

**UNIFEI - UNIVERSIDADE FEDERAL DE ITAJUBÁ**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**



# **UM SISTEMA DE CONTROLE E REGISTRO ACADÊMICO PARA UMA UNIVERSIDADE**

**Vera Cristina Correale Tavares**

**DISSERTAÇÃO DE MESTRADO**

**Itajubá**

**2002**

**Vera Cristina Correale Tavares**

# **UM SISTEMA DE CONTROLE E REGISTRO ACADÊMICO PARA UMA UNIVERSIDADE**

**Dissertação apresentada ao  
Programa de Pós-Graduação em  
Engenharia Elétrica da UNIFEI  
como requisito parcial para obtenção  
do título de mestre em Engenharia Elétrica.**

**Orientador: Prof<sup>o</sup> Otávio Augusto Carpinteiro, Dr.**

**Itajubá  
2003**

Tavares, Vera Cristina Correale. Um Sistema de Controle e Registro Acadêmico para uma Universidade. Itajubá: UNIFEI, 2003. 108 Páginas (Dissertação de mestrado submetida ao Programa de Pós-graduação em Engenharia Elétrica da UNIFEI).

Palavras-chaves: Engenharia de software, técnicas de orientação a objeto, gestão acadêmica, RUP e UML.

## SUMÁRIO

<b>SUMÁRIO.....</b>	<b>I</b>
<b>LISTA DE FIGURAS .....</b>	<b>III</b>
<b>LISTA DE TABELAS.....</b>	<b>IV</b>
<b>LISTA DE ABREVIATURAS.....</b>	<b>V</b>
<b>RESUMO .....</b>	<b>VI</b>
<b>ABSTRACT .....</b>	<b>VII</b>
<b>1 INTRODUÇÃO .....</b>	<b>1</b>
1.1 ASSUNTO ABORDADO .....	1
1.2 A IMPORTÂNCIA DO ESTUDO .....	2
1.3 DESCRIÇÃO DO PROBLEMA .....	2
1.4 OBJETIVOS DO TRABALHO .....	3
1.5 JUSTIFICATIVAS .....	5
1.6 AMBIENTE DE ESTUDO DA PESQUISA.....	5
1.7 DESENVOLVIMENTO DO TRABALHO .....	6
<b>2 FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>7</b>
2.1 TÓPICOS EM GESTÃO ACADÊMICA.....	7
2.1.1 A IES como organização .....	7
2.1.2 Metodologia de gestão das IES.....	9
2.1.3 Fases de implementação do planejamento estratégico nas IES .....	12
2.1.4 Considerações finais.....	16
2.2 TÓPICOS EM SOFTWARE E ENGENHARIA DE SOFTWARE .....	16
2.2.1 Técnicas de Orientação a Objetos - OO.....	17
2.2.2 UML .....	19
2.2.3 Processo de desenvolvimento do software .....	27
<b>3 SISTEMA PROPOSTO DE REGISTRO ACADÊMICO .....</b>	<b>40</b>
3.1 LEVANTAMENTO DE REQUISITOS.....	40
3.2 PROPOSTA DE DESENVOLVIMENTO DO SISTEMA .....	41
3.2.1 Descrição do sistema atual.....	41
3.2.2 Descrição do sistema proposto .....	43
3.2.3 Relação do sistema com o ambiente.....	45
3.3 ANÁLISES DE REQUISITOS DO SOFTWARE .....	46
3.3.2 Escopo do produto .....	46
3.3.3 Missão do produto .....	46
3.3.4 Limites do produto.....	47
3.3.5 Descrição geral do produto.....	47
3.3.6 Requisitos funcionais do sistema.....	48
3.3.7 Requisitos não funcionais .....	66
3.4 PROJETO DA ARQUITETURA DO AMBIENTE.....	67
3.4.1 Modelo de implementação.....	68
3.4.2 Subsistemas e componentes.....	70
3.4.3 Diagrama de componentes.....	74
<b>4 AVALIAÇÃO DA FERRAMENTA DE REGISTRO ACADÊMICO .....</b>	<b>78</b>

4.1	ANÁLISE DOS DADOS .....	78
4.2	RESULTADOS DA AVALIAÇÃO .....	80
4.2.1	Opinião dos usuários em relação ao módulo Internet .....	80
4.2.2	Opinião dos usuários em relação ao módulo Intranet .....	82
4.2.3	Opinião dos usuários em relação ao sistema Online .....	85
<b>5</b>	<b>CONCLUSÕES .....</b>	<b>87</b>
5.1	CONCLUSÃO .....	87
5.2	TRABALHOS FUTUROS .....	88
	<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>89</b>

## LISTA DE FIGURAS

Fig1.	Sistemática de Planejamento Estratégico em Universidades. _____	13
Fig2.	Visões e seus diagramas. _____	20
Fig3.	Diagrama de classe. _____	21
Fig4.	Diagrama de use case. _____	22
Fig5.	Diagrama de seqüência. _____	23
Fig6.	Ciclo de vida clássico. _____	28
Fig7.	Ciclo de vida - Prototipagem. _____	28
Fig8.	Modelo incremental _____	29
Fig9.	Modelo espiral _____	29
Fig10.	Fases do processo RUP. _____	32
Fig11.	Principais workflows. _____	33
Fig12.	Artefatos da análise de requisitos. _____	34
Fig13.	Diagrama de Grantt. _____	45
Fig14.	Diagrama geral de use case. _____	50
Fig15.	Use case - Realizar Matrícula _____	51
Fig16.	Tela de login do usuário. _____	55
Fig17.	Tela de abertura do sistema SisRA. _____	56
Fig18.	Tela Alunos _____	58
Fig19.	Tela Matrícula _____	61
Fig20.	Tela gera turmas. _____	63
Fig21.	Tela do aluno - Consultar notas. _____	65
Fig22.	Tela do professor -Inserir notas. _____	65
Fig23.	Diagrama de seqüência – Realizar matrícula. _____	66
Fig24.	Diagrama de implantação _____	68
Fig25.	Diagrama de colaboração – Módulo Intranet. _____	69
Fig26.	Camadas do sistema. _____	70
Fig27.	Diagrama de interface - Intranet. _____	71
Fig28.	Diagrama de Interface – Sistema Online. _____	72
Fig29.	Diagrama de classe persistente _____	73
Fig30.	Diagrama de componentes. _____	74
Fig31.	Público-alvo do sistema online. _____	78
Fig32.	Qualidade de aplicações Web. _____	80
Fig33.	Requisitos relacionado ao conteúdo do site. _____	81
Fig34.	Requisitos relacionado à velocidade de acesso as informações. _____	82
Fig35.	Requisitos de usabilidade. _____	83
Fig36.	Requisitos de funcionalidade. _____	84
Fig37.	Requisitos de manutenibilidade. _____	84
Fig38.	Requisitos de confiabilidade. _____	85
Fig39.	Facilidade de acesso aos dados. _____	85
Fig40.	Importância da Internet no controle acadêmico. _____	86

## LISTA DE TABELAS

Tab1.	<i>Formulário para realização de entrevistas</i> .....	40
Tab2.	<i>Benefícios do produto</i> .....	47
Tab3.	<i>Requisitos funcionais - Módulo Intranet</i> .....	48
Tab4.	<i>Requisitos funcionais – Módulo Internet</i> .....	49
Tab5.	<i>Atores e sistemas internos</i> .....	49
Tab6.	<i>Cartão use case - Realizar matrícula</i> .....	51
Tab7.	<i>Cartão use case – Matricular novatos</i> .....	52
Tab8.	<i>Cartão use case - Efetuar matrícula.</i> .....	52
Tab9.	<i>Cartão use case - Trancar matrícula</i> .....	53
Tab10.	<i>Levantamentos das interfaces do módulo Intranet</i> .....	53
Tab11.	<i>Levantamentos das interfaces do módulo Internet.</i> .....	54
Tab12.	<i>Campos da interface Login</i> .....	56
Tab13.	<i>Comando da interface Login</i> .....	56
Tab14.	<i>Comando da interface Abertura.</i> .....	57
Tab15.	<i>Campos da interface cadastro de alunos</i> .....	59
Tab16.	<i>Comando da interface cadastro de alunos.</i> .....	60
Tab17.	<i>Campos da interface Matrícula.</i> .....	61
Tab18.	<i>Comando da interface Matrícula</i> .....	62
Tab19.	<i>Campos da interface Gera turmas</i> .....	63
Tab20.	<i>Comando da interface Gera turmas.</i> .....	64
Tab21.	<i>Requisitos do módulo Intranet</i> .....	67
Tab22.	<i>Requisitos do módulo Internet.</i> .....	68
Tab23.	<i>Tratamento de erros.</i> .....	75
Tab24.	<i>Teste dos use cases.</i> .....	76
Tab25.	<i>Procedimento de teste</i> .....	76
Tab26.	<i>Critérios de completeza e sucesso.</i> .....	77
Tab27.	<i>Procedimento de teste – Realizar Matrícula</i> .....	77
Tab28.	<i>Questionário de análise de resultados</i> .....	82

**LISTA DE ABREVIATURAS**

<b>ABREVIATURAS</b>	<b>NOME</b>
BD	Banco de Dados
DRA	Diretório de Registro Acadêmico
HTML	Hypertext Markup Language
IES	Instituição de Ensino Superior
ODBC	Open Database Connectivity
OMG	Object Management Group
OMT	Object Modeling Technique.
OO	Orientação a objeto
OOSE	Object-Oriented Software Engineering.
PHP	Hypertext Preprocessor
POO	Programação orientada a objeto
RAD	Rapid Application Development
RUP	Rational Unified Process
SGBD	Sistema Gerenciador de Banco de Dados
SGML	Generalized Markup Language.
SisRA	Sistema de Registro Acadêmico
UML	Unified Modeling Language
UNIVERSITAS	Centro Universitário de Itajubá



# UM SISTEMA DE CONTROLE E REGISTRO ACADÊMICO PARA UMA UNIVERSIDADE

## RESUMO

As Universidades em geral precisam manipular uma grande quantidade de informações armazenadas em diversos arquivos que, com o tempo, tornam o controle acadêmico lento e incapaz de oferecer qualidade no atendimento aos alunos. Algumas instituições preferem adquirir sistemas comerciais para o gerenciamento do diretório de registro acadêmico.

O Centro Universitário de Itajubá - UNIVERSITAS, com o propósito de adquirir um melhor desempenho, optou por desenvolver uma ferramenta adequada às suas necessidades.

O objetivo principal deste trabalho é avaliar o uso da gestão acadêmica nos processos de administração do registro acadêmico - DRA. Com base na investigação realizada, foi proposto um modelo, implementado através de uma ferramenta capaz de controlar a vida acadêmica do aluno, o processo de matrícula, o cadastro de docentes, além de fornecer, para maior comodidade de alunos e professores, a possibilidade de consulta e inserção de notas e frequência através do sistema on-line.

Para o desenvolvimento do modelo e ferramenta foram utilizados conceitos e técnicas de gestão acadêmica e de Engenharia de Software.

## PALAVRAS - CHAVE

Engenharia de software, técnicas de orientação a objeto, gestão acadêmica, RUP e UML.

# UM SISTEMA DE CONTROLE E REGISTRO ACADÊMICO PARA UMA UNIVERSIDADE

## ABSTRACT

Schools and Universities in general, need to manipulate a great amount of information stored in several files that, with time, make the control academic slow and unable to offer an acceptable service quality. Some institutions prefer to acquire commercial software for the administration of the directory of academic registration.

Several Academic Center with the purpose of acquiring a better performance opted to develop an appropriate tool to their needs.

The main objective of this work is to evaluate the use of the academic administration in the processes of administration in the Directory of Academic Registration -DRA. With base in the accomplished investigation, a model was proposed, implemented through a tool capable to control the student academic information, the registration process, the teacher registers, besides supplying for larger comfort of students and teachers, the consultation possibility and insert of notes and frequency through the system online.

For the development of the model and tool were used concepts and techniques of academic administration and of software engineering.

## Keywords

Software Engineering, Oriented-Object Techniques, system of academic administration, RUP and UML.

# ***Capítulo primeiro***

## **1 INTRODUÇÃO**

### **1.1 ASSUNTO ABORDADO**

As necessidades dos novos tempos, criados pelo contexto de uma economia globalizada com forte poder de concorrência, requerem estudos onde se busque avaliar as universidades, os negócios e as necessidades de controle das informações. Conceitos como flexibilidade, integrabilidade, criatividade, qualidade total, reengenharia e planejamento estratégico, entre outros, constituem a partir do século passado o tom das organizações, influenciando as universidades no que se refere à gestão de serviços educacionais.

As universidades e faculdades que há alguns anos atuavam de forma passiva nas questões educacionais, principalmente nas relações com o mercado, hoje estão sendo forçadas a ser pró-ativas em suas ações estratégicas, principalmente na identificação e satisfação das expectativas e necessidades de um mercado cada vez mais seletivo e exigente.

A gestão acadêmica diz respeito a todas as atividades organizacionais e documentais necessárias para assegurar que os cursos sejam oferecidos com todos os seus requisitos, recebam e identifiquem os alunos e certifiquem seu progresso, com o mínimo de erros, o máximo de eficiência e participação dos alunos e professores.

No ensino superior, desde a primeira Lei de Diretrizes e Bases da Educação Nacional, sempre foi necessário que cada instituição tivesse seus arquivos rigorosamente atualizados. A atualização dos arquivos de qualquer secretaria acadêmica faz-se indispensável, pois estes registram toda a vida acadêmica do aluno.

Um dos grandes problemas enfrentados por administradores de instituições de ensino que não possuem um sistema informatizado é, na maioria das vezes, relacionado com o controle da vida acadêmica do aluno, em que cada um possui particularidades e pode frequentar várias disciplinas de um curso ministrado por diversos professores. Da matriz tridimensional (alunos, disciplinas e professores) surgem todas as dificuldades de um controle acadêmico, que deve conciliar, para sua consecução, os interesses pessoais dos alunos, professores e instituição.

Considerando a importância de se renovar e buscar a criação de um vínculo mais forte com os alunos e docentes da instituição, o UNIVERSITAS iniciou seu processo de modernização da gestão, revendo e reestruturando o processo administrativo do Instituto e

investindo na criação de um modelo e no desenvolvimento de uma ferramenta para controlar o sistema acadêmico.

## 1.2 A IMPORTÂNCIA DO ESTUDO

Um dos problemas importantes em qualquer Instituição de Ensino Superior - IES é a necessidade de tornar o sistema de administração adequado para garantir, entre outras coisas, a satisfação dos membros da comunidade universitária. Segundo KARIDIMA [1992], o êxito das universidades não depende somente das atividades realizadas pelos professores e pesquisadores, senão também da administração acadêmica e das atividades de tipo estritamente administrativo.

Desta forma, este estudo tem importância teórica e prática. Teórica, porque, através da análise do sistema de administração acadêmica, podem ser apontadas todas as deficiências do processo atual existente no UNIVERSITAS, permitindo assim, a especificação dos requisitos necessários para a criação da nova ferramenta.

De uma perspectiva prática, o estudo fornecerá subsídios sobre as questões da organização, aprimoramento dos serviços prestados e o alcance da qualidade do software desenvolvido.

## 1.3 DESCRIÇÃO DO PROBLEMA

A primeira dificuldade encontrada relacionada ao controle de registro acadêmico é a administração da grade horária, pois para elaborá-la é preciso considerar as disponibilidades das salas, docentes, laboratórios e demais equipamentos. A partir da criação de horários, são emitidos diversos documentos tais como lista de presença, necessidades de cadeiras de cada sala, número de aprovados em cada disciplina.

Durante o período de realização de matrícula e re-matrícula, os funcionários do DRA são dispensados de suas atividades diárias para dedicar-se ao processo de matrícula. O aluno realiza a matrícula através de um formulário que, depois de preenchido, precisa ser encaminhado à Tesouraria para a efetuação do pagamento e confirmação da matrícula.

Os funcionários, depois do período de inscrição, digitam todas as informações relacionadas aos alunos no sistema existente no departamento.

A breve descrição dos procedimentos para a realização da matrícula tem como objetivo ilustrar deficiências e a fragilidade do sistema atual, bem como os procedimentos administrativos que o envolvem.

#### 1.4 OBJETIVOS DO TRABALHO

Não se pode afirmar que exista um único sistema de controle acadêmico capaz de atender a todo e qualquer tipo de instituição de Ensino. Os processos envolvidos em cada caso são diferentes, ou seja, cada universidade possui características específicas.

O objetivo precípua deste trabalho é avaliar o uso de gestão acadêmica no processo decisório do UNIVERSITAS. O diagnóstico obtido neste cenário possibilitará dispor aos dirigentes universitários uma análise para desenvolvimento futuro de uma ferramenta capaz de controlar a vida acadêmica do aluno.

Através da investigação do processo de administração existente no UNIVERSITAS será desenvolvida uma ferramenta específica para a o processo de matrícula, manutenção de informações referentes aos alunos, professores, turmas, cursos, controle de notas e integração com o sistema de vestibular existente na mesma com o objetivo de realizar a matrícula automática dos candidatos aprovados no vestibular.

A ferramenta apresentará ainda um módulo, baseado na Internet, que permitirá uma significativa redução de custos citados abaixo:

- Custo de secretaria e atendimento - A disponibilidade das informações em qualquer lugar e hora reduz significativamente a necessidade do atendimento pessoal na secretaria, permitindo que os funcionários possam dedicar-se a funções prioritárias.
- Material de escritório e suprimentos - A possibilidade de consulta às notas e frequência e envio de relatórios pela Internet, diminuem os gastos com papel, tinta de impressora, etiquetas, envelopes, e outros materiais de escritório e suprimentos de informática.
- Digitação e entrada de dados - Com lançamento de notas pela Internet, os custos com digitação e entrada de dados no sistema são praticamente reduzidos a zero. Neste caso, todas as informações relativas a frequência, notas e avaliações, que consistem no maior volume de dados em um sistema acadêmico, são lançados

diretamente pelos professores através do sistema, eliminando a necessidade da digitação destas informações na secretaria.

