

UNIVERSIDADE FEDERAL DE ITAJUBÁ - UNIFEI

INSTITUTO DE ENGENHARIA ELÉTRICA - IEE

DEPARTAMENTO DE ELETRÔNICA - DON



Um sistema de admissão acadêmica para uma universidade

MESTRANDA: Lilia Ramos Carneiro.

ORIENTADOR: PROF. DR. Otávio Augusto Salgado Carpinteiro.

Itajubá – MG

Março 2003.

Um sistema de admissão acadêmica para uma universidade

Dissertação submetida a **UNIFEI**
para obtenção do Título de Mestre
em Engenharia Elétrica.

Área de Concentração:
Automação de Sistemas.

Mestranda:
Lilia Ramos Carneiro.

ORIENTADOR: PROF. DR. OTÁVIO AUGUSTO SALGADO CARPINTEIRO.

Itajubá – MG

2003.

DEDICATÓRIA

*À minha querida mãe (in memoriam), que apesar da
ausência, deixou sua marca de luta e persistência e
permanece viva nos momentos mais importantes
de minha vida;*

AGRADECIMENTOS

A Deus, pela possibilidade de realizar este trabalho e tê-lo concluído.

Com muito carinho, um agradecimento especial a os meus pais, Laercio e Gizélia, que me educaram com muita dignidade para enfrentar todos os desafios da vida e chegar onde eu cheguei.

Um agradecimento especial a minha amiga Vera, que foi a que mais me orientou deste o começo do mestrado, dando todo seu apoio e conhecimento.

Agradeço a minha irmã Fernanda e meus sobrinhos , pelo carinho, apoio e incentivo para realização deste trabalho.

Um agradecimento especial ao Prof. Pinheiro pela prestativa colaboração, ajuda e estímulos, sempre presentes.

Ao Prof. . Jair e o pessoal do UNIVERSITAS que deu apoio e cedeu o laboratório para realização deste trabalho.

Agradeço ao meu amigo, Agnaldo, que me socorreu nos momentos finais da dissertação, adicionando conceitos e sugerindo novas idéias para o fechamento deste trabalho.

SUMÁRIO

ÍNDICE DE FIGURAS.....	VI
ÍNDICE DE TABELAS	VII
LISTA DE ABRUVIATURAS.....	VIII
RESUMO	IX
ABSTRACT.....	X
CAPÍTULO 1 - INTRODUÇÃO.....	1
1.1 CONSIDERAÇÕES INICIAIS.....	1
1.2 APRESENTAÇÃO DO PROBLEMA	2
1.3 OBJETIVOS DO TRABALHO.....	2
1.4 IMPORTÂNCIA DO ESTUDO	3
1.5 JUSTIFICATIVA	3
1.6 DESENVOLVIMENTO DO TRABALHO	3
CAPÍTULO 2 - REVISÃO BIBLIOGRÁFICA.....	4
2.1 GESTÃO DA INFORMAÇÃO NAS INSTITUIÇÕES DE ENSINO.....	4
2.1.1 Estrutura organizacional universitária.....	4
2.1.2 O papel da informação na IES.....	6
2.1.3 A IES e o Marketing integrado.....	7
2.1.4 Modelo de gestão nas IES.....	8
2.1.5 Elaborando o perfil futuro da Instituição	9
2.2 TÓPICOS EM SOFTWARE E ENGENHARIA DE SOFTWARE.....	10
2.2.1 Processo de desenvolvimento do sistema.....	11
2.2.2 Modelos de Ciclo de Vida	12
2.2.3 Modelo unificado de desenvolvimento de software.....	16
2.2.4 Orientação a objeto.....	21
2.2.5 UML.....	24
2.2.6 Tecnologias de desenvolvimento para Web	32
2.2.7 Sistema gerenciador de banco de dados - SGBD.....	33
2.2.8 Linguagem de Programação.....	34
CAPÍTULO 3 - SISTEMA PROPOSTO.....	36
3.1 INTRODUÇÃO	36
3.2 FASE INICIAL – CONCEPÇÃO	36
3.2.1 Coleta de informações - entrevistas	37
3.3 PROPOSTA DE DESENVOLVIMENTO.....	39
3.3.1 Introdução.....	39
3.3.2 Descrição do sistema atual.....	39
3.3.3 Expectativa com o processo de automação.....	40
3.3.4 Descrição do sistema proposto	40
3.3.5 Benefícios	41
3.3.6 Viabilidade de implementação.....	41
3.3.7 Relação do sistema com o ambiente	42
3.4 FASE DE ELABORAÇÃO	42
3.4.1 Introdução.....	42
3.4.2 Escopo do produto:.....	43
3.4.3 Missão do produto.....	43
3.4.4 Limites do produto	43
3.4.5 Especificação do sistema	43
3.4.6 Requisitos funcionais do sistema.....	44
3.4.7 Descrição dos atores e use cases	46

3.4.8	<i>Detalhamento dos use cases</i>	46
3.4.9	<i>Levantamento de interface com usuário – Módulo intranet</i>	50
3.4.10	<i>Levantamento de interface com usuário – Módulo Internet</i>	52
3.4.11	<i>Interfaces do software</i>	53
3.4.12	<i>Desenvolvimento do Diagrama de Seqüência</i>	63
3.4.13	<i>Requisitos não funcionais</i>	66
3.5	FASE DA CONSTRUÇÃO	66
3.5.1	<i>Projeto da arquitetura do ambiente</i>	66
3.5.2	<i>Desenvolvimento do diagrama de colaboração – Intranet</i>	68
3.5.3	<i>Subsistemas e Componentes</i>	69
3.5.4	<i>Modelo de implementação</i>	73
3.5.5	<i>Plano de testes</i>	75
CAPÍTULO 4 - RESULTADOS OBTIDOS		78
4.1	INTRODUÇÃO	78
4.2	RESULTADOS DA AVALIAÇÃO	80
4.2.1	<i>Avaliação dos funcionários em relação ao Sistema de Admissão Acadêmica</i>	80
4.2.2	<i>Avaliação dos candidatos em relação ao sistema de inscrição on-line</i>	84
CAPÍTULO 5 - CONCLUSÕES		85
5.1	CONCLUSÃO	85
5.2	SUGESTÕES PARA PRÓXIMOS TRABALHOS	86
CAPÍTULO 6 - REFERÊNCIAS BIBLIOGRÁFICAS		87

INDICE DE FIGURAS

Figura 1 -A IES e o ambiente externo. _____	5
Figura 2 - Modelo de gestão nas universidades. _____	8
Figura 3 - Ciclo de vida clássico. _____	12
Figura 4 - Modelo de Prototipagem. _____	14
Figura 5 - Modelo incremental. _____	15
Figura 6 - Modelo Espiral. _____	16
Figura 7 - Modelo RUP _____	17
Figura 8 - Fases e workflows do processo RUP. _____	18
Figura 9 - Visões do sistema. _____	25
Figura 10 - Exemplo de diagrama de classe. _____	27
Figura 11 - Exemplo de diagrama de objetos. _____	28
Figura 12 - Exemplo de diagrama de componentes. _____	28
Figura 13 - Exemplo de diagrama de implantação. _____	29
Figura 14 - Exemplo de diagrama use case. _____	29
Figura 15 - Exemplo de diagrama de seqüência. _____	30
Figura 16 - Exemplo de diagrama de colaboração. _____	30
Figura 17 - Exemplo de diagrama de estado. _____	31
Figura 18 - Exemplo de diagrama de atividade. _____	32
Figura 19 - Diagrama de contexto. _____	47
Figura 20 - Use-Case – Controlar candidatos. _____	47
Figura 21 - Use-Case – Controle de candidatos _____	49
Figura 22 - Tela Login. _____	53
Figura 23 - Tela: Principal. _____	54
Figura 24 - Tela: Áreas. _____	55
Figura 25 - Tela: Inscrição de candidato. _____	56
Figura 26 - Tela: Incluir Critérios. _____	57
Figura 27 - Tela: Critérios cadastrados. _____	58
Figura 28 - Tela: Classificação de candidatos – colocação. _____	59
Figura 29 - Tela: Inicial. _____	60
Figura 30 - Tela: Inscrição –Opção. _____	60
Figura 31 - Tela: Inscrição – Ficha Cadastral. _____	61
Figura 32 - Tela: Inscrição – Questionário. _____	61
Figura 33 - Tela: Classificação. _____	62
Figura 34 - Tela: Classificação por área. _____	63
Figura 35 - Diagrama de seqüência – Controle de candidatos. _____	64
Figura 36 - Diagrama de seqüência – confirmar inscrição. _____	65
Figura 37 - Diagrama de seqüência – Inscrição do candidato. _____	65
Figura 38 - Diagrama de colaboração. _____	68
Figura 39 - Subsistemas e componentes _____	69
Figura 40 - Diagrama de interface – Intranet _____	70
Figura 41 - Diagrama de Interface - Internet _____	71
Figura 42 - Diagrama de dados persistentes. _____	72
Figura 43 - Diagrama de implantação. _____	73
Figura 44 - Diagrama de componentes _____	74
Figura 45 - Usabilidade do Sistema _____	81
Figura 46 - Funcionalidade do Sistema. _____	82
Figura 47 - Manutenibilidade do Sistema. _____	82
Figura 48 - Confiabilidade do Sistema. _____	83
Figura 49 - Eficiência do Sistema. _____	83
Figura 50 - Avaliação do Sistema on-line. _____	84

INDICE DE TABELAS

Tabela 1 - Formulário de perguntas para especificação de requisitos. _____	39
Tabela 2 - Formulário de respostas para especificação de requisitos. _____	39
Tabela 3 - Cronograma do projeto. _____	43
Tabela 4 - Requisitos funcionais do sistema. _____	46
Tabela 5 - Cartão cenário use case: Controlar candidatos. _____	49
Tabela 6 - Cartão cenário use case: Controlar candidatos. _____	49
Tabela 7 - Cartão Cenário Use case: Controlar classificação. _____	50
Tabela 8 - Cartão cenário use case: Imprimir candidatos excedentes. _____	51
Tabela 9 - Levantamento de interface com usuário-intranet. _____	52
Tabela 10 - Levantamento da interface com usuário - (Internet). _____	53
Tabela 11 - Requisitos de arquitetura – (Internet). _____	68
Tabela 12 - Requisitos de arquitetura – (Intranet). _____	68
Tabela 13 - Relações dos use cases testados. _____	78
Tabela 14 - Procedimentos de teste. _____	78
Tabela 15 - Critérios de completeza e sucesso. _____	79
Tabela 16 - Procedimento de teste: Inscrever candidato. _____	79
Tabela 17 - Procedimento de teste: Consultar inscrição _____	82
Tabela 18 - Resultado obtido - funcionários do setor administrativo. _____	83
Tabela 19 - Características dos requisitos não funcionais. _____	83

LISTA DE ABRIVIATURAS

IES	Instituição de Ensino Superior
Universitas	Centro Universitário de Itajubá
AOO	Análise Orientada a objetos
OMT	Técnica de Modelagem de Objetos.
OOSE	Object-Oriented Software Engineering.
BD	Banco de Dados.
CASE	Computer-Aided Software Engineering
CGI	Common Gateway Interface.
PERL	Practical Extraction and Report Language.
HTML	Hypertext Markup Language.
DRA	Diretório de Registro Acadêmico.
HTTP	Hipertext Transfer Protocol.
ODBC	Open Database Connectivity.
OO	Orientação a Objetos.
OLE	Object Linking and Embeldding.
PHP	Personal Home Page Tools.
RUP	Rational Unified Process.
SGBD	Sistema Gerenciador de Banco de Dados.
SQL	Structured Query Language.
UML	Unified Modeling Language.
VB	Visual Basic.
Web	World Wide Web.

RESUMO

O fator mais relevante no sistema educacional moderno é o ser humano. Por essa razão, as universidades precisam dinamizar-se e devem estar preparadas para atender à urgência das necessidades educacionais, além de ter como premissa o aprimoramento constante do corpo docente e a preocupação com o produto final - o conhecimento que ministra.

O processo de admissão dos candidatos e a gerência das informações constantes da vida acadêmica requerem uma dinâmica na manipulação e organização de informações. O Centro Universitário de Itajubá - Universitas pode ser citado como uma instituição que passa por problemas diversos iniciados com o processo seletivo não devidamente informatizado. É nesse contexto que se insere esse trabalho cuja proposta é desenvolver uma ferramenta computacional que possibilite o controle de todas as atividades do processo seletivo com o objetivo de torná-lo mais ágil e flexível. Isso implicará na melhoria das condições de trabalho, com repercussão positiva para a instituição e para a população de usuários.

O modelo de sistema abordado neste documento foi desenvolvido tendo por base conceitos de gestão acadêmica, engenharia de software, técnicas de orientação a objeto e Metodologia Unified Modeling Language - UML. O sistema foi implantado e testado no Universitas. Os testes, resultados e as principais conclusões desse trabalho são apresentados.

PALAVRAS - CHAVE

Engenharia de software, Técnicas de orientação a objeto, Sistema de admissão acadêmica, Gestão acadêmica, Unified Modeling Language.

ABSTRACT

The most relevant factor in the modern education system is the human being. For this reason, universities should be kept very well organized and prepared to meet the recent education changes. Moreover, an incessant concern with the improvement of the quality of their services is mandatory.

The process of candidates' admission (which involves a considerable amount of data) and the management of information related to academic issues calls for a very reliable tool in order to keep that database as organized and updated as possible. Universitas (a private university in Itajubá) can be quoted as an institution that experiences several problems related to the process of admission due to the fact that its process still depends on some hand-made procedures. Therefore, the main goal of this work is to develop a computational tool that automates the whole process of admission.

The software package was developed based on concepts of academic administration, software engineering, object-oriented techniques and Unified Modeling Language - UML. It has been installed and tested at Universitas. Its evaluation process and the main conclusions of this work can be found herein.

Keywords

Software Engineering, Oriented-Object Technical, system of academic administration, Rup and Unified Modeling Language.

Capítulo 1 Introdução

Este capítulo apresenta as dificuldades e as justificativas que motivaram o desenvolvimento deste trabalho, assim como, o objetivo estabelecido.

1.1 CONSIDERAÇÕES INICIAIS

Se o homem é agente elaborador da economia moderna, o fator mais relevante é o capital humano. Dessa forma, o conhecimento passa a ser o grande diferencial. Os países que mais se desenvolveram foram aqueles que promoveram o crescimento humano de maneira mais eficaz, baseado na busca e no aperfeiçoamento dos conhecimentos. Tradicionalmente, as universidades têm sido as grandes propulsoras de material humano gerador do conhecimento.

A escassez de recursos, de forma geral, é uma das principais dificuldades enfrentadas, tanto para as universidades públicas quanto para as privadas. Ainda, pela responsabilidade social das instituições, observa-se a necessidade de um maior envolvimento da comunidade para promover o fortalecimento da Instituição como uma organização efetiva capaz de se manter produtiva e com a qualidade exigida.

As universidades devem estar preparadas para atender às recentes mudanças nos sistemas educacionais e estarem constantemente preocupadas com a melhoria na qualidade do seu atendimento. Verifica-se a necessidade de uma administração universitária competente, preocupada com as alterações do ambiente e disposta a implantar as mudanças necessárias para alcançar os objetivos da Instituição.

O processo de admissão de candidatos administra uma quantidade significativa de informações e o bom gerenciamento deste processo é fundamental para a manutenção e expansão da Instituição. Geralmente, o processo seletivo é realizado de maneira precária, fazendo com que os serviços do setor administrativo se tornem lentos, o que dificulta o atendimento de uma grande quantidade de candidatos que desejam realizar suas inscrições.

O Universitas passa pelos diversos problemas relacionados ao processo seletivo, pois conta com um sistema parcialmente informatizado. Mas o Universitas optou pela adoção de uma ferramenta computacional, que possibilite o controle de todas as atividades do processo seletivo.

1.2 APRESENTAÇÃO DO PROBLEMA

Apesar do esforço e da evolução dos processos administrativos acadêmicos, as instituições têm demonstrado uma dificuldade em armazenar com mais eficácia as informações das quais dependerão da avaliação do seu desempenho como organização e o cumprimento das metas traçadas frente às exigências da realidade. A correção dos rumos organizacionais depende das decisões tomadas que se baseiam na precisão e agilidade com que as informações compiladas são utilizadas.

Até o momento, a Instituição não possui uma administração completamente informatizada. Isto indica os principais pontos que deverão ser atacados para o desenvolvimento de um novo sistema mais robusto.

1.3 OBJETIVOS DO TRABALHO

O sistema proposto deverá permitir, uma vez implantado, a rápida consistência dos dados dos candidatos inscritos no processo seletivo, possibilitando a agilização de todos os processos envolvidos na inscrição.

Esse sistema deverá contemplar os seguintes objetivos:

- Conceber um novo sistema de classificação dos alunos inscritos no processo seletivo;
- Agilizar o serviço dos funcionários do setor administrativo;
- Otimizar o processo de chamadas;
- Agilizar os processos da secretaria, diminuindo o custo;
- Reduzir as despesas operacionais através do controle de produtividade e da avaliação dos serviços realizados;
- Reduzir o tempo médio de atendimento e de execução dos serviços operacionais;
- Minimizar o número de reclamações por parte dos alunos em consequência das melhorias provenientes da implantação do sistema proposto;
- Possibilitar a integridade das informações;
- Possibilitar segurança dos dados armazenados;
- Diminuir o tempo de emissão de documentos e relatórios (listagens referentes ao processo seletivo).

1.4 IMPORTÂNCIA DO ESTUDO

Um dos problemas preocupantes em qualquer IES é a necessidade de estabelecer um sistema de administração adequado e racional que garanta uma organização flexível para responder as mudanças requisitadas, definir a missão da Instituição, estipular metas e satisfazer as expectativas pessoais dos membros da comunidade universitária.

Através do desenvolvimento deste trabalho, será possível examinar a aplicação do planejamento estratégico no Universitas revelando os pontos positivos e negativos do processo de admissão.

Pode-se dizer, ainda, que a importância do presente estudo está na necessidade de implementar um novo sistema que agilize o processo seletivo.

1.5 JUSTIFICATIVA

Apesar de existirem no mercado diversos softwares disponíveis nesta área, apenas alguns módulos atendiam às necessidades específicas do Universitas. Portanto, o Universitas optou pelo desenvolvimento de um modelo totalmente informatizado, que atendesse às suas próprias necessidades. Um outro fator considerado foi a necessidade de se obter um sistema que pudesse ser integrado aos outros módulos existentes na Instituição.

1.6 DESENVOLVIMENTO DO TRABALHO

Uma vez descritos os objetivos e as justificativas deste estudo, busca-se detalhar o desenvolvimento do trabalho nos capítulos subseqüentes.

- No Capítulo 2, segue uma revisão bibliográfica dos conceitos das diversas tecnologias utilizadas para o desenvolvimento do sistema proposto.
- No Capítulo 3, é apresentado o desenvolvimento do sistema proposto para atender às necessidades da Instituição durante todo o processo seletivo.
- No Capítulo 4, são apresentados e discutidos os resultados obtidos durante o período de implantação do sistema.
- No Capítulo 5 encontram-se as conclusões obtidas da realização deste trabalho, juntamente com as principais contribuições e sugestões para trabalhos futuros.

Capítulo 2 Revisão Bibliográfica

Este capítulo apresenta uma revisão dos principais conceitos das diversas tecnologias utilizadas para desenvolver o sistema proposto.

2.1 GESTÃO DA INFORMAÇÃO NAS INSTITUIÇÕES DE ENSINO

O objetivo deste capítulo é fornecer uma visão teórica sobre os seguintes assuntos: estrutura organizacional universitária, o papel da informação no sistema universitário brasileiro, a IES e o marketing integrado e o modelo de gestão de informações na IES.

Os conceitos relacionados à engenharia de software são utilizados para justificar a importância de se utilizar as diversas metodologias para o desenvolvimento de um sistema com qualidade.

2.1.1 Estrutura organizacional universitária

O complexo mundo das organizações tem alterado os modelos clássicos de administração e mudado as formas antes não flexíveis e exatas, por formas imprecisas e flexíveis. Estruturas organizacionais, funções administrativas e hierarquia adquirem características de flexibilidade pouco conhecidas ou não praticadas [Finger, 1997]. A evolução do meio social, econômico e político, no qual as organizações encontram-se inseridas, são os fatores que provocam essas mudanças.

As mudanças que ocorrem no ambiente afetam também as organizações universitárias. Sendo assim, estas devem estar sensíveis a tais transformações. Anualmente e semestralmente os estudantes, que ingressam em uma IES, trazem consigo novas exigências, novos critérios e projetos, aos quais a universidade deve estar atenta.

A universidade tem um papel importante com a sociedade, cabendo a ela definir o produto mais adequado, bem como entender, em termos de quantidade e qualidade, as necessidades dessa sociedade, na qual se encontra inserida.

As organizações universitárias, apesar de terem finalidades específicas e objetivos diferentes, possuem semelhanças, pois podem ser administradas segundo os princípios e modelos propostos pelas teorias da administração [Finger, 1997]. Analisadas pelo ângulo da administração, uma indústria, uma loja, um museu e uma universidade possuem semelhanças.

Do ponto de vista administrativo, não há algo que individualize ou singularize essas organizações, pois são todas passíveis de serem administradas de forma semelhante. Entende-se por ambiente o conjunto de elementos que não pertencem à organização, conforme exemplifica a figura abaixo.

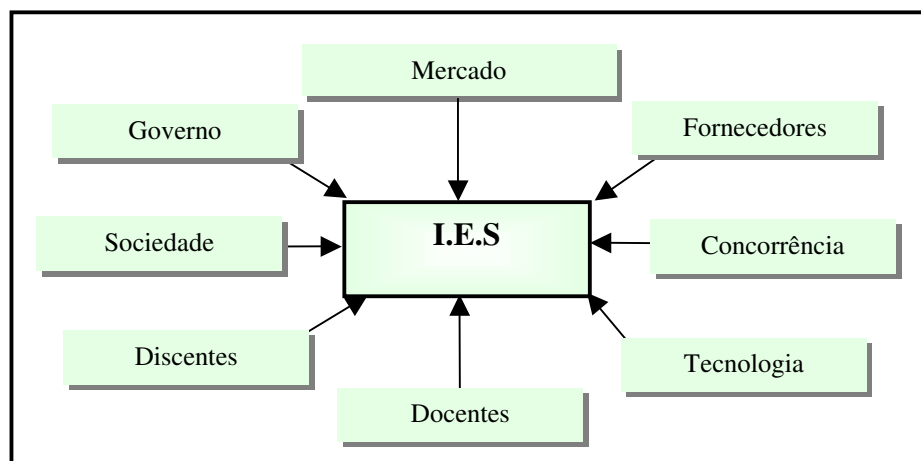


Figura 1 -A IES e o ambiente externo.

Existem muitos aspectos semelhantes entre as universidades e as empresas de qualquer área, com exceção da missão específica. As universidades se diferem das empresas nos aspectos relacionados a negócios como: valores e forma de contribuição com a sociedade [Finger, 1997].

A liderança e as decisões na organização universitária permitem rotulá-la como uma "anarquia organizada", onde existe pouca coordenação e controle, onde cada indivíduo é um autônomo tomador de decisões. Destacam-se como características especiais: [Finger, 1997]

- Ambigüidade de objetivos - os objetivos são vagos e difusos;
- Clientela - alunos com necessidades específicas e diversificadas;
- Tecnologia problemática - utilização de uma variedade de métodos e técnicas para atender a uma clientela diferenciada;
- Vulnerabilidade ao ambiente - sensibilidade aos fatores ambientais externos que poderá afetar a sistemática e padrões de administração universitária;
- A dificuldade de se mensurar os produtos resultantes da ação organizacional;
- A ausência de padrões de desempenho e compromisso com o resultado.

2.1.2 O papel da informação na IES

Com a globalização e a velocidade das inovações tecnológicas, a procura por informação se tornou um ponto comum de toda a sociedade. Surge, então o conceito de uma sociedade da informação, ou seja, possuir informação ou ao menos ter o privilégio garantido de acesso a ela [Alvim, 1998].

As universidades hoje precisam ter a preocupação em obter qualidade de informação, e não quantidade. O importante é obter a informação correta, ou melhor, adequada à determinada necessidade, em um tempo razoável e com um custo compatível. Os gestores universitários estão vivendo em um ambiente inteiramente novo, no qual vêm-se obrigados a tomar decisões, não só em maior número, mas de forma cada vez mais rápida.

A tecnologia nas IES se tornou o fator principal para torná-las competitivas e também constitui-se de um elemento diferenciador que garante a sobrevivência dessas instituições. Diante disso cada vez mais se faz necessário investir no processo de capacitação tecnológica, que precisa ser entendido como algo dinâmico e fazer parte das estratégias dos gestores.

