incorporadas ao processo pela organização sejam trazidas para dentro da ferramenta e possam ser consultadas por todos os usuários.

6.2.3 Requisitos identificados ao longo do projeto

- **melhoria do desenho da camada de persistência:** o desenho da camada de persistência acabou ficando pouco robusto, devido ao curto período de tempo para sua elaboração e ao foco dado à simplicidade em sua implementação;
- **geração de relatórios de acompanhamento de tarefas do projeto:** tais relatórios seriam de grande importância para manter informados tanto os desenvolvedores quanto os gerentes das equipes a respeito do andamento do projeto;
- **geração de relatórios de distribuição de tarefas por membro da equipe:** este tipo de relatório também é muito interessante para facilitar a gerência do projeto e o esclarecimento de dúvidas de qualquer membro da equipe em relação à determinada tarefa.

Bibliografia

- [Ambler97] Scott W. Ambler. *An Object-Oriented Visual Glossary*. AmbySoft, 1997.
- [Ambler99] Scott W. Ambler. *The Design of a Robust Persistence Layer for Relational Databases*. AmbySoft, 1999.
- [BFPUG] *Brazilian Function Point Users Group*. <http://www.bfpug.com.br>.
- [Blaha+98] Michael Blaha and William Premerlani. *Implementing UML Models with Relational Databases*. Rose Architect – Rational Software Corporation, 1998.
- [Boehm00] Barry W. Boehm et al. *Software Cost Estimation with Cocomo II*. Prentice-Hall, Upper Saddle River – NJ, 2000.
- [Booch91] Grady Booch. *Object-Oriented Design*. Benjamin/Cummings, 1991.
- [Booch94] Grady Booch. *Object-Oriented Analysis and Design with Applications 2nd*. Ed. Benjamin/Cummings, Redwood City – CA, 1994.
- [Booch96] Grady Booch. *Object Solutions: Managing the Object-Oriented Project*. Addison-Wesley, Reading –MA, 1996.
- [Booch+99] Grady Booch, Ivar Jacobson and James Rumbaugh. *The Unified Modeling Language User Guide*. Addison-Wesley, Reading –MA, 1999.
- [Coad+99] Peter Coad and Mark Mayfield. *Java Design: Building Better Apps and Applets 2nd*. Ed. Prentice-Hall, 1999.
- [Eckel00] Bruce Eckel. *Thinking in Java*. Prentice-Hall, , Upper Saddle River – NJ, 2000.
- [Delphi98] Borland International. *Borland Delphi v.4.0 – Documentação*. Borland Press, 1998.
- [Fowler+97] Martin Fowler and Kendall Scott. *UML Distilled: Applying the Standard Object Modeling Language*. Addison-Wesley Publishing Company, 1997.
- [Fowler00] Amy Fowler. *A Swing Architecture Overview – The Inside Story on JFC Component Design*. <http://java.sun.com/products/jfc/articles/architecture/index.html>, 2000.
- [Gamma+95] Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995.
- [Grand98] Mark Grand. *Patterns in Java: a catalog of reusable design patterns illustrated with UML*. Wiley, 1998.
- [Humphrey95] Watts S. Humphrey. *A Discipline for Software Engineering*. Addison-Wesley, Reading, MA, 1995.
- [Humphrey99] Watts S. Humphrey. *Introduction to the Team Software Process*. Addison-Wesley, Reading, MA, 1999.
- [IDEF1x00] *IDEF1x Overview*. Knowledge Based Systems, Inc. <http://www.idef.com/idef1x.html>, 2000.
- [IEEE94] IEEE. *IEEE Standards Collection – Software Engineering*. IEEE, New York – NY, 1994.