As principais vantagens que podem ser alcançadas com a implantação da ferramenta informatizada são:

- Diminuição das filas.
- Disponibilização das notas e frequência para consulta na Internet.
- Facilidade na criação das grades horárias dos cursos.
- Agilização no processo de alteração das turmas e classes.
- Otimização da ocupação física das classes.
- Redução das despesas operacionais através do controle de produtividade e da avaliação dos serviços realizados.
- Redução do tempo médio de atendimento e de execução dos serviços operacionais;
- Controle informatizado de todas as fases do processo de matrícula.
- Redução do número de reclamações por parte dos alunos em consequência das melhorias provenientes da implantação do sistema proposto.
- Controle de toda a vida acadêmica do aluno.
- Integridade das informações.
- Segurança dos dados armazenados.
- Melhora no tempo de emissão de documentos e relatórios.
- Comodidade ao professor que poderá inserir as notas e frequências da turma pela Internet.

Por outro lado, para o desenvolvimento de soluções informatizadas que venham a contribuir para a consecução desses objetivos, há que se considerar alguns dos problemas que fatalmente afetam o processo como um todo, principalmente:

- Disponibilidade dos professores;
- Capacidade física das classes;
- Número de alunos dependentes por disciplina;

Para o desenvolvimento do modelo proposto, foram utilizados conceitos de Engenharia de software, apontando os desafios e implicações que envolvem a transição entre o sistema atual e o sistema proposto.

## 1.5 JUSTIFICATIVAS

Muitos sistemas de controle acadêmico começaram a disputar o mercado. Alguns, infelizmente, são preparados em função de outras realidades e não funcionam efetivamente quando implementados. Outros, ao contrário, têm um desempenho excelente visto serem construídos a partir de um contexto previamente estudado. Cabe salientar, entretanto, que um bom sistema é aquele preparado em função da realidade do UNIVERSITAS.

Um software bastante discutido para ser implementado foi o SAGU que é um sistema Open Source desenvolvido em PHP, mas que apesar de ser gratuito, é desenvolvido de maneira generalizada. Para torná-lo específico seria necessário um estudo de todo o código fonte, o que exige um tempo inútil, além de pessoal para manutenção e treinamentos aos funcionários e plataforma restrita.

Um outro fator a ser considerado é a possibilidade de integração com os módulos que futuramente serão implantados no instituto.

Dessa forma, a escolha do tema decorreu da necessidade de se desenvolver uma ferramenta computacional específica para a instituição e contribuir com a melhora do atendimento aos alunos.

Quanto aos custos adicionais necessários, estima-se que a simples eliminação dos gastos com o pessoal contratado para o desenvolvimento e implantação compensará todo o investimento a ser feito no sistema.

## 1.6 AMBIENTE DE ESTUDO DA PESQUISA

Sobre o centro universitário - A Fundação de Ensino e Pesquisa de Itajubá - FEPI, criada pela Lei Estadual nº 3.009, de 17 de dezembro de 1963, é uma instituição de ensino, pesquisa e formação profissional. Instalada em um amplo campus de 27.845 m<sup>2</sup> de área, localizado à Av. Dr. Antônio Braga Filho, 687, com uma área construída de 7.001m<sup>2</sup>, a FEPI mantém o UNIVERSITAS e o Colégio de Aplicação.

## 1.7 DESENVOLVIMENTO DO TRABALHO

Este trabalho está estruturado em 5 capítulos distribuídos da seguinte forma: O capítulo em questão apresenta o assunto abordado, assim como a importância deste, objetivo do trabalho, as justificativas para a realização deste estudo, o ambiente de pesquisa e sua estrutura.

A seguir, realiza-se uma fundamentação teórica sobre a metodologia aplicada, de modo a conhecer e aprofundar os conceitos de gestão acadêmica e as técnicas de desenvolvimento de software como processos de desenvolvimento, técnicas de Orientação a Objeto - OO, Unified Modeling Language - UML, buscando-se destacar a importância da implantação e acompanhamento dos processos de desenvolvimento, modelagem e gerenciamento de software.

No Capítulo 3, serão apresentadas as características que tornam a ferramenta singular em sua proposta. As fases de análise e projeto são decompostas em uma seqüência de passos que serão seqüencialmente explicitadas. Cada passo consiste em um aspecto particular descrito nos padrões da UML.

No capítulo 4 serão abordadas questões relacionadas à implementação, análise das opiniões dos usuários e o resultados obtidos.

O capítulo 5, o último da dissertação, é constituído pelas conclusões inferidas do trabalho desenvolvido, juntamente com as futuras pesquisas a serem desenvolvidas a partir da presente dissertação.



## Capítulo segundo

### 2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo aborda uma visão teórica, contextualizando as IES em relação a temas: contexto das instituições sob a visão organização; estudos de metodologias de gestão das universidades brasileiras; elaboração do perfil futuro da instituição.

A seguir será feita uma breve descrição dos recursos, paradigmas, métodos e, principalmente a demonstração das vantagens de se utilizar a Engenharia de Software no processo de desenvolvimento de sistemas acadêmicos, enfatizando a melhoria da qualidade dos processos e produtos gerados.

#### 2.1 TÓPICOS EM GESTÃO ACADÊMICA

A gestão é um fator estratégico de sucesso em qualquer tipo de empreendimento. O ensino superior não é exceção, pois do planejamento depende a implementação das estratégias que dão à instituição o seu diferencial de mercado.

Empresas dos mais variados tipos investem atualmente em *business intelligence*, ou seja, em sistemas que permitem aos dirigentes ter acesso a mecanismos de monitoramento e avaliação do seu negócio, contribuindo para decisões bem fundamentadas.

A inteligência do negócio em termos de sistema de gestão acadêmica é que, além de permitir a informatização dos processos administrativos, diminuindo custos operacionais, facilita a organização das grades curriculares flexíveis e monitora continuamente o desempenho acadêmico e financeiro da instituição.

##### 2.1.1 A IES como organização

As IES são organizações que desenvolvem uma lógica distinta de outras organizações econômicas, pois sua atenção principal está voltada para a formação e disseminação de conhecimento através de práticas educativas. No entanto, apresentam estruturas que exigem a definição de práticas de gestão que possam garantir o alcance de resultados esperados pela sua comunidade interna e demais grupos de interesse.

BALDRIDGE [1971] escreveu que organizações variam significativamente em muitos aspectos como: tipos de clientes, tecnologias, habilidades dos trabalhadores, estruturas e estilos

de coordenação e relacionamento com seu ambiente externo. Existem muitos elementos comuns na operação de faculdades e universidades, empresas, organizações governamentais, mas nunca duas organizações são iguais. Este autor afirma ainda que universidades são organizações singulares, diferindo na maioria dos seus aspectos das empresas industriais, empresas de serviços e organizações governamentais.

Enquanto organização complexa, a universidade apresenta cinco características específicas, as quais são descritas por BALDRIDGE [1971] como: ambigüidade das metas, *client service*, tecnologia problemática, profissionalismo e vulnerabilidade ambiental.

#### Ambigüidade das metas

As universidades geralmente têm metas vagas e ambíguas. Enquanto nas organizações lucrativas, onde os objetivos e metas são definidos, se constroem as estruturas de decisão para atingir sua finalidade, nas universidades, as estruturas de decisão têm que ser construídas para enfrentar a incerteza e o conflito com as metas.

#### Client service

Segundo BALDRIDGE [1971], As universidades são consideradas como instituições processadoras de pessoas. Os “clientes” entram na organização com necessidades específicas e são "alimentados" dentro dela. As instituições agem sobre ele e depois devolvem-nos à sociedade. Na educação superior, os "clientes" são completamente capazes de falar por si mesmos e freqüentemente o fazem. Eles buscam voz no processo decisório e normalmente conseguem, tornando este mais complexo e menos sujeito à lógica da escolha racional.

#### Tecnologia problemática

Uma organização manufatureira desenvolve uma tecnologia específica que pode ser segmentada e realizada através de rotinas. Mas é difícil construir uma tecnologia simples para uma organização que lida com pessoas. Se às vezes faculdades e universidades não sabem claramente o que elas estão tentando fazer, elas freqüentemente não sabem também como fazê-lo.

### Profissionalismo

Muitas organizações para lidar com objetivos ambíguos e tecnologia problemática empregam profissionais altamente treinados. No caso das universidades esses profissionais são professores, os quais usam um amplo repertório de habilidades para lidar com os problemas de seus "clientes". Ao invés de dividir uma tarefa complicada num conjunto de procedimentos de rotina, trabalhos profissionais requerem que uma ampla variedade de tarefas seja desenvolvida por um único empregado. Outra implicação disso é que muitas vezes esses profissionais têm sua lealdade dividida entre as normas da corporação (profissão) e os objetivos da organização.

### Vulnerabilidade ao ambiente

Todas as organizações interagem com seu ambiente social em alguma extensão. Mas, ainda que nenhuma organização seja completamente autônoma, algumas têm consideravelmente maior liberdade de ação do que outras. O grau de autonomia que uma organização tem em relação ao seu ambiente é um dos determinantes críticos de como ela será gerenciada. Importante para uma abordagem institucionalista é a afirmação de BALDRIDGE [1971], de que quando organizações profissionais estão bem separadas das pressões do ambiente externo, os valores e normas profissionais exercem um papel dominante em moldar o caráter da organização. Por outro lado, quando são exercidas pressões sobre as faculdades e universidades, a autonomia operacional dos profissionais acadêmicos é seriamente reduzida. As alterações na vulnerabilidade ambiental da organização mudam significativamente seu padrão de gestão.

#### **2.1.2 Metodologia de gestão das IES**

MACHADO e SILVEIRA [1998], consideram que, ao longo dos séculos, as instituições de ensino foram estruturadas para mudar lentamente, como forma de perenizar suas atividades. Segundo BUARQUE [1994], diferente do cenário mundial, a universidade brasileira mostra-se estática e se acomoda, limitando-se à repetição. O autor sugere que, seja qual for o caminho da humanidade, ela passa pela universidade, que terá que assustar-se consigo e promover as transformações que a dotem da agilidade e flexibilidade requeridas para uma atuação mais efetiva, como forma de vislumbrar soluções alternativas a fim de superar os sustos e a perplexidade deste final de século.

Hoje, pelo mesmo motivo (perenização), as universidades devem mudar, tornar-se mais ágeis e flexíveis, para melhor atender a sociedade com a otimização dos seus recursos, sejam eles humanos, tecnológicos, físicos ou financeiros. Torna-se importante racionalizar, visando ao retorno do resultado dessa otimização para a própria sociedade.

Segundo MACHADO e SILVEIRA [1998], é importante entender que as universidades possuem objetivos complexos, empregam tecnologias complexas e, conseqüentemente, adotam uma multiplicidade de critérios de estruturação, como forma de viabilizar o seu funcionamento e atingir seus objetivos estratégicos.

Quando se fala em gestão, devem-se estudar as formas de mudanças (reformas) e continuidade. Segundo MARCOVITCH [1998], se a instituição incorporar a cada ano, medidas que aprimorem a forma de executar os recursos escassos, no prazo de cinco anos, ocorrerá uma verdadeira revolução. Uma revolução que tem por objetivo explícito a qualidade dos serviços, em todas as dimensões. A universidade deve ser um exemplo de renovação, mas deve garantir a continuidade do que é positivo em cada gestão.

Outra questão presente na gestão universitária é a da velocidade das decisões. As decisões são lentas em face da simultaneidade de problemas tópicos, que devem ser resolvidos. Isto faz o gestor priorizar as atividades que estão no cronograma. Desta forma, é resolvido aquilo que, circunstancialmente, está com prazo marcado. Na universidade, vive-se com uma sobrecarga de vários processos em mãos para serem decididos.

O gestor tem um desafio extra: descobrir até onde pode empreender sem infringir as normas. Isso não é trivial, mas muitas vezes bloqueia o processo decisório. Como sugere NAKAGAWA [1994], ultimamente as universidades estão passando por um questionamento profundo e buscando uma nova identidade que as capacite a serem uma resposta efetiva às expectativas e necessidades da comunidade. A preocupação das universidades brasileiras com qualidade total em seus serviços é cada vez maior. Em alguns casos prende-se ao fato de que órgãos de financiamento exigem aperfeiçoamento contínuo do pessoal vinculado à instituição em que serão alocados os recursos. As universidades estão despertando para a necessidade de começar a buscar a qualidade nos diversos setores que compõem a instituição, como forma de atender adequadamente à comunidade a que servem.

Nesse sentido, diversas instituições já iniciaram um processo de melhoria de qualidade, ainda que nem todas o tenham adequadamente estruturado, porque estão mais centradas na

avaliação da estrutura do que no processo e nos resultados e porque não dispõem ainda de uma função ou órgão especificamente estruturado para esta finalidade.

O planejamento estratégico, como é abordado por MEYER [1991], é muito mais do que uma metodologia; ele implica uma nova postura organizacional, uma forma nova de administrar. O planejamento pode se constituir num importante instrumento de gestão ao permitir a análise da organização através do estabelecimento da sua missão e do levantamento das potencialidades e fraquezas dos diversos cursos, programas, atividades e instalações.

KARIDIMA [1992], também adverte que, para enfrentar os desafios apresentados, as universidades deverão passar da administração e planejamento operativo pontual e contingente para os modelos estratégicos amplos e integrais apoiados na evidência empírica e em análises institucionais permanentes.

Um dos problemas importantes em qualquer IES é a necessidade de estabelecer um sistema de administração adequado e racional que garanta, entre outras coisas, que seja mantida uma organização flexível para responder às mudanças requeridas, que a missão e os objetivos sejam atuais, viáveis e operacionais e que também trate de satisfazer às expectativas pessoais dos membros da comunidade universitária. Segundo KARIDIMA [1992], o êxito das universidades não depende somente das atividades realizadas pelos professores e pesquisadores, senão também da administração acadêmica e das atividades de tipo estritamente administrativo.

Na concepção de sistemas administrativos próprios, que possam ser utilizados por universidades e institutos de ensino, é importante ressaltar os trabalhos desenvolvidos, nos últimos anos, pelo NCHEMS - *National Center for Higher Educational Management Systems* da Universidade Colorado, pelo NUPEAU - Núcleo de Pesquisas e Estudos em Administração Universitária, da UFSC - Universidade Federal de Santa Catarina, além de outras Universidades e organizações.

Entretanto, ainda são muito poucos os trabalhos sobre planejamento estratégico na área da administração universitária, e são menos ainda, ou praticamente inexitem, os que do desenvolvimento do processo de implementação do mesmo.

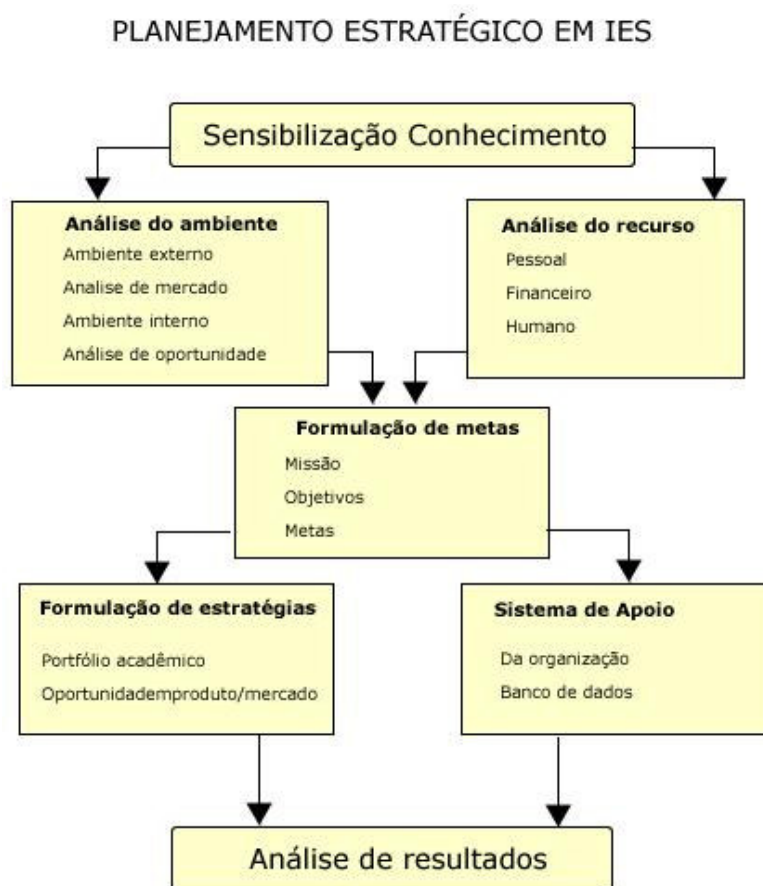
No que se refere a sua administração, o processo de mudança se realiza através da implantação e adaptação de técnicas de gestão administrativa já utilizadas com sucesso pelas organizações privadas, pois na concepção de MEYER [1991], assim como as empresas, as

universidades brasileiras estão necessitando de uma reformulação gerencial de maneira a posicionar-se de forma mais competitiva no ambiente.

### **2.1.3 Fases de implementação do planejamento estratégico nas IES**

Os processos de planejamento estratégicos dentro de uma IES, devem ser primeiramente trabalhados em relação à sensibilização e conhecimento dos seus conceitos e objetivos, para que posteriormente não sejam colocadas barreiras corporativas e reacionárias. O ideal é que sejam formadas equipes multifuncionais para permitir visões de diversos ângulos da instituição e através dessas equipes desenvolver preparação e treinamento em situações inerentes ao sistema, seguindo-se trabalhos de definição da instituição como um todo "como se está" e "onde se quer chegar".

Segundo BODINI [2000], a universidade, já na sua fundação, possui alguns objetivos principais bem definidos e muitos inerentes aos princípios do ensino superior. Estes objetivos norteiam o processo e podem sofrer modificações ou avanços à procura de novas oportunidades ou a adaptações a novos tempos. As etapas fundamentais que devem ser cumpridas estão descritas abaixo.