Hoje, uma quantidade significativa de informações é manipulada por sistemas de informação e a cada dia se tornam mais importantes para a universidade. As grandes empresas estão gastando mais dinheiro do que nunca na obtenção de informação, mas muito dinheiro é perdido na construção de sistemas ineficientes, baseados em bancos de dados não apropriados ou de informações não relevantes [Borges, 1993].

Diante da afirmativa, deduz-se que não basta somente definir a informação necessária, mas também especificar quando e em que formato a informação deverá ser disponibilizada. Os usuários devem participar do processo de definição de necessidades e avaliar o valor que a informação terá como ferramenta para a tomada de decisão. Portanto, antes de se planejar e implementar sistemas de informação torna-se necessário conscientizar-se que o sistema deverá atender as necessidades do usuário.

Quando o usuário necessita de informações agregadas e transformadas para apoiar decisões gerenciais, ele possui um papel diferenciado. Nestes casos, o ciclo de vida é alterado pelo fato de o usuário desconhecer todas suas necessidades, no início do projeto. É o conhecimento gradual que faz com que ele informe ao projetista da ferramenta as informações importantes. Portanto, os dirigentes universitários devem acompanhar o desenvolvimento do

sistema e constantemente ajudar o projetista, para buscar melhores opções para o desenvolvimento do sistema.

2.1.3 A IES e o Marketing integrado

A função de marketing tem como objetivo o estabelecimento de trocas voluntárias de valor entre a organização e seus mercados. Do ponto de vista administrativo e do processo decisório, o principal impacto da adoção do conceito de marketing reside no deslocamento do pólo decisório da produção e vendas, para o mercado. A discussão que envolve o planejamento estratégico precisa passar pela ação de marketing no processo de integração com as outras áreas da organização. No caso das universidades, as estratégias de marketing apresentam-se hoje como parte do processo decisório, na prestação de serviços referente à educação [Toledo e Silva, 1992].

Em relação às estratégias de marketing nas IES, qualquer análise de tendência do ambiente é útil. Torna-se necessária uma avaliação dos pontos positivos e negativos que derivam da análise interna e externa do ambiente. Com estes pontos avaliados, são identificadas as ameaças e as oportunidades ambientais, definindo os objetivos organizacionais e determinando as estratégias que serão adotadas [Leite e Silveira, 1998].

Para alcançar os objetivos de marketing em uma IES, devem ser desenvolvidas algumas estratégias como: segmentação de mercado; marketing competitivo; desenvolvimento de estratégias de crescimento; ciclo de vida de produtos; entre outros.

Quando uma IES implementa procedimentos formais para reduzir ou desativar programas improdutivos, deve começar pesquisando as ofertas mais significantes, os novos programas que têm chance de sucesso, e a própria posição da universidade, no futuro. Estes procedimentos são, no entanto, uma parte das estratégias de marketing [Lamb Júnior e Cravens, 1987].

A ausência da prática do marketing por parte das IES brasileiras pode ser originária do próprio desconhecimento de seus dirigentes sobre os aspectos teóricos elementares de marketing. Uma visão das universidades sob a visão do marketing implica em uma mudança de mentalidade. Uma orientação da universidade para marketing significa transformá-la numa organização dinâmica, adequada ao seu tempo e ao seu público-alvo, contribuindo para diminuir a distância entre universidade e a sociedade [Ávila, 1990].

2.1.4 Modelo de gestão nas IES

O modelo descrito na figura abaixo procura demonstrar a delimitação da gestão nas universidades.

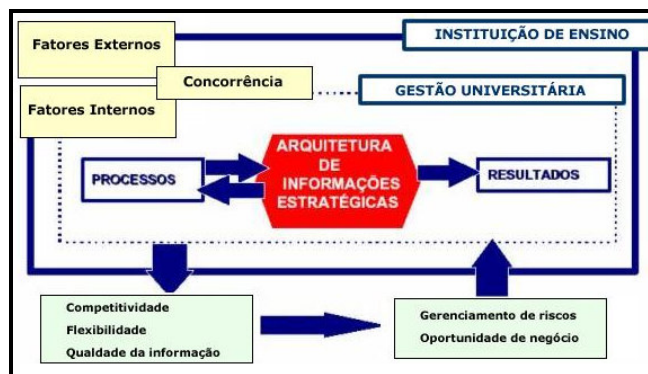


Figura 2 - Modelo de gestão nas universidades.

Para o esclarecimento da funcionalidade de cada elemento do diagrama, descreve-se a seguir, as características que tornam o ambiente ativo:

Informações externas – As informações pertinentes aos negócios são localizadas fora do âmbito da organização. O desafio passa a ser a identificação das principais informações que suportam decisões e sua estruturação para consumo do gestor universitário.

Informações internas – São as informações de dentro da organização (não-estruturais) e que influenciam no processo de tomada de decisões.

Processos internos - Estes são todos os processos operacionais, que fluem na gestão universitária: acadêmicos, administrativos, financeiros, extensão, pesquisa, biblioteca, suprimentos, recursos humanos, etc. Viabilizam as operações da organização, e são os que produzem informações gerenciais. Desta forma, facilitam a alimentação da arquitetura de informações estratégicas, que entende estes dados em formato de indicadores e sinalizam o desempenho da organização.

Resultados - Retratam as condições da organização nos seus diversos segmentos. Acompanhados e monitorados, a partir da arquitetura de informações estratégicas, deverá refletir o desempenho de: clientes, setores, concorrentes, recursos financeiros, humanos, infraestrutura, sociais, etc. A análise dos resultados suportada pela arquitetura de informações auxilia o dirigente universitário a tomar decisões baseadas em fatos e dados concretos e precisos.

Características gerenciais - O sistema de informação estratégico deverá fornecer as ferramentas para que o dirigente universitário possa interagir com o banco de informações e extrair dele os indicadores que detectem ou confirmem a existência destas novas condições de mercado, as áreas de preferências dos alunos em potencial, padrões de consumo, etc. Portanto, o processo de gestão universitária deve conduzir à organização para alcançar maior: competitividade, agilidade, flexibilidade e qualidade da informação nas suas decisões.

Ações gerenciais - O suporte da arquitetura da Instituição possibilitará o gerenciamento de oportunidades e riscos dos negócios. Assim, permite ao dirigente interagir na organização, acompanhando as ações estratégicas e tomando decisões baseadas em informações atualizadas e análises de tendências futuras.

Arquitetura de informações estratégicas - O tipo de informação a ser apresentada deverá oferecer a possibilidade de controle gerencial da universidade, permitindo que os administradores universitários possam comandar uma representação modelada da empresa.

Dinâmica do modelo - Possibilita ao dirigente acompanhar os resultados de sua Instituição e as tendências futuras de comportamento da mesma, frente às influências estabelecidas externa e internamente. O objetivo é tornar a gestão universitária mais competitiva, com agilidade, flexível às exigências de mercado e com qualidade de informação, para que as decisões tomadas proporcionem um gerenciamento com excelência e identificação de oportunidades de negócios continuamente.

2.1.5 Elaborando o perfil futuro da Instituição

Para elaborar o perfil estratégico da Instituição é necessário levar em consideração três principais elementos que refletem verdadeiramente a sua estratégia, sua direção e marketing: Produtos e serviços, mercado e clientes.

Para efetuar uma análise destas principais áreas, será imprescindível conhecer profundamente os seguintes aspectos:

- A natureza de seus produtos e serviços (a Instituição caracteriza-se pela prestação de serviços educacionais, que envolve não só resultados quantitativos, mas principalmente os qualitativos e os que não são mensuráveis);
- A natureza dos seus clientes (alunos - caracterização do ambiente cultural, social, expectativas, crenças e valores da comunidade);
- A natureza de seus segmentos de mercado (concorrência, novos produtos e serviços);

Assim, as perguntas mais importantes que a equipe de dirigentes deve saber responder, a fim de modelar a sua Instituição, são:

- Que produtos e serviços nós oferecemos?
- Que serviços nós não oferecemos?
- A que grupos de clientes, oferecemos nossos serviços? Mas, principalmente, a que grupo de clientes, não oferecemos os serviços?
- Quais as necessidades e expectativas dos clientes que estamos atendendo? Que necessidades e expectativas não atendemos ou nunca ousamos implementar?
- Quais produtos e serviços receberão maior ou menor ênfase no futuro?

Para modelar o perfil de uma Instituição, a equipe de dirigentes deve fazer as perguntas certas como resultado de um sólido processo de raciocínio estratégico. Este exercício possibilitará determinar a força propulsora que representa a principal estratégia de uma Instituição.

Nos últimos anos, a tendência das organizações é buscar uma melhora significativa no atendimento ao cliente. Toda organização deve ser sensível ao cliente. Isto significa dizer, que se os seus produtos e serviços não atendem a certas necessidades dos clientes sua empresa está predestinada ao fracasso.

As instituições não possuem uma maneira de estrutura organizacional, por isso não se pode afirmar que um modelo seja mais eficiente que outro. Cada um dos modelos existentes explica melhor a realidade de uma universidade, ou parte dela, mas universidade alguma reflete, de forma pura, qualquer modelo [Finger, 1997].

2.2 TÓPICOS EM SOFTWARE E ENGENHARIA DE SOFTWARE

Uma primeira definição de engenharia de software foi proposta por Fritz Bauer:

O estabelecimento e uso de sólidos princípios de engenharia para que se possa obter um software economicamente viável, que seja confiável e que funcione eficientemente em máquinas reais [Bauer, 1972].

Engenharia de software é a aplicação de uma abordagem sistemática e disciplinada para o desenvolvimento e operação de software [IEEE, 1993].

2.2.1 Processo de desenvolvimento do sistema

O processo de desenvolvimento do software é um conjunto completo de atividades necessárias para transformar os requisitos do cliente em um conjunto consistente de artefatos que representam um produto de software [Jacobson, 1999].

Definição: Focaliza o “*o que*”

Durante a fase de definição o engenheiro de software tenta identificar as informações que são processadas, as funções, o desempenho requisitado, o comportamento do sistema esperado, as interfaces estabelecidas, as restrições do projeto e os critérios de validação necessários para definir o sucesso do sistema. Os métodos aplicados durante esta fase, podem variar de acordo com o paradigma (ou combinação de paradigmas aplicados) [Pressman, 2001].

Desenvolvimento: Focaliza-se o “*como*”

Durante a fase do desenvolvimento será definido como os dados estarão estruturados, como as funções serão implementadas de acordo com a arquitetura do software, como as interfaces serão caracterizadas, como o projeto se transformará em uma linguagem de programação e como os testes serão desempenhados.

Manutenção: Concentra-se nas “*mudanças*” associadas à correção de erros, adaptações requisitadas e entre outras baseadas nos requisitos do cliente [Pressman, 2001]. O processo de desenvolvimento de software deve ser capaz de evoluir ao longo de vários anos e devendo a evolução acompanhar as inovações de tecnologias, ferramentas, pessoas e padrões organizacionais.

O processo de desenvolvimento de software necessita:

- Ordenar as atividades de cada equipe de trabalho.
- Indicar tarefas de cada pessoa e cada equipe como um todo.
- Especificar os artefatos a serem desenvolvidos.
- Estipular critérios de verificação da evolução das atividades e artefatos [Pressman, 1995].

No processo de desenvolvimento de um software, um conjunto de etapas deve ser definido, o qual pode ser denominado de paradigmas da engenharia de software ou ciclo de vida do software [Pressman, 1995].

2.2.2 Modelos de Ciclo de Vida

Um modelo de processo do software é uma visão das atividades que ocorrem durante o desenvolvimento de um software, definindo, assim, os estágios ou as fases que um produto de software atravessa ao ser desenvolvido. Este modelo possibilita um melhor entendimento das atividades básicas e características do software em desenvolvimento. Desta forma, um modelo de processo: descreve as principais fases do desenvolvimento de software; define as funções primárias que devem ser executadas durante as fases; auxilia os gerentes a acompanhar o progresso do projeto e fornece uma estrutura para a definição de um processo de desenvolvimento de software detalhado [Davis, 1997].

Muitos modelos de processo têm sido propostos na literatura e em sua maioria apresentam características complementares entre si, acrescentando novos conceitos que corrigem deficiências de outros. Abaixo são definidos alguns modelos de processo de software.

2.2.2.1 Modelo de ciclo de vida clássico - (Waterfall)

Este ciclo apresenta progressão seqüencial e linear do desenvolvimento. Isso implica que não são executadas atividades paralelas, e nenhuma atividade pode ser iniciada sem que a anterior seja concluída. Este paradigma é considerado o mais antigo e mais usado na engenharia de software [Pressman, 1995].

Vantagens: O paradigma pode ser considerado simples e fácil de seguir.

Desvantagens: Muitas vezes é difícil para o cliente declarar todas as exigências explicitamente. O ciclo de vida clássico exige isso e tem dificuldade de acomodar a incerteza natural que existe no começo de muitos projetos.

A figura abaixo fornece uma descrição visual do modelo.

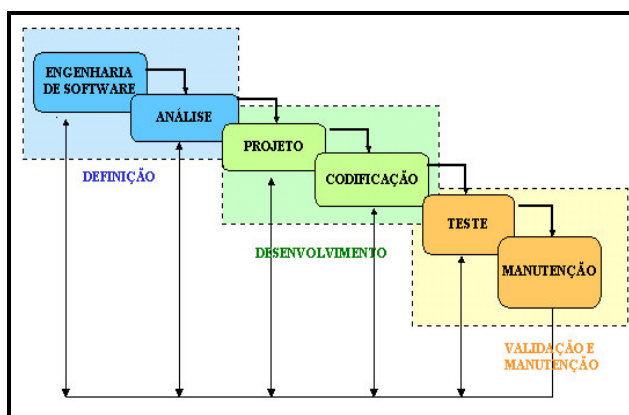


Figura 3 - Ciclo de vida clássico.

Na fase de análise de requisitos, são estabelecidos os requisitos do sistema a ser desenvolvido. Nesta fase, o engenheiro de software necessita compreender o domínio do problema no qual está trabalhando. A análise de requisitos começa ao se reconhecer que um problema necessita da tecnologia de informação como solução ou ao surgir uma idéia de um novo negócio ou de um novo sistema de informação na empresa. Esta fase é concluída ao se ter uma descrição completa do comportamento do software a ser documentado na especificação de requisitos.

A fase de projeto tem por objetivo traduzir os requisitos definidos em uma representação de software com detalhes suficientes para que possa ser implementado. Pode-se verificar que quanto mais detalhada for a fase de projeto, mais clara será a fase de construção.

Na fase de construção os programas são codificados, as bases de dados criadas e os módulos do software integrados. Uma vez que o código foi gerado, segue-se a fase de avaliação da qualidade do software.

Os testes concentram-se em validar os procedimentos lógicos internos do programa, garantindo que todos os comandos foram testados e que o comportamento funcional externo do sistema produz os resultados esperados para determinadas entradas de dados. Teste representa uma revisão final das fases de análise, projeto e construção. A intenção da realização de testes é achar erros e para que os testes sejam eficazes é necessário que estes sejam planejados com seus objetivos claramente definidos [Pressman, 1995].

2.2.2.2 Prototipagem

O modelo de prototipagem se baseia na utilização de um protótipo do sistema real, para auxiliar na determinação de requisito. Um protótipo deve ser de baixo custo e de rápida obtenção, para que possa ser avaliado. Para isto, uma determinada parte do sistema é desenvolvida com o mínimo de investimento, mas sem perder as características básicas, para ser analisada juntamente com o usuário [Pressman, 1995].

Vantagens: Os requisitos do cliente podem ser identificados com facilidade e rapidez; os clientes podem validar o protótipo na primeira fase e fornecer suas entradas e *feedback*; análise refinada e envolvimento dos usuários.

Desvantagens: Soluções menos importantes podem se tornar parte integrante do sistema; o cliente pode querer o protótipo precário e não o sistema otimizado.

A prototipagem é uma forma de desenvolvimento incremental e contém quatro tipos diferentes: Ilustrativo (telas); Simulado (acesso ao banco de dados); Funcional (subconjunto limitado); Evolucionário (começa pequeno e cresce). Os três primeiros tipos são construídos para atingir primeiramente os objetivos propostos, sendo considerados protótipos descartáveis. O protótipo evolucionário se tornará ao final um produto operacional.

Na figura abaixo observa-se a seqüência de eventos verificada no paradigma da prototipagem [Pressman,1995].

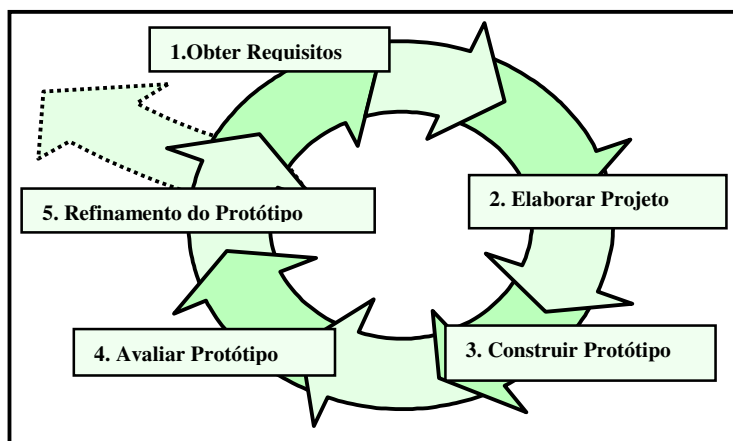


Figura 4 - Modelo de Prototipagem.

1- Obtenção dos Requisitos: projetista e cliente definem os objetivos gerais do software, identificam quais requisitos são conhecidos e as áreas que necessitam de definições adicionais.

2- Projeto Rápido: representação dos aspectos do software que são visíveis ao usuário (abordagens de entrada e formatos de saída)

3- Construção Protótipo: implementação rápida do projeto.

4- Avaliação do Protótipo: cliente e projetista avaliam o protótipo

5- Refinamento do Protótipo: cliente e projetista refinam os requisitos do software a ser desenvolvido.

6- Construção Produto: identificados os requisitos, o protótipo deve ser descartado e a versão de produção deve ser construída considerando os critérios de qualidade.

Ainda que possam ocorrer problemas, a prototipagem é um modelo de processo de software eficiente. O cliente e o projetista devem ambos concordar que o protótipo seja construído para servir como um mecanismo a fim de definir os requisitos [Pressman, 2001].

2.2.2.3 Modelo incremental

O modelo incremental corresponde à idéia de aumentar pouco-a-pouco a esfera do sistema, ou seja, alargar o sistema em sucessivos incrementos [Silva, 2001].

Este modelo combina a interação do modelo de prototipagem com a eficiência do modelo de ciclo de vida clássico [Sommerville, 2000].

O objetivo é trabalhar junto do usuário para descobrir seus requisitos, de maneira incremental, até que o produto final seja obtido. Este tipo de modelo é mais apropriado para sistemas pequenos e difere da prototipagem, pois a cada incremento produz uma versão operacional do software.

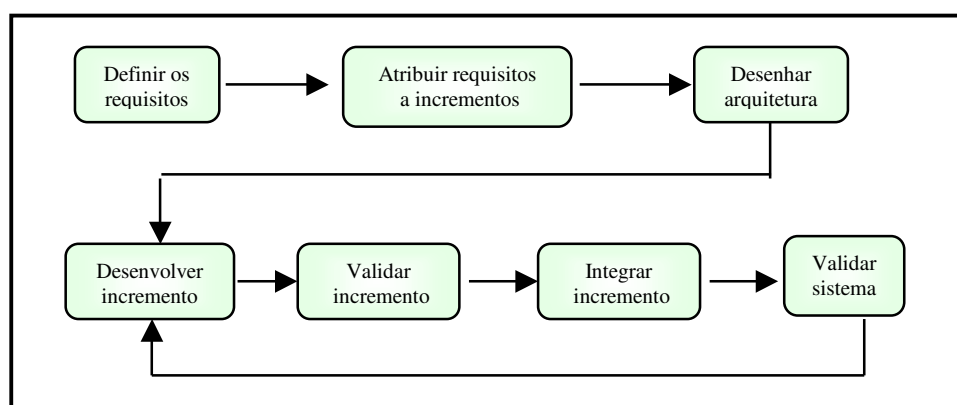


Figura 5 - Modelo incremental.

2.2.2.4 Modelo Espiral

É uma inovação que combina os modelos: de ciclo de vida clássico, incremental e prototipagem para ser usado em várias partes do desenvolvimento [Boehm, 1988].

O modelo, representado pela espiral, define quatro importantes atividades abaixo citadas:

- 1-Planejamento: determinação de requisitos, alternativas e restrições;
- 2-Análise de riscos: análise de alternativas e identificação dos riscos.
- 3-Engenharia: desenvolvimento do produto no nível seguinte (nível crescente de complexidade e de carga de trabalho à medida que se evolui no modelo espiral).
- 4-Avaliação feita pelo cliente: avaliação dos resultados da engenharia [Pressman, 1995].

Vantagens: Incorpora a abordagem de prototipagem e testes para cada etapa do ciclo clássico; Cada ciclo é encerrado por uma atividade de revisão, onde todos os produtos do ciclo são avaliados.

Desvantagens: Quanto mais voltas se criam no modelo espiral mais caro será o sistema, conforme é mostrado na figura abaixo:

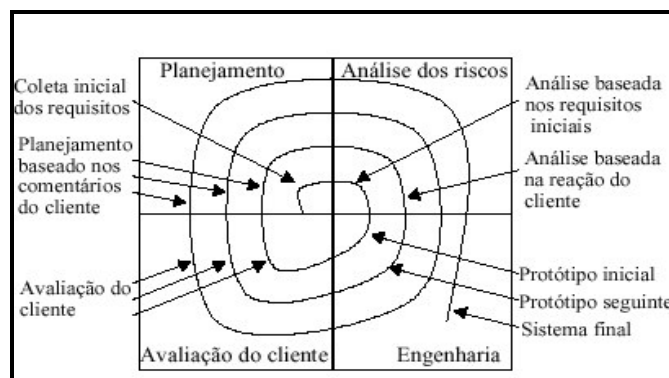


Figura 6 - Modelo Espiral.

2.2.3 Modelo unificado de desenvolvimento de software

É importante ressaltar que existem no mercado diversas metodologias de engenharia de software que criaram novos paradigmas, combinando e aproveitando os melhores conceitos de outras metodologias. Devemos destacar a metodologia Rational Unified Process - RUP da Rational, sendo uma das principais metodologias mais utilizadas atualmente no mundo, principalmente em conjunto com a ferramenta Computer-Aided Software Engineering - CASE, engenharia de software auxiliado por computador.

O RUP é um processo de desenvolvimento de software e seu objetivo é transformar os requisitos do usuário em sistema de software [Jacobson, et. 1999].

O RUP é dividido em ciclos, cada qual trabalhando na geração de uma nova versão do produto. O processo de desenvolvimento do RUP consiste em quatro fases:

Concepção: Define o escopo do projeto e a viabilidade econômica do projeto.

Elaboração: Define os requisitos e uma arquitetura estável, permitindo ainda eliminar os principais riscos.

Construção: Desenvolvimento do sistema, preparado para a transição para os usuários.

Transição: Implanta o sistema e a fase de treinamento dos usuários.

A característica principal do RUP é ser baseada em componentes, ou seja, o software desenvolvido é formado por componentes que se comunicam através de interfaces bem definidas. O padrão adotado para representação dos modelos é a UML.

Os aspectos que distinguem o processo unificado são capturados em três conceitos:

Dirigido por use case: Um use case é uma parte da funcionalidade do sistema que fornece ao usuário um resultado de valor [Jacobson, et. 1999]. Os use cases são utilizados para a captura e definição dos requisitos funcionais do sistema. São também utilizados para projetar os casos de teste.

Centrado em arquitetura: No RUP os use cases e a arquitetura são desenvolvidos em paralelo. A conceito de arquitetura engloba os aspectos mais relevantes, tanto estáticos, como dinâmicos do sistema.

Iterativo e Incremental: O RUP utiliza pequenos ciclos de projeto (mini-projetos) que correspondem a uma iteração e que resultam em um incremento do software.

2.2.3.1 Principais workflows do RUP

Cada fase do RUP é dividida em iterações e finalizada com um marco (milestone) bem definido, conforme mostra a figura abaixo:

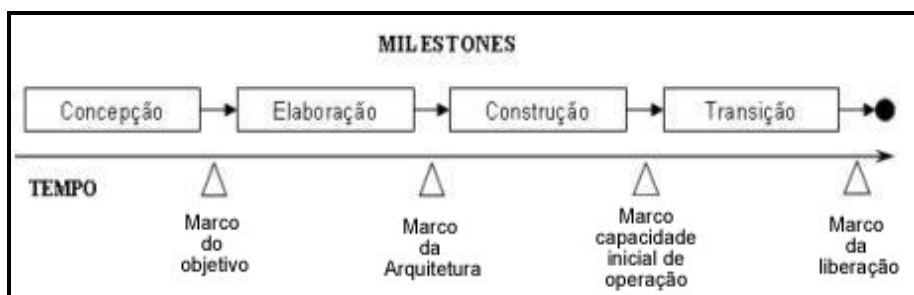


Figura 7 - Modelo RUP

Toda iteração é organizada em termos de workflows, isto é, conjunto de atividades realizadas por responsáveis que produzem artefatos.