- [Jacobson94] Ivar Jacobson. *Object-Oriented Software Engineering*. Addison-Wesley, Reading – MA, 1994.
- [Jacobson+94a] Ivar Jacobson, Maria Ericsson and Agneta Jacobson. *The Object Advantage: Business Process Reengineering with Object Technology*. Addison-Wesley, Reading – MA, 1994.
- [Jacobson97] Ivar Jacobson. *The Objectory Process*. Rational Software Corporation, 1997.
- [Jacobson+97] Ivar Jacobson, Martin Griss and Patrik Jonsson. *Software Reuse: Architecture, Process and Organization for Business Success*. Addison-Wesley, Reading –MA, 1997.
- [Jacobson+99] Ivar Jacobson, James Rumbaugh and Grady Booch. *Unified Software Development Process*. Addison-Wesley, Reading –MA, 1999.
- [JBuilder00] Documentação do produto *JBuilder 4 Foundation*. <http://www.borland.com/jbuilder/foundation> , 2000.
- [Jones94] Capers Jones. *Assessment and Control of Software Risks*. Yourdon Press - Prentice-Hall, Upper Saddle River - NJ, 1994.
- [Jtest00] Documentação do produto *Jtest v.3.2*. <http://www.parasoft.com/products/jtest>.
- [McClure89] Carma McClure. *CASE is Software Automation*. N.J.: Prentice Hall, 1989.
- [McConnell96] Steve McConnell. *Rapid Development*. Microsoft Press, 1996.
- [Paula+98a] Wilson de Pádua Paula Filho e Cláudio Ricardo Guimarães Sant’Ana. *Manual de Engenharia de Produtos de Software - Parte I: Recomendações*. RT – DCC – 008/1998.
- [Paula+98b] Wilson de Pádua Paula Filho e Cláudio Ricardo Guimarães Sant’Ana. *Manual de Engenharia de Produtos de Software - Parte II: Padrões e Modelos*. RT – DCC – 009/1998.
- [Paula+98c] Wilson de Pádua Paula Filho e Cláudio Ricardo Guimarães Sant’Ana. *Manual de Engenharia de Processos de Software - Parte I: Políticas*. RT – DCC – 015/1998.
- [Paula+98d] Wilson de Pádua Paula Filho e Cláudio Ricardo Guimarães Sant’Ana. *Manual de Engenharia de Processos de Software - Parte II: Padrões e Modelos*. RT – DCC – 016/1998.
- [Paula01] Wilson de Pádua Paula Filho. *Engenharia de Software: Fundamentos, Métodos e Padrões*. LTC Editora. Rio de Janeiro – RJ, 2001.
- [Paulk+93] Mark C. Paulk, Bill Curtis, Mary Beth Chrissis and Charles V. Weber. *Capability Maturity Model for Software, Version 1.1*. CMU/SEI-93-TR-24. Software Engineering Institute, Pittsburgh – PA, Feb. 1993.
- [Paulk+93a] Mark C. Paulk, Charles V. Weber, Suzanne M. Garcia, Mary Beth Chrissis and Marilyn Bush. *Key Practices of the Capability Maturity Model, Version 1.1*. CMU/SEI-93-TR-25. Software Engineering Institute, Pittsburgh – PA, Feb. 1993
- [Paulk+95] Mark C. Paulk, Charles V. Weber, Bill Curtiss and Mary Beth Chrissis. *The Capability Maturity Model: Guidelines for Improving the Software Process*. Addison-Wesley, Reading –MA, 1995.
- [Quatrani98] Terry Quatrani. *Visual Modeling with Rational Rose and UML*. Addison-Wesley, Reading –MA, 1998.
- [RationalSuite00] Documentação do produto *Rational Suite Enterprise v 2000.02.10*.

- <http://www.rational.com/products/rs>, 2000.
- [Rumbaugh91] James Rumbaugh, et al. *Object-Oriented Modeling and Design*. Ernst & Young, 1991.
- [Rumbaugh+99] James Rumbaugh, Ivar Jacobson and Grady Booch. *Unified Modeling Language Reference Manual*. Addison-Wesley, Reading –MA, 1999.
- [RUP00] Documentação do produto *Rational Unified Process v.2000.02.10*.
<http://www.rational.com/products/rup>, 2000.
- [SourceSafe96] Documentação do produto *Microsoft Visual SourceSafe v.5.0*.
<http://msdn.microsoft.com/ssafe>, 1996.
- [Together00] Documentação do produto *Together v.4.1*. <http://www.togethersoft.com/together>, 2000.
- [Travis01] Greg Travis. *The Persistent Hashtable: A Quick-and-Dirty Database*.
<http://www.gamelan.com>, 2001.
- [UMLDM00] *The UML and Data Modeling*. Rational Software Corporation.
<http://www.rational.com/products/whitepapers/101516.jsp>, 2000.