*Fig1. Sistemática de Planejamento Estratégico em Universidades.*

Segundo FRANCO [2000], as IES devem se relacionar com o segmento interno (dirigentes, pessoal técnico especializado, pessoal administrativo e de apoio, professores e alunos) e com o segmento externo (candidatos ao vestibular, organizações empresariais, órgãos públicos, escolas, outras instituições públicas e particulares e a mídia).

Análise do ambiente - A definição ou estudo das principais tendências, oportunidades e ameaças, bem como suas implicações, devem ser examinadas em todos os setores do ambiente da instituição.

- Ambiente externo – Consiste na verificação de ameaças ou oportunidades que estão em seu ambiente e as melhores formas de evitar ou usufruir destas situações. O olhar a universidade como um todo e do lado de fora, inclui itens como aspectos sociais, econômicos, evolução tecnológica, desempenho institucional, competitividade entre outros.

- Ambiente interno - Consiste na verificação dos pontos fortes e fracos da instituição. A análise interna deverá contar com a participação do conselho diretor, reitoria, corpo docente, corpo discente e área administrativa, que deverá considerar fatores como sistema de informação, eficiência de ensino, recursos humanos, materiais, tecnologia, estrutura organizacional e imagem institucional.
- Ambiente de mercado - Num mundo em rápido processo de evolução tecnológica e globalização, a universidade precisa competir não só com outras universidades, mas também com um sistema de informações e pesquisa, assumindo seu papel decisivo no processo de desenvolvimento tecnológico, cultural, social e econômico da sociedade.
- Análise de oportunidades - Quando conhecemos os pontos fortes, fracos e mesmo neutros da instituição, sua estrutura organizacional e o ambiente em que ela está inserida, define as oportunidades de mercado e áreas de ação importantes nas quais poderá obter vantagens competitivas.

Análise de recursos - Adoção de metas, propósitos e estratégias devem ser compatíveis com os recursos humanos, financeiros e materiais existentes. Investir em formação e atualização docente, aproveitar melhor os recursos humanos e materiais existentes, fomentar interdisciplinaridade e multidisciplinaridade, e busca de outras fontes de recursos, especialmente para pesquisa, são as ações mais recomendadas neste procedimento.

Formulação de metas - A análise do ambiente e dos recursos, dão subsídios às instituições, com a possibilidade de avaliar se sua missão, objetivos e metas, que no início de suas atividades eram claros, continuam sendo com as mudanças no macroambiente.

- Missão - A missão da universidade permeia todas as decisões estratégicas. Proporcionando a instituição senso de oportunidade, direção, importância e conquista.
- Objetivos: A definição de objetivos e metas pode fazer parte do mesmo processo de definição da missão. Porém os objetivos devem ser mais específicos e de alcance em menor tempo. Como os objetivos de uma IES são muitos e existe a dificuldade de trabalhar com todos ao mesmo tempo, deverá ser enfatizado algum deles



considerando o outro como limitações. Desta forma os objetivos podem variar de ano para ano, dependendo da forma como vão sendo supridas as dificuldades e necessidades.

Formulação de estratégias - Consiste em definir estratégias para a consecução das metas a serem alcançadas.

Para melhorias significativas muitas vezes, a universidade precisa traçar uma estratégia que envolva todas as suas áreas. Para que a análise possa ser efetuada é interessante organizar o portfólio acadêmico, avaliando os atuais programas e a decisão de o que fazer com eles.

- Portfólio acadêmico - A análise periódica do portfólio, deverá avaliar: programas acadêmicos, o conteúdo das disciplinas e o fluxograma da instituição, podendo levar a decisões como: expandir, manter, reduzir ou eliminar informações.
- Oportunidade produto/mercado: As oportunidades podem ser definidas a partir de itens como expansão do número de matrículas, expansão e modificação programas para atrair novos mercados.

Sistemas de Apoio - As universidades possuem um procedimento organizacional difícil de ser mudado, mas, com a consciência da necessidade de crescimento deve direcionar esta fase do processo. Cabe a reitoria definir modos de mudar a cultura da organização através de: treinamentos de pessoas em cargos principais, melhoria do corpo docente, melhoria da prestação de serviços, ações para melhoria do ensino e desenvolvimento da orientação de mercado.

- Banco de Dados - BD - Existe um número muito grande de informações que devem ser analisadas quando está sendo realizado o planejamento da instituição, por isso o desenvolvimento de um BD sobre: alunos, índices dos cursos e departamentos, ex-alunos, comunidade, público alvo, desenvolvimento tecnológico e econômico, e pesquisas de mercado é de suma importância para o embasamento de decisões.

Análise de Resultados - O planejamento estratégico só é útil se sair do papel, ou seja, se for implementado, acompanhado, avaliado e reestruturado, se necessário. Os propósitos podem não

ser totalmente alcançados em um primeiro ciclo, por isto a visão de contínuos resultados de um plano de ação permite correções de rota e mesmo troca de estratégias, em face de um mercado em constante evolução.

Para ROBERT [1998], existe apenas um componente da instituição que domina a estratégia e a impulsiona na conquista da vantagem competitiva. Até que a direção compreenda claramente qual é este componente, terá dificuldade de tomar decisões inteligentes e coerentes com relação à alocação dos recursos e escolha de oportunidades.

#### **2.1.4 Considerações finais**

O planejamento estratégico é essencial a sobrevivência da universidade, assegurando um mínimo de visão global e ação, a partir de uma definição adequada de suas finalidades (objetivos e metas), coerentes com os objetivos e estratégias nacionais e com o cenário internacional.

Segundo BODINI [2000], a organização e otimização dos meios para a consecução de suas finalidades devem contar com a participação da comunidade acadêmica e da sociedade na decisão de onde devem ser alocados os recursos.

## **2.2 TÓPICOS EM SOFTWARE E ENGENHARIA DE SOFTWARE**

Um fator essencial para a conclusão de um projeto de desenvolvimento de software é a adoção de uma metodologia, que pode ser entendida como um conjunto de regras e padrões que orientam as abordagens utilizadas em todas as tarefas associadas com o ciclo de desenvolvimento de sistemas. A existência de uma metodologia padronizada de desenvolvimento de sistema garante que as especificações serão consistentes para toda a empresa.

Existem diversas técnicas de análise e modelagem, porém as mais dominantes são o desenvolvimento estruturado e o orientado a objeto.

Segundo PRESSMAN [2001], o desenvolvimento estruturado é uma atividade de construção de modelo, que utilizando uma notação própria, cria modelos que retratam o fluxo e o conteúdo da informação. Os sistemas são divididos em partições funcionais e comportamentais que descrevem a essência do produto a ser construído.

O desenvolvimento orientado a objetos é uma continuação da abordagem modular do desenvolvimento estruturado combinado com a modelagem e programação orientada a objeto.

### **2.2.1 Técnicas de Orientação a Objetos - OO**

Segundo MARTIN & ODELL [1996], as técnicas OO mudam a visão que os analistas têm do mundo. Em vez de pensarem em processos e na sua decomposição, eles pensam em objetos e no comportamento destes. O objeto internamente pode ser complexo, porém o analista não precisa entender esta complexidade e sim, saber como se comporta tal objeto e como utilizá-lo.

Segundo RUMBAUGH [1994], a modelagem e o projeto baseado em objetos são um novo modo de estudar problemas com utilização de modelos fundamentados em conceitos do mundo real. A estrutura básica é o objeto, que combina a estrutura e o comportamento dos dados em uma única entidade. Os modelos baseados em objetos são úteis para a compreensão de problemas, para a comunicação com os peritos em aplicações, para modelar empresas, preparar documentação e projetar programas e bancos de dados.

As vantagens da OO são:

- Maior facilidade para reutilização de códigos e por consequência do projeto;
- Utilização de um padrão único durante todo o processo de criação do software;
- Maior adequação à arquitetura cliente /servidor;
- Ciclo de vida mais longo para os sistemas;
- Menor custo para o desenvolvimento e manutenção de sistemas.

#### **2.2.1.1 Conceitos Fundamentais**

Alguns conceitos que diferenciam a OO dos outros métodos são mostrados abaixo:

- Abstração - Segundo COOD [1991], é o princípio que leva a ignorar os aspectos de um assunto não relevante para o propósito em questão, tornando possível uma concentração maior nos assuntos principais. Consiste na seleção que o analista faz de alguns aspectos, ignorando outros. As formas de abstração existentes são: procedimentos e dados.

- Encapsulamento - Segundo FURLAN [1998], significa omitir informações pelo princípio de que uma determinada entidade esconde informações as quais são necessárias apenas à mesma. É fundamental que o objeto proteja seus dados, não permitindo que o usuário do objeto os acesse diretamente, mas sim através de métodos.
- Herança - Segundo PRESSMAN [2001], é um mecanismo que permite a propagação das responsabilidades de um objeto para o outro.
- Polimorfismo - Conceito usado em Programação OO para denotar a característica segundo a qual linguagem suporta a utilização do mesmo identificador (o mesmo nome) para métodos de classes diferentes. A rigor, o polimorfismo aparece quando se usa o mesmo nome para tarefas similares em classes diferentes. Segundo PRESSMAN [2001], é uma característica que reduz o esforço necessário para aumentar o sistema existente.

### 2.2.1.2 Metodologias

As principais metodologias de OO que se tornaram populares nos anos 90 foram:

Método de Booch - Segundo BOOCH [1994], este método engloba projetos de desenvolvimento micros e macros. O nível micro identifica classes e objetos de acordo com um certo nível de abstração, identifica a semântica dos objetos e das classes, os relacionamentos entre classes e objetos e sua implementação. O macroprocesso é usado para controlar o microprocesso.

Método de Coad e Yordon - Segundo COAD [1991], este método é considerado como um dos mais fáceis da análise orientada a objetos. Sua notação é relativamente simples e as instruções são fáceis de compreender. Ele consiste de cinco atividades principais: identificação de classes e objetos usando o critério "o que procurar"; identificação de estruturas; identificação de assuntos; definição de atributos e serviços.

Método de Jacobson - Também chamado de Object-Oriented Software Engineering - OOSE Segundo JACOBSON [1992], é uma versão simplificada do método proprietário *Objectory*,

também desenvolvido por ele. Este modelo é diferenciado dos outros devido à forte ênfase nos use cases.

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 de objetos. Na descrição da OMT, o sistema é descrito a partir de três pontos de vista diferentes: visão dos objetos (descreve o fluxo de informações), visão dinâmica (fluxo de controle) e funcional (especificação dos processos).

## 2.2.2 UML

A utilização de uma metodologia de desenvolvimento de sistema para captar desde os primeiros contatos, até a conclusão das etapas de desenvolvimento de software, é de extrema importância.

Segundo FURLAN [1998], com a evolução dos processos, sentiu-se a necessidade de se ter uma linguagem unificada que se tornasse poderosa o suficiente para modelar qualquer tipo de aplicação. Dessa necessidade surgiu a UML, uma linguagem padrão para especificar, visualizar, documentar e construir artefatos de um sistema e pode ser utilizada com todos os processos ao longo do ciclo de desenvolvimento e através de diferentes tecnologias de implementação. A UML é a junção das três mais conceituadas metodologias (Booch, OOSE e OMT), no qual foram aproveitados os que de melhor cada uma delas possuem.

As partes que compõem a UML são:

### 2.2.2.1.1 Visões

Segundo PENDER [2002], Uma visão é a apresentação da coleção de diagramas que descrevem os aspectos similares do projeto. Frequentemente existem três conjuntos distintos de visões, chamadas de visão estática, dinâmica e funcional.

As visões também podem servir de ligação entre a linguagem de modelagem e o método/processo de desenvolvimento escolhido.

A figura abaixo ilustra a natureza complementar das visões e os diagramas pertencentes a cada visão.

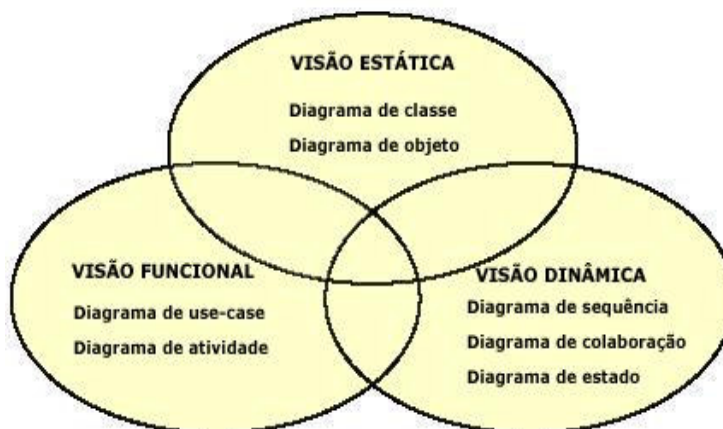


Fig2. Visões e seus diagramas.

#### 2.2.2.1.2 Modelos de elementos

Os conceitos usados nos diagramas são chamados de modelos de elementos. Alguns exemplos desses modelos são as classes, objetos, estados, pacotes, componentes e relacionamentos. Um elemento pode existir em diversos tipos de diagramas, mas existem regras que definem quais elementos podem ser mostrados em que tipos de diagramas.

- Classes - Segundo PRESSMAN [2001], uma classe é uma descrição de um conjunto de objetos que compartilham os mesmos atributos, operações, relacionamentos e semântica. Podem ser implementadas em uma ou mais interfaces.
- Atributos - Segundo PRESSMAN [2001], os atributos são anexados às classes e objetos com o objetivo de descrevê-las.
- Operações - Segundo PENDER [2002], definem o comportamento que uma classe de objetos pode realizar.
- Objetos - São elementos que podemos manipular, acompanhar seu comportamento, criar, interagir, ou até destruí-los.
- Estado - Segundo BOOCH [2000], é uma condição ou situação na vida de um objeto durante a qual o objeto satisfaz alguma condição, realiza alguma atividade ou aguarda um evento. Um estado pode ter três compartimentos: nome do evento, atributos e atividades.
- Pacotes - Segundo PENDER [2002], é um mecanismo de propósito geral para organizar elementos de modelo em grupos, tipicamente por funções ou use-cases parecidos com o contexto do sistema.

- Componente - Pode ser tanto um código em linguagem de programação como um código executável já compilado.
- Relacionamento - liga classes ou objetos entre si criando relações lógicas entre estas entidades. Existem três tipos de relacionamento: associação, generalização e dependências.

### 2.2.2.1.3 Mecanismos gerais

Fornecem comentários suplementares e informações sobre os elementos que compõem os modelos além de fornecer mecanismos de extensão para estender a UML a um método, organização ou usuário específico.

- Ornamentos - 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.
- Notas - Permitem adicionar informações a um modelo e podem ser colocadas em qualquer lugar do diagrama.

### 2.2.2.1.4 Diagramas

Um conjunto de diagramas compreende o arsenal técnico definido pela UML para a representação gráfica dos modelos de software orientado a objetos. Nem todos são sempre necessários. A parte estática de um sistema pode ser visualizada a partir dos seguintes diagramas:

- Diagrama de classe - Segundo FURLAN [1998], 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. O diagrama mostra um conjunto de classes, interfaces, colaborações e seus relacionamentos.

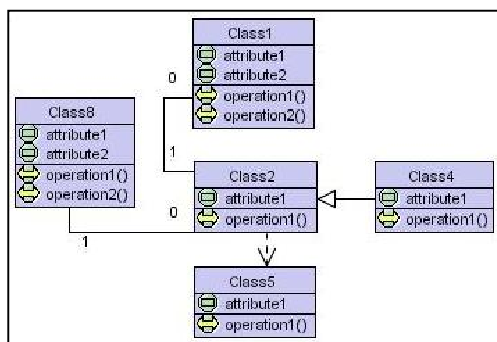


Fig3. Diagrama de classe.

- Diagrama de objeto - Mostra um conjunto de objetos e seus relacionamentos. Esses diagramas são usados para ilustrar as estruturas estáticas de dados, registros estáticos de instâncias dos itens encontrados nos diagramas de classes.
- Diagrama de componentes - Mostra um conjunto de componentes e seus relacionamentos e é utilizado para ilustrar a visão estática da implementação de um sistema.
- Diagrama de implantação - Mostra um conjunto de nós e seus relacionamentos. São utilizados para ilustrar a visão estática da implantação de uma arquitetura. Está relacionado aos diagramas de componentes.

A parte dinâmica de um sistema será visualizada a partir dos seguintes diagramas:

- Diagrama de use case - Segundo JACOBSON [1994], A modelagem use case é uma análise técnica para extrair informações, entender e definir funcionalmente os requisitos de um sistema. O modelo consiste de atores, use cases e o relacionamento entre eles. Os atores são objetos que residem fora do sistema modelado e os use cases são objetos que residem no sistema.

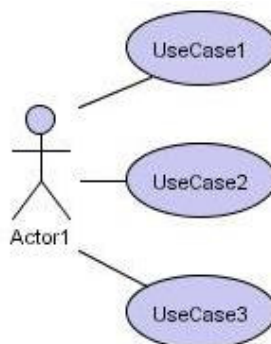


Fig4. Diagrama de use case.

- Diagrama de seqüência - É um diagrama de interação que dá ênfase à ordenação temporal de mensagens. O diagrama mostra um conjunto de objetos e a troca de mensagens entre eles.



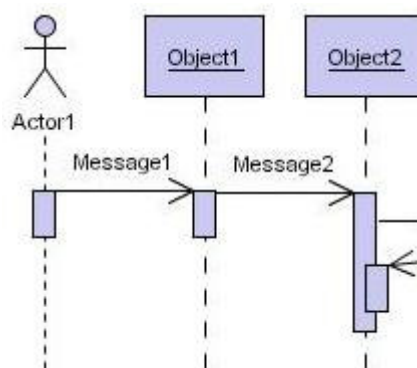


Fig5. Diagrama de seqüência.

- Diagrama de colaboração - Dá ênfase à organização estrutural dos objetos que trocam mensagens.
- Diagrama de estado - Mostra uma máquina de estados, que apresenta estados, transições, eventos e atividades. Esses diagramas são importantes realizar a modelagem do comportamento de uma interface, de uma classe ou de uma colaboração.
- Diagrama de atividade - Mostra o fluxo de uma atividade para outra em um sistema. Esses diagramas são importantes para se fazer a modelagem da função de um sistema.