O modelo de processo RUP fornece **seis workflows** de processo (perspectiva técnica) e **três workflows** de suporte (perspectiva gerencial), conforme mostra a figura a seguir:

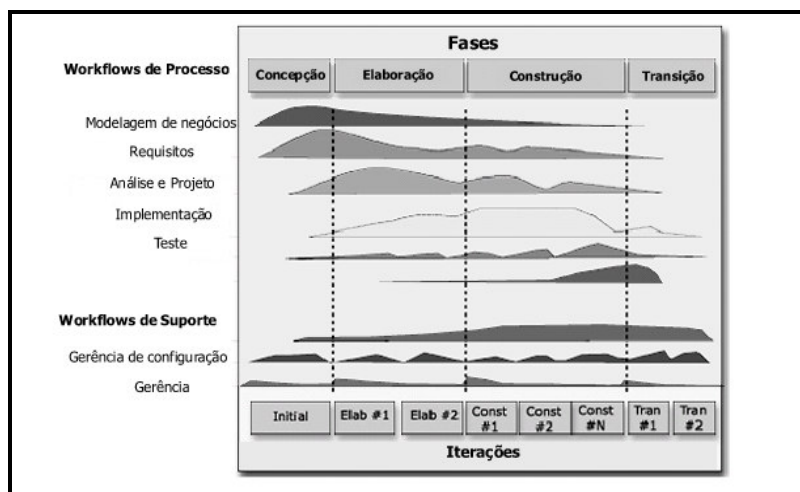


Figura 8 - Fases e workflows do processo RUP.

Os workflows definem um conjunto de atividades que produzem um resultado. Cada workflow requer, usualmente, a interação de diversos participantes. Os resultados produzem artefatos e, estes são qualquer informação produzida por um participante (modelos, documentos).

Workflows de processo: Definem a execução do processo, que são:

Modelagem do Negócio: Envolve o entendimento da estrutura e dinâmica da organização, garantindo que clientes, usuários e projetistas tenham a mesma visão da organização para a qual será feito o desenvolvimento.

As atividades relacionadas a esta fase são:

- Identificar as necessidades do usuário;
- definir o escopo do produto;
- levantar os requisitos preliminares;
- estimar custos e prazos;
- descrever a arquitetura ou modelo do sistema.

O artefato principal gerado neste workflow é:

- Proposta do desenvolvimento do sistema.

Requisitos: o fluxo de requisitos reúne as atividades que visam obter o enunciado completo, claro e preciso dos requisitos de um produto de software. Estes requisitos devem ser levantados pela equipe do projeto, em conjunto com representantes do cliente, usuários chaves

e outros especialistas da área de aplicação. O conjunto de técnicas empregadas para levantar, detalhar, documentar e validar os requisitos de um produto forma a Engenharia de Requisitos [Paula, 2001].

As atividades relacionadas a esta etapa são:

- Determinação do contexto: levanta os aspectos dos processos de negócio ou de um sistema maior, que sejam relevantes para a determinação dos requisitos do produto.
- Definição de requisitos funcionais: Os requisitos funcionais descrevem as funções que o produto deverá realizar em benefício dos usuários. Existem muitas maneiras de descrição destas funções. Cada função será descrita por um use case. A descrição dos fluxos dos use case define os detalhes dos requisitos funcionais [Paula, 2001].
- Definição de requisitos não funcionais: Os requisitos não funcionais incluem os requisitos de desempenho e outros atributos de qualidade do produto.
- Definição do layout sugerido: levanta os aspectos das interfaces do produto que os usuários consideram como requisitos. Normalmente, é feito um esboço das interfaces de usuário, levantado através de um protótipo executável. Também são detalhados as interfaces com outros sistemas e componentes de sistema [Paula, 2001].
- Revisão dos requisitos: determina se todos eles satisfazem aos critérios de qualidade de requisitos, e se a Especificação dos Requisitos do Software está clara e bem entendida por todas as partes interessadas.

O artefato principal gerado neste workflow é:

- Modelo use case que é a base de todo o processo de desenvolvimento, o que facilita a avaliação da concordância do sistema final com os requisitos iniciais para os quais foi construído,
- Protótipo da interface,
- Documentação dos requisitos,
- Especificação complementar (requisitos não-funcionais).

Análise e Projeto: O fluxo da análise possui os seguintes objetivos: verificar a qualidade dos requisitos obtidos através do fluxo de requisitos e detalhá-los o suficiente para que atinjam as necessidades dos projetistas.

As atividades relacionadas a etapa de análise são: identificação das classes, organização das classes, identificação dos relacionamentos, identificação dos atributos e realização dos use case.

Os artefatos principais gerados neste workflow são:

- Modelo de análise;
- um ou mais diagramas de classes, de seqüência e de colaboração;
- um conjunto de pacotes lógicos que particiona as classes de forma adequada;
- diagrama de pacotes lógicos.

O fluxo de projeto tem por objetivo definir uma estrutura implementável para um produto de software que atenda aos requisitos especificados. O desenho de um produto de software deve considerar os seguintes aspectos:

- atendimento dos requisitos não funcionais, como os requisitos de desempenho;
- definição detalhada de classes e outros elementos de modelos para a respectiva implementação;
- decomposição do produto em componentes cuja construção seja relativamente independente, de forma que possa eventualmente ser realizada por pessoas diferentes;
- definição adequada e rigorosa das interfaces entre os componentes do produto, minimizando os efeitos que os problemas em cada um desses possa trazer aos demais elementos;
- documentação das decisões do projeto, de forma que estas possam ser comunicadas e entendidas por quem vier a implementar e manter o produto;
- reutilização de componentes, mecanismos e outros artefatos, para aumentar a produtividade e confiabilidade [Paula,2001].

As atividades relacionadas à fase de projeto são: desenho arquitetônico, estudos de usabilidade, projeto das interfaces de usuário, definição dos dados persistentes, detalhamento das classes do projeto, realização dos use cases e projeto de liberações.

Os artefatos principais gerados neste workflow são:

- Diagrama de implantação.
- Diagrama de componentes.
- Documento de projeto do software.

Implementação: Envolve o desenvolvimento de código: classes, objetos, etc., teste de unidades e integração de subsistemas.

O artefato principal gerado neste workflow é:

- Modelo da implementação.

Teste: Envolve a verificação do sistema como um todo, com testes de integração e conformidade com os requisitos especificados.

O artefato principal gerado neste workflow é:

- Modelo de teste.

Workflows de suporte: Possibilitam a coordenação e gerência do processo, que são:

Gerência de projeto: Envolve o gerenciamento de riscos, planejamento e acompanhamento do projeto.

Gerência de configuração e mudanças: Envolve o gerenciamento dos artefatos gerados durante o desenvolvimento.

Configuração do ambiente: Envolve a organização do ambiente de trabalho para a equipe do projeto e a configuração do RUP para o projeto.

2.2.4 Orientação a objeto

Dizer que um software é orientado a objetos significa que ele é organizado como uma coleção de objetos separados, que incorporam tanto a estrutura como o comportamento dos dados [Furlan, 1998]. Um objeto é uma entidade que possui:

Atributos ou propriedades, que descrevem o estado de um objeto do mundo real;

Métodos ou ações que representam os processos associados ao objeto no mundo real;

Nome ou identificador que designa univocamente o objeto na representação.

Uma das principais vantagens da orientação a objeto é a possibilidade de se construir software que seja **reutilizável** (capacidade de reutilização do mesmo código para componentes de sistemas diferentes em várias aplicações diferentes), **robusto** (capacidade de produzir saídas corretas para todas as entradas recebidas) e **adaptável** (capacidade de se adaptar à mudanças em seu ambiente).

A Orientação a objetos trouxe vários novos conceitos ao desenvolvimento de software, como:

Abstração: Denota as características essenciais de um objeto que o distingue dos demais [Booch, 2000].

Encapsulamento: Os métodos em um objeto podem somente manipular diretamente os dados do próprio objeto. Esta propriedade denomina-se encapsulamento e é de grande importância na integração de objetos heterogêneos.

Objeto: Um objeto é simplesmente alguma coisa que faz sentido no contexto de uma aplicação [Rumbaugh, 1994]. É um elemento que podemos manipular, acompanhar seu comportamento, criar, destruir, etc.

Classe: É um tipo definido pelo usuário que contém o molde, a especificação para os objetos.

Atributo: Um atributo é uma propriedade nomeada de uma classe que descreve um intervalo e valores que as instâncias da propriedade podem apresentar, podendo a classe ter qualquer número de atributos ou mesmo nenhum atributo.

Operação: Uma operação é a implementação de um serviço que pode ser solicitado por algum objeto da classe para modificar o comportamento.

Método: São funções que estão associadas aos objetos. Suas chamadas são feitas através de mensagens. Um método pode estar relacionado a um evento.

Mensagem: São estímulos disparados contra um objeto para que ele responda com algum comportamento (método).

Evento: É um estímulo externo que pode estar relacionado a um objeto qualquer, este estímulo vai ser tratado pelo objeto através de um método.

Interface: É a visão externa de uma classe, objeto ou módulo, os quais enfatizam suas abstrações enquanto esconde suas estruturas e os segredos de seus comportamentos.

Generalização: É um relacionamento entre itens gerais (chamados superclasses ou classes-mãe) e tipos mais específicos desses itens (chamadas subclasses ou classes-filha). A generalização significa que os objetos da classe filha podem ser utilizados em qualquer local em que a classe mãe ocorra, mas não vice-versa [Booch, 2000].

Herança: O mecanismo pelo qual os elementos mais específicos incorporam a estrutura e o comportamento dos elementos mais gerais [Booch, 2000].

Polimorfismo: Trata-se da possibilidade que uma mesma operação possui de atuar de modos diferentes em classes diferentes. Isto é possível quando uma operação esteja declarada em

classes diferentes, porém com o mesmo nome, executando processamentos diferentes para atender os requisitos semânticos de sua classe.

2.2.4.1 Análise e Projeto Orientado a Objeto

Durante a fase de análise orientada a objetos - AOO a ênfase está em achar e descrever objetos ou conceitos de forma a representar entidades do mundo real e otimizar a compreensão do problema.

O objetivo da AOO é desenvolver um modelo que descreva como o software trabalha para satisfazer um conjunto de requisitos definidos pelo cliente. Para realização da AOO, o engenheiro de software deve seguir os seguintes passos:

- Tornar explícito os requisitos que forem especificados pelo cliente,
- Identificar cenários e use cases,
- Selecionar classes e objetos, baseados nos requisitos do sistema,
- Identificar os atributos e as operações para cada objeto do sistema,
- Organizar as classes em estruturas e hierarquias,
- Construir um modelo de relacionamento do projeto,
- Revisar os use cases do objeto ou cenários [Pressman, 2001].

Os principais benefícios em se utilizar AOO são:

- Maior compreensão de domínios de problemas;
- organização da análise e especificação usando os métodos que fazem parte do modo de pensar das pessoas (interação entre analista e especialistas);
- redução da distância entre as diferentes atividades de análise, tratando os atributos e os serviços como um todo;
- organização dos resultados com base em construções de domínios de problemas, permitindo a reutilização presente e futura. [Yourdon, 1992].

Durante a fase do Projeto Orientado a Objeto - POO, a ênfase está em achar objetos lógicos de software que poderão ser eventualmente implementados usando uma linguagem de programação orientada a objeto. Nesta fase é descrito como o software será implementado, podendo ser vista como um aprofundamento da análise caminhando em direção a implementação. É durante a fase de projeto que são definidos a estrutura geral, estilo do sistema e sua arquitetura [Rumbaugh, 1994].

2.2.4.2 Metodologias

Uma variedade de métodos para análise e projeto OO foram propostos e utilizados durante os anos 80 e 90. Esses métodos estabeleceram uma fundamentação para a notação de análise e projeto OO, que são [Pressman, 2001]:

Método de Rumbaugh: O método, também chamado de *Object Modeling Technique* - OMT, divide a análise e o projeto em três partes: análise, projeto do sistema e projeto dos objetos. Na descrição da OMT, o sistema é descrito de três pontos de vista diferentes: visão dos objetos (descreve o fluxo de informações), visão dinâmica (fluxo de controle) e funcional (especificação dos processos) [Rumbaugh, 1994].

Método de Booch: Este método definiu a noção de que um sistema é analisado a partir de um número de visões, onde cada uma é descrita por um número de modelos e diagramas. Este método trazia uma simbologia complexa de ser desenhada a mão, continha também o processo pelo qual, sistemas são analisados por macro e micro visões [Eriksson, 1998].

Método de Jacobson: Os métodos Object-Oriented Software Engineering - OOSE e o Objectory foram desenvolvidos baseados no mesmo ponto de vista formado por Jacobson. O método OOSE é a visão de Jacobson de um método orientado a objetos, já o Objectory é usado para a construção de sistemas. Ambos os métodos são baseados na utilização de use cases, que definem os requisitos iniciais do sistema vistos por um ator externo.

Cada um dos métodos descritos acima possui uma notação própria para representar os modelos, processos e ferramentas [Eriksson, 1998].

2.2.5 UML

Com o conhecimento na área de modelagem, as metodologias de Booch, a OMT e a OOSE, citadas anteriormente, evoluíram levando a criação de uma linguagem de notação unificada, chamada UML.

A UML é uma linguagem-padrão para a elaboração da estrutura de projetos de software, podendo ser empregada para visualização, especificação, construção e documentação de artefatos que façam uso de sistemas complexos de software. Sendo assim, a linguagem pode ser considerada expressiva e abrangente [Booch, 2000].

No desenvolvimento da UML são utilizados cinco tipos de visões, nove diagramas e vários modelos de elementos e todos estarão envolvidos na criação dos diagramas e mecanismos gerais. Todos estes componentes em conjunto, especificam e exemplificam a definição do sistema. Abaixo são mostradas as partes que compõe o UML:

2.2.5.1 Visões de sistemas

As visões mostram diferentes aspectos do sistema que está sendo modelado. Cada visão é descrita por um número de diagramas que contém informações que dão ênfase aos aspectos particulares do sistema. As visões são compostas pelos modelos de elementos do sistema conforme mostra a figura abaixo:

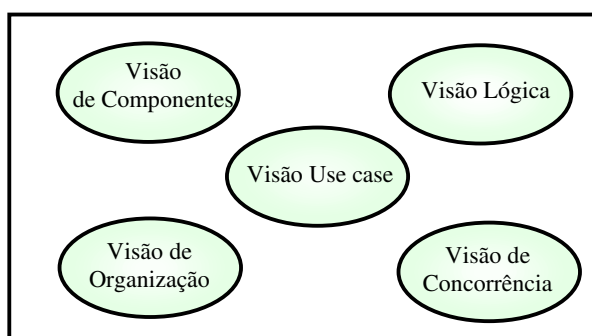


Figura 9 - Visões do sistema.

2.2.5.2 Modelos de elementos

Os conceitos usados nos diagramas são chamados de modelos de elementos. Um modelo de elemento é definido com a semântica e a definição formal do elemento com o exato significado do que ele representa sem definições duvidosas ou ambíguas e também define sua representação gráfica que é mostrada nos diagramas da UML. Alguns exemplos de modelos de elementos são: classes, objetos, estados, pacotes e componentes [Barros, 2001].

2.2.5.3 Mecanismos gerais

A UML utiliza alguns mecanismos em seus diagramas para tratar informações adicionais como ornamentos e notas.

Os ornamentos gráficos são anexados aos modelos de elementos em diagramas e adicionam semânticas ao elemento. Um exemplo de um ornamento é o da técnica de separar um tipo de uma instância.

Para permitir adicionar informações a um modelo, UML permite a utilização de notas, que são símbolos gráficos para a representação de restrições e comentários anexados a um elemento ou uma coleção de elementos [Booch, 2000].

2.2.5.4 Diagramas

O diagrama é uma apresentação gráfica de um conjunto de elementos, geralmente representados como um gráfico conectado de vértices (itens) e arcos (relacionamentos). Uma vez que nenhum sistema complexo pode ser compreendido em sua totalidade em uma única perspectiva, a UML define um número de diagramas que permite dirigir o foco para aspectos diferentes de seus sistemas de maneira independente [Booch, 2000].

Os modelos são desenvolvidos a partir de blocos como: classes, colaborações, interfaces, componentes, dependências, generalizações, associações, etc. Os diagramas utilizados pela UML são compostos de nove tipos citados abaixo, que são meios para visualização de blocos de construção.

Visão de use case

- Diagramas de use case (Modelagem comportamental);
- Diagramas de atividades (Modelagem comportamental).

Visão de projeto

- Diagramas de classes (Modelagem estrutural);
- Diagramas de colaboração (Modelagem comportamental);
- Diagramas de estados (Modelagem comportamental).

Visão de processo

- Diagramas de objeto (Modelagem estrutural);
- Diagramas de seqüência (Modelagem comportamental).

Visão de implementação

- Diagramas de componentes (Modelagem estrutural).

Visão de implantação

- Diagramas de execução ou implantação (Modelagem estrutural).

Diagramas estruturais: Os diagramas estruturais da UML existem para visualizar, especificar, construir e documentar os aspectos estáticos de um sistema. [Booch, 2000]. Os diagramas estruturais são:

Diagrama de classes

É uma estrutura lógica estática em uma superfície de duas dimensões mostrando uma coleção de elementos declarativos de modelo, como classes, tipos e seus respectivos conteúdos e relações [Furlan, 1998].

As classes podem se relacionar com outras através de diversas maneiras: Associação (conectadas entre si); dependência (uma classe depende ou usa outra classe); especialização (uma classe é uma especialização de outra classe); em pacotes (classes agrupadas por características similares) [Eriksson, 1998].

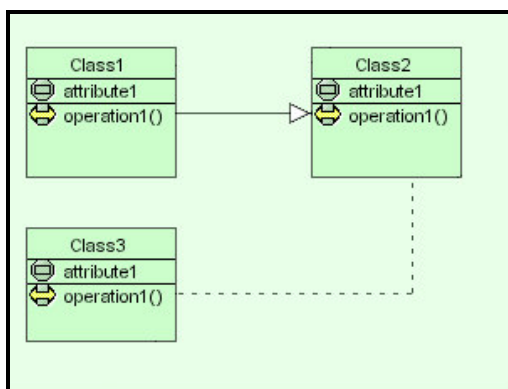


Figura 10 - Exemplo de diagrama de classe.

Diagrama de objetos

Um diagrama de objetos mostra um conjunto de objetos e seus relacionamentos em determinado ponto no tempo. Estes diagramas não são importantes apenas para a visualização, especificação e documentação de modelos estruturais, mas também para a construção de aspectos estáticos de sistemas por meio de engenharia de produção e engenharia reversa [Booch, 2000].

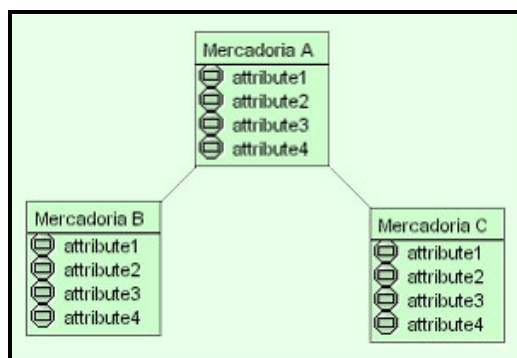


Figura 11 - Exemplo de diagrama de objetos.

Diagrama de componentes

Diagrama físico que mostra os vários componentes em um sistema e suas dependências. É um diagrama similar a um pacote, mas com um enfoque ao empacotamento físico do código. Uma classe pode estar em vários componentes, mas só pode estar definida em um único pacote. Os componentes podem ser: arquivos executáveis, bibliotecas estáticas e Dll's [Erickson, 1999].

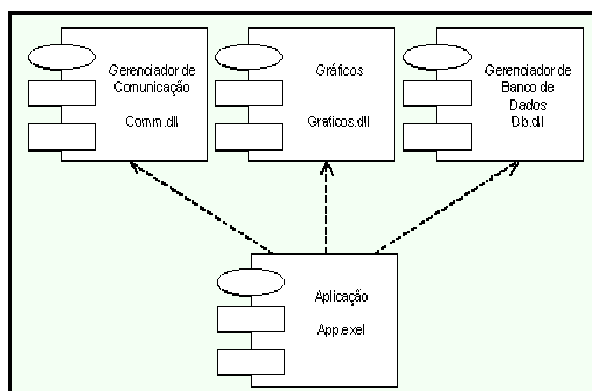


Figura 12 - Exemplo de diagrama de componentes.

Diagrama de implantação

O diagrama de implantação mostra a arquitetura física do hardware e software no sistema. Apresenta os atuais computadores e periféricos, juntamente com as conexões que eles estabelecem entre si e pode mostrar também os tipos de conexões entre esses computadores e periféricos. Os diagramas de implantação são empregados para modelagem da visão estática de implantação de um sistema [Booch, 2000].

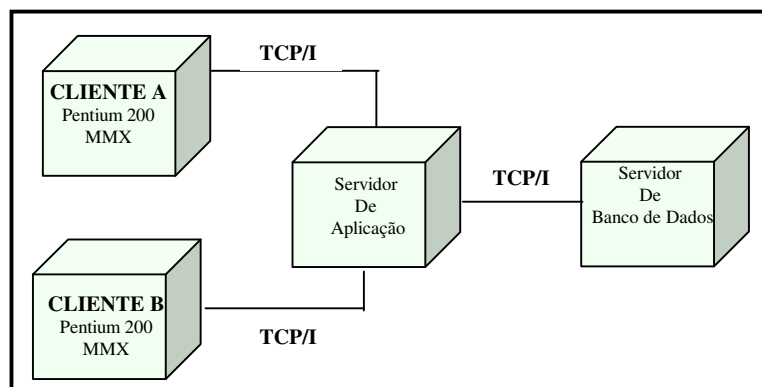


Figura 13 - Exemplo de diagrama de implantação.

Diagramas comportamentais: Os diagramas comportamentais da UML existem para visualizar, especificar, construir e documentar os aspectos dinâmicos de um sistema. Abaixo serão apresentados os diagramas adicionais para visualizar as partes dinâmicas de um sistema.

Diagrama Use case

O diagrama de use case consiste em cinco gráficos simples que representam o sistema, atores e use cases, associações e dependência do projeto. O objetivo desse diagrama é fornecer uma explicação do relacionamento entre o sistema e ambiente externo [Pender, 2002]. Os atores representam usuários e outros sistemas que interagem com sistema modelado e os use cases mostram o comportamento do sistema, cenários que o sistema percorre em resposta ao estímulo de um ator. Este diagrama faz com que sistemas, subsistemas e classes fiquem acessíveis e compreensíveis, por apresentarem uma visão externa sobre como esses elementos podem ser utilizados no contexto [Booch, 2000].

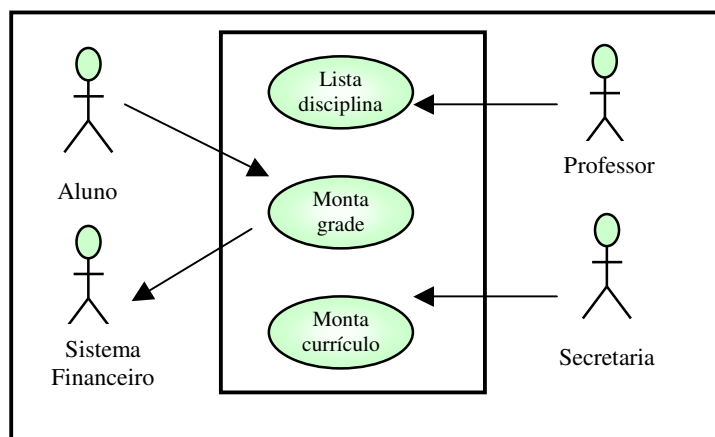


Figura 14 - Exemplo de diagrama use case.

Diagrama Seqüência

Um diagrama comportamental que mostra uma interação, dando ênfase à ordenação temporal das mensagens. O diagrama de seqüência captura o comportamento de um único use case e apresenta os objetos e as mensagens que são passadas entre estes objetos dentro do use case, conforme mostra a figura abaixo [Booch, 2000].

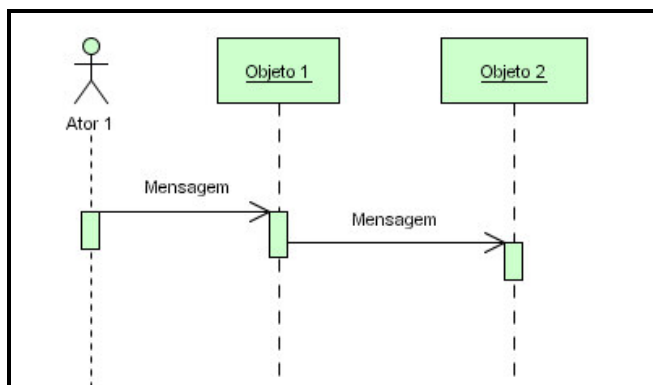


Figura 15 - Exemplo de diagrama de seqüência.

Diagrama de colaboração

Um diagrama comportamental que mostra uma interação, dando ênfase à organização estrutural de objetos que enviam e recebem mensagens [Booch, 2000].

Um diagrama de colaboração pode ser uma outra alternativa para a modelagem de interações entre objetos de um sistema. Diferentemente do diagrama de seqüência que focaliza na ordem cronológica do cenário que está sendo modelado, o diagrama de colaboração focaliza no relacionamento entre os objetos e na compreensão dos efeitos sobre um objeto durante um cenário, conforme mostra a figura abaixo: [Eriksson, 1998].

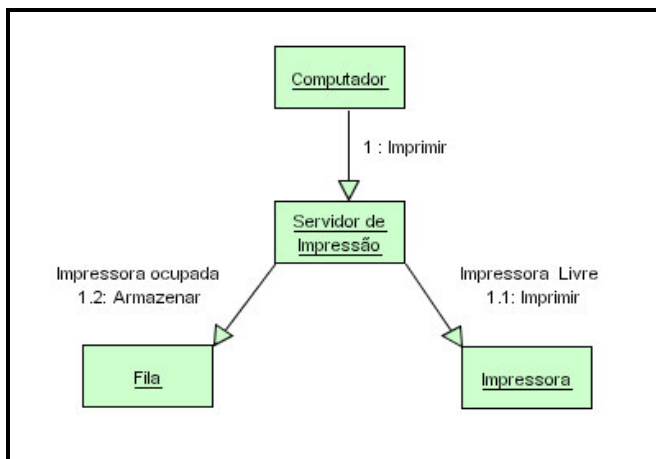


Figura 16 - Exemplo de diagrama de colaboração.

Diagrama de estados

Os diagramas de estados são usados para modelar o comportamento dinâmico de um sistema. Mostram o ciclo de vida de um objeto em níveis de detalhe arbitrariamente simples ou complexos [Larman, 2000].

Este diagrama mostra todos os estados possíveis que os objetos de uma certa classe podem se encontrar e também quais são os eventos dos sistemas que provocam tais mudanças. Os diagramas de estado não são escritos para todas as classes de um sistema, mas apenas para aquelas que possuem um número definido de estados conhecidos e onde o comportamento das classes é afetado e modificado pelos diferentes estados.

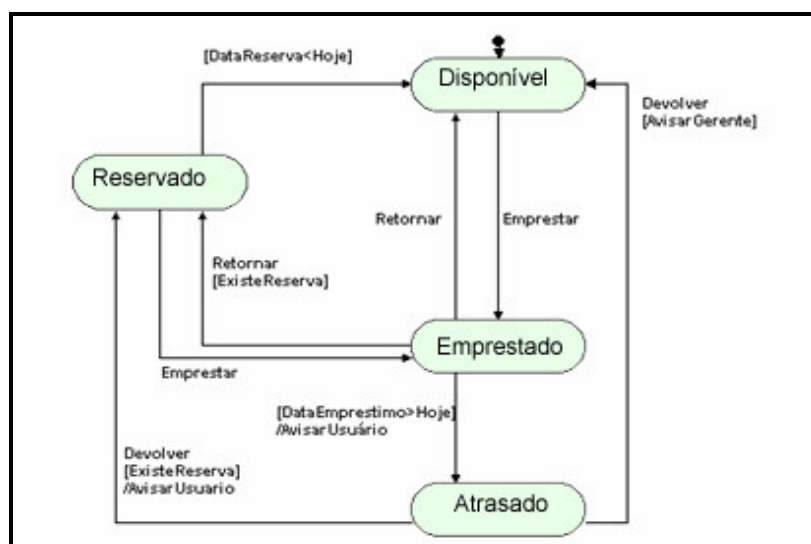


Figura 17 - Exemplo de diagrama de estado.

Diagrama de atividades

Um diagrama comportamental que mostra uma máquina de estados, dando ênfase ao fluxo de uma atividade para outra [Booch, 2000].

Os diagramas de atividades são usados para detalhar classes, implementação de operações e use case. Os diagramas de atividades representam o que acontece, mas não representam quem as realiza. Isso significa que o diagrama não diz qual classe é responsável por cada atividade. O uso comum do diagrama de atividade é de capturar o trabalho interno em um objeto, para mostrar como um grupo de ações relacionadas podem ser executadas, e como elas vão afetar os objetos em torno delas e para mostrar como um negócio funciona em

termos de trabalhadores (atores), fluxos de trabalho, organização, e objetos (fatores físicos e intelectuais usados no negócio).

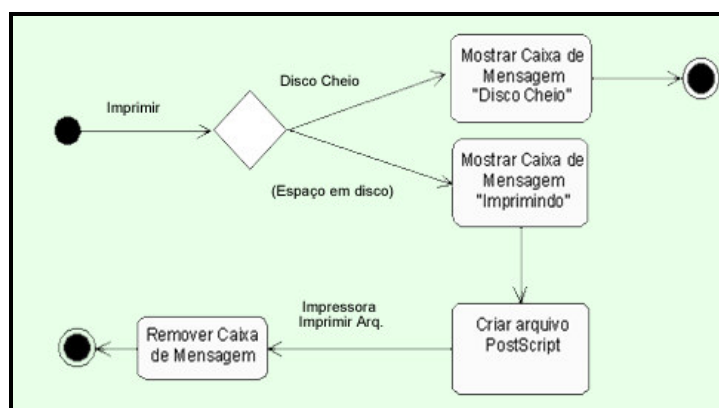


Figura 18 - Exemplo de diagrama de atividade.

2.2.6 Tecnologias de desenvolvimento para Web

Existem diversas tecnologias para *Web*, cada qual com sua filosofia. Serão abordadas algumas destas tecnologias, dando uma visão geral do processo de desenvolvimento para a *Web*.

2.2.6.1 Tecnologia para interface

Existem algumas tecnologias que adicionadas ao *Hypertext Markup Language* - HTML proporcionam uma interface mais poderosa e rica, pois ela sozinha, não é capaz de oferecer tais recursos. Algumas tecnologias conhecidas que são responsáveis pela construção da interface com o usuário são: DHTML, Active X, Applet Java e VBScript.

2.2.6.2 Tecnologia para processamento no servidor

Na camada *middleware* (software intermediário), ocorre a execução do aplicativo *Web*. A responsabilidade desta camada é processar as informações solicitadas pelo cliente, interagir com o banco de dados, extrair respostas (quase sempre na forma de uma página HTML) e enviá-la ao cliente. Existem várias tecnologias para a construção dessa camada como, *CGI-Common Gateway Interface*, *PHP-Hypertext Preprocessor*, *ASP-Application Service Provider*, *Servlets*, etc.

2.2.7 Sistema gerenciador de banco de dados - SGBD

O SGBD permite que os dados sejam armazenados em um só lugar, tornando-os disponíveis para diferentes aplicações. O software de SGBD serve como interface entre o BD comum e diversos programas aplicativos [Laudon, 1998].

2.2.7.1 Banco de dados orientado a objeto

O BD orientado a objetos armazenam dados como objetos que podem ser automaticamente recuperados e compartilhados. Esses objetos podem conter diversos tipos de dados, inclusive sons, gráficos e vídeos, bem como dados tradicionais e procedimentos de processamento. Os objetos podem ser compartilhados e reutilizados [Laudon, 1998].

2.2.7.2 Banco de dados relacional

O modelo relacional representa todos os dados do BD em tabelas simples bidimensionais denominadas relações. As tabelas parecem semelhantes a arquivos simples, mas as informações em mais de um arquivo podem ser extraídas e combinadas com facilidade.

A vantagem do modelo relacional está em que um elemento de dados de um arquivo ou tabela pode ser relacionado a qualquer fragmento de dado em outro arquivo ou tabela desde que ambas as tabelas compartilhem um elemento de dado comum [Laudon, 1998].

Existem BD *open-source*, como Interbase e o PostgreSQL e *freewares* como MySQL e comerciais como Oracle, DB2, etc. A escolha quanto ao tipo depende da necessidade e das limitações do cliente. É importante destacar que cada tipo de projeto tem suas necessidades específicas.

MySQL: É um SGBD relacional *open-source*. Foi originalmente desenvolvido para manipular grandes bases de dados com maior rapidez que as soluções existentes. Tem sido utilizado com sucesso em ambiente de produção com alta demanda. Apesar de estar em constante desenvolvimento, o MySQL oferece um rico conjunto de funções. Sua velocidade e segurança tornam o MySQL altamente apropriado para acesso a BD na Internet. É um sistema cliente/servidor que consiste de um servidor SQL que suporta vários programas clientes, bibliotecas, ferramentas administrativas e uma interface de programação. Suas principais características:

- Suporta vários sistemas operacionais como Linux e Windows;
- Suporta consulta com tabelas de diferentes bases de dados;
- Manipula grandes bases de dados [MySQL, 2002].

2.2.8 Linguagem de Programação

Durante a escolha de uma linguagem de programação alguns critérios devem ser avaliados, como:

- A área de aplicação geral;
- A complexidade computacional e algorítmica;
- O ambiente em que o software será executado;
- Considerações de desempenho;
- A complexidade da estrutura de dados;
- O conhecimento da equipe de desenvolvimento do software;
- A disponibilidade de um bom compilador [Pressman, 1995].

Visual Basic - VB: É uma linguagem de programação para Windows, baseada em eventos gerando aplicativos cuja operação é determinada pela escolha de algum objeto gráfico, que representa uma ação, por parte do usuário.

O VB oferece recursos para construir janelas contendo todos os elementos da interface visual padrão do Windows: botões, caixas de diálogo, menus, marcadores de opção, listas de itens para seleção, etc. Além disto, módulos adicionais, chamados de VB *extensions* que permitem acrescentar novas funções, que podem ser desenvolvidas por terceiros, àquelas consideradas padrão e distribuídas com o sistema.

O VB oferece vários recursos como, acesso ao banco de dados (com vários formatos suportados), funções de comunicação e suporte a recursos avançados do Windows e ligação e incorporação de objetos (*OLE - Object Linking and Embedding*).

PHP: É uma linguagem de programação de ampla utilização, interpretada, que é especialmente interessante para desenvolvimento para a *Web* e pode ser mesclada dentro do código HTML. A sintaxe da linguagem lembra Visual C, Java e Perl, e é de fácil compreensão. O objetivo principal da linguagem é permitir ao projetista escrever páginas que serão geradas dinamicamente [PHP, 2002].

A característica mais forte e mais significativa do PHP é que ele tem suporte à maioria dos BD populares (suporta mais de vinte BD diferentes), tanto em plataforma Linux como Windows. O PHP também tem suporte para utilização de outros serviços, utilizando protocolos como SNMP, POP3, HTTP, IMAP, entre outros [PHP, 2002].

Capítulo 3 Sistema Proposto

Este capítulo apresenta uma descrição das fases de desenvolvimento do sistema de admissão acadêmica baseado em metodologias de engenharia de software.

3.1 INTRODUÇÃO

Como visto no capítulo anterior, a UML é uma linguagem para especificação, visualização, documentação e construção tornando possível expressar modelos orientados a objetos.

Serão utilizados o processo de desenvolvimento RUP e o ciclo de vida clássico para organização das atividades. O modelo progride através das fases citadas a seguir:

Concepção: Descrição do sistema atual e do sistema proposto a partir do entendimento do problema, estabelecimento de metas e prazos.

Elaboração: Levantamento dos requisitos do sistema, criação de modelos abstratos que visam demonstrar o tratamento das informações a serem trabalhadas pelo sistema e o protótipo da interface com o usuário.

Construção: Criação do código fonte a partir da elaboração e criação do manual do usuário.

Transição: Simulação do funcionamento do sistema para detecção de possíveis erros e correção dos mesmos, e alterações para um melhor funcionamento do sistema. Treinamento dos usuários do sistema e entrega do manual do usuário.

3.2 FASE INICIAL – CONCEPÇÃO

Nesta seção será apresentado o resultado da fase inicial do ciclo de vida de um software, onde o artefato gerado é a proposta de desenvolvimento. Esta proposta faz parte da documentação do projeto, e visa descrever a situação atual do sistema e permitir ao usuário a possibilidade de executar um estudo de viabilidade com base nas alternativas de implementação, benefícios e objetivos.

3.2.1 Coleta de informações - entrevistas

Nesta parte da documentação do projeto, será discutido apenas entrevistas que aconteceram no decorrer da coleta de informações, tendo como objetivo maior capturar o máximo de informações dos funcionários envolvidos no processo seletivo.

Durante esta etapa de desenvolvimento do sistema, onde mostra todo o procedimento da entrevista com vários tipos de usuários, será utilizado questionário, de onde serão extraídos os primeiros requisitos do sistema.

Através das reuniões realizadas com os usuários foi possível obter as seguintes informações:

- informações gerais do processo seletivo;
- criação de novas idéias;
- maneira de ser administrado o processo de classificação dos candidatos;
- definição de metas;
- avaliação do sistema atual;
- resolução de futuros e existentes problemas;
- planejamento ou definição de estratégias;
- tomada de decisões;
- obtenções de resultados.

Entrevista:

A primeira entrevista aconteceu com o coordenador, projetista e funcionários que trabalham diretamente com o sistema atual.

Os principais objetivos da entrevista: coletar dados pertinentes ao assunto a ser tratado o quanto antes possível e documentar o funcionamento do sistema atual, esclarecer as dúvidas e tratar dos assuntos relacionados ao desenvolvimento do novo sistema, que terá como função controlar o processo seletivo.

O material adquirido será revisado e estudado podendo assim surgir um modelo descritivo, para que a entrevista subsequente possa verificar se realmente não houve engano no entendimento; se o usuário não mudou de opinião e se surgiram novos esclarecimentos e eventuais dúvidas quanto ao sistema a ser desenvolvido. O formulário abaixo mostra a especificação de requisitos respondida durante a entrevista com o usuário.

Tabela 1 - Formulário de perguntas para especificação de requisitos.

ESPECIFICAÇÃO DE REQUISITOS	
Analista:	Local:
Participantes:	Data:
QUESTIONÁRIO: Perguntas	
01. Quais os tipos de atividades que são realizadas neste departamento?	
02. Quais as pessoas responsáveis pelas atividades dentro do departamento?	
03. Quem é o responsável por todas as atividades?	
04. Quais os benefícios de solução bem sucedida?	
05. Que problemas essa solução vai atacar?	
06. Na criação de um sistema de armazenamento de dados, quem será o responsável pelo sistema?	
07. Quais serão os tipos de informações que serão produzidas?	
08. Quais são os fatores que afetam o atendimento do candidato?	
09. Quais funções e desempenho são exigidos?	

Tabela 2 - Formulário de respostas para especificação de requisitos

ESPECIFICAÇÃO DE REQUISITOS	
Analista:	Local:
Participantes:	Data:
QUESTIONÁRIO: Respostas	
01. Controle das informações relacionadas aos candidatos, como: cursos, vagas, edital, prova, classificação, documentos, etc. e outros fatores que estão relacionados com o processo seletivo.	
02. As pessoas responsáveis serão os próprios funcionários do setor administrativo.	
03. Chefe do departamento administrativo do Universitas é o responsável pelo processo seletivo.	
04. Confiabilidade, agilidade, segurança, redução de custos e melhor desempenho.	
05. Com a implantação do novo sistema, o maior problema que será atacado será em relação à redundância de dados.	
06. O chefe de departamento ficará responsável.	
07. Aos candidatos deverão ser fornecidas informações em relação ao processo seletivo do ano corrente, bem como o acesso às inscrições e consultas à classificação. Já os funcionários deverão ter acessos as informações referentes ao sistema Intranet.	
08. Os fatores que afetam o atendimento do candidato estão relacionados com a falta de opção em realizar inscrições, fazendo com que o candidato se desloque até o Universitas.	
09. Todos os tipos de informações referentes às atividades executadas pelo departamento responsável pelo processo seletivo.	

3.3 PROPOSTA DE DESENVOLVIMENTO

3.3.1 Introdução

Objetivo do documento

O documento tem como objetivo descrever o funcionamento do sistema atual do Universitas, seus problemas e suas necessidades, bem como apresentar uma proposta para o desenvolvimento de um novo sistema de admissão acadêmica, objetivando melhorar o sistema de gerenciamento de informações sobre o processo seletivo.

No decorrer do processo de desenvolvimento do sistema vale ressaltar que serão aceitas modificações na estrutura do projeto, bem como no cronograma de atividades, se necessário, desde que ambas as partes estejam de acordo.

O documento atual servirá de referência para futuras modificações e correções de requisitos dos usuários.

3.3.2 Descrição do sistema atual

Atualmente o sistema apresenta falhas em sua estrutura de desenvolvimento, tais como:

- O sistema não possui documentação adequada (Ex: manual do usuário);
- apresenta deficiência para diagnosticar e eliminar falhas;
- as tabelas não são relacionais;
- o sistema é composto por várias bases de dados, causando assim redundância de dados;
- dificuldade em recuperar qualquer tipo de informação pertinente ao processo seletivo anterior;
- não possui tarefas necessárias para um bom desempenho e como consequência perde-se rendimento (Ex: insuficiência de relatórios);
- ausência de um critério de consistência de dados que permita a integração deste com outros sistemas da instituição.

3.3.3 Expectativa com o processo de automação

O processo de informações se tornará mais rápido e seguro, eliminando as falhas do sistema, possibilitando um controle eficiente dos dados e principalmente evitando informações redundantes.

3.3.4 Descrição do sistema proposto

Objetivos

O sistema proposto terá como principal objetivo otimizar as atividades realizadas durante o processo de admissão acadêmica, através do controle das informações descritas abaixo:

- Possibilitar a inscrição do candidato via *Web*;
- divulgar a lista de classificados via *Web*,
- emitir relatórios como: lista de candidatos confirmados e não confirmados para a prova, distribuição de candidatos por sala e bloco, classificação geral de candidatos aprovados e excedentes;
- distribuir candidatos por salas e blocos existentes na Instituição,
- simplificar as rotinas do setor administrativo, com objetivo de acelerar e aprimorar a qualidade dos serviços prestados pelo Universitas.
- cadastrar as seguintes informações:
 - Candidatos inscritos no vestibular;
 - salas/blocos onde serão realizadas as provas;
 - áreas (Biologia, exatas e humanas);
 - cursos existentes na Instituição;
 - disciplinas pertencentes a cada área;
 - critérios de classificação do candidato;
 - peso de cada disciplina da área;
 - notas referente a prova realizada pelo aluno.

3.3.5 Benefícios

De forma geral, podemos dizer que o sistema proposto terá as seguintes vantagens:

- Coerência e clareza das informações;
- economia de tempo e recurso;
- rapidez de acesso às informações;
- redução na taxa de erro de dados;
- resolução das principais deficiências do sistema atual;
- interação entre módulos do sistema, futuramente entre (DRA);
- confiabilidade do sistema.

Além da secretaria, o departamento financeiro também será beneficiado com a informatização do sistema, uma vez que o boleto de pagamento das inscrições também poderá ser emitido através da Internet, proporcionando maior comodidade para o aluno e economizando o tempo do funcionário responsável por sua entrega.

3.3.6 Viabilidade de implementação

Expectativas com relação ao sistema proposto

O sistema proposto deverá atender as especificações citadas acima, proporcionando rapidez, agilidade e segurança na obtenção de informações.

O sistema será eficaz, de fácil aprendizado, com capacidade para adaptação (flexibilidade) e facilidade para realizar alterações, etc. Também deverá expressar com clareza a realidade representada pelo sistema e suprir as expectativas e necessidades do usuário.

O sistema proposto desconsidera as características próprias do sistema atual, já que o mesmo não consegue satisfazer o seu cliente.

Recursos necessários

Cronograma de recursos: Não será necessária a aquisição de software e hardware, pois no laboratório de processamento de dados, possui uma rede local, computadores e softwares suficientes para o desenvolvimento do sistema.

Alternativas de implementação

O cronograma define as metas que devem ser atingidas, nos quais os envolvidos deverão empenhar-se na execução de suas atividades para o cumprimento dos mesmos. A tabela abaixo exprime o tempo dedicado a cada tarefa, determinando assim o prazo final do projeto.

Tabela 3 - Cronograma do projeto.

FASES	18/8	25/8	01/9	08/9	15/9	22/9	29/9	06/10	13/10	20/10	27/10	03/11	10/11	17/11	24/11	05/12
Concepção																
Elaboração																
Construção																
Transição																

3.3.7 Relação do sistema com o ambiente

Singularidade: Existem diversas ferramentas de controle de vestibular disponíveis no mercado, entretanto este projeto será desenvolvido especificamente para atender as necessidades da Instituição.

Integrabilidade: Este projeto propõe-se a desenvolver o sistema de forma que ele possa ser integrado a outros sistemas para futuras expansões.

Com a análise e aprovação da proposta de desenvolvimento realizada pela comissão do processo seletivo do Universitas, foi possível dar início à fase de elaboração.

3.4 FASE DE ELABORAÇÃO

3.4.1 Introdução

Nesta fase serão desenvolvidas atividades como: desenvolvimento do modelo descritivo, definição dos requisitos funcionais e não funcionais, definição dos atores, criação e documentação dos use cases através do modelo de cartões e a criação do layout sugerido pelo cliente.

3.4.2 Escopo do produto:

Nome do produto e seus componentes principais

Módulo Internet: Sistema de inscrição on-line.

Módulo Intranet: Controle geral das atividades realizadas pelo funcionário - Sistema de admissão acadêmica.

3.4.3 Missão do produto

Controlar o processo seletivo da Instituição, proporcionando uma maior agilidade na manipulação dos dados, no qual os principais objetivos são:

- Otimizar e tornar mais eficaz o sistema de seleção dos candidatos.
- Proporcionar mais comodidade aos candidatos com sistema on-line.

3.4.4 Limites do produto

Apenas algumas funções do sistema de admissão acadêmica serão desenvolvidas para *Web*.

3.4.5 Especificação do sistema

O sistema de admissão acadêmica tem como objetivo principal controlar o processo de inscrição do aluno e gerar a lista de classificação dos aprovados.

O sistema deverá permitir aos funcionários da secretaria a inclusão, consulta, alteração e exclusão dos itens abaixo:

- Inscrição dos candidatos;
- Áreas (humanas, exatas e biológicas);
- Cursos oferecidos em cada área;
- Critérios de classificação do aluno;
- Disciplinas de cada área, notas dos alunos, pesos de cada disciplina e salas onde serão realizadas as provas.

Deverá existir uma opção de listagens para permitir a consulta dos candidatos aprovados, confirmados, excedentes e o envio de relatório de notas para o MEC.

No sistema on-line, deverá existir uma opção de inscrição e envio de boleto, onde o aluno possa realizar sua inscrição e adquirir o boleto para efetuar o pagamento. Além disso, o aluno poderá consultar a lista de classificação pela Internet.

3.4.6 Requisitos funcionais do sistema

Os requisitos funcionais definem as ações fundamentais através das quais o produto aceita e processa as entradas especificadas, gerando as respectivas saídas. Nesta seção será feito o detalhamento desses requisitos, em nível suficiente para o projeto do produto, de seus testes de aceitação e de seu manual do usuário.

Tabela 4 - Requisitos funcionais do sistema.

<i>Use case</i>	<i>Descrição</i>
Cadastrar usuário	Permite ao administrador cadastrar usuários que terão acesso ao sistema.
Autenticar usuário.	Permite a entrada de usuário no sistema.
Realizar estatísticas.	Permite exportar dados do questionário sócio-cultural cadastrado na tela candidatos, com o objetivo de manipular os dados no Excel.
Controlar cursos.	Permite a inserção, exclusão, alteração e consulta dos cursos.
Realizar manutenção.	Permite a compactação da base de dados do sistema.
Distribuir salas/blocos.	Permite inserção, exclusão, alteração e consulta das salas existentes na Instituição.
Controlar disciplinas.	Permite inserção, exclusão e consulta de disciplinas.
Controlar áreas.	Permite a inserção, alteração, exclusão ou somente consultas das áreas cadastradas.
Controlar candidatos.	Permite controlar as informações referentes aos candidatos cadastrados no processo seletivo.
Distribuir cand./sala.	Permite consultar e imprimir a distribuição dos candidatos por sala e bloco.
Definir pesos.	Permite a inserção, alteração, exclusão ou somente consultas dos pesos cadastradas.
Inscrever candidato.	Permite a realização do candidato no processo seletivo.
Estipular critérios.	Permite estipular critérios para a desclassificação do candidato no processo seletivo.
Emitir relatórios.	Permite emitir Listagem: (Candidatos confirmados, candidatos não confirmados, distribuição de candidatos, classificação geral dos candidatos, candidatos aprovados, candidatos excedentes e relatório notas MEC).
Gerar backup	Permite fazer backup do processo que esta sendo usado.
Controlar notas.	Permite inclusão das notas obtidas por cada candidato nas provas e onde será dada a média final para a classificação.
Controlar classificação.	Permite a visualização das notas e o peso por disciplina/área, para a classificação dos candidatos em cada processo seletivo. Permite também visualizar a colocação por ordem de opção escolhida pelo candidato na inscrição.
Divulgar edital	Interface que permite disponibilizar informações ao candidato sobre o Processo Seletivo.
Efetuar inscrição on-line.	Permite o candidato se inscrever no processo seletivo através da <i>Web</i> .
Divulgar cursos e vagas	Permite disponibilizar informações ao candidato quanto aos cursos/vagas existentes na Instituição.
Realizar inscrição	Interface que permite disponibilizar informações ao candidato quanto ao período, local, horário, documentação, em relação ao processo seletivo.