#### 2.2.2.2 Linguagem de programação orientada a objeto

Segundo DEITEL [2001], a POO modela objetos do mundo real com duplicatas em software. Ela se aproveita das relações de classe, nas quais os objetos de uma certa classe possuem as mesmas características das outras.

A POO nos dá uma maneira mais natural e intuitiva para visualizar o processo de programação, a saber, modelando objetos do mundo real, seus atributos e comportamentos. A POO também modela a comunicação entre objetos. Da mesma maneira que as pessoas enviam mensagens umas às outras, objetos também se comunicam através de mensagens.

A POO encapsula dados (atributos) e funções (comportamento) em pacotes chamados objetos; os dados e funções de um objeto estão intimamente amarrados. Os objetos têm a propriedade de ocultação de informação. Embora os objetos possam saber como se comunicar uns com os outros através de interfaces bem definidas, normalmente não é permitido a esses saberem como outros objetos são implementados.

#### 2.2.2.2.1 Visual Basic - VB

Embora a linguagem VB não seja orientada a objetos, ela incorpora diversas funções e características típicas das linguagens de OO, em especial nas áreas de lógica de programação.

Uma das deficiências que a linguagem apresenta está relacionada ao fato de não aceitar o conceito de herança no seu aspecto mais puro. As linguagens concorrentes permitem a reutilização de forma completa, rápida e flexível. Apesar dessa desvantagem, a linguagem utiliza a tecnologia OLE - Object Modeling Technique, ou seja, uma tecnologia que permite ao programador de plataforma Windows criar um programa que apresente dados a outras aplicações, permitindo que o usuário altere os mesmos diretamente, não precisando finalizar a aplicação que está sendo executada.

#### 2.2.2.3 *BD orientados a objetos*

É basicamente um sistema em que a unidade de armazenamento é o objeto, com o mesmo conceito das linguagens de POO. A diferença fundamental é a persistência dos objetos, ou seja, os objetos continuam a existir mesmo após o encerramento do programa, através das construções orientadas a objeto. Os programadores podem esconder os detalhes da implementação de seus módulos, compartilhar a referência a objetos e expandir seus sistemas através de módulos existentes. O BD orientado a objeto combina os benefícios e conceitos da orientação a objetos com a funcionalidade dos bancos de dados.

#### 2.2.2.4 *BD relacionais*

Atualmente, o modelo de dados lógico mais utilizado por Sistema Gerenciador de Banco de Dados - SGBD comercial é o modelo relacional, baseado em extensões do conceito de relações. Segundo DELOBEL [1995], os dados são representados em forma de tabelas, que contêm uma coleção de linhas. Uma linha contém valores. Cada valor corresponde a um atributo que pode ser do tipo *integer*, *float*, *character*, *string*, entre outros.

O objetivo do BD relacional é a independência dos dados, que são separados do processamento e normalizados. Dessa forma os dados podem ser usados por diversas aplicações mesmo não previstas no projeto. Já o objetivo do BD orientado a objetos é o encapsulamento. Os dados são associados a uma classe específica que usa métodos específicos. Dessa forma os dados só podem ser usados pelos métodos dessa classe. A classe por sua vez é que pode ser usada por

diferentes aplicações. Outra diferença é que os relacionais armazenam apenas dados enquanto os orientados a objetos armazenam estruturas com dados e métodos. A redundância de dados é tratada no BD relacional através da normalização, que visa eliminar a incoerência das informações. No BD orientado a objetos o encapsulamento e a herança fornecem dados não redundantes. A herança além de facilitar a diminuição da redundância, também propicia a reutilização de classes.

#### 2.2.2.5 Ferramentas para desenvolvimento Web

Para se desenvolver soluções para a Internet, podem ser utilizadas várias tecnologias, como protocolos de rede, aplicações para servidor, BD e programação de interfaces gráficas para os usuários.

No momento de decidir quais tecnologias serão utilizadas no desenvolvimento de um aplicativo *Web*, parâmetros como custo, habilidade da equipe, desempenho, multiplataforma, segurança e adesão a padrões e tecnologias abertas devem ser considerados.

As aplicações *Web* podem ser realizadas em várias camadas citadas abaixo:

Camada de interface com o usuário - A primeira camada utiliza, em geral, um navegador para interpretar as páginas HTML vindas do servidor. Algumas tecnologias conhecidas são: HTML, DHTML, Applet Java, Active X, JavaScript e XML.

- HTML - Utiliza os conceitos do hipertexto e da hipermídia para apresentar, num mesmo ambiente, dados, imagens e outros tipos de mídia, como vídeos, sons e gráficos. O HTML é um subconjunto do *Standard Generalized Markup Language* (SGML) e utiliza rótulos (*tags*) que definem a aparência e o formato dos dados, sendo padronizado pelo OMG. É interpretado por qualquer navegador, em qualquer plataforma.

Camada Middleware - Esta camada é responsável pelo processamento das informações no sistema. As solicitações dos usuários são recebidas por ela que, realiza a interação com o BD e envia a resposta ao usuário na forma de páginas HTML. Alguns exemplos são: CGI, SSI, ASP, ColdFusion e PHP.

- PHP - Segue a mesma filosofia do ASP, porém pode ser executada por diferentes servidores, principalmente na plataforma Unix (Solaris, Linux, etc.). Diferentemente do ASP, o PHP utiliza sintaxe baseada em C, Java e Perl. É uma tecnologia não proprietária.

Camada de BD – Esta camada armazena as informações do sistema. Como BD, as opções são muitas. Existem BD open-source como o Interbase e o PostgreSQL, freewares, como o MySQL, e comerciais, como Oracle, Informix, DB2 e Sybase, sendo estes mais voltados ao mercado corporativo. A escolha de cada tipo de BD vai depender da necessidade de segurança, desempenho, escalabilidade da aplicação e das limitações financeiras.

- MySQL - O MySQL é um sistema de BD relacional de código aberto, projetado para oferecer um suporte ao BD cliente/servidor, oferecendo rapidez e flexibilidade. Pode ser capaz de rodar em qualquer servidor e suportar diferentes aplicações cliente, além de fornecer interação com diferentes ferramentas de administração e interfaces de programação, tais como C++, Java, Perl, PHP, Python e Tcl. Dentre os componentes principais do MySQL destaca-se o MySQLAdmin, que pode ser utilizado para uma variedade de tarefas tais como: criação e exclusão de BD, controle de conexões em memória, controle de contas de usuário e monitoramento de servidor. Na versão Windows, foi incorporado o WinMySQLAdmin, que permite através de uma interface gráfica, visualizar e alterar as configurações do MySQL, além de criar novos bancos de dados. Outro utilitário, o MySQLManager, permite executar instruções SQL nos bancos de dados criados pelo MySQL.

Camada de comunicação - Esta camada é transparente para aquele que desenvolve o sistema, visto que o navegador e o servidor se encarregam de utilizá-la, sem a necessidade da interferência do daquele. É na camada de comunicação onde os protocolos de rede atuam. O protocolo utilizado na Internet é o TCP/IP.

### 2.2.3 Processo de desenvolvimento do software

Desenvolver software de qualidade assegurada, com elevada produtividade, dentro do prazo estabelecido e sem necessitar de mais recursos que os alocados, tem sido o grande desafio da engenharia de software. A principal causa dos problemas é a falta de um processo de desenvolvimento de software claramente definido e efetivo. Conhecer os processos significa conhecer como os produtos e serviços são planejados, produzidos e entregues. Cabe ressaltar que, a partir da definição do processo, é possível definir medições e coletar dados de execução. Isto dá visibilidade aos gerentes e técnicos sobre o andamento dos projetos, possibilitando ações para controlar as variações do projeto e dos processos por ele utilizados.

Segundo JACOBSON [1999], um processo de software é a definição de um conjunto completo de atividades necessárias para transformar os requisitos do usuário num conjunto consistente de artefatos que representam um produto de software e, mais tarde, transformar alterações nesses requisitos num novo conjunto consistente de artefatos.

Segundo PRESSMAN [1995], no processo de desenvolvimento de um software, um conjunto de etapas deve ser definido, sendo denominado paradigmas da engenharia de software, também conhecido como modelos de ciclo de vida de software. O processo é um caminho para a construção de software de alta qualidade além de fornecer uma estrutura para o gerenciamento de atividades que facilmente podem sair do controle.

Segundo PAULA [2001], em um processo de desenvolvimento de software, o ponto de partida para a arquitetura é a escolha de um modelo de ciclo de vida.

Destacam-se alguns paradigmas:

Ciclo de vida clássico (*waterfall*) - Segundo PRESSMAN [2001], este modelo possui uma abordagem seqüencial e sistemática para o desenvolvimento que progride através da análise, projeto, codificação, testes e manutenção.

Segundo PAULA [2001], no modelo de ciclo de vida clássico, os principais processos são executados em estrita seqüência, o que permite demarcá-las com pontos de controle bem definidos. Estes pontos de controle facilitam a gestão dos projetos, o que faz com que este seja, em princípio, confiável e utilizável em projetos de qualquer escala. Por outro lado, se interpretado literalmente, é um processo rígido e burocrático, onde as atividades de requisitos, análise e projeto precisam ser muitas bem dominadas, pois não são permitidos erros. O modelo de cascata puro é de baixa visibilidade para o cliente, que só recebe o resultado final do projeto.

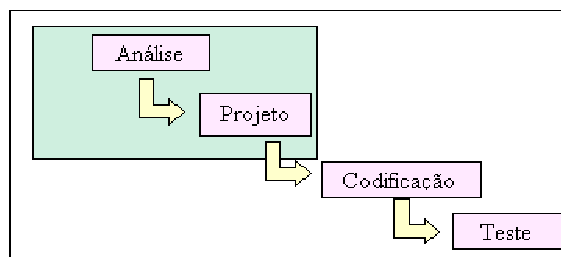


Fig6. Ciclo de vida clássico.

Prototipagem - Segundo LAUDON [1998], a prototipagem consiste na construção de um sistema experimental, de forma rápida e barata para avaliação dos usuários finais. Interagindo com o protótipo, os usuários podem ter uma melhor idéia das informações requeridas. O protótipo validado pelos usuários pode ser usado como um modelo para a criação do sistema final.

A prototipagem como paradigma pode ser problemática por muitas razões. Muitas vezes faz concessões de implementação a fim de colocar um protótipo em funcionamento rapidamente. Com o passar do tempo, as funções do sistema acabam se tornando familiares e o projetista acaba se esquecendo de todas as razões pelas quais elas eram inadequadas, tornando desta forma, a opção menos ideal parte como integrante do sistema.

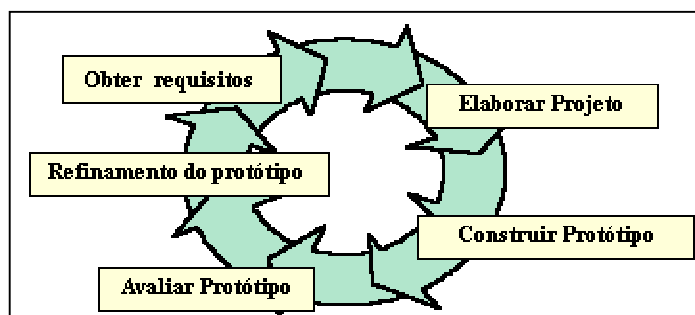


Fig7. Ciclo de vida - Prototipagem.

Modelo incremental – Segundo PRESSMAN [2001], combina elementos do modelo cascata com a filosofia de repetição do modelo de prototipagem. O modelo é bastante útil quando não se pode reunir um grupo de pessoas para concluir uma dada tarefa até o prazo final de entrega.

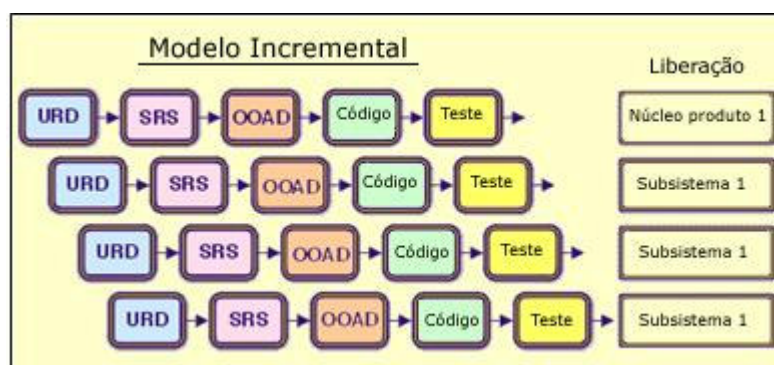


Fig8. Modelo incremental

Modelo espiral - Segundo BOEHM [1988], o modelo é uma inovação que combina os modelos de prototipagem, convencional e incremental para ser usado em várias partes do desenvolvimento.

Segundo PAULA [2001], o produto é desenvolvido em uma série de iterações. Cada nova iteração corresponde a uma volta na espiral. Isto permite construir produtos em prazos curtos, com novas características e recursos que são agregados na medida em que a experiência descobre sua necessidade. As atividades de manutenção são usadas para identificar problemas; seus registros fornecem dados para definir os requisitos das próximas liberações. O principal problema desse ciclo de vida é a necessidade de uma gestão sofisticada para ser previsível e confiável.



Fig9. Modelo espiral

Modelo baseado em componente (tecnologia orientada a objeto) - Esse modelo incorpora algumas das características do modelo espiral, entretanto, ele é composto previamente de componentes de software, chamados de classes.

Modelo concorrente – DAVIS E SITARAM [1994], defendem que as fases do processo de software devem ocorrer de maneira concorrente e não sequencialmente. A descrição do modelo é

feita utilizando um estudo de caso modelado em *statecharts*. O mecanismo pelo qual o processo ocorre é baseado em eventos que sinalizam alterações de estado dentro de cada fase.

Segundo PRESSMAN [1995], independente do paradigma a ser seguido, existem as fases genéricas do desenvolvimento do software que devem ser atingidas:

- Fase de definição - Focalizada no “que” (engenharia de informação, planejamento do projeto de software, análise de requisitos).
- Fase de desenvolvimento - Focalizada em “como” (projeto de software, criação de códigos).
- Fase de suporte - Focalizada na mudança (manutenção corretiva, manutenção preventiva).

### **2.2.3.1 Processo unificado de desenvolvimento de software**

Segundo SCHNEIDER [1998], é um processo de desenvolvimento de software especificado que pode ser especializado para diferentes tamanhos, áreas e classes de projetos. Suas origens incluem, em ordem cronológica: a abordagem usada pela empresa sueca Ericsson para a área de telecomunicações em 1967; a linguagem de especificação e descrição, padronizada pela CCI TT (órgão internacional de padronização em telecomunicações) em 1976; o processo *Objectory* de desenvolvimento de software, da empresa Objectory AB, descrito por Ivar Jacobson em 1988, que difunde o conceito de use cases e outros modelos relacionados; o processo *Rational Objectory* surgido da aquisição da Objectory AB pela empresa Rational, em 1995 e a linguagem de modelagem UML desenvolvida por Grady Booch, Ivar Jacobson e James Rumbaugh em 1995 e considerada um padrão pelo OMG (*Object Management Group*) em 1997. Em 1998 o processo passou a chamar-se RUP - Rational Unified Process.

Segundo JACOBSON [1999], o RUP fornece uma abordagem disciplinada para determinar tarefas e responsabilidades em um desenvolvimento. A meta é assegurar a produção de software de alta qualidade e suprir as necessidades do usuário final, com cronogramas e orçamentos previsíveis além de fornecer uma orientação em como usar a UML.

O RUP se apoia em três conceitos básicos:



- Dirigido por use case - Capturam os requisitos funcionais do sistema e são a base para as atividades de análise, projeto, implementação e teste.
- Centrado na arquitetura - Incorpora os aspectos estáticos e dinâmicos de um sistema e é descrita como diferentes visões do sistema.
- Iterativo e incremental - O software é dividido em projetos pequenos onde cada um deles é uma iteração controlada que resulta em um incremento.

O desenvolvimento de software efetuado através do RUP é incremental e cada incremento é desenvolvido utilizando-se quatro fases:

Concepção - Segundo ROYCE [1998], na fase de concepção são estabelecidos o caso de negócio e o escopo do projeto. O caso de negócios inclui critérios de sucesso, avaliação de riscos, estimativa de recursos necessários e um plano para a fase. Durante a concepção é comum a criação de um protótipo executável, servindo como teste para a concepção.

Elaboração - As metas para a fase de elaboração são a análise do domínio do problema, o desenvolvimento do plano do projeto e a eliminação dos elementos de alto risco. Na fase final de elaboração são examinados o escopo e os objetivos detalhados do sistema, a escolha da arquitetura e a solução para os principais riscos, além de decidir se deve prosseguir com a construção.

Construção - Durante a fase de construção é desenvolvido de maneira incremental e iterativa um produto pronto para a transição à sua comunidade de usuários. Isso implica uma descrição dos requisitos restantes e de critérios de aceitação, dando corpo ao projeto e concluindo a implementação e o teste do software.

Transição - Nesta fase o software será disponível para a comunidade.

Após a fase de transição, o produto pode voltar a percorrer cada uma das fases, constituindo a fase de evolução, na qual se produz uma nova geração do produto.

Cada fase dentro do ciclo se encerra com um *milestone* que ajuda a controlar a evolução do trabalho de desenvolvimento do software, além de definir os critérios de início e fim de cada fase.

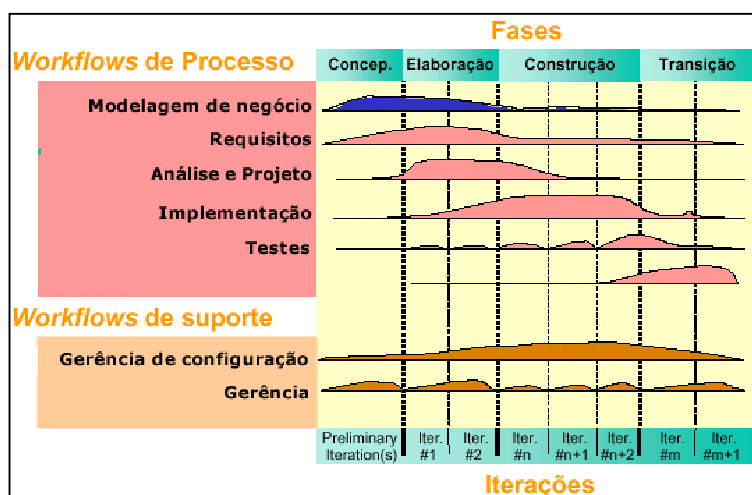


Fig10. Fases do processo RUP.