Continuação da Tabela 4 - Requisitos funcionais do sistema.

<i>Use case</i>	<i>Descrição</i>
Divulgar provas	Interface que permite disponibilizar informações ao candidato quanto ao programa das provas do processo seletivo.
Consultar classificação On-line.	Interface que permite disponibilizar informações aos candidatos de duas maneiras: classificação geral e classificação por área.
Exibir resultado	Interface que permite disponibilizar informações aos candidatos sobre a data de divulgação do resultado do processo seletivo.
Exibir critérios de desempate	Permite disponibilizar informações aos candidatos quanto aos critérios adotados pela Instituição a respeito de empate.
Divulgar matrícula	Interface que permite disponibilizar informações aos candidatos sobre a data, local e documentos necessários para realização da matrícula.
Exibir regulamentação	Interface que permite disponibilizar as normas da Instituição.

3.4.7 Descrição dos atores e use cases

Este tópico mostrará um diagrama de contexto contendo os principais use cases do sistema referente aos requisitos funcionais.

Atores do sistema: O sistema possui senhas e existem os seguintes tipos de usuários:

- SGBD: Servidor de banco de dados do sistema de admissão acadêmica.
- Funcionário: Terá acesso a todas as telas do software.
- Candidato: Terá acesso à tela de cadastro de candidatos, na consulta da classificação e outras informações referente ao processo seletivo.

Use cases: O diagrama de contexto descrito na figura 19 apresenta os atores e os use cases do sistema. Os use cases são compostos por fluxos de eventos (fluxo principal, subfluxo alternativo e subfluxo de exceção).

3.4.8 Detalhamento dos use cases

A **figura 20** apresenta os subfluxos do use case controlar candidato.

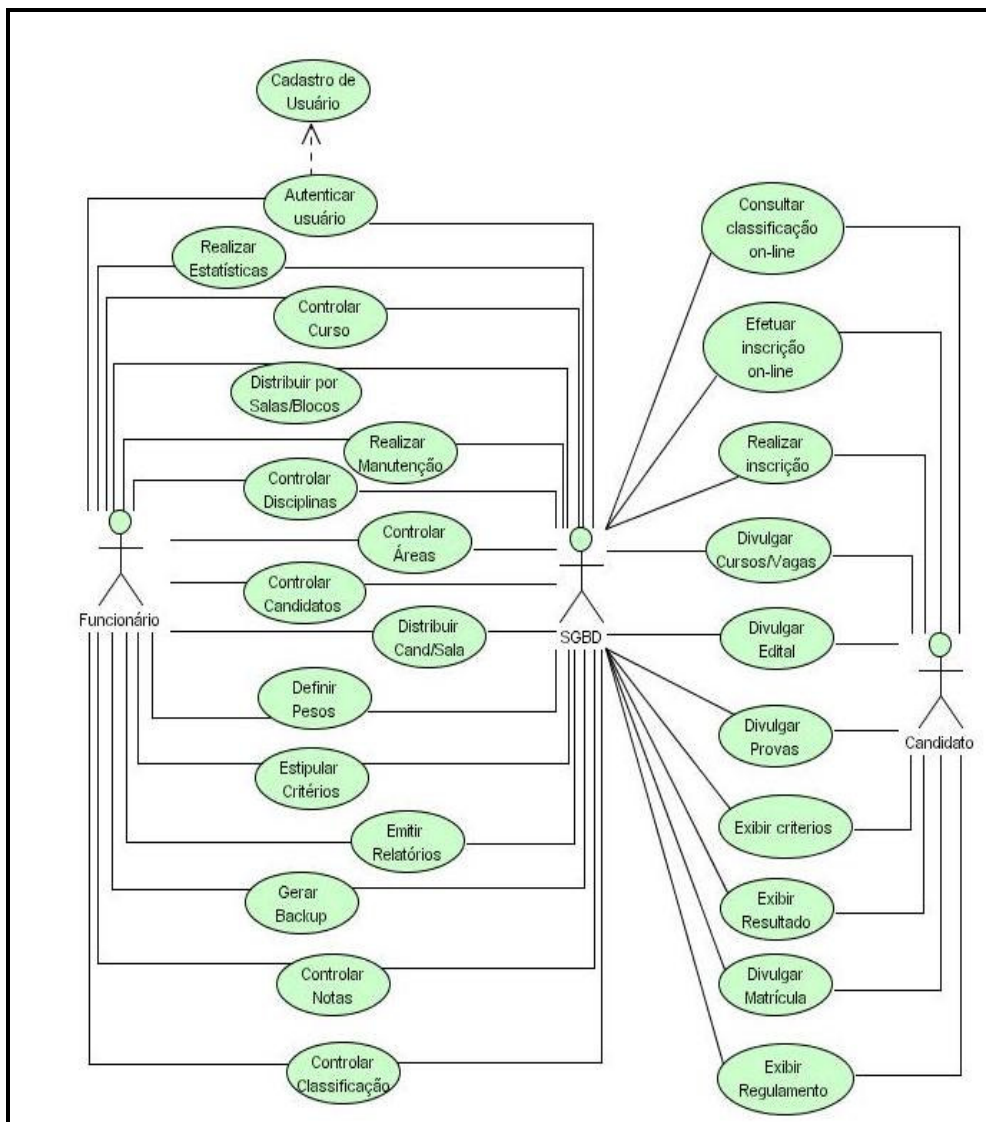


Figura 19 - Diagrama de contexto.

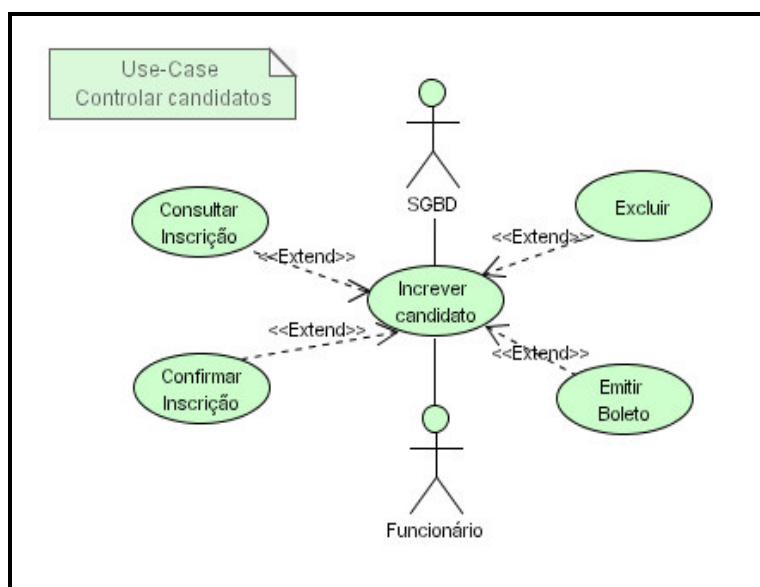


Figura 20 - Use-Case – Controlar candidatos.

1. Fluxo de evento principal – Controlar candidatos

Tabela 5 - Cartão cenário use case: Controlar candidatos.

Use case:		Controlar Candidatos	
Atores:	Funcionário quem inicia.		
Propósito:	Efetuar inscrição do candidato no BD.		
Descrição:	Candidato passa informações pessoais e escolha do curso, e o Funcionário o cadastra.		
Curso de Eventos Típicos			
Tipo:	Principal		
Fluxo de evento principal			
Ação do Ator		Resposta do Sistema	
1. Candidato passa informações pessoais e escolha da área/curso. Funcionário abre tela cadastro de candidatos.		2. O sistema exibe os campos a serem preenchidos.	
3. O Sistema fornece ao funcionário algumas outras atividades como: Alteração, Exclusão, Localização, Impressão, Confirmação e Sair.			
4. Se a atividade selecionada for: * Alterar, * Excluir, * Localizar, * Imprimir, * Confirmar, * Sair.		5. * O subfluxo (Alterar inscrição do candidato) é executado. * O subfluxo (Excluir inscrição cadastrada) é executado. * O subfluxo (Localizar candidato) é executado. * O subfluxo (Imprimir boleto/ficha cadastral) é executado. * O subfluxo (Confirmar inscrição do candidato) é executado. * O Use-Case é encerrado.	
Subfluxos de exceções			
* Caso já tenha encerrado o processo seletivo, não será possível o funcionário alterar ou cadastrar candidatos.			
Subfluxos Alternativos			
* Alterar dados do candidato, * Excluir inscrição do candidato, * Localizar candidato, * Imprimir inscrição, * Confirmar inscrição do candidato, * Sair do sistema.			

2. Subfluxo Alternativo – Alterar dados do candidato

Tabela 6 - Cartão cenário use case: Controlar candidatos.

Use case:		Alterar dados do candidato	
Atores:	Funcionário (quem inicia).		
Propósito:	Efetuar Alteração dos dados do candidato.		
Descrição:	Funcionário altera os dados do candidato.		
Curso de Eventos Típicos			
Tipo: Alternativo.			
Subfluxos Alternativos			
Ação do Ator		Resposta do Sistema	
1. O Funcionário solicita alteração selecionando o campo do dado desejado para alteração.		2. O sistema deixa o campo pronto para edição.	
3. O Funcionário altera o dado desejado.		4. O dado é alterado na base de armazenamento persistente.	
Subfluxos de exceções			
* Usuário desiste das alterações e pressiona tecla enter. O Sistema não executa nenhuma alteração dos dados retornando para o valor que existia anteriormente.			

Use case - Controlar classificação

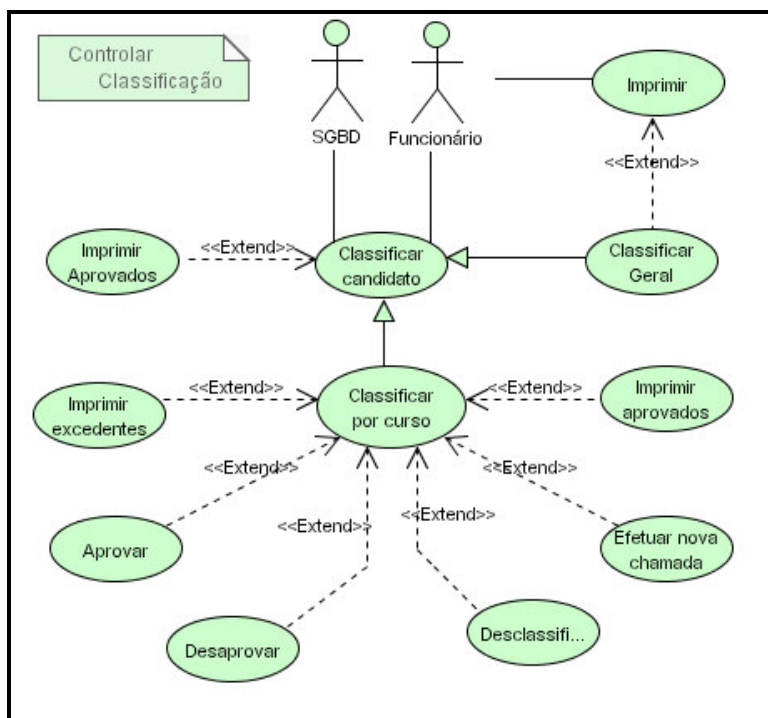


Figura 21 - Use-Case – Controle de candidatos

1. Fluxo de evento principal – Controlar classificação

Tabela 7 - Cartão Cenário Use case: Controlar classificação.

Use case:	Controlar Classificação	
Atores:	Funcionário (quem inicia).	
Propósito:	Controlar classificação dos candidatos.	
Descrição:	Funcionário controla classificação.	
Curso de Eventos Típicos		
Tipo: Principal		
Fluxo de evento principal		
Ação do Ator	Resposta do Sistema	
1. Use-Case inicia quando o Funcionário seleciona a opção Classificar Candidato.	2. O sistema exibe a tela com as seguintes opções: * Classificação Geral * Classificação	
3. O Funcionário escolhe a opção Classificação Geral.	4. O sistema exibe uma tela com uma lista de todos os candidatos classificados nos diversos cursos oferecidos.	
Fluxo de evento principal		
Ação do Ator	Resposta do Sistema	
1.O Funcionário escolhe a opção Classificação.	2. O sistema exibe uma tela, onde o funcionário seleciona o curso desejado para que possa exibir uma lista de todos os candidatos classificados.	
3. O funcionário escolhe o curso desejado.	4. O Sistema exibe os alunos classificados no curso específico.	
5. O Sistema fornece ao funcionário algumas atividades: Imprimir Excedentes, Imprimir Aprovados, Desaprovar, Desclassificar, Aprovar, Efetuar Chamada e Sair.		

6. Se a atividade selecionada for: * Imprimir Excedentes; * Imprimir Aprovados; * Desaprovar; * Desclassificar; * Aprovar; * Efetuar Chamada; * Sair.	7. * O subfluxo (Imprimir Excedentes) é executado. * O subfluxo (Imprimir Aprovados) é executado. * O subfluxo (Desaprovar) é executado. * O subfluxo (Desclassificar) é executado. * O subfluxo (Aprovar) é executado. * O subfluxo (Efetuar Chamada) é executado. * O Use-Case é encerrado.
Subfluxos de exceções	
Subfluxos Alternativos	
* Imprimir Candidatos Excedentes; * Imprimir Candidatos Aprovados; * Desaprovar Candidatos; * Desclassificar Candidatos; * Aprovar Candidatos; * Efetuar Chamada; * Sair.	

2. Subfluxo Alternativo – Imprimir candidatos excedentes

Tabela 8 - Cartão cenário use case: Imprimir candidatos excedentes.

Use case:	Imprimir candidatos excedentes	
Atores:	Funcionário (quem inicia).	
Propósito:	Efetuar impressão dos candidatos excedentes.	
Descrição:	Funcionário imprimir candidatos excedentes.	
Curso de Eventos Típicos		
Tipo: Alternativo.		
Subfluxos Alternativos		
Ação do Ator	Resposta do Sistema	
1. O funcionário abre a tela “Imprimir Excedentes”.	2. O sistema exibe os campos “Nome do Curso” e “Chamada para seleção”.	
3. Após a seleção escolha a forma de impressão do botão: <ul style="list-style-type: none"> • Listagem de excedentes. 	4. O sistema imprime apenas candidatos excedentes.	
Subfluxos de exceções		

3.4.9 Levantamento de interface com usuário – Módulo intranet

Foi mostrado no item acima as funções e atores do diagrama, iremos identificar as possíveis classes de interface com usuário na tabela abaixo:

Tabela 9 - Levantamento de interface com usuário-intranet.

<i>Nome</i>	<i>Ator</i>	<i>Use case</i>	<i>Descrição</i>
Tela senha.	Funcionário	Autenticar usuário.	Interface que permite a entrada de usuário no sistema.
Tela inicial.	Funcionário	Autenticar usuário.	Interface que permite a navegação para as demais interfaces do sistema.
Tela área.	Funcionário	Controlar áreas.	Interface que permite a inserção, alteração, exclusão ou somente consultas das áreas cadastradas.
Tela disciplina.	Funcionário	Controlar disciplinas.	Interface que permite inserção, exclusão ou somente a consulta das disciplinas cadastradas.
Tela curso.	Funcionário	Controlar cursos.	Interface que permite inserção, exclusão, alteração e consulta dos cursos cadastrados.
Tela sala.	Funcionário	Controlar salas/blocos.	Interface que permite inserção, exclusão, alteração e consulta das salas existentes no Universi-tas.
Tela Candidato.	Funcionário	Controlar candidato.	Interface que permite controlar as informações referentes aos candidatos cadastrados no proces-so seletivo.
Tela Critério	Funcionário	Estipular critérios.	Interface que permite a estipulação para a des-classificação do candidato no processo seletivo.
Telas Notas	Funcionário	Controlar notas	Interface que permite a inclusão das notas obti-das por cada candidato nas provas e onde será dada a média final para a classificação.
Tela distribuição de candidatos	Funcionário	Distribuir cand./sala.	Interface que permite a consulta e impressão da distribuição dos candidatos por sala e bloco.
Tela de classifi-cação de candi-datos.	Funcionário	Controlar classificação.	Interface que permite a visualização das notas e o peso por disciplina/área, para a classificação dos candidatos em cada processo seletivo.
Tela de impres-são de relatórios	Funcionário	Emitir relatórios.	Interface que permite emitir diversos relatórios.
Tela manutenção	Funcionário	Realizar manutenção.	Interface que permite a compactação da base de dados do sistema.
Tela estatística	Funcionário	Realizar Estatísticas.	Interface para exportação de dados do questio-nário Sócio-Cultural cadastrado na tela Candi-datos, com o objetivo de manipular os dados no Excel.
Tela de cadastro do usuário.	Administrador	Administrador.	Interface que permite a manipulação dos usuá-rios cadastrados no sistema.
Tela Backup	Funcionário	Gerar backup	Interface que permite fazer backup do processo que esta sendo usado.

3.4.10 Levantamento de interface com usuário – Módulo Internet

Tabela 10 - Levantamento da interface com usuário - (Internet).

<i>Nome</i>	<i>Ator</i>	<i>Use case</i>	<i>Descrição</i>
Tela Edital	Candidato	Divulgar Edital	Interface que permite disponibilizar informações ao candidato informando quando as inscrições para o Processo Seletivo estarão abertas.
Tela Cursos e Vagas	Candidato	Divulgar Cursos e Vagas	Interface que permite disponibilizar informações ao candidato quanto aos cursos/vagas existentes na Instituição.
Tela Inscrições	Candidato	Inscrever-se	Interface que permite disponibilizar informações ao candidato quanto ao Período, Local, Horário, Documentação, em relação ao processo seletivo.
Tela provas	Candidato	Divulgar Provas	Interface que permite disponibilizar informações ao candidato quanto ao programa das provas do processo seletivo e será disponibilizada no ato da inscrição.
Tela consulta Classificação	Candidato	Consultar Classificação	Interface que permite disponibilizar informações aos candidatos de duas maneiras Classificação geral e classificação por área.
Tela efetuar inscrição on-line	Candidato	Efetuar inscrição on-line	Interface que permite disponibilizar a ficha de inscrição, o questionário sócio cultural e a emissão do boleto aos candidatos para que sejam preenchidos.
Tela Critérios de desempate	Candidato	Exibir Critérios de desempate	Interface que permite disponibilizar informações aos candidatos quanto aos critérios adotados pela Instituição a respeito de empate.
Tela Resultado	Candidato	Exibir Resultado	Interface que permite disponibilizar informações aos candidatos sobre a data de divulgação do resultado do Processo Seletivo.
Tela Resultado	Candidato	Exibir Resultado	Interface que permite disponibilizar informações aos candidatos sobre a data de divulgação do resultado do Processo Seletivo.
Tela Matrícula	Candidato	Divulgar Matrícula	Interface que permite disponibilizar informações aos candidatos sobre a data, local e documentos da realização da matrícula.
Tela Regulamentação	Candidato	Exibir Regulamentação	Interface que permite disponibilizar as normas da Instituição.

3.4.11 Interfaces do software

Nesta etapa será detalhado o layout sugerido para interface com o usuário. Mostrando-se a realização dos use cases, assim como a estrutura dos componentes de interfaces. As idéias para interfaces do sistema foram sugeridas durante a entrevista com o cliente.

Módulo Intranet

TELA DE LOGIN

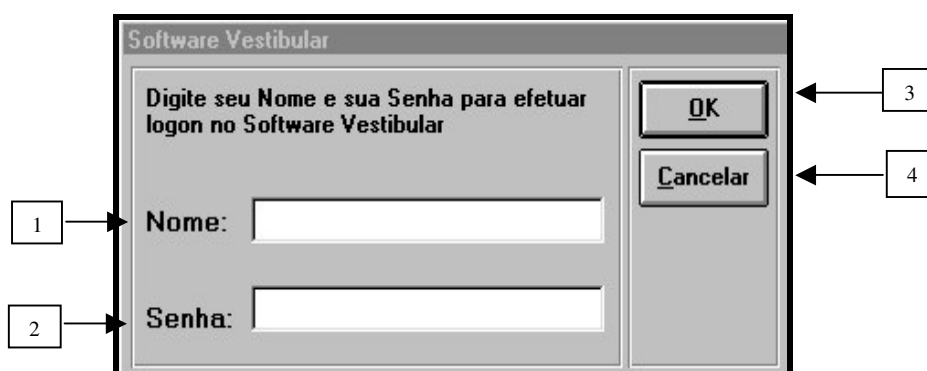


Figura 22 - Tela Login.

Relacionamento com outras Interfaces:

1. Campo para a entrada e visualização do nome do usuário.
2. Campo para a entrada da senha do usuário.
3. Permite ao usuário entrar no sistema.
4. Permite o cancelamento da entrada do usuário no sistema.

TELA INICIAL (PRINCIPAL)

Figura 23 - Tela: Principal.

Relacionamento com outras Interfaces:**1. Opção Processo Seletivo permite escolher as seguintes opções:**

- Abrir um novo processo.
- Fechar processo atual.
- Abrir um processo existente.
- Voltar ao processo atual.

2. Opção Cadastros permite escolher as seguintes opções:

- Cadastrar Áreas.
- Cadastrar Candidatos.
- Cadastrar Cursos.
- Cadastrar Critérios.
- Cadastrar Disciplinas.
- Cadastrar Notas.
- Cadastrar Pesos.
- Cadastrar Salas.

3. Opção Distribuição de Candidatos permite escolher a seguinte opção:

Distribuir Candidatos.

4. Opção Classificação permite escolher a seguinte opção:

Classificar Candidatos.

5. Opção Listagens permite escolher as seguintes opções:

Candidatos confirmados.

Candidatos não confirmados.

Distribuição de candidatos.

Classificação geral dos candidatos.

Candidatos aprovados.

Candidatos excedentes.

Relatório notas MEC.

6. Opção Estatística permite escolher a seguinte opção:

Exportar dados.

7. Opção Manutenção permite escolher as seguintes opções:

Backup do sistema.

Cadastro de usuário.

8. Opção Ajuda permite escolher as seguintes opções:

Ajuda do software vestibular.

Sobre o software vestibular.

9. Opção Sair permite escolher a seguinte opção:

Sair do sistema.

TELA ÁREAS

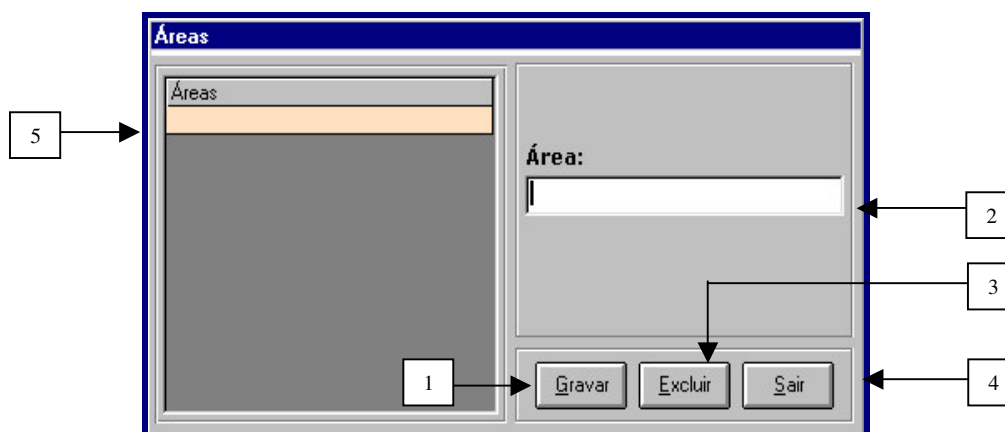


Figura 24 - Tela: Áreas.

As áreas não podem ter seus dados excluídos, alterados e consultados após o início do cadastrado dos candidatos, antes fica a critério do usuário.

Relacionamento com outras interfaces:

1. Permite ao usuário gravar uma nova área no sistema.
2. Permitti a visualização das áreas cadastradas.
3. Permite ao usuário excluir áreas existentes.
4. Permitti a saída do sistema.
5. Campo onde mostrará todas as áreas que foram cadastradas no sistema.

TELA INSCRIÇÃO DE CANDIDATO

The screenshot shows a software window titled "Alteração de Candidatos" with a sub-tab "Inscrição do Candidato". The form is divided into several sections:

- Dados Pessoais do Candidato:** Includes fields for Name, RG, CPF, Natural de, Endereço, Cidade, UF, CEP, E-Mail, Conclusão Ens. Médio, Treineiro, and Língua Estrangeira. There are also dropdown menus for Sexo and Data Nascimento.
- Escolha de Cursos:** Four dropdown menus labeled 1ª Opção, 2ª Opção, 3ª Opção, and 4ª Opção.
- Socio Economico:** A section with the question "1. Em que escola você cursou o 2º Grau?" and two radio button options: "Somente em escola particular" and "Somente em escola pública".
- Footer:** A row of buttons including navigation arrows, a status bar showing "Total de Candidatos 0", and buttons for "Incluir", "Excluir", "Imprimir", "Boleto", "Conf. Inscrição", "Localizar", and "Sair".

Numbered callouts (1-13) point to various elements: 1 points to the CEP field; 2 points to the Socio Economico section; 3 points to the "Incluir" button; 4 points to the "Boleto" button; 5 points to the "Conf. Inscrição" button; 6 points to the "Imprimir" button; 7 points to the "Localizar" button; 8 points to the "Excluir" button; 9 points to the left navigation arrow; 10 points to the "Total de Candidatos 0" status bar; 11 points to the left navigation arrow; 12 points to the right navigation arrow; 13 points to the "Sair" button.

Figura 25 - Tela: Inscrição de candidato.

Relacionamento com outras interfaces:

1. Campos onde serão incluídos todos os dados dos candidatos.
2. Campos onde será respondido o questionário sócio cultural.
3. Permite ao usuário gravar o candidato cadastrado no sistema.

4. Imprimir o Boleto da inscrição feita pelo candidato.
5. Confirmar a inscrição do candidato após o pagamento do boleto.
6. Imprimir os candidatos cadastrados.
7. Permite ao usuário localizar o candidato. Abre a interface “Busca de Candidato”, no qual exibi os dados do candidato.
8. Permite ao usuário excluir candidato.
9. Retornar a 1ª opção da lista de candidatos cadastrados.
10. Avança para o próximo Candidato da lista.
11. Retorna ao Candidato anterior da lista.
12. Avança para o último Candidato da lista.
13. Permite retornar a tela principal do sistema.

TELA CRITÉRIOS

A imagem mostra uma janela de software intitulada "Critérios de desclassificação de candidatos". No topo, há duas abas: "Critérios Cadastrados" e "Incluir Critério", com a segunda estando selecionada. O formulário principal contém o seguinte texto: "Eliminar os candidatos que optaram pela Área", seguido de um menu suspenso (indicado pelo número 1). Abaixo, "e que na disciplina", seguido de outro menu suspenso (indicado pelo número 2). Em seguida, "tenham obtido nota", seguido de um menu suspenso (indicado pelo número 3) e "a", seguido de um campo de entrada de texto. Na base do formulário, há dois botões: "Gravar" (indicado pelo número 4) e "Sair" (indicado pelo número 5).

Figura 26 - Tela: Incluir Critérios.

Relacionamento com outras interfaces:

1. Campo onde serão exibidas as áreas.
2. Campo onde serão exibidas as disciplinas.
3. Campo onde serão exibidos os pesos.
4. Permite ao usuário gravar um novo critério no sistema.
5. Permite retornar a tela principal do sistema.

TELA CRITÉRIOS



Figura 27 - Tela: Critérios cadastrados.

Relacionamento com outras interfaces:

1. Nome da área a ser selecionada.
2. Nome da disciplina a ser selecionada
3. Critério a ser selecionado.
4. Valor a ser selecionado.
5. Campo onde serão exibidas as áreas, disciplinas e critérios que foram cadastrados.
6. Permite alterar os campos.
7. Permite excluir os campos desejados.
8. Permite retornar a tela principal do sistema.

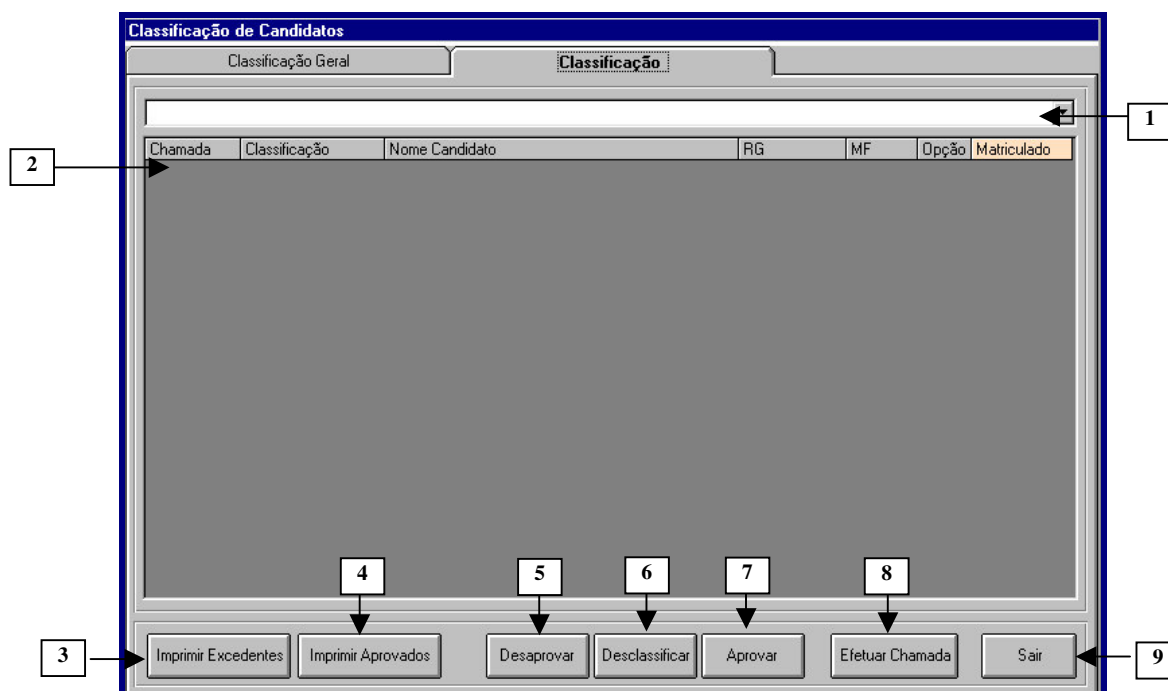
TELA CLASSIFICAÇÃO DE CANDIDATOS - COLOCAÇÃO

Figura 28 - Tela: Classificação de candidatos – colocação.

Relacionamento com outras interfaces:

1. Campo onde será selecionada a área específica.
2. Campo que exibirá todas informações da classificação da área selecionada.
3. O botão **Imprimir Excedentes** permite visualizar a tela “Imprimir Excedentes”, com sua devida opção: Imprimir listagem de excedentes por curso e chamada.
4. O botão **Imprimir Aprovados** permite visualizar a tela “Imprimir Aprovados”, com as devidas opções Listagem para candidato A-Z; Listagem para secretaria; listagem de classificação; Aprovados Manualmente.
5. Permite ao usuário desclassificar um determinado candidato manualmente.
6. Permite ao usuário desclassificar um determinado candidato.
7. Fica a critério do usuário aprovar um candidato.
8. Faz o processo de classificação (1ª, 2ª e as demais chamadas).
9. O botão **Sair** permite retornar a tela principal do sistema.

Módulo Internet

TELA INICIAL – Processo Seletivo

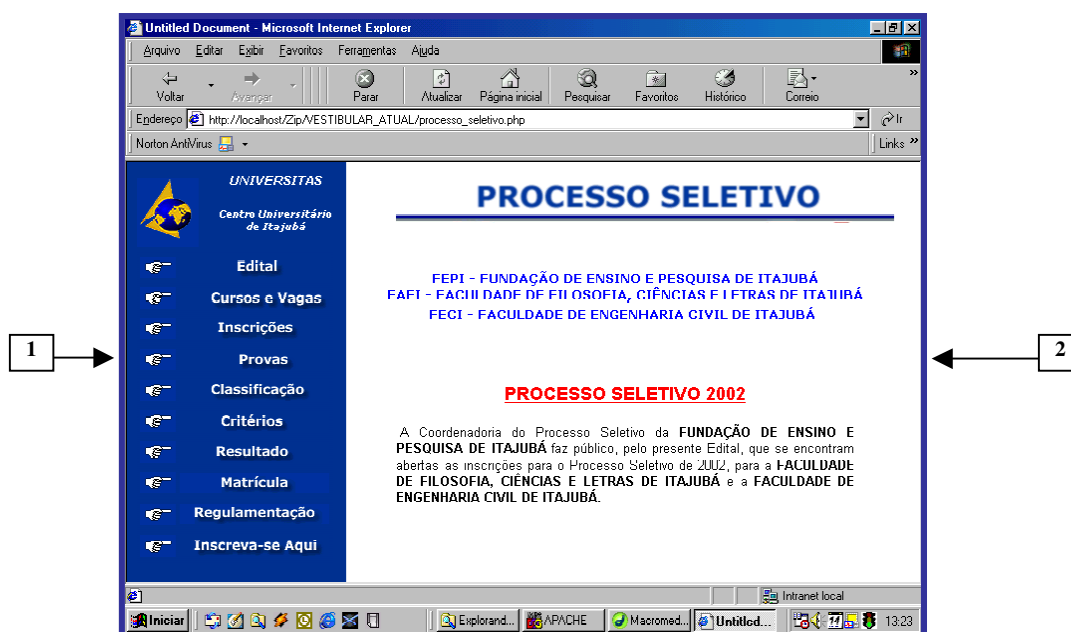


Figura 29 - Tela: Inicial.

1. Todos os **links** oferecidos na área de visitação do candidato.
2. Campo que exibirá todas informações que forem selecionadas.

TELA INSCRIÇÃO: Opção

1. Nesta tela o candidato terá a opção de escolher as seguintes áreas: **Exatas** ou **Humanas**.



Figura 30 - Tela: Inscrição –Opção.

TELA INSCRIÇÃO – Ficha Cadastral

Figura 31 - Tela: Inscrição – Ficha Cadastral.

1. Mostra a opção escolhida pelo candidato no ato da inscrição.
2. Depois da escolha pela área, o candidato deverá indicar a sua ordem de preferência, podendo fazer até quatro opções. Mas só poderá escolher cursos de uma única área: **Humanas** ou **Exatas**.
3. Todos os campos do formulário tanto o de **Exatas** como o de **Humanas** devem ser preenchidos **OBRIGATORIAMENTE**.

TELA INSCRIÇÃO – Questionário

Figura 32 - Tela: Inscrição – Questionário.

1. Nesta tela o candidato deverá preencher todos os campos, depois de preenchidos deverá clicar no botão **ENVIAR DADOS**, se todos os campos forem preenchidos corretamente abrirá uma página confirmando que os dados foram enviados com sucesso, caso algum dos dados tiver faltando ou for preenchido de forma incorreta abrirá uma página mostrando qual ou quais dados estão faltando e/ou incorretos.

TELA CLASSIFICAÇÃO

Nesta tela o candidato terá a opção de consultar sua classificação no processo seletivo. O candidato poderá visualizar os resultados gerais ou por área. Cada lista de classificação pode ser impressa pelo candidato.

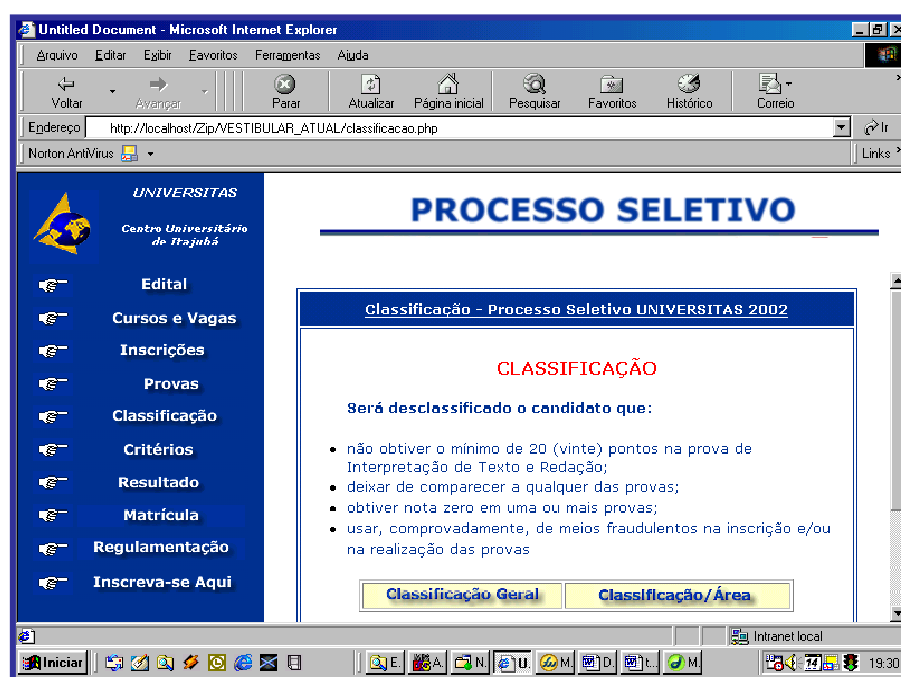


Figura 33 - Tela: Classificação.

TELA CLASSIFICAÇÃO POR ÁREA

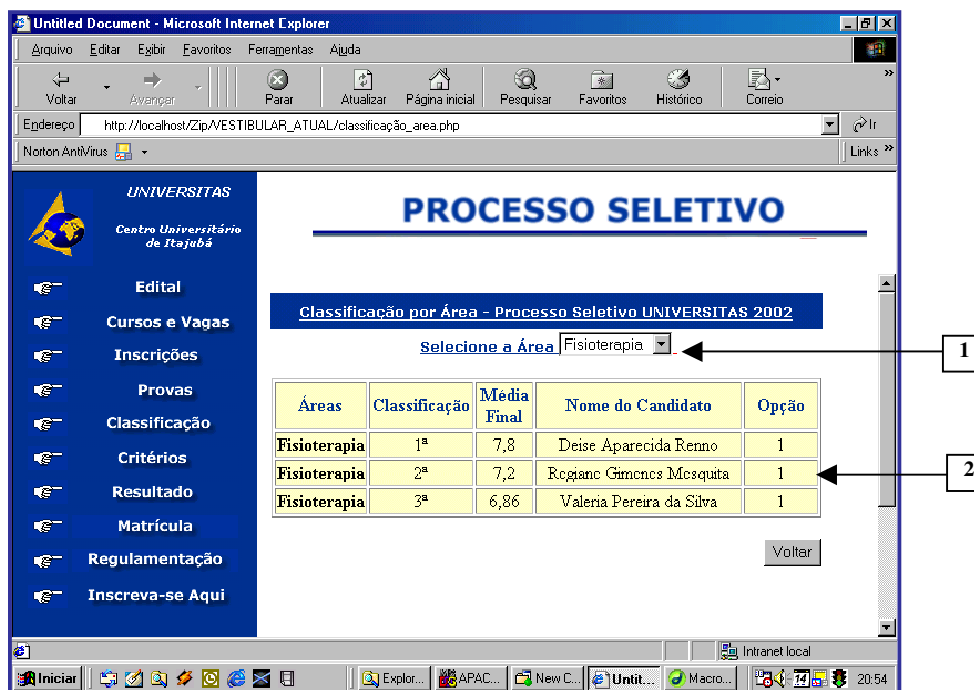


Figura 34 - Tela: Classificação por área.

1. O candidato deverá selecionar a área desejada para a visualização da listagem de classificação.
2. Campo onde serão listados todos os candidatos classificados de acordo com área que foi selecionada.

3.4.12 Desenvolvimento do Diagrama de Seqüência

O diagrama abaixo esta relacionado com o use Case: Controle de candidatos - Inscrição do candidato (Intranet):

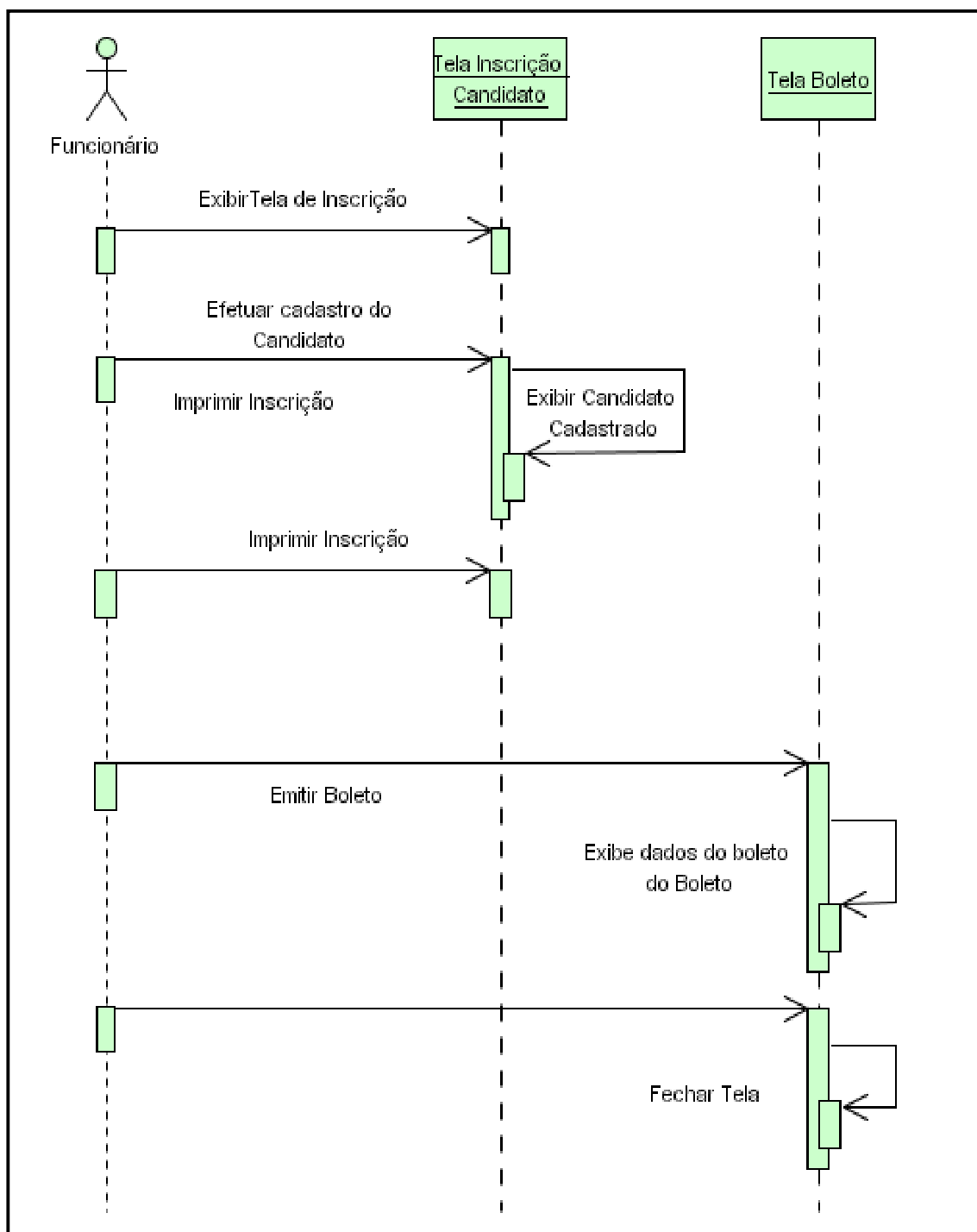


Figura 35 - Diagrama de seqüência – Controle de candidatos.

O diagrama abaixo está relacionado com o use case: Controle de candidatos - confirmar inscrição (Intranet):

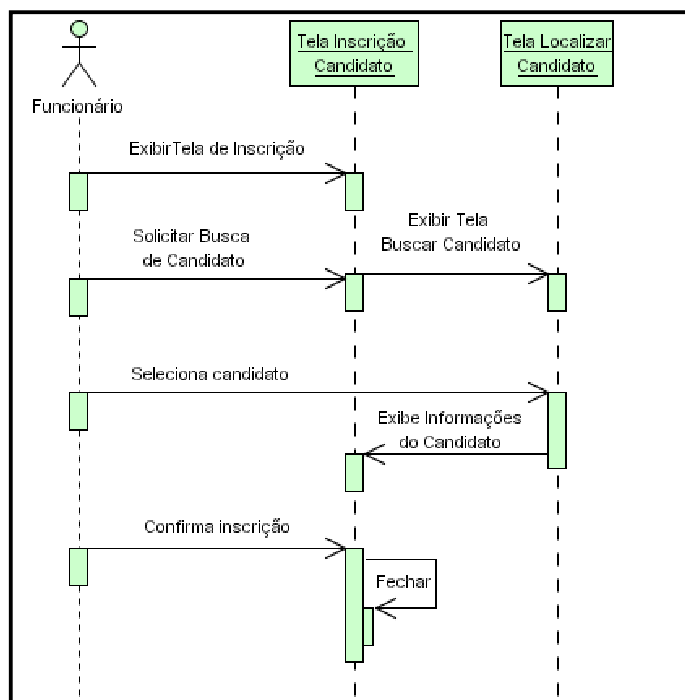


Figura 36 - Diagrama de seqüência – confirmar inscrição.

O diagrama abaixo está relacionado com o use case: Controle de candidatos – Inscrição do candidato (Internet):

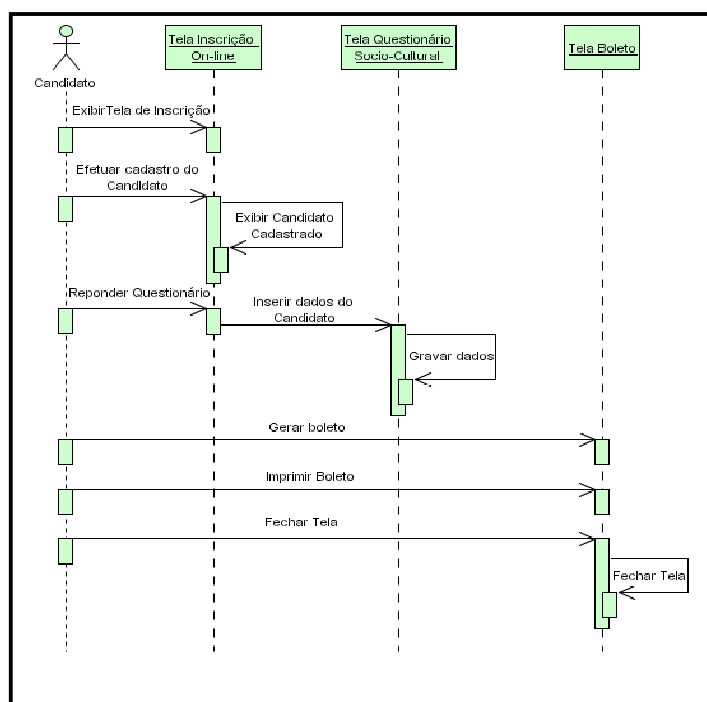


Figura 37 - Diagrama de seqüência – Inscrição do candidato.

3.4.13 Requisitos não funcionais

Os requisitos não funcionais são requisitos que declaram restrições, ou atributos de qualidade para um software e/ou para o processo de desenvolvimento deste sistema. Como: segurança, desempenho, usabilidade, funcionalidade, manutenibilidade e confiabilidade são exemplos de requisitos não funcionais.

Segurança: As medidas de segurança básica aplicável ao sistema são descritas abaixo:

- Gerar cópias de segurança (backup). O armazenamento de todos os dados vital do sistema deve ficar em outra localidade, e se possível, em outro prédio.
- Uso de senhas: o sistema possuirá um controle por meio de senhas, de maneira a impedir que usuários não autorizados venham a obter informações confidenciais, alterar ou destruir dados;
- Conscientização dos funcionários sobre a importância do sigilo e da privacidade a respeito do sistema.
- Uso de software antivírus.
- Uso de NO-BREAK e/ou Estabilizadores para evitar perdas devido à falta ou problemas de energia elétrica.

Desempenho

Tempo de resposta para consulta no BD : O tempo para realização de qualquer operação de consulta de candidatos, áreas salas, pesos, cursos, não pode ser maior do que 10 segundos.

Tempo de resposta para cadastro : O tempo para realização de qualquer operação de cadastro de candidatos, áreas, salas, pesos, cursos, notas, disciplinas e critérios não pode ser maior do que 10 segundos.

3.5 FASE DA CONSTRUÇÃO

3.5.1 Projeto da arquitetura do ambiente

Para atender os requisitos, o ambiente foi modelado seguindo a modelagem UML, conforme apresentada no capítulo anterior, e são constituídos dos seguintes componentes da Internet e Intranet.

Internet: Cliente www, servidor http, bases de dados e protocolos de comunicação, manipulados através da linguagem PHP).