O RUP é representado usando 4 elementos primários:

- *Workers* ‘Quem’ - Descreve as funções de cada participante do projeto.
- Atividades ‘Como’ - Tarefas entregue a cada *Worker*.
- Artefatos ‘O que’ - São as entradas e saídas das atividades (*use cases*, códigos, etc).
- *Workflows* ‘Quando’ - Sequência de atividades (Análise de requisitos, projeto, implementação e teste).

Segundo a RATIONAL [1998], os *workflows* são divididos em um grupo de 6 *workflows* de Engenharia e 3 de Suporte, descritos abaixo:

### Principais *workflows* do processo RUP

Cada uma das quatro fases do RUP é adicionalmente dividida em iterações e finalizada com um ponto de checagem que verifica se os objetivos de cada fase foram alcançados. Toda iteração é organizada em termos de *workflows* de processo, que são conjuntos de atividades realizadas por responsáveis que produzem artefatos, conforme ilustrado na figura abaixo:

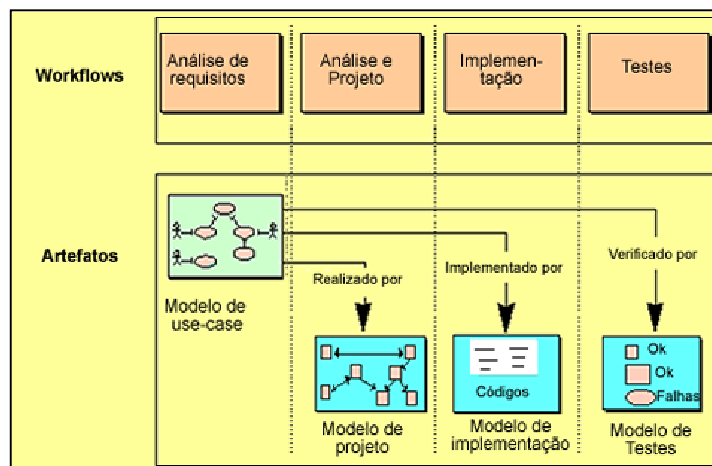


Fig11. Principais workflows.

## Workflows de engenharia

### 2.2.3.1.1 Modelagem do negócio

Durante a fase de concepção, são estabelecidos o modelo de negócio e o escopo do projeto para o sistema. Para efetuar isto, são identificadas todas as entidades externas com a qual o sistema irá interagir (atores) e definido a natureza dessa interação. Isto envolve a identificação de todos use cases e a descrição de cada um. O objetivo é entender os requisitos do sistema que será desenvolvido, com perfeito entendimento do domínio dentro do qual ele irá atuar, identificando subsídios essenciais para que o planejamento do projeto de software seja realizado de forma efetiva. As atividades básicas previstas são:

- Identificar as necessidades do usuário;
- Estabelecer a viabilidade do projeto;
- Alocar funções a hardware, software, pessoas, BD e outros elementos da arquitetura;
- Descrever a arquitetura ou modelo do sistema;

Artefato gerado:

Proposta de desenvolvimento do Sistema - Contém a uma visão geral do documento com a lista de requisitos funcionais, restrições de custos, resultados do estudo de viabilidade técnica e econômica e o cronograma a ser seguido.

### 2.2.3.1.2 Especificação de requisitos

BOEHM [1989] define a fase de especificação de requisitos como uma disciplina, engenharia de requisitos, cujo objetivo é desenvolver uma especificação completa, consistente e não ambígua, servindo de base para um acordo entre todas as partes envolvidas e descrevendo *o quê* o produto de software irá fazer ou executar, mas não como ele será feito.

Segundo PAULA [2001], o fluxo de requisitos reúne as atividades que visam obter o enunciado completo, claro e preciso das necessidades 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.

O objetivo desse *workflow* é conseguir o acordo, entre os usuários finais e o projetista do sistema, sobre a descrição do que o sistema deve fazer.

O processo de captura de requisitos envolve atividades genéricas como: a criação de uma lista de requisitos candidatos (entrevistas), compreensão do contexto do sistema, captura de requisitos funcionais (comportamento do sistema) e não funcionais (propriedades do sistema).

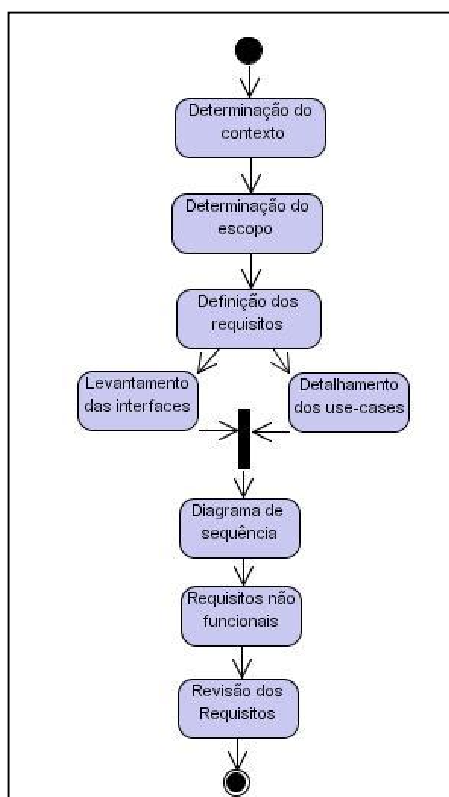


Fig12. Artefatos da análise de requisitos.

Determinação do contexto: O ponto focal do escopo de um produto é a missão que sintetiza o valor acrescentado aos clientes e usuários. Um outro fator que deve ser considerado é a determinação dos limites – evitando falsas expectativas por parte do cliente – e os benefícios do produto. No aspecto gerencial devem ser determinadas as faixas de custos e prazos do projeto.

Definição dos requisitos funcionais: Os requisitos funcionais descrevem as funções que o produto deverá realizar em benefício do usuário. Neste artefato são identificados os atores e os use cases, além do relacionamento entre os dois representados pelo diagrama geral de use case.

Segundo PAULA [2001], os relacionamentos entre use cases e atores são representados através do diagrama de use case, dos quais o principal é o diagrama de contexto do produto.

No diagrama de contexto devem ser mostrados apenas os use cases que se comunicam diretamente com os atores através de interfaces.

Levantamento das interfaces com o usuário:

Segundo PAULA [2001], nas interfaces gráficas de usuário, existem questões que claramente representam requisitos do produto, tais como formatos de dados e comandos. Entretanto, pode ser recomendável que a análise de requisitos contenha um esboço gráfico da interface, já que estes ajudam a identificar mais claramente os requisitos. Caso estes esboços sejam incluídos, fica entendido que eles representam sugestões, pois o detalhamento definitivo será realizado na fase de projeto.

Os use cases geralmente possuem um fluxo principal e vários subfluxos alternativos que representam as variantes executadas sob certas condições. Os subfluxos descrevem detalhes de iterações, ou de condições executadas com frequência.

Detalhamento dos use cases: O detalhamento dos fluxos dos use cases constitui geralmente em uma das tarefas mais demoradas, difíceis e importantes do fluxo de requisitos.

Para cada use case, é preciso levantar as precondições, ou seja, as condições que devem ser satisfeitas ao iniciar a execução de um use case. Em seguida, levanta-se o fluxo principal, que representa a execução mais normal da função, os subfluxos e fluxos alternativos, que representam variantes que são executadas sob certas condições.

Definição de requisitos não-funcionais: Os requisitos não-funcionais incluem requisitos de desempenho e outros atributos de qualidade de produto. Incluem-se também requisitos lógicos e as restrições da fase de projeto.

Existem diversas técnicas que podem ser usadas para capturar os requisitos básicos do cliente e então definir um modelo de análise para o sistema.

O processo pode iniciar com a definição de use cases que descrevem como o sistema será usado. A técnica CRC - Classes, Responsabilidades e Colaborações é aplicada para documentar as classes, atributos e suas operações, além de fornecer uma visão inicial das colaborações que ocorrem entre objetos.

O próximo passo no processo é a classificação de objetos e a criação de classes hierárquicas. Os subsistemas ou pacotes podem ser usados para encapsular os objetos relatados. O modelo de relacionamento entre objetos fornece a indicação de como as classes estão conectadas e o modelo de comportamento indica o comportamento dos objetos individuais e o comportamento do sistema por inteiro.

Usando a notação UML, uma variedade de diagramas de classes pode ser criada com as seguintes características:

- Representação de classes hierárquicas;
- Criação de modelos de relacionamentos entre objetos;
- Criação do modelo de comportamento do objeto;

#### 2.2.3.1.3 Análise e Projeto

Segundo RUMBAUGH [1994], a fase de projeto também pode ser vista como um aprofundamento da análise caminhando em direção a implementação do sistema.

De acordo com a definição da IEEE [1990], projeto é a definição de arquiteturas, componentes, interfaces e outras características resultantes do processo.

Durante a fase de projeto são definidos a estrutura geral, o estilo do sistema e a sua arquitetura.

O processo de projeto do sistema realiza as seguintes atividades:

- Dividir o modelo de análise em subsistemas;
- Identificar a concorrência que é ditada pelo problema;
- Alocar subsistema ao processador e tarefas;
- Escolher uma estratégia básica para implementar o gerenciamento de dados;
- Identificar os recursos e mecanismos de controle necessários para acessá-los;
- Desenvolver um mecanismo de controle para o sistema, incluindo tarefas de gerenciamento.

O projeto do sistema pode ser dividido em duas partes:

- Projeto arquitetural - Relacionado à arquitetura do sistema.
- Projeto detalhado - Lida com objetos individuais e suas interações.

Artefatos gerados:

Relatório com configuração da plataforma e justificativas

Diagrama de implementação

Diagrama de componentes - Modelam os itens físicos que residem em um nó como executáveis, bibliotecas, tabelas, arquivos e documentos.

O principal resultado desta fase é o documento de projeto do software.

#### 2.2.3.1.4 Implementação

Segundo RUMBAUGHN [1994], esta fase é uma simples questão de tradução do projeto para um código, já que as decisões mais difíceis já foram tomadas durante a fase de projeto.

No mercado existem ferramentas do tipo RAD – Rapid application Development que permitem ao usuário um rápido desenvolvimento do sistema baseado em conceitos de reusabilidade. Linguagens como Java, Visual Basic e C++ são as mais utilizadas atualmente.

Na fase de implementação, as classes provenientes do projeto são convertidas para o código da linguagem orientada a objetos escolhida.

Dependendo da capacidade da linguagem utilizada, essa conversão pode ser uma tarefa fácil ou muito complicada. No momento da criação de modelos de análise e projeto em UML, é melhor evitar traduzí-los mentalmente em código. Nas fases anteriores, os modelos criados são o significado do entendimento e da estrutura do sistema, assim, no momento da geração do código

onde o analista conclua antecipadamente sobre modificações em seu conteúdo, seus modelos não estarão mais demonstrando o real perfil do sistema. A implementação é uma fase separada e distinta, onde os modelos criados são convertidos em código.

Durante esta fase são realizadas as seguintes tarefas:

- Definir a organização dos códigos em termos de objetos e classes;
- Implementar classes e objetos em componentes (código fonte, binários, executáveis e outros).

#### 2.2.3.1.5 Teste orientado a objeto

O teste orientado a objeto não difere dos demais desenvolvidos por métodos tradicionais como o estruturado. Em qualquer método, é necessário verificar se o sistema tem sido corretamente desenvolvido de acordo com as especificações dos requisitos do sistema.

Segundo VON MAURHAUSER [1990], várias estratégias de testes podem ser implementadas para assegurar que o software está em acordo com suas especificações e livre de erros. Teste de unidade, integração, sistema, instalação e aceitação são exemplos de técnicas que podem ser utilizadas.

Segundo PRESSMAN, [1995], uma estratégia de teste de software deve estar dividida nas seguintes fases:

- Planejamento de teste - Deve ser definida uma equipe de teste antes de iniciá-lo. Os testes devem ser baseados nos use cases desenvolvidos na fase de análise, verificando se este atingiu seu propósito.
- Execução - Nesta fase são executados os testes de unidade (cada unidade de software é testada separadamente), integração (descobrir os erros associados à interface), validação (verificar se o software como um todo cumpre os requisitos do sistema) e sistema (garantir segurança, desempenho, etc.).
- Avaliação dos resultados - Podemos apontar falhas em determinadas fases do projeto baseando-se nesta avaliação, bem como se certificar da qualidade e/ou bom desempenho do projeto.

Os objetivos da realização de teste são os seguintes:

- Verificar a interação entre os objetos;



- Verificar a integração apropriada de todos os componentes do software;
- Verificar se todos os requisitos foram implementados;
- Identificar os defeitos e garantir que estes estejam endereçados antes do software ser desenvolvido.

### Workflows de Suporte

- 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.

Segundo a RATIONAL [1998], cada atividade do RUP possui vários artefatos associados. Existem nove modelos que, em conjunto, abrangem todas as decisões importantes para a visualização, especificação, construção e documentação de um sistema complexo de software:

- Modelo de negócio - Estabelece uma abstração da empresa.
- Modelo de domínio - Estabelece o contexto do sistema.
- Modelo de use case - Estabelece os requisitos funcionais do sistema.
- Modelo de análise (opcional) - Estabelece o projeto de uma idéia.
- Modelo de projeto - Estabelece o vocabulário do problema e de sua solução.
- Modelo de processo (opcional) - Estabelece os mecanismos de concorrência e de sincronização d sistema.
- Modelo de implantação - Estabelece a topologia do hardware em que o sistema é executado.
- Modelo de implementação - Estabelece as partes utilizadas para montar e liberar o sistema físico.
- Modelo de teste - Estabelece os caminhos pelos quais o sistema é validado e verificado.
- 

Modelo de teste - Estabelece os caminhos pelos quais o sistema é validado e verificado.

## Capítulo Terceiro

### 3 SISTEMA PROPOSTO DE REGISTRO ACADÊMICO

O desenvolvimento e implementação de uma ferramenta para o controle acadêmico envolvem muitos detalhes que devem ser observados. Neste capítulo, serão abordadas algumas etapas do desenvolvimento do sistema baseadas no processo RUP que podem ser executadas em organizações educacionais ou corporativas, dos setores públicos ou privados.

O paradigma adotado para a realização das atividades foi o ciclo de vida clássico.

O processo de desenvolvimento de software segue as seguintes fases descritas abaixo.

#### 3.1 LEVANTAMENTO DE REQUISITOS

A coleta de informações necessárias para a construção do sistema foi realizada através de entrevistas e documentos disponibilizados pelos funcionários do DRA. Segue abaixo o modelo de questionário utilizado para a captura dos requisitos.

Tab1. Formulário para realização de entrevistas

<b>FORMULÁRIO PARA A REALIZAÇÃO DAS ENTREVISTAS.</b>
<b>Perguntas:</b>
<ol style="list-style-type: none"><li>1. Quais as atividades executadas neste departamento em relação ao controle acadêmico?</li><li>2. Quem são as pessoas responsáveis por cada atividade?</li><li>3. Quem pode ser considerado responsável por todas as atividades?</li><li>4. Na montagem de um sistema de controle acadêmico, quem será usuário?</li><li>5. Quais os principais fatores que prejudicam o atendimento aos alunos?</li><li>6. Quais os principais fatores que afetam o funcionamento do DRA?</li><li>7. Em qual ambiente será implantado o sistema?</li><li>8. No seu ponto de vista, quais os benefícios de uma solução bem sucedida?</li><li>9. Quais informações devem ser fornecidas?</li><li>10. Quais informações devem ser armazenadas?</li><li>11. Que funções e desempenhos são exigidos?</li></ol>

**Respostas:**

1. Controle das informações relacionadas aos professores, alunos, notas, horários, disciplinas, regime, calendário, cursos, matrícula, salas e laboratórios, emissão de relatórios, documentos, mapas e diário de classe.
2. As funcionárias do DRA.
3. A secretária chefe do departamento.
4. O módulo interno será utilizado pelos funcionários descritos acima. O sistema Online estará disponível para alunos e professores com suas devidas restrições.
5. Lentidão no envio de documentos e disponibilização de notas; dificuldade na escolha de horário de dependências.
6. A lentidão e a falta de recursos do sistema existente.
7. O módulo Intranet deverá ser implantado em ambiente Windows.
8. Agilidade, confiabilidade, segurança e integrabilidade e redução de custos.
9. Aos alunos deverão ser fornecidas informações como: calendário anual, notas, frequências, Os funcionários deverão ter acesso a todas informações referentes ao sistema. Os professores deverão realizar o controle de notas e frequência através da Internet.
10. Todas as informações referentes às atividades executadas pelo DRA.

### 3.2 PROPOSTA DE DESENVOLVIMENTO DO SISTEMA

Com base nos questionários respondidos pelos funcionários foi possível capturar informações sobre o sistema atual e elaborar uma proposta de desenvolvimento da ferramenta.

A proposta contém uma visão preliminar que será utilizada apenas para iniciar o fluxo de requisitos.

#### 3.2.1 Descrição do sistema atual

Com o aumento de informações, o sistema parcialmente automatizado existente na instituição, tem causado atrasos na disponibilização de informações aos alunos e tornado o gerenciamento de dados cada vez mais complicado.

Os dados pessoais dos alunos ficam armazenados em banco de dados Database File - DBF que não possuem uma separação adequada.

Para a geração da grade horária, o professor envia primeiro ao DRA um plano de curso juntamente com sua disponibilidade de horários que será encaminhado à coordenação do instituto para que o coordenador de cada curso possa montar a grade curricular do curso.

Para a realização da matrícula, o instituto disponibiliza uma sala e com funcionários do DRA onde os alunos devem comparecer, preencher a ficha de matrícula, levar à tesouraria para realizar o pagamento, voltar à sala para finalizar o processo e obter a grade curricular do curso.

O DRA emite uma lista de todos os alunos matriculados, verifica a disponibilidade de cada sala e bloco e realiza a montagem das turmas.

Os procedimentos para a disponibilização de notas são o seguinte:

- O professor recebe um diário de classe contendo todos os alunos matriculados na disciplina lecionada por ele, onde serão inseridas as notas de frequência de cada um;
- O professor, no final de cada bimestre, encaminha o diário com as notas e frequência dos alunos ao DRA;
- O DRA faz a verificação do mapa de notas do professor com os alunos matriculados na disciplina, gera uma cópia do documento e as disponibiliza no quadro de aviso;
- A cópia gerada é arquivada para que no término do ano letivo sejam somadas e emitidas as notas finais.

O DRA possui um modelo de documentos gerais em formato do Word que ao serem requisitados, são salvos com os dados específicos do aluno e ao ser retirado são dados baixa em um caderno de anotações.

### **3.2.1.1 Objetivos**

O sistema atual tem como objetivo principal gerenciar as seguintes informações do DRA, como: controle de alunos, controle de matrícula, montagem do mapa de notas e armazenamento das mesmas, montagem de calendários, estipulação de regimes, controle de disciplinas por curso e série, controle de professores e da instituição em si.

### **3.2.1.2 Problemas observados**

Os principais problemas apresentados pelo sistema atualmente utilizado no DRA são:

- Falta de comunicação com os demais sistemas existentes na instituição;
- Utiliza linguagem de programação Clipper, que se encontra atualmente obsoleta;
- O BD utiliza um SGBD não relacional, aumentando a lentidão das rotinas de acordo com aumento da base de dados;
- Não existe o controle do corpo docente da instituição;

- Não permite acesso do docente para lançamento de notas e frequência;
- Não apresenta nenhum tipo de controle estatístico;
- Não emite todos os relatórios e documentos necessários;
- Não possui um tipo de documentação técnica, inviabilizando melhorias e ampliações por outra pessoa que não seja o desenvolvedor do sistema atual.

### **3.2.1.3 Importância do sistema atual**

Apesar dos problemas observados com o sistema atual, as rotinas que hoje fazem parte deste sistema, são indispensáveis para o controle das atividades pertinentes ao DRA.

## **3.2.2 Descrição do sistema proposto**

### **3.2.2.1 Objetivos**

O sistema proposto terá como principal objetivo melhorar e otimizar as atividades realizadas no DRA, através do armazenamento e gerenciamento das seguintes informações:

- Ano letivo e respectivo calendário;
- Matrícula e rematrícula;
- Informações pessoais do corpo docente;
- Vida acadêmica do aluno;
- Ex-alunos;
- Alunos desistentes;
- Alunos transferidos;
- Trancamento de matrícula;
- Histórico profissional do corpo docente;
- Atividades realizadas pelo docente na instituição;
- Institutos;
- Cursos;
- Disciplinas;
- Mapa de notas;
- Frequência;

- Horários de aula;
- Distribuição do espaço físico
- Controle estatístico;
- Emissão de declarações e relatórios;
- Sistema de consulta de notas e frequência online;
- Sistema de inserção de nota online.

### **3.2.2.2 Premissas e restrições**

Este projeto está sendo desenvolvido para fins acadêmicos, sem fins lucrativos para rodar em ambiente Cliente/Servidor. O desenvolvimento abrangerá recursos humanos, de software e de hardware limitados sob as condições listadas abaixo.

- Sistema Operacional Windows 9X, ME ou XP;
- A linguagem de codificação será VB 6.0;
- A linguagem utilizada para o desenvolvimento das páginas será HTML e PHP.
- O Driver ODBC será utilizado para a comunicação do sistema Intranet com o BD;
- Para o desenvolvimento serão exigidas três máquinas com processador 450Mhz 128 MB de RAM;
- Para utilização será exigida uma máquina com processador 450Mhz 64 MB de RAM, uma impressora matricial 80 colunas;
- O servidor de páginas será rodado no sistema operacional LINUX;
- Servidor de BD será o MySQL;
- A metodologia utilizada para documentação do sistema será a UML;

### **3.2.2.3 Benefícios**

De forma geral, podemos dizer que o sistema proposto terá como principais benefícios:

- Rapidez no acesso as informações;
- Eliminação das principais deficiências do sistema atual;
- Integração com os demais sistemas existentes na instituição;

- Personalização de acordo com os requisitos do cliente;
- Melhor gerenciamento das informações;
- Eliminação das filas no período de matrícula;
- Agilidade no processo de emissão de notas;
- Agilidade na emissão de declarações e relatórios;
- Comodidade ao aluno e professor através do sistema online;

### 3.2.2.4 Alternativas de implementação

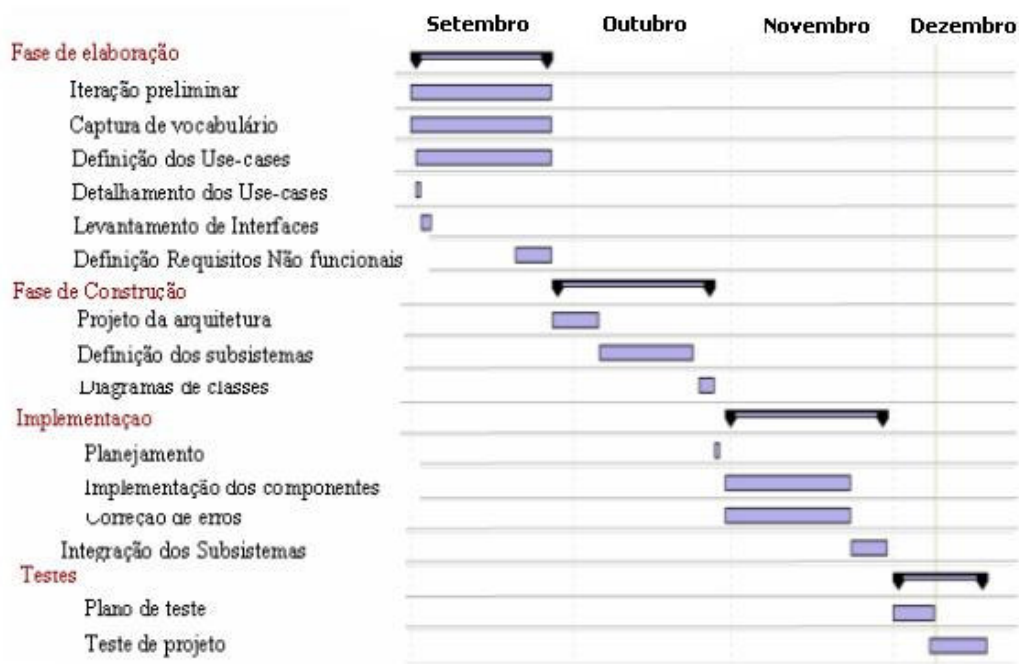


Fig13. Diagrama de Grantt.

### 3.2.3 Relação do sistema com o ambiente

#### 3.2.3.1 Singularidade

Existem no mercado diversas soluções disponíveis para compra e gratuitas que serão utilizadas apenas como referência para o desenvolvimento.

#### 3.2.3.2 Integrabilidade

Este projeto propõe-se a desenvolver parte de uma base de dados global de Informações, integrando outras bases e/ou sistemas existentes na instituição.

### 3.3 ANÁLISES DE REQUISITOS DO SOFTWARE

As atividades realizadas nesta fase são:

- Determinação do contexto.
- Definição do escopo.
- Definição dos requisitos funcionais.
- Definição dos atores e criação dos use cases.
- Detalhamento dos use cases através de cartões.
- Levantamento das classes candidatas.
- Especificação do *layout* sugerido.
- Diagrama de seqüência de atividades

#### 3.3.1.1 *Objetivos do documento*

Descrever e especificar as necessidades do UNIVERSITAS que deverão ser atendidas através do uso do **SisRA na Intranet e Internet**, além de definir as atividades do desenvolvedor.

Público Alvo - Clientes, usuários e desenvolvedor do **SisRA**.

#### 3.3.2 **Escopo do produto**

Nome do produto e seus componentes principais

**SisRA** - Módulo Intranet

**SisRA Online** - Módulo disponível na Internet

#### 3.3.3 **Missão do produto**

Apoio à gestão acadêmica do UNIVERSITAS, com ênfase no gerenciamento de cursos, disciplinas, turmas e alunos, no qual o objetivo principal é melhorar e otimizar as atividades realizadas pela secretária.



### 3.3.4 Limites do produto

O **SisRA** não terá uma função para realizar o controle financeiro do processo de matrícula.

Apenas parte do **SisRA** será desenvolvido para *Web*.

O **SisRA** não fará a emissão do comprovante de matrícula e comprovante de cancelamento de matrícula durante a operação de matrícula e cancelamento de matrícula respectivamente.

#### 3.3.4.1 Benefícios do produto

Na tabela abaixo são identificados os benefícios que se espera obter com o produto e o valor destes para o cliente.

O levantamento dos benefícios torna-se necessário para determinar se o valor do produto compensará o investimento no projeto.

Tab2. Benefícios do produto

Número de ordem	Benefício	Valor para o cliente
1	Agilidade no processamento de matrícula	Essencial
2	Rapidez de acesso às informações	Essencial
3	Diminuição dos gastos	Essencial
4	Confiabilidade nas informações pessoais do aluno.	Essencial
5	Diminuição de erros no processo de matrícula.	Essencial
6	Integração com o sistema de Vestibular	Essencial
7	Qualidade na emissão da grade curricular.	Desejável
8	Agilidade na divulgação de resultados de notas e faltas.	Essencial
9	Diminuição de erros na montagem de turmas.	Essencial
10	Conhecimento mais detalhado das informações dos alunos.	Essencial

### 3.3.5 Descrição geral do produto

#### 3.3.5.1 Especificação do sistema

No sistema de informação DRA, o módulo aluno deverá efetuar a importação dos candidatos aprovados através do módulo vestibular, permitindo que seja feito uma pré - matrícula do mesmo no curso e período escolhido.

O sistema deverá permitir a consulta, alteração e adição de novos dados ao módulo aluno, como: dados pessoais, histórico acadêmico, situação de ex-alunos (transferidos, desistentes, formandos e desistentes), registro de notas e frequências.

O sistema deverá registrar a matrícula das disciplinas referentes a série que será cursada, disponibilizando locais, horários de aulas e previsão de turmas. A montagem do horário será feita através da verificação de pré - requisitos.

Os candidatos aprovados no vestibular deverão cursar o pacote de disciplinas oferecidas pelo curso. Os alunos das demais séries terão a possibilidade de se matricular em disciplinas de outros cursos que possuem a mesma carga horária.

O cálculo de nota deverá ser realizado de acordo com o regime de cada curso.

O sistema deverá permitir a geração e emissão da documentação escolar do aluno.

Além disso, para maior comodidade do professor e do aluno, deverá existir um sistema online para consulta e inserção de notas e visualização das turmas formadas.

### 3.3.6 Requisitos funcionais do sistema

A forma de descrição funcional adotada nesse padrão é a modelagem de use case, baseada na notação UML citada no capítulo 2.

Tab3. Requisitos funcionais - Módulo Intranet

Número de ordem	Use case	Descrição
1	Disponibilizar eventos	Disponibilização dos eventos que ocorrerão no decorrer do ano.
2	Controlar horários	Controle dos horários utilizados pela instituição.
3	Gerar turmas	Geração de turmas, com sua disciplina, curso série e horários.
4	Estipular de regime	Estipulação dos regimes seguidos pela instituição.
5	Controlar de cursos	Processamento de inclusão, exclusão, alteração dos diversos cursos oferecidos na instituição.
6	Controlar disciplinas	Processamento de inclusão, exclusão, alteração e consulta de disciplinas.
7	Controlar docentes	Processamento de inclusão, exclusão, alteração e consulta de docentes.
8	Controlar instituição	Processamento de inclusão, exclusão, alteração de dados referentes à instituição.
9	Controlar institutos	Processamento de inclusão, exclusão, alteração de dados referentes aos institutos.

Número de ordem	Use case	Descrição
10	Controlar de salas e laboratórios.	Processamento de inclusão, exclusão, alteração e consulta de salas e laboratórios da instituição.
11	Realizar matrícula	Realização de matrículas, rematrículas e trancamento de matrículas.
12	Emitir relatórios e documentos	Emissão dos relatórios com as informações da base de dados do SisRA.
13	Controlar notas	Processamento de inclusão, exclusão, alteração de notas referentes a cada bimestre cursado pelos alunos.
14	Controlar de frequência	Processamento de inclusão, exclusão, alteração de frequências referentes a cada bimestre cursado pelos alunos.
15	Gerar <i>Backup</i>	Geração de cópias do sistema.
16	Autenticar funcionários	Permite a autenticação do usuário para a entrada no sistema.
17	Controlar alunos	Processamento de inclusão, exclusão, alteração dos dados pessoais referentes a cada aluno.

Tab4. Requisitos funcionais – Módulo Internet

Número de ordem	Use case	Descrição
1	Consultar notas	Permite ao aluno realizar a consulta de notas através do SisRA online
2	Inserir notas	Permite ao docente realizar o processamento de inclusão, exclusão, alteração de notas referentes a cada bimestre cursado pelos alunos através do SisRA online.
2	Inserir frequência	Permite ao docente realizar o processamento de inclusão, exclusão, alteração de frequência relacionada a cada bimestre cursado pelos alunos através do SisRA online.
3	Realizar cadastro	Permite a realização de cadastro dos docentes e alunos do UNIVERSITAS.
4	Consultar calendário	Possibilita o professor e aluno consultar o calendário da instituição.
5	Consultar horário	Possibilita o professor e aluno consultar os horários das aulas.

### 3.3.6.1 Descrição dos atores

Tab5. Atores e sistemas internos

Número de ordem	Ator	Definição
1	Funcionário	Funcionário responsável pelo cadastro de usuários, controle de alunos, controle de disciplinas, cursos, horários, turmas, realização e cancelamento de matrícula e Emissão de documentos e relatórios.
2	Docente	Profissional responsável por lecionar as disciplinas e pelo controle de notas e faltas do aluno.
3	Aluno	Aluno devidamente matriculado com direito para consulta de notas e faltas de acordo com o direito de acesso cadastrado.

### 3.3.6.2 Diagrama geral de use cases

O diagrama geral de use case (Fig 14), descrito abaixo indica os use cases, usuários e o sistema externo que deverá interagir com o sistema proposto.

### 3.3.6.3 Detalhamento dos use cases

A documentação de um use case é composta por uma descrição textual resumida e fluxos de eventos (fluxo principal, subfluxos alternativos e subfluxos de exceção).

A figura 15 descreve os subfluxos do use case - Realizar matrícula.

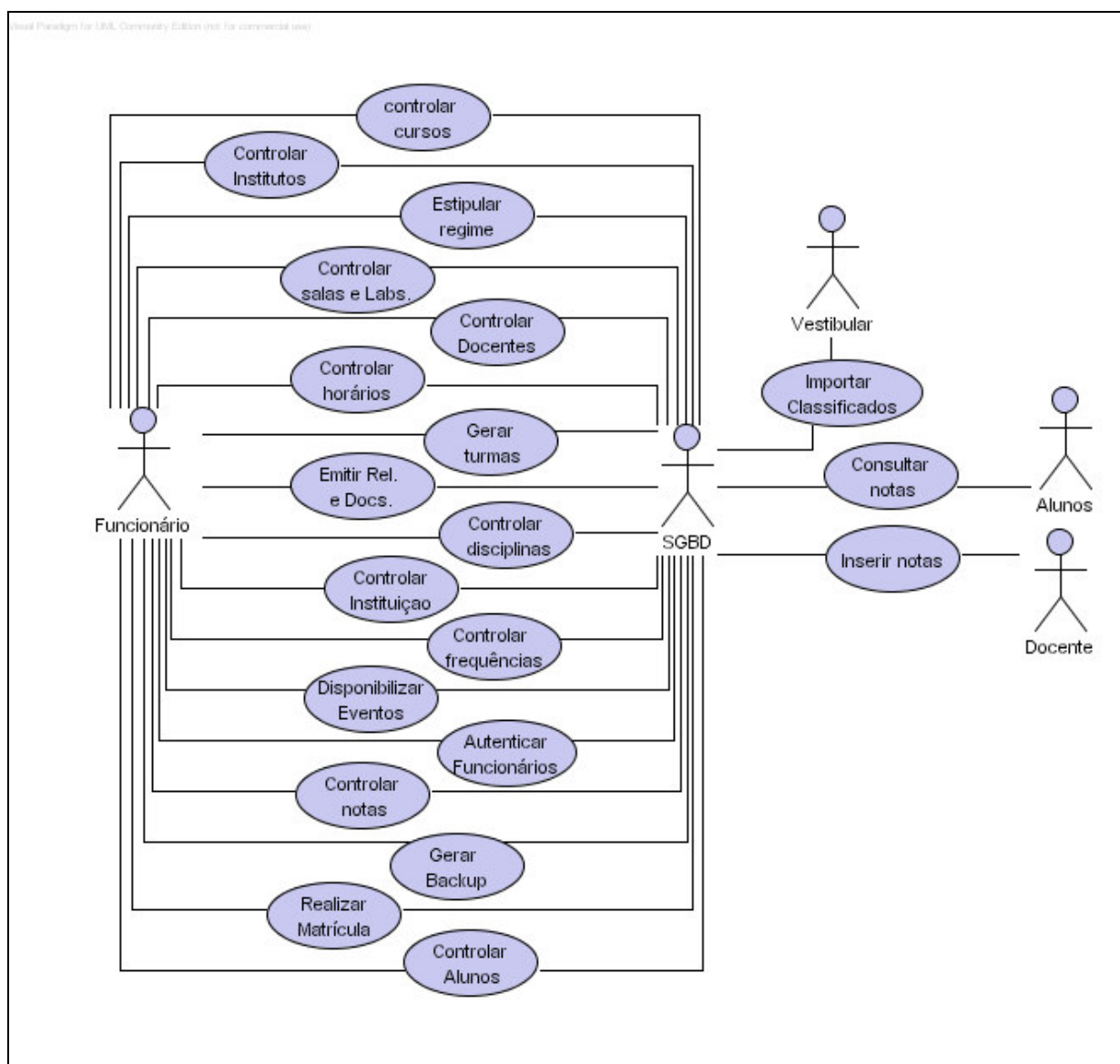


Fig14. Diagrama geral de use case.