Tabela 11 - Requisitos de arquitetura – (Internet).

Restrição	Descrição
Sistema Operacional Linux	Escolhido pela segurança que o sistema oferece.
Servidor Web	O Apache foi escolhido por ser open source.
Segurança	A restrição de acessos deverá ser feita através de senhas que serão fornecidas aos usuários.
Camada de BD	O MySQL foi escolhido, principalmente pela sua facilidade.
Camada de Interface com usuário	Será usada HTML, pelo fato de ser interpretada por qualquer browser.
Camada Middleware	Adotamos a linguagem PHP, pela sua simplicidade, já que os scripts ficam no próprio código HTML, e possuem uma sintaxe bastante simples.

Intranet: (VB 6.0, ODBC, que possibilita a conexão com o banco de dados relacional Mysql).

Tabela 12 - Requisitos de arquitetura – (Intranet).

Restrição	Descrição
Sistema Operacional do Software	Sistema operacional compatível – Windows 9x.
Ambiente	Configuração mínima para execução do sistema: Celeron 300 MHz, 64 RAM. Com impressora de tecnologia laser ou jato de tinta, para ser usada nas impressões de relatórios, boletos e listagens relacionadas ao processo seletivo.
Servidor BD	O servidor onde ficará o BD será usado o linux, foi escolhido por medidas de segurança.
Linguagem	Será utilizada o VB, escolha da Instituição.

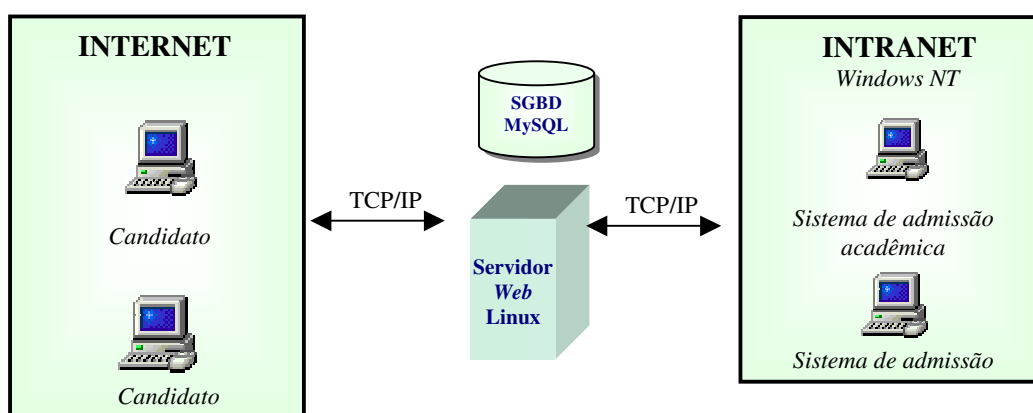


Figura 38 - Arquitetura genérica do ambiente.

3.5.2 Desenvolvimento do diagrama de colaboração – Intranet

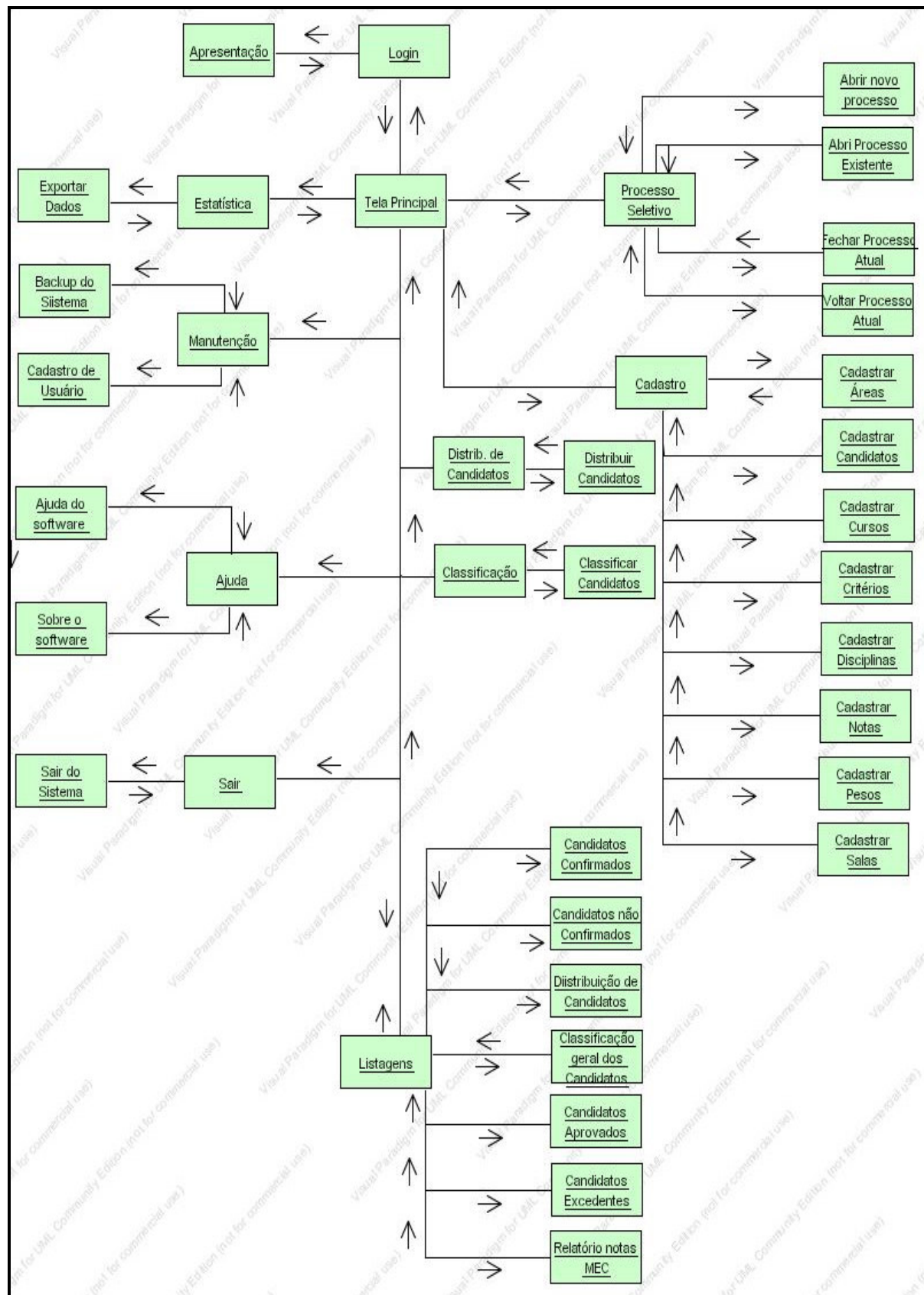


Figura 39 - Diagrama de colaboração.

Este diagrama nos auxilia no entendimento de um comportamento de um conjunto de objetos que trocam mensagens entre si para realizar um propósito em uma interação global, através deste diagrama podemos ver somente os objetos e as mensagens envolvidas, conforme é mostrado na figura39.

3.5.3 Subsistemas e Componentes

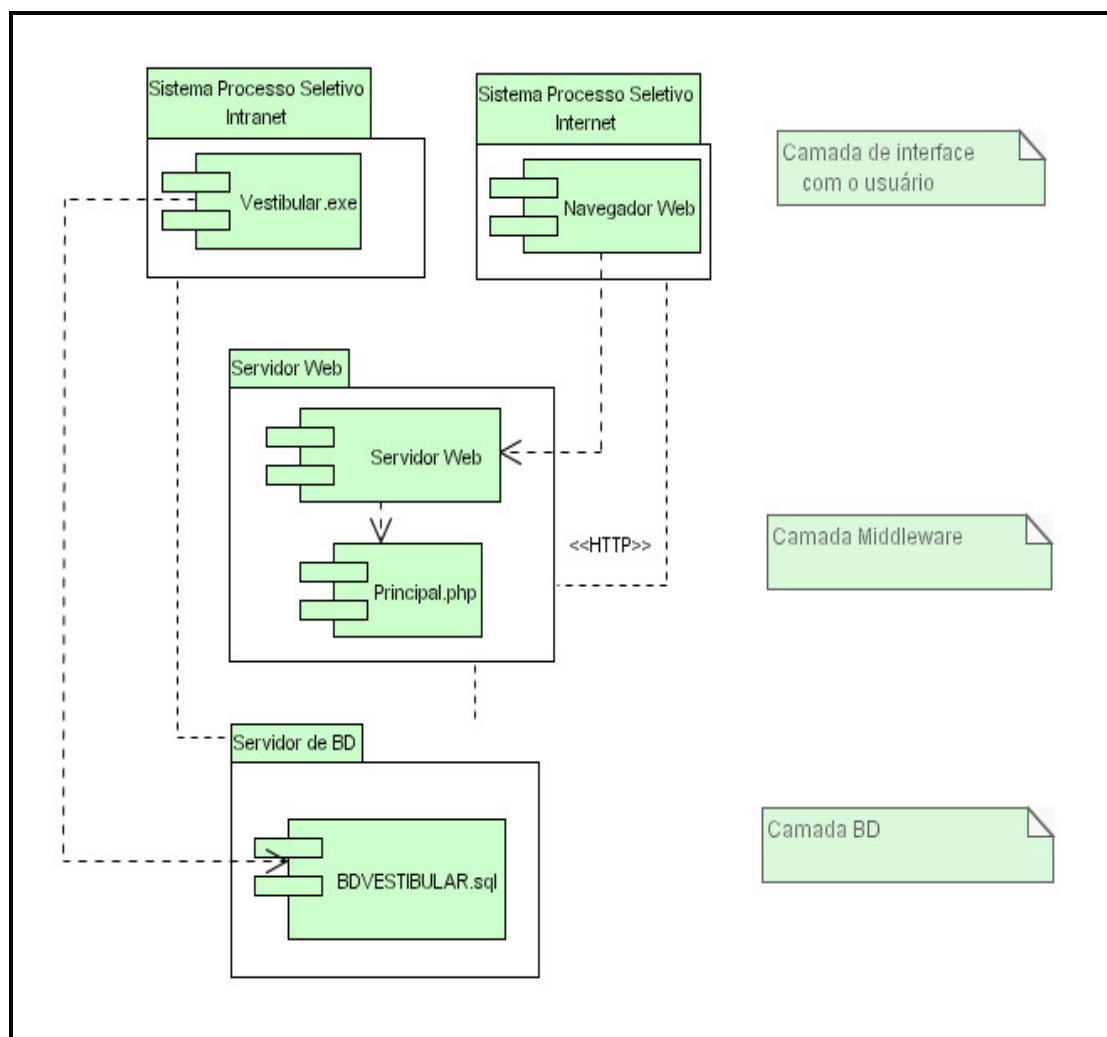


Figura 40 - Subsistemas e componentes

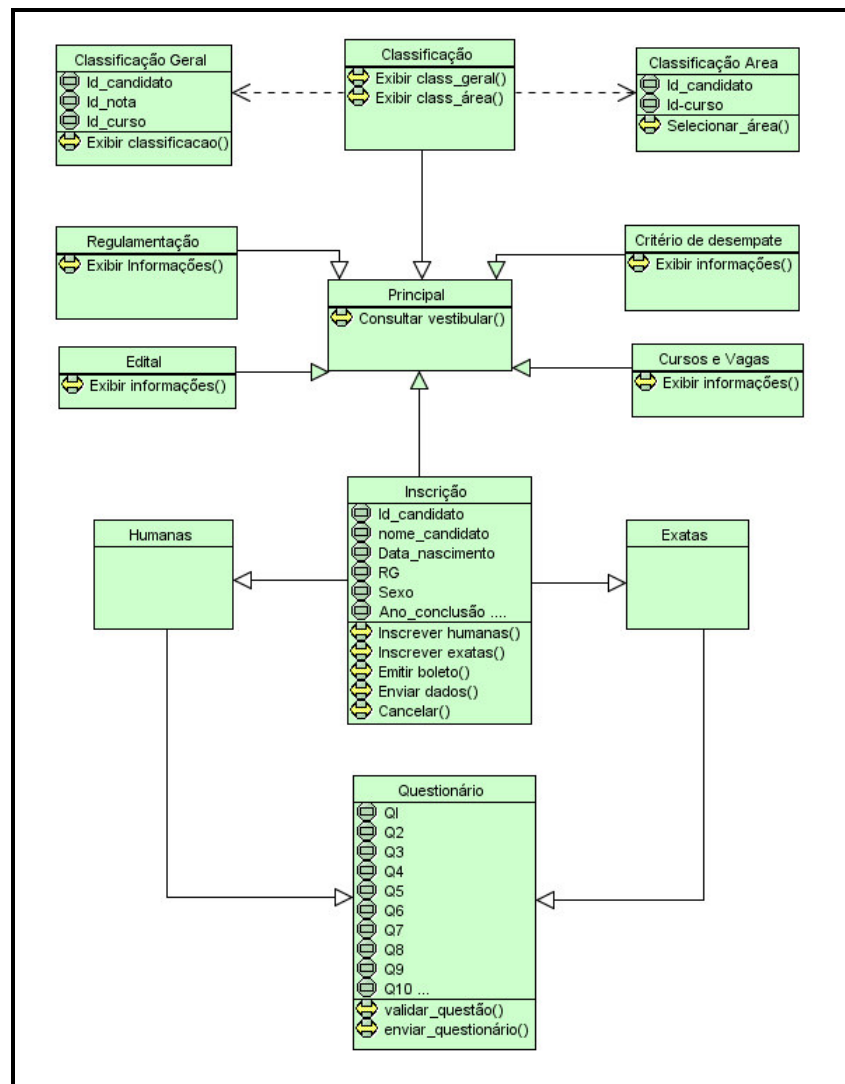
Diagrama de interface (Internet)

Figura 42 - Diagrama de Interface - Internet

3.5.4 Modelo de implementação

Os diagramas de implementação modelam os aspectos físicos do sistema. E se divide em: Diagramas de implantação e Diagramas de componentes conforme é mostrado abaixo:

Desenvolvimento do diagrama de implantação

São diagramas empregados para modelagem da visão estática de implantação de um sistema. Este diagrama nos mostra a organização do hardware e a ligação do software aos dispositivos físicos.

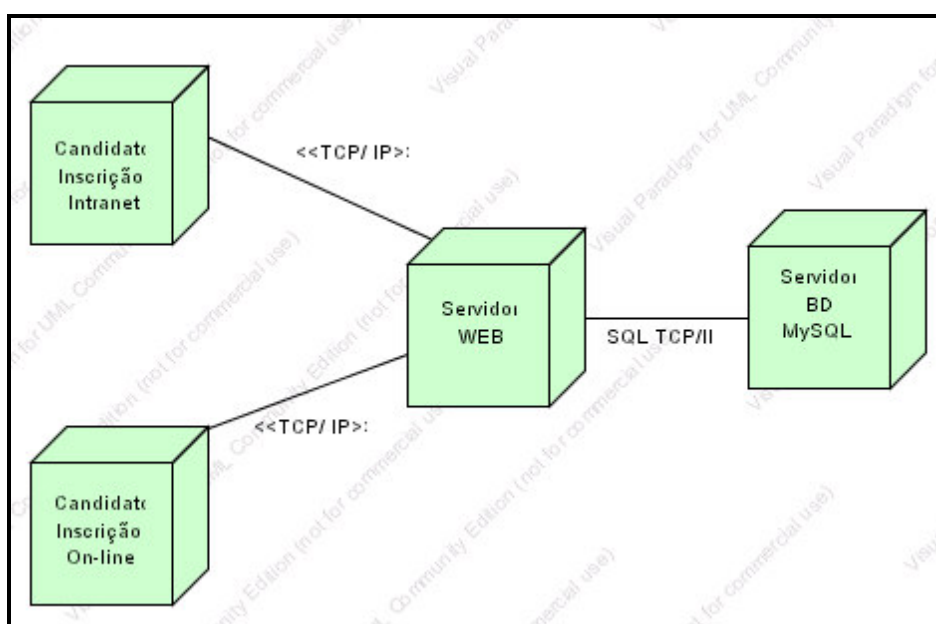


Figura 44 - Diagrama de implantação.

Desenvolvimento do diagrama de componentes

O diagrama de componente é utilizado para modelar o código fonte, versões de códigos executáveis e a estrutura física do banco de dados (bancos de dados, tabelas), conforme é mostrado abaixo:

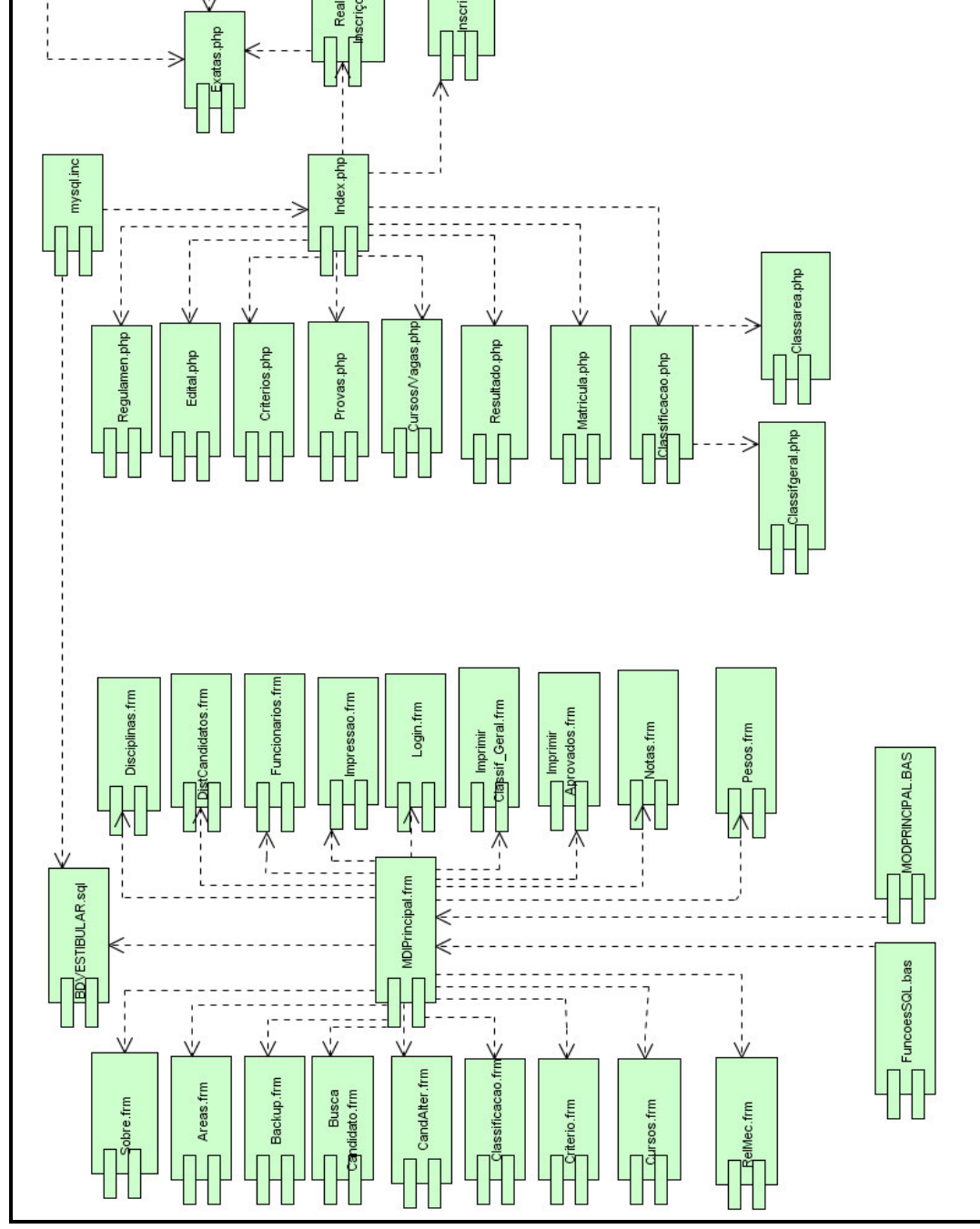


Figura 45 - Diagrama de componentes

3.5.5 Plano de testes

O teste é o elemento de grande importância para o controle de qualidade do software, pois pode garantir a confiança do desempenho das funções específicas. O tempo a ser gasto no planejamento deve ser visto como investimento.

Objetivos do teste

Testar a funcionalidade, completeza e correção da implementação do Sistema de admissão acadêmica, comparando-o com a respectiva da especificação de requisitos.

Escopo do teste

Os use cases foram todos testados, conforme é mostrado na tabela abaixo, e os resultados totalmente satisfatórios. Os erros encontrados durante os testes foram todos corrigidos. Não será documentado o teste relacionado à instalação e configuração do SGBD e das impressoras utilizadas pelo produto.

Tabela 13 - Relações dos use cases testados.

<i>Use case</i>	<i>Funcionou</i>	<i>Não Funcionou</i>
Cadastrar usuário	✓	
Autenticar usuário	✓	
Realizar Estatísticas	✓	
Controlar cursos	✓	
Distribuir Salas/blocos	✓	
Realizar Manutenção	✓	
Controlar disciplinas	✓	
Controlar áreas	✓	
Controlar candidatos	✓	
Distribuir cand./sala	✓	
Definir Pesos	✓	
Inscrever Candidato	✓	
Estipular critérios	✓	
Emitir relatórios	✓	
Gerar backup	✓	
Controlar notas	✓	
Controlar classificação	✓	
Divulgar edital	✓	
Efetuar inscrição on-line	✓	

Continuação da Tabela 13 - Relações dos use cases testados

<i>Use case</i>	<i>Funcionou</i>	<i>Não Funcionou</i>
Divulgar cursos e vagas	✓	
Inscriver-se	✓	
Divulgar provas	✓	
Consultar classificação	✓	
Exibir critérios de desempate	✓	
Exibir resultado	✓	
Divulgar matrícula	✓	
Exibir regulamentação	✓	

Especificações de teste

Mostraremos o teste detalhado realizado de apenas um use case (como exemplo):

- Especificação do teste do use-case Controlar candidatos.

Identificação dos testes

Procedimentos de teste

Tabela 14 - Procedimentos de teste.

NÚMERO	PROCEDIMENTO DE TESTE
1	Inscriver candidato.
2	Consultar inscrição.
3	Confirmar inscrição.
4	Excluir candidato.
5	Emitir boleto.

CrITÉrios de completiza e sucesso

Tabela 15 - CrITÉrios de completiza e sucesso.

NÚMERO	CRITÉRIOS
1	Funcionou corretamente o cadastro de candidatos feito pelo funcionário.
2	Depois de cadastrado os candidatos a função consulta também funcionou corretamente, consultando assim as inscrições pelo nome do candidato.
3	Após o preenchimento da ficha cadastral, o sistema confirma a inscrição do candidato que é feita pelo funcionário. Está função de confirmação também funcionou corretamente, ou seja, atendeu a todos os requisitos do sistema.

Continuação da Tabela 15 - Critérios de completeza e sucesso

NÚMERO	CRITÉRIOS
4	Todos os candidatos excluídos foram corretamente eliminados do banco de dados, ou seja, funcionou corretamente.
5	A emissão do boleto após a confirmação da inscrição também atendeu a todos os requisitos do sistema, funcionando corretamente.

Procedimento de teste < nome do procedimento: Inscrever candidato >

Tabela 16 - Procedimento de teste: Inscrever candidato.

Identificação	Use case Controlar candidato - Inscrever candidato.
Objetivo	Verificar se a inclusão de um candidato que foi cadastrado pelo funcionário foi feita corretamente no “Sistema Processo Seletivo”.
Requisitos especiais	Nenhum.
Fluxo	Abrir interface: Tela de Inscrição do candidato. Preencher campos: Nome, CPF,... Acionar Incluir. Acionar Imprimir. Acionar Imprimir Boleto.

Procedimento de teste: < nome do procedimento: Consultar Inscrição >

Tabela 17 - Procedimento de teste: Consultar inscrição

Identificação	Use case Controlar de candidato – Consultar inscrição.
Objetivo	Verificar se a consulta de um candidato é feita corretamente no “Sistema de admissão acadêmica”.
Requisitos especiais	Nenhum.
Fluxo	* Acionar Localizar . * Abre Interface Tela Busca Candidato . * Digitar <u>N</u> ome ou <u>R</u> G para consulta. * Selecionar candidato desejado para consulta.

Capítulo 4 Resultados Obtidos

4.1 INTRODUÇÃO

Neste capítulo serão apresentadas as vantagens de cada fase do processo de desenvolvimento do sistema.

Após a implementação do sistema, foi possível obter alguns resultados os quais julgamos ser necessário destacar. A avaliação não deverá se limitar aos resultados dos dados gerados, mas também na avaliação das metas propostas relacionadas aos requisitos não funcionais do sistema, como o desempenho, segurança e outros fatores que influenciam na qualidade do sistema.