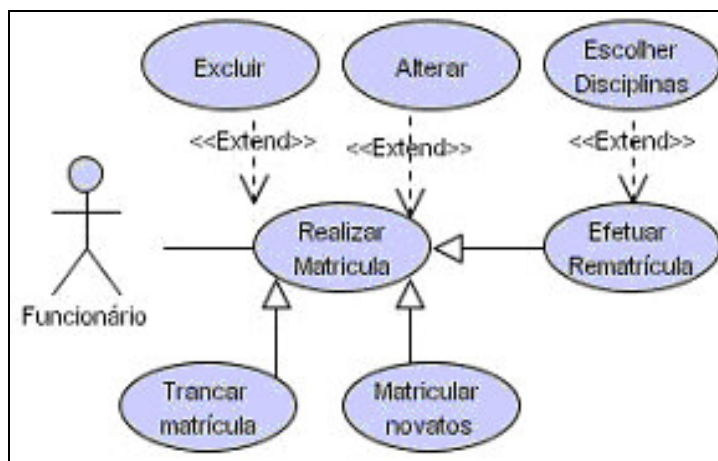


Fig15. Use case - Realizar Matrícula

Tab6. Cartão use case - Realizar matrícula

<b>Use case:</b>	<b>Realizar matrícula.</b>
<b>Atores:</b>	Funcionários.
<b>Proposta:</b>	Realizar a matrícula dos alunos.
<b>Descrição:</b>	O funcionário realiza matrícula dos alunos cadastrados.
<b>Tipo:</b>	Principal.
<b>Use-cases relacionados:</b>	
<b>Fluxos de Evento principal</b>	
<b>Ação do Ator</b>	<b>Resposta do Sistema</b>
1. Use case começa quando o funcionário deseja efetuar a matrícula do aluno	2. O sistema exibe uma tela de matrícula com o status a escolher.
3. Se a atividade seleciona for: Matrícula Re-matrícula Trancamento	4. Subfluxo (Matricular novatos) é executado. Subfluxo (Efetua re-matrícula) é executado. Subfluxo (trancar matrícula) é executado
<b>Subfluxos de exceções</b>	
E1.O funcionário matriculou um aluno errado.	O funcionário pode excluir a matrícula do aluno.
E2.O funcionário fornece algum dado errado.	Os campos podem ser alterados através do botão Alterar.
<b>Subfluxos alternativos</b>	
Matricular novatos. Efetua re-matrícula. Trancar matrícula.	

Subfluxo alternativo - Matricular Novatos

Tab7. Cartão use case – Matricular novatos.

<b>Use case:</b>	<b>Matricular novatos</b>	
<b>Atores:</b>	Funcionários	
<b>Proposta:</b>	Realiza a matrícula dos alunos aprovados no processo seletivo.	
<b>Descrição:</b>	Funcionário matricula os alunos novatos.	
<b>Tipo:</b>	Alternativos	
<b>Subfluxos alternativos</b>		
<b>Ação do Ator</b>		<b>Resposta do Sistema</b>
1. O funcionário escolhe a opção matrícula.		2. O sistema exibe a tela com os campos para a realização de matrícula.
3. O funcionário informa o curso, série, período.		4. O sistema exibe o conjunto de disciplinas pertencente ao curso.
5. O funcionário aciona o botão Ok.		6. O sistema atualiza os campos.
<b>Subfluxos de exceções</b>		
E1.O funcionário matriculou um aluno errado.		O funcionário pode excluir a matrícula do aluno.
E2.O funcionário fornece algum dado errado.		O funcionário pode alterar os campos através do botão alterar.

Subfluxo Alternativo – Efetuar Re-matrícula

Tab8. Cartão use case - Efetuar re-matrícula.

<b>Use case:</b>	<b>Efetuar re-matrícula</b>	
<b>Atores:</b>	Funcionários	
<b>Proposta:</b>	Realiza a re-matrícula dos alunos veteranos.	
<b>Descrição:</b>	Funcionário re-matricula os alunos veteranos.	
<b>Tipo:</b>	Alternativos	
<b>Subfluxos alternativos</b>		
<b>Ação do Ator</b>		<b>Resposta do Sistema</b>
1. O funcionário escolhe a opção re-matrícula.		2. O sistema exibe a tela com os campos para a realização de re-matrícula.
3. O funcionário informa o curso, série, período.		4. O sistema exibe o conjunto de disciplinas de todas as séries pertencente ao curso.
5. O funcionário seleciona as disciplinas desejadas e a série a qual deseja cursar.		6. O sistema lista as disciplinas escolhidas.
5. O funcionário aciona o botão Ok.		6. O sistema atualiza os campos.
<b>Subfluxos de exceções</b>		
E1.O funcionário matriculou um aluno errado.		O funcionário pode excluir a matrícula do aluno.
E2.O funcionário fornece algum dado errado.		O funcionário altera os campos.

Subfluxo - Trancar matrícula

Tab9. Cartão use case - Trancar matrícula.

<b>Use case:</b>	<b>Trancar matrícula</b>	
<b>Atores:</b>	Funcionários	
<b>Proposta:</b>	Realiza o trancamento dos alunos.	
<b>Descrição:</b>	Funcionário efetua trancamento de matrícula.	
<b>Tipo:</b>	Alternativos	
<b>Subfluxos alternativos</b>		
<b>Ação do Ator</b>	<b>Resposta do Sistema</b>	
1. O funcionário escolhe a opção trancamento.	2. O sistema exibe a tela com os campos para a realização de trancamento.	
3. O funcionário escolhe o comando importar alunos novatos.	4. O sistema exibe todos os alunos aprovados no sistema de vestibular.	
5. O funcionário seleciona o aluno desejado e cadastra os campos requeridos.	6. O sistema atualiza os campos.	
7. O funcionário seleciona o botão salvar.	7. O sistema cadastro o aluno.	

**3.3.6.4 Levantamento de classes de interface com o usuário - Módulo Intranet**

Tab10. Levantamentos das interfaces do módulo Intranet

Nome	Ator	Use case	Descrição
Tela de Login	Funcionário	Autenticação de usuários	Interface para autenticar usuários autorizados.
Tela inicial	Funcionário	Abertura do sistema.	Tela que permite navegar pelas demais funcionalidades do sistema.
Tela de eventos	Funcionário	Disponibilizar eventos	Interface para inclusão, impressão, alteração e exclusão dos eventos que ocorrem na instituição.
Tela de horários	Funcionário	Controlar horários	Interface para inclusão consulta, alteração e exclusão de horários.
Tela de regime	Funcionário	Estipular regime	Interface que permite estipular o regime, sistema de peso e cálculo de notas.
Tela de instituição	Funcionário	Controlar instituição	Interface que permite o cadastro dos dados da instituição.
Tela de institutos	Funcionário	Controlar institutos	Interface que permite o cadastro dos institutos pertencentes à instituição.
Tela de cursos	Funcionário	Controlar cursos	Interface para inclusão consulta, alteração e exclusão de cursos e sua grade curricular.
Tela de disciplinas	Funcionário	Controlar disciplinas	Interface para inclusão consulta, alteração e exclusão de disciplinas.

Nome	Ator	Use case	Descrição
Tela de docentes	Funcionário	Controlar docentes	Interface para inclusão consulta, alteração e exclusão de instrutores.
Tela de salas e laboratórios	Funcionário	Controlar Salas e Laboratórios.	Interface que permite controlar as informações referentes às salas e laboratórios existentes na instituição.
Tela gera turmas	Funcionário	Gerar turmas	Interface que permite a criação de turmas.
Tela de Matrícula	Funcionário	Realizar matrícula	Interface para efetivação da matrícula do aluno.
Tela de Relatórios	Funcionário	Emitir relatórios e documentos	Interface para impressão de relatórios e documentos solicitado pelos alunos.
Tela de Lançamento de Notas	Docentes	Controlar notas e frequência	Interface para lançamento das notas e faltas do aluno pelo instrutor.
Tela de <i>backup</i>	Funcionário	Realizar <i>backup</i>	Interface que permite a realização de <i>backup</i> do sistema.

### 3.3.6.5 Levantamento de classes de interface com o usuário - Sistema online

Tab11. Levantamentos das interfaces do módulo Internet.

Nome	Ator	Use case	Descrição
Tela Espaço do aluno	Aluno	Acessar alunos	Permite a entrada no menu principal do aluno.
Tela Espaço professor	Professor	Acessar professor	Permite a entrada no menu principal do professor.
Tela de cadastro do professor	Professor	Realizar cadastro	Permite a realização do cadastro dos professores.
Tela de horário	Professor	Cadastrar horário disponível	Permite cadastrar os horários disponíveis do professor.
Tela de eventos	Professor / Aluno	Consultar eventos	Permite consultar os principais eventos que ocorrerão na instituição durante o ano letivo.
Tela inserção notas Online	Professor	Inserir notas Online.	Permite a inserção Online das notas referentes aos alunos matriculados
Tela de cadastro provas	Professor	Cadastrar provas	Permite o cadastrar as datas de realização de provas.
Tela de calendário	Professor / Aluno	Consultar calendário	Permite consultar o calendário escolar da instituição.
Tela cadastro frequência	Professor	Cadastrar frequência	Permite a inserção das frequências referentes aos alunos matriculados.
Documentação	Aluno	Requisitar documentos	Permite realizar a requisição de documentos e históricos.



Nome	Ator	Use case	Descrição
Tela Consulta Notas Online	Aluno/ Professor	Consultar notas Online.	Permite a consulta Online das notas referentes aos alunos matriculados.
Tela Provas	Aluno	Consultar provas	Permite realizar a consulta das datas de provas referentes às disciplinas cursadas.
Horário	Aluno	Consultar horário	Permite consultar os horários de aula de cada curso.
Tela consulta de frequência	Aluno	Consultar frequência	Permite consultar as frequências referentes à cada disciplina do curso.

### 3.3.6.6 Requisitos relacionados à interface do sistema

Descrevem-se aqui, de forma detalhada, todas as entradas e saídas do produto. As interfaces externas não incluem arquivos de trabalho usados apenas pelo produto, mas incluem qualquer tipo de dados compartilhados com outros produtos e componentes de sistema. O layout da tela deve ser descrito em nível de detalhe suficiente para que o cliente possa aprová-lo ou não, mas sem entrar em considerações de desenho para a usabilidade. Por exemplo, o destaque que se dá a um item de uma janela pode ser item de requisitos. A lista dos campos deve detalhar todos os campos requeridos na interface. Fica entendido que, no projeto da interface definitiva, esses campos podem ser substituídos por soluções funcionalmente equivalentes, desde que isso contribua para facilitar o uso do produto.

Para cada campo, devem constar: nome; valores válidos; formato (por exemplo, tamanho máximo, divisões do campo etc.); tipo (por exemplo, inteiro, real, moeda, data etc.); restrições (por exemplo, opcional, alterável, calculado etc.). Para os comandos ou equivalentes, devem ser descritos: nome; ação (o que deve acontecer quando o comando é acionado); restrições (por exemplo, quanto o comando está habilitado).

As interfaces a seguir, especificam as telas do sistema SisRA que estará disponível para os funcionários do DRA.

#### Tela de login

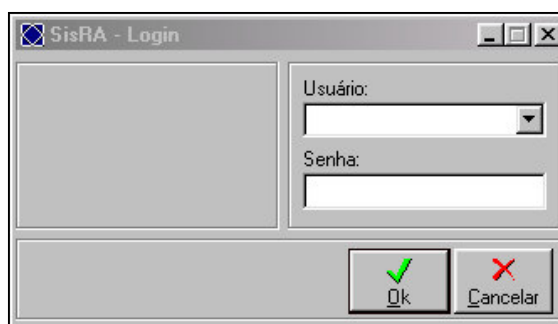


Fig16. Tela de login do usuário.

Relacionamento com outras interfaces - O botão *OK* permite a abertura da tela inicial.

Tab12. Campos da interface Login

Nome	Valores válidos	Formato	Tipo	Restrições
Usuário	Caracteres alfanuméricos.	Até 50 caracteres	Varchar.	Obrigatório
Senha	Caracteres alfanuméricos.	Até 6 caracteres	Varchar.	Obrigatório

Tab13. Comando da interface Login

Nome	Ação	Restrições
Ok	Permite a abertura do sistema.	O uso da senha é obrigatório.
Cancelar	Cancela a abertura do sistema.	Sempre habilitado

### Tela de abertura

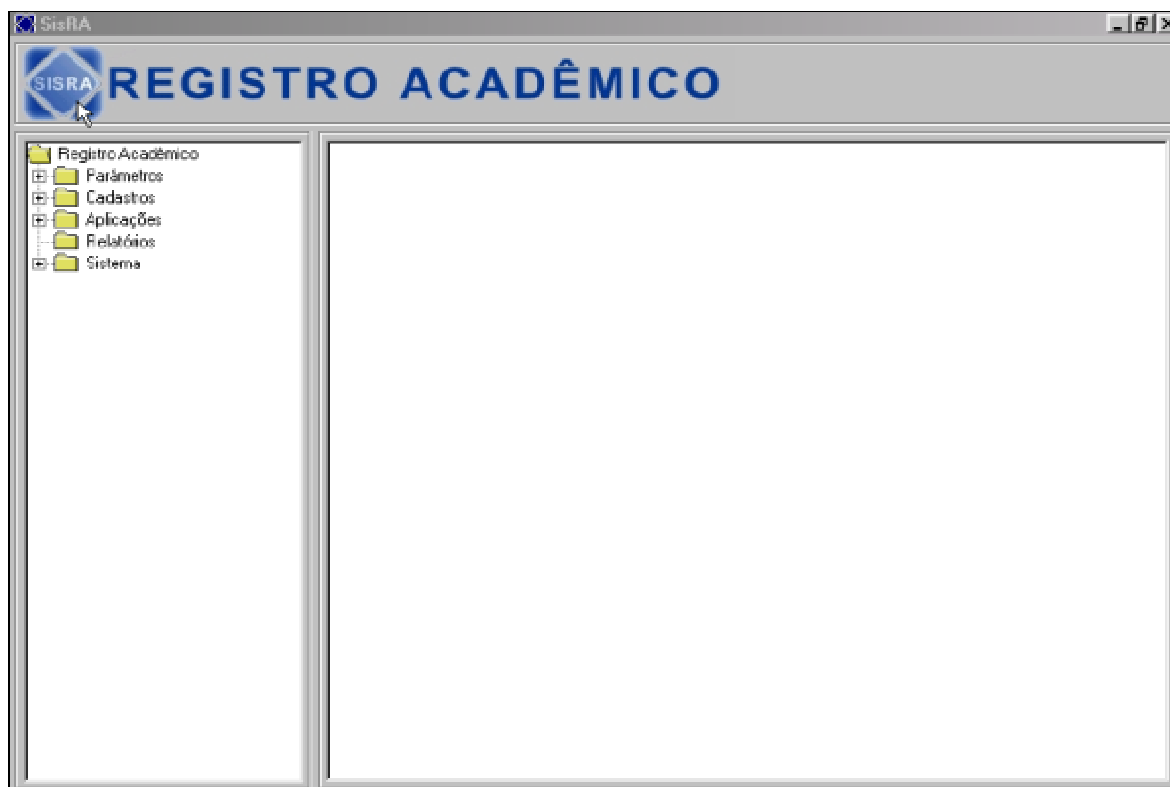


Fig17. Tela de abertura do sistema SisRA.

Relacionamento com outras interfaces

A opção *parâmetros* permite a entrada nas seguintes opções: tabelas de eventos, tabela de horários e tabela de regimes.

A opção *cadastro* permite a entrada nas seguintes opções: cadastro de institutos, instituição, cursos, disciplinas, professores alunos, ex-alunos e salas e laboratórios.

A opção *aplicações* permite a manipulação das seguintes opções: cadastro de ano letivo, calendário, horário, matrícula, gerar turmas, notas, frequência, formandos e histórico.

A opção *relatórios* permite a impressão de diversos documentos relacionados a vida acadêmica do aluno.

A opção *sistema* possibilita a realização de *backup* do sistema e o cadastro dos usuários do sistema.

Tab14. Comando da interface Abertura.

Nome	Ação	Restrições
Parâmetros	Permite escolher as seguintes opções de cadastro: horários eventos e regimes.	Sempre habilitado.
Cadastros	Permite escolher as seguintes opções de cadastro: instituição, curso, disciplina, Professor, aluno, ex-aluno, salas e laboratórios.	Sempre habilitado.
Aplicações	Permite manipular as seguintes opções: instituição, curso, disciplina, Professor, aluno, ex-aluno, salas e laboratórios.	Sempre habilitado.
Relatórios	Permite escolher as seguintes opções de relatórios.	Sempre habilitado.
Sistemas	Manutenção do sistema	Sempre habilitado.

Relacionamento com outras interfaces - O botão *Sair* retorna à interface principal.