Através dos resultados obtidos da avaliação do sistema, novas funcionalidades sugeridas pelos usuários poderão ser implementadas. Esse procedimento deve ser repetido indefinidamente, pois as tecnologias mudam da mesma forma que as necessidades da organização e do público-alvo.

Eventos Iniciais:

De um modo geral atendeu as necessidades, pois desde o primeiro contato com o cliente, tudo foi registrado, facilitando as próximas etapas.

Entrevista:

Com os tópicos abordados na reunião, os objetivos gerais e específicos pretendidos com o sistema puderam ser obtidos.

Na entrevista com os usuários, todos os requisitos necessários para a criação da documentação foram especificados. As metas que o sistema deverá atingir foram claramente definidas.

Documentação:

A criação do documento: Plano de desenvolvimento do sistema auxiliou na comunicação entre usuários e o projetista.

Todas as metas definidas na fase de entrevista foram documentadas, servindo de base para consulta de informações relacionadas ao cronograma de desenvolvimento do sistema.

Análise e projeto:

O uso da UML para a modelagem do sistema auxiliou no entendimento de todas as funções que o sistema deverá executar. Os diagramas de use cases foram bem detalhados, de modo que qualquer pessoa envolvida no projeto entendesse o seu funcionamento. O detalhamento dos use cases e os diagramas de seqüências facilitaram no desenvolvimento durante a fase de implementação.

Seguindo as definições da metodologia, o diagrama de classe foi facilmente desenvolvido e compreendido.

A definição de subsistemas serviu como base para o entendimento dos usuários em relação às ferramentas que deveriam ser adquiridas.

O diagrama de implantação auxiliou no entendimento da estrutura física, bem como informações essenciais para o software como, por exemplo, o servidor de banco de dados.

O diagrama de componentes auxiliou na estruturação do código fonte, separando seus objetos de acordo com sua finalidade.

Implementação:

Para a implementação do sistema, considera-se que o VB foi a principal linguagem de desenvolvimento e pôde-se observar que o seu desempenho e facilidade de uso supriram todas as necessidades exigidas.

Também deve ser destacada a utilização de linguagem/ferramenta para a *Web*, como é o caso do PHP, que permitiu a interação entre BD e o usuário, mostrou seu ótimo desempenho e principalmente sua flexibilidade, pois o PHP tem a liberdade de escolha em relação ao sistema operacional e suporte a uma ampla variedade de banco de dados.

A documentação do código-fonte foi de grande importância durante o desenvolvimento do sistema, pois na criação de trabalhos futuros, este poderá ser de grande utilidade.

Em relação ao banco de dados, a escolha é quase sempre guiada pelas necessidades de desempenho, segurança e principalmente pelo custo. Como já visto no capítulo anterior, bancos de dados *freewares* ou *open-source*, foi à opção para o projeto, o MySQL é um exemplo e por isso que foi o motivo pelo qual foi escolhido, obviamente, pelo fator financeiro, visto que o custo para a implementação desse BD é praticamente nulo, quando comparado aos demais.

Manual do usuário:

Com o objetivo de apoiar a fase de implementação, um esboço do Manual do usuário foi elaborado. O objetivo desse manual é orientar o futuro usuário do sistema no que se refere à utilização do programa, explicitando o seu objetivo e os recursos oferecidos.

O manual do usuário contribuiu para a diminuição do tempo e esforço necessário para que o usuário atingisse o nível de desempenho elevado.

4.2 RESULTADOS DA AVALIAÇÃO

Público alvo: Como pôde ser analisada, através dos capítulos anteriores, a ferramenta foi desenvolvida com objetivo de atender as necessidades da Instituição proporcionando maiores benefícios aos funcionários e candidatos.

4.2.1 Avaliação dos funcionários em relação ao Sistema de Admissão Acadêmica

Para uma melhor análise da opinião dos funcionários em relação ao sistema, foi aplicado um formulário apresentado abaixo.

Tabela 18 - Resultado obtido - funcionários do setor administrativo.

FORMULÁRIO APLICADO A QUINZE FUNCIONÁRIOS ENVOLVIDOS NO PROCESSO SELETIVO.			
Característica – Requisitos não funcionais	Desempenho		
	Razoável	Bom	Ótimo
Usabilidade			
Facilidade de uso	3	3	9
Facilidade de aprendizado	4	5	6
Satisfação do usuário	2	2	11
Flexibilidade	1	7	7
Produtividade	1	5	9
Funcionalidade	Razoável	Bom	Ótimo
Geração de resultados	2	3	10
Capacidade de interagir com outro sistema	3	4	8
Segurança de acesso	1	3	11
Manutenibilidade	Razoável	Bom	Ótimo
Facilidade de diagnosticar causas de falhas	4	4	7
Facilidade de remoção e modificação dos erros	3	4	8
Estabilidade	3	5	7
Testabilidade	2	6	7
Confiabilidade	Razoável	Bom	Ótimo
Frequência de falhas	3	3	6
Tolerância à falhas	3	3	6

Recuperabilidade	2	6	7
Eficiência	Razoável	Bom	Ótimo
Tempo de resposta da operação solicitada pelo usuário	2	4	9
Tempo de resposta para pesquisa no BD	4	6	5
Tempo relacionado à utilização de recursos do sistema	3	4	8

Características dos requisitos não funcionais:

Tabela 19 - Características dos requisitos não funcionais.

Usabilidade	Evidencia a facilidade para utilização do software.
Funcionalidade	Conjunto de funções que atende às necessidades para a finalidade a que se destina o produto.
Manutenibilidade	Facilidade de correções, alterações e atualizações do sistema.
Confiabilidade	Desempenho mantido ao longo do tempo e em condições estabelecidas.
Eficiência	Tempo envolvido são compatíveis com o nível de desempenho requerido para o produto.

Certas informações que não são apresentadas graficamente também possuem grande importância na pesquisa realizada, como é o caso dos resultados obtidos em cada fase do processo de desenvolvimento do sistema, conforme foi citado na seção anterior.

São apresentadas abaixo, em forma de gráficos, as opiniões dos funcionários em relação aos requisitos de qualidade do software.

Usabilidade do Sistema

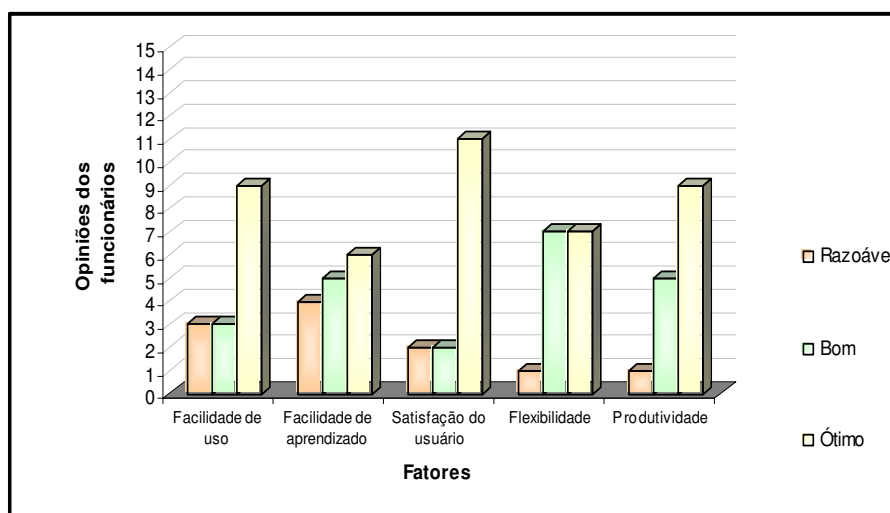


Figura 46 - Usabilidade do Sistema

De acordo com o gráfico apresentado, os funcionários demonstraram uma grande satisfação com o novo sistema e surgiram algumas dificuldades normais no período de adaptação, que foram e serão eliminadas gradativamente conforme a utilização do sistema.

Funcionalidade do Sistema

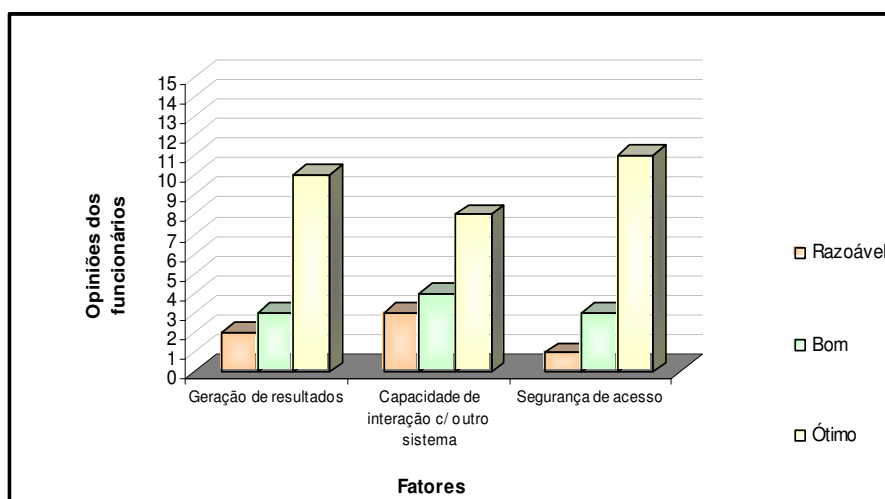


Figura 47 - Funcionalidade do Sistema.

Como foi apresentado acima, o sistema demonstrou-se bastante seguro, gerando bons resultados e interagindo bem com outros sistemas.

Manutenibilidade do Sistema

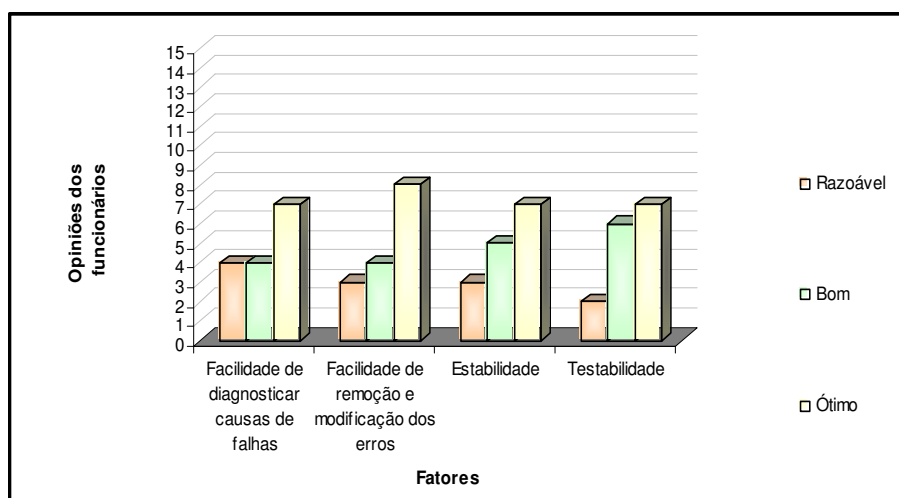


Figura 48 - Manutenibilidade do Sistema.

Através do gráfico apresentado acima, foi possível verificar que o sistema teve uma boa aceitação entre os funcionários. O outro dado apresentado foi a facilidade com que o sistema teve em diagnosticar as falhas ocorridas durante os testes e suas causas. Essa característica apresentada pelo sistema faz com que o sistema se torne mais estável e de fácil manutenção.

Confiabilidade do Sistema

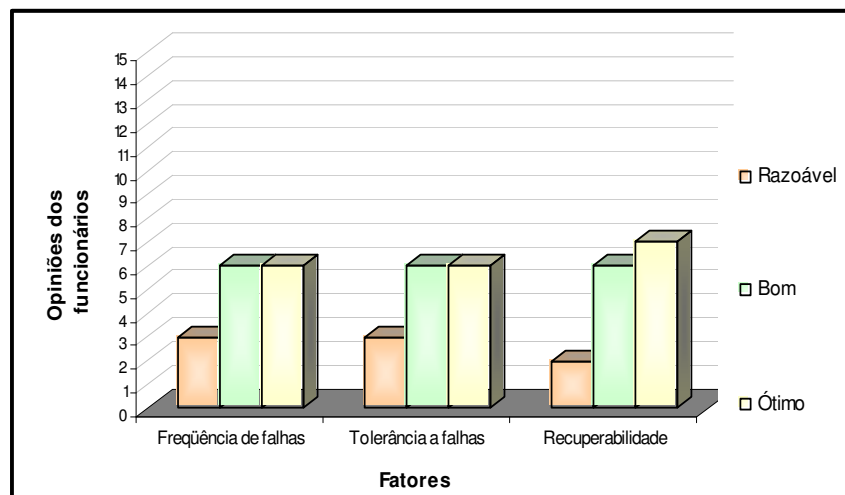


Figura 49 - Confiabilidade do Sistema.

O sistema não apresentou muitas falhas e as que surgiram puderam ser contornadas com o pronto restabelecimento e restauração dos dados. Sendo assim considerado confiável.

Eficiência do Sistema

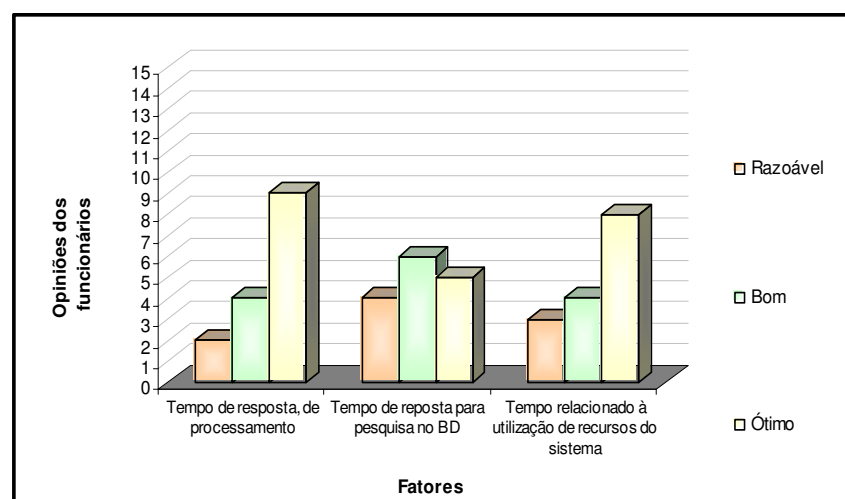


Figura 50 - Eficiência do Sistema.

Como mostra o gráfico acima, os recursos oferecidos pelo sistema foram suficientes, resultando um ótimo tempo de processamento e de pesquisa ao banco de dados, tornando seu desempenho e eficiência satisfatórios.

4.2.2 Avaliação dos candidatos em relação ao sistema de inscrição on-line.

Com o sistema de inscrição on-line o candidato pôde fazer sua inscrição pela Internet, efetuando o pagamento da taxa de inscrição em uma agência bancária da sua própria cidade. Depois confirmar sua inscrição enviando o documento (boleto padrão da Instituição) autenticado pela agência bancária por email ou fax. Esse sistema proporcionou ao candidato um maior conforto e comodidade.

Para comprovar estes dados foi realizada uma pesquisa com os candidatos inscritos no processo seletivo, pedindo sua opinião sobre o sistema de inscrição on-line. Foram levantados os seguintes resultados:

- 80% das opiniões acharam o sistema prático, facilitando as inscrições.
- 15% encontraram dificuldades em realizar suas inscrições via *Web*.
- 5% não opinaram sobre o assunto.

Graficamente:

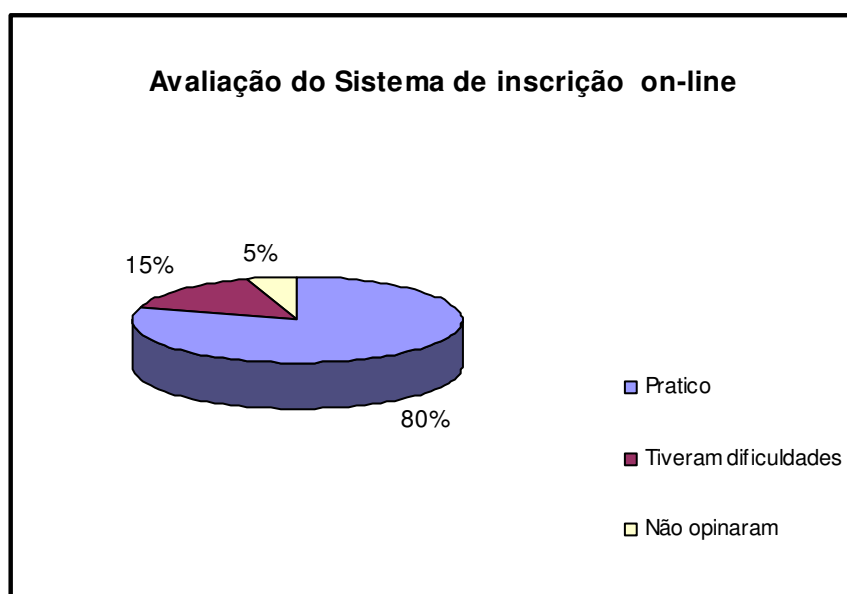


Figura 51 - Avaliação do Sistema on-line.

Capítulo 5 Conclusões

Este capítulo sumariza as conclusões advindas deste estudo. E também são sugeridos temas para futuras pesquisas que complementam e dão prosseguimento ao projeto iniciado com este trabalho.

5.1 CONCLUSÃO

A proposta desse capítulo é apresentar as conclusões obtidas com a aplicação dos conceitos relacionados à gestão acadêmica e métodos de engenharia de software no desenvolvimento do sistema de admissão acadêmica.

O sistema de admissão acadêmica surgiu devido às dificuldades encontradas pelo DRA de se obter informações corretas, ou seja, adequadas às necessidades em um tempo ágil e com custos compatíveis. Através da abordagem de planejamento estratégico na IES, foi possível elaborar um plano de gestão acadêmica e identificar as metas que deveriam ser atendidas pelo sistema proposto.

Os tópicos relacionados à engenharia de software abordados no capítulo 2 forneceram a fundamentação teórica necessária para que o sistema pudesse ser desenvolvido com qualidade a fim de atender às necessidades dos usuários.

Com a implementação do sistema de admissão acadêmica foi possível afirmar que houve uma contribuição significativa na intenção de otimizar os serviços prestados e de torná-los cada vez mais acessíveis a todos os usuários através de meios tecnológicos modernos.

O foco principal foi incentivar a criação e melhoria de sistemas de controle de admissão, com interfaces amigáveis e ergonômicas, na intenção de diminuir a distância entre usuário e sistema, através da informatização sem perda de confiabilidade.

A proposta desenvolvida permitiu, em todos os processos realizados na secretaria, o abandono do físico (papel, por exemplo) pelo simbólico (digitalização de documentos e armazenamento em bancos de dados).

O sistema proposto uma vez implantado, permitiu mais rapidez no armazenamento e recuperação dos dados dos candidatos inscritos, agilizando os processos envolvidos na realização do vestibular, eliminando-se, assim, a principal fonte causadora de atrasos.

Neste contexto conclui-se que o trabalho apresentou uma melhora significativa para a Instituição, visto que o controle dos alunos inscritos no vestibular, distribuição de alunos por

salas, processo de classificação e a listagem dos alunos aprovados merecem consideração especial, pois, embora não sejam complexos, tem demandado uma grande parcela de tempo e energia despendida pelos funcionários.

Além da secretaria, o departamento financeiro também se beneficiou com a automação do sistema, uma vez que permite que o boleto de pagamento das inscrições seja emitido através da Internet, tornando sua retirada mais cômoda para o aluno e economizando o tempo do funcionário responsável pela sua entrega, ou economizando custo da impressão do documento.

5.2 SUGESTÕES PARA PRÓXIMOS TRABALHOS

Fica de incentivo para trabalhos futuros a continuação do estudo e a possível realização de alguns trabalhos complementares, dentre eles a criação das funções descritas abaixo:

- Implementação de rotina de emissão de etiquetas com o código de barras para identificação de candidatos.
- Correção das notas via interface com leitora ótica.
- Utilização de notas do ENEM e de outros processos seletivos para classificar os alunos.
- Emissão de boleto integrado ao sistema bancário.

Capítulo 6 Referências bibliográficas

[ALVIM, 1998] ALVIM, P. C. R. *O papel da informação no processo de capacitação tecnológica das micro e pequenas empresas*. Ciência da Informação, Brasília, v.27 n.1, Janeiro 1998.

[ÁVILA, 1990] AVILA, J. de. *O papel do marketing nas universidades brasileiras: Um estudo prospectivo fundamentado no método "DELPHI"*. Porto Alegre, 1990. Dissertação (Mestrado em Administração) – UFRS,1990.

[BAUER, 1972] BAUER, F.L. *Software Engineering*. Information Processing, 1972.

[BORGES, 1993] BORGES, A. *Inteligência empresarial informacional*. Rio de Janeiro, v.33,1993.

[BARROS, 2001] BARROS, P. *Linguagem de Modelagem Unificada (em português)*. Disponível em <http://www.uml.com.br/arquivos/tutoriais/apostilas%20UML.doc> Acesso em 10/11/2002.

[BOEHM, 1988] BOEHM, B.W. *A Spiral Model of Software Development and Enhancement*. IEEE Computer, v. 21, n. 5, May 1988.

[BOOCH, 2000] BOOCH G, et al. *UML: Guia do Usuário, O mais avançado tutorial sobre Unified Modeling Language*. Rio de Janeiro. Campus, 2000.

[DAVIS, 1998] DAVIS, M.W. *Computerizing Healthcare Information: Developing Electronic Patient Information Systems*. Revised edition. New York: Mcgraw-Hill, 1998.

[ERIKSSON,1998] ERIKSSON, H. E. & PENKER, M. *UML Toolkit*. Editora Wiley,1998.

FINGER, A. P. *Gestão de Universidades: novas abordagens*. Champagnat, Curitiba, 1997.

[FURLAN, 1998] FURLAN, J.D. *Modelagem de objetos através da UML - The Unified Modeling Language*. São Paulo, Makron Books, 1998.

[IEEE, 1993] IEEE, *Standard Glossary of Software Engineering Terminology*. IEEE Std 610.12-1990, 1993.

[JACOBSON, 1999] JACOBSON, I.; BOOCH, G. and RUMBAUGH, J. *Unified Software Development Process*. Addison-Wesley, 1999.

[LAMB JUNIOR E CRAVENS, 1987] LAMB JUNIOR, C. W., CRAVENS, D.W. *The retrenchment challenge in higher education*. Journal professional services marketing, v.2, n.4, 1987.

[LARMAN, 2000] LARMAN, C. *Utilizando UML e Padrões: Uma introdução a análise e ao projeto orientados a objetos*. Tradução L. A. M. Salgado. Revisão R. T. Price. Porto Alegre, Bookman, 2000.

[LAUDON, 1998] LAUDON, K, C. and LAUDON, J. P. *Management information systems: new approaches to organization & technology*. 5 Edition, USA, Prentice Hall Inc., 1998.

[LEITE E SILVEIRA, 1998] LEITE, E.P.C. e SILVEIRA, A. *Estratégias de marketing em organizações universitárias: estudo da ULBRA e da UNISINOS*. Administração universitária: estudos brasileiros. Florianópolis: Insular, 1998.

[MYSQL, 2002] MySQL. *What is Mysql?* Disponível na rede Internet <http://www.mysql.com>. Acesso em 12/11/2002.

[PAULA, 2001] PAULA W. P. F. *Manual do engenheiro de Software – Métodos Gerenciais*. DDC – UFMG. 2000.

[PENDER, 2002] PENDER, T. A., *UML Weekend Crash Course*. Wiley Publishing, Inc, 2002.

[PRESSMAN, 1995] PRESSMAN, R. S. *Engenharia de software*. São Paulo, Makron Books, 1995.

[PRESSMAN, 2001] PRESSMAN, R. S. *Software Engineering a Practitioner's Approach*. 4 Edition New York: Mc Graw Hill – 2001.

[PHP, 2002] *Manual do PHP*. Disponível na rede Internet <http://www.php.net>. Acesso em 21/11/2002.

[RUMBAUGH, 1994] RUMBAUGH, J.; BLAHA, M.; PREMERLANI, W; EDDY, F. e LORENSEN, W. *Modelagem e Projetos Baseados em Objetos*. Rio de Janeiro, Campus, 1994.

[TOLEDO,1992] TOLEDO, G. L., SILVA, F. S. *Marketing e competitividade*. in: Encontro anual da Associação dos Programas de Pós-Graduação em Administração, 16, 1992, Canela.Anais Salvador, ANPAD, 1992.

[SILVA, 2001] Alberto M. R. VIDEIRA Carlos A. E. *UML, Metodologias e Ferramentas Case*. Lisboa: Centro Atlântico, 2001.

[SOMMERVILLE,2000] SOMMERVILLE, I. *Software Engineering*, 6th ed., Addison-Wesley,2000.

[YOURDON, 1992] YOURDON, E. *Análise Estruturada Moderna*. Rio de Janeiro: Campus, 1992.