## Tela de cadastro de alunos

The screenshot shows the 'SisRA' application window titled 'REGISTRO ACADÊMICO'. On the left, a tree view shows the system structure under 'Registro Acadêmico', including 'Parâmetros', 'Cadastros' (with sub-items like 'Instituição', 'Institutos', 'Cursos', 'Disciplinas', 'Professores', 'Alunos', 'Ex-Alunos', 'Salas e Laboratórios'), 'Aplicações', 'Relatórios', and 'Sistema'. The main area features a toolbar with 'Incluir', 'Alterar', 'Excluir', 'Consultar', 'Imprimir', and 'Sair'. Below the toolbar are tabs for 'Dados Pessoais', 'Dados Escolares', 'Dados Profissionais', 'Documentos', and 'Dados da Matrícula'. The 'Dados Pessoais' tab is active, displaying a form with fields for personal information: 'Nome Aluno', 'Logradouro', 'Número', 'Complemento', 'Bairro', 'CEP', 'Cidade', 'UF' (dropdown), 'Telefone', 'Fax', 'Estado Civil' (radio buttons for Solteiro, Casado, Separado Judicialmente, Divorciado, Viúvo), 'Naturalidade', 'Sexo' (dropdown), 'Nacionalidade', 'Data Nascimento' (date picker), 'RG', 'Orgão Exp.', 'CPF', 'Nome do Pai', and 'Nome da Mãe'. At the bottom, there are fields for 'Data do Cadastrado' and 'Data Alteração', and 'Ok' and 'Cancelar' buttons.

Fig18. Tela Alunos

## Relacionamento com outras interfaces

O botão *Sair* retorna à interface principal.

A opção *Dados escolares* permite a entrada na interface que contem os dados escolares do aluno.

A opção *Dados Profissionais* permite a entrada na interface que contem os dados profissionais do aluno.

A opção *Documentos* permite a entrada na interface que contem os documentos pessoais do aluno.

A opção *Dados da matrícula* permite a entrada na interface que contem os dados da matrícula do aluno.

Tab15. Campos da interface cadastro de alunos.

Nome	Valores válidos	Formato	Tipo	Restrições
Nome do aluno	Caracteres alfanuméricos.	Até 100 dígitos	Varchar.	Informado/alterável
Logradouro	Caracteres alfanuméricos.	Até 10 dígitos	Varchar.	Opcional /alterável
Nome do aluno	Caracteres alfanuméricos.	Até 100 dígitos	Varchar.	Informado/alterável
Logradouro	Caracteres alfanuméricos.	Até 10 dígitos	Varchar.	Opcional /alterável
Número	Caracteres numéricos	Até 10 dígitos	Inteiro	Opcional /alterável
Complemento	Caracteres numéricos	Até 10 dígitos	Inteiro	Opcional /alterável
Bairro	Caracteres alfanuméricos.	Até 10 dígitos	Varchar	Opcional /alterável
CEP	Caracteres numéricos.	Até 10 dígitos	Inteiro	Opcional /alterável
Cidade	Caracteres alfanuméricos.	Até 10 dígitos	Varchar	Opcional /alterável
UF	Caracteres alfanuméricos	Até 2 dígitos	Varchar	Opcional /alterável
Telefone	Caracteres numéricos	Até 10 dígitos	Inteiro	Opcional /alterável
Fax	Caracteres numéricos	Até 10 dígitos	Inteiro	Opcional/alterável
Estado Civil	Casado/solteiro	Opção	Lógico	Obrigatório / alterável.
Naturalidade	Caracteres alfanuméricos.	Até 10 dígitos	Varchar	Opcional /alterável
Sexo	Masculino/feminino	Opção	Varchar	Obrigatório / alterável.
Nacionalidade	Caracteres alfanuméricos.	Até 10 dígitos	Varchar	Opcional /alterável
Data de nascimento	Caracteres numéricos	Até 8 dígitos	Date	Opcional /alterável
RG	Caracteres numéricos	Até 10 dígitos	Inteiro	Obrigatório/alterável
Órgão Exp.	Caracteres alfanuméricos.	Até 3 dígitos	Varchar	Obrigatório /alterável

Nome	Valores válidos	Formato	Tipo	Restrições
CPF	Caracteres numéricos	Até 10 dígitos	Inteiro	Obrigatório/alterável
Nome do pai	Caracteres alfanuméricos.	Até 20 dígitos	Varchar	Opcional /alterável
Nome da mãe	Caracteres alfanuméricos.	Até 10 dígitos	Varchar	Opcional /alterável

Tab16. Comando da interface cadastro de alunos.

Nome	Ação	Restrições
Incluir	Incluir um novo aluno na tabela.	Sempre habilitado, porém quando selecionado, desabilita todos os comando exceto o consultar.
Alterar	Altera um aluno existente na tabela.	Sempre habilitado, porém quando selecionado, desabilita todos os comando exceto o consultar.
Excluir	Exclui um aluno existente na tabela.	Sempre habilitado com confirmação quando houver risco de perda de dados
Consultar	Consulta um aluno existente na tabela.	Sempre habilitado, porém quando selecionado, desabilita todos os comando exceto o consultar.
Imprimir	Imprime uma lista de alunos existentes na instituição.	Sempre habilitado, porém quando selecionado, desabilita todos os comando exceto o consultar.
Sair	Sai da tela de cadastro de aluno.	Sempre habilitado, porém quando selecionado, desabilita todos os comando exceto o consultar.
Cancelar	Cancelar as ações realizadas pelo funcionário.	Habilitado somente quando houver um comando selecionado.
Ok	Confirma as ações realizadas pelo funcionário.	Habilitado somente quando houver um comando selecionado.
Avançar	Avança para o próximo registro.	Sempre habilitado, porém quando selecionado, desabilita todos os comando exceto o consultar.
Retornar	Retorna ao registro anterior.	Sempre habilitado
Ir para o primeiro registro	Retorna ao primeiro registro.	Sempre habilitado
Ir para o último registro	Vai para o último registro.	Sempre habilitado

## Tela de realização de matrícula

Fig19. Tela Matrícula

Relacionamento com outras interfaces

O botão *Sair* retorna à interface principal.

Tab17. Campos da interface Matrícula.

Nome	Valores válidos	Formato	Tipo	Restrições
Opção	Matrícula/re-matrícula e trancamento	Opção	Lógico	Obrigatório / alterável.
Nº de matrícula	Caracteres Numéricos	Até 11 dígitos	Inteiro	Obrigatório / alterável.
Nome aluno	Caracteres alfanuméricos.	Até 10 dígitos	Varchar	Obrigatório / alterável.
Código do curso	Caracteres numéricos	Até 3 dígitos	Inteiro	Obrigatório / alterável.

Nome	Valores válidos	Formato	Tipo	Restrições
Série	Caracteres numéricos.	Até 11 dígitos	Inteiro	Opcional /alterável
Disciplina	Caracteres alfanuméricos.	Até 15 dígitos	Varchar	Opcional /alterável
Re-matrícula em	Caracteres numéricos	Até 8 dígitos	Data	Obrigatório / alterável.
Disciplinas selecionadas	Caracteres alfanuméricos.	Até 15 dígitos	Varchar	Fornecido pelo programa.

Tab18. Comando da interface Matrícula.

Nome	Ação	Restrições
Incluir	Incluir uma nova matrícula, re-matrícula ou trancamento do aluno.	Sempre habilitado, porém quando selecionado, desabilita todos os comando exceto o consultar.
Alterar	Altera um aluno existente na tabela.	Sempre habilitado, porém quando selecionado, desabilita todos os comando exceto o consultar.
Excluir	Exclui um aluno existente na tabela.	Sempre habilitado com confirmação quando houver risco de perda de dados
Consultar	Consulta um aluno existente na tabela.	Sempre habilitado, porém quando selecionado, desabilita todos os comando exceto o consultar.
Sair	Sai da tela de cadastro de aluno.	Sempre habilitado, porém quando selecionado, desabilita todos os comando exceto o consultar.
Cancelar	Cancelar as ações realizadas pelo funcionário.	Habilitado somente quando houver um comando selecionado.
Ok	Confirma as ações realizadas pelo funcionário.	Habilitado somente quando houver um comando selecionado.
Avançar	Avança para o próximo registro.	Sempre habilitado, porém quando selecionado, desabilita todos os comando exceto o consultar.
Retornar	Retorna ao registro anterior.	Sempre habilitado
Ir para o primeiro registro	Retorna ao primeiro registro.	Sempre habilitado
Ir para o último registro	Vai para o último registro.	Sempre habilitado



## Tela de gerar turmas

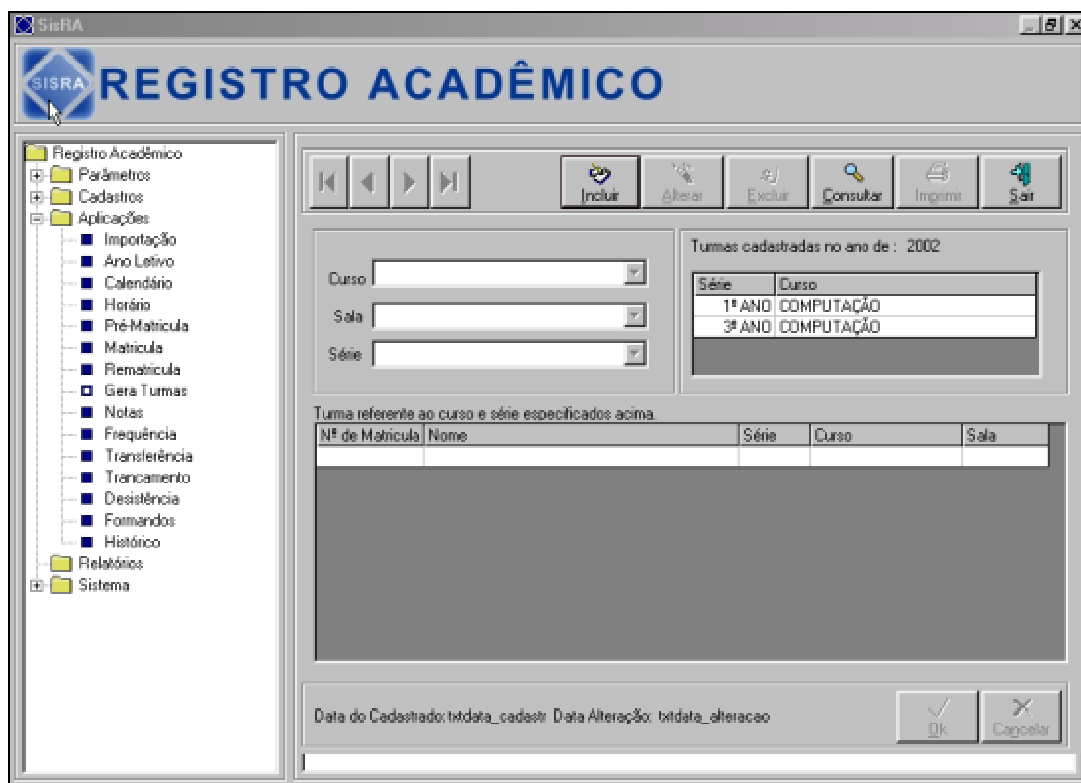


Fig20. Tela gera turmas.

Tab19. Campos da interface Gera turmas.

Nome	Valores válidos	Formato	Tipo	Restrições
Código do curso	Caracteres numéricos	Até 3 dígitos	Inteiro	Obrigatório / alterável.
Sala	Caracteres alfanuméricos.	Até 1 dígito	Varchar	Obrigatório / alterável.
Série	Caracteres numéricos.	Até 11 dígitos	Inteiro	Opcional /alterável
Disciplina	Caracteres alfanuméricos.	Até 15 dígitos	Varchar	Opcional /alterável

Tab20. Comando da interface Gera turmas.

Nome	Ação	Restrições
Incluir	Incluir uma nova matrícula, re-matrícula ou trancamento do aluno.	Sempre habilitado, porém quando selecionado, desabilita todos os comando exceto o consultar.
Alterar	Altera um aluno existente na tabela.	Sempre habilitado, porém quando selecionado, desabilita todos os comando exceto o consultar.
Excluir	Exclui um aluno existente na tabela.	Sempre habilitado com confirmação quando houver risco de perda de dados
Consultar	Consulta um aluno existente na tabela.	Sempre habilitado, porém quando selecionado, desabilita todos os comando exceto o consultar.
Imprimir	Imprime uma lista de alunos existentes na instituição.	Sempre habilitado, porém quando selecionado, desabilita todos os comando exceto o consultar.
Sair	Sai da tela de cadastro de aluno.	Sempre habilitado, porém quando selecionado, desabilita todos os comando exceto o consultar.
Cancelar	Cancelar as ações realizadas pelo funcionário.	Habilitado somente quando houver um comando selecionado.
Ok	Confirma as ações realizadas pelo funcionário.	Habilitado somente quando houver um comando selecionado.
Avançar	Avança para o próximo registro.	Sempre habilitado, porém quando selecionado, desabilita todos os comando exceto o consultar.
Retornar	Retorna ao registro anterior.	Sempre habilitado
Ir para o primeiro registro	Retorna ao primeiro registro.	Sempre habilitado
Ir para o último registro	Vai para o último registro.	Sempre habilitado

As interfaces a seguir, especificam as telas do sistema online que estará disponível para os professores e alunos do UNIVERSITAS.

As *interfaces* descritas a seguir foram desenvolvidas utilizando HTML e PHP. No endereço <http://www.fepi.br/>, clicando em uma das opções (visão do aluno/ professor) tem-se acesso à página de *login* do sistema. A figuras abaixo apresentam o layout da visão do aluno e professor.

### Tela de consulta de notas - Visão do aluno

The screenshot shows the SISRA (Sistema de Registro Acadêmico) interface for a student. The browser address bar shows `http://localhost/DRA2/notas.php`. The page title is "SISRA SISTEMA DE REGISTRO ACADÊMICO". The main heading is "Relatório de Notas".

**DADOS DO ALUNO**

Matrícula: 1 Aluno: Jose Curso: Informática

Relação de notas referentes ao Ano Letivo de 2002.

**REGIME ANUAL**

Disciplina	1º Bimestre	2º Bimestre	3º Bimestre	4º Bimestre	Final	TOTAL
Engenharia de Software	80	70	70	70		-
Banco de dados	50	30	30	20		-
Sistema de Informação	60	80	50	50		-
Cálculo Numérico	70	70	70	70		-
Processamento de Dados	50	60	35	0		-

OBS: A nota referente à disciplina de Processamento de dados ainda foi disponibilizada pelo professor.

The left sidebar contains navigation links: VISÃO DO ALUNO, Matrículas, Ex-alunos, Eventos, Documentação, Notas, Provas, Horários, Calendário Escolar, and Frequência. The date 12/17/2002 is displayed at the bottom of the sidebar.

Fig21. Tela do aluno - Consultar notas.

### Tela de inserção de notas - Visão do professor

The screenshot shows the SISRA (Sistema de Registro Acadêmico) interface for a professor. The browser address bar shows `http://localhost/DRA2/Professor/inserir_notas.php`. The page title is "SISRA SISTEMA DE REGISTRO ACADÊMICO". The main heading is "INSERIR NOTAS".

**INSERIR NOTAS**

Seleção de curso: Computação Seleção de série: 2º Ano Seleção de período: 2º Bimestre

Seleção de disciplina: Lógica de Programação

Seleção de curso: Ângela Martins Insira a nota: 73.5

Gravar Cancelar

The left sidebar contains navigation links: VISÃO DO DOCENTE, Cadastro, Horários, Eventos, Controle de Notas, Provas, Agenda, Calendário Escolar, and Frequência. The date 12/17/2002 is displayed at the bottom of the sidebar.

Fig22. Tela do professor -Inserir notas.

### 3.3.6.7 Diagrama de seqüência do use case: Realizar matrícula

O diagrama de seqüência abaixo mostra a interação entre os objetos ao longo do tempo. São apresentados os objetos que participam da interação e a seqüência de mensagens trocadas.

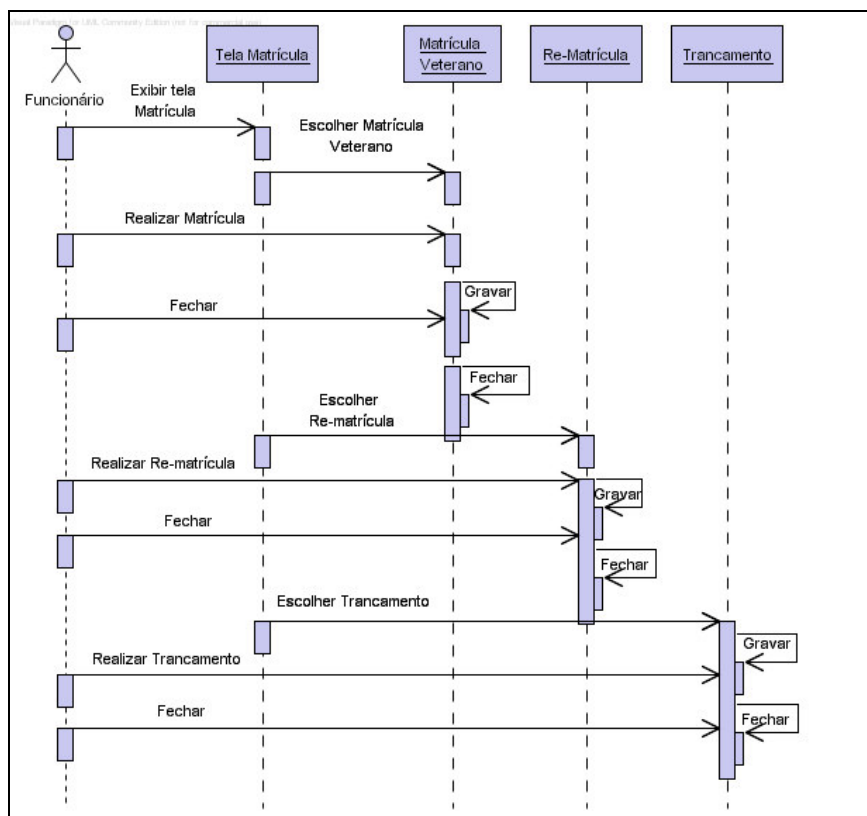


Fig23. Diagrama de seqüência – Realizar matrícula.

### 3.3.7 Requisitos não funcionais

Conforme visto no capítulo anterior, os requisitos não funcionais dizem respeito à qualidade sistema. Destacam-se desempenho, integridade, confiabilidade, segurança, projetos de interfaces incluindo o uso de cores e outros.

#### 3.3.7.1 Segurança

As operações do sistema só podem ser realizadas por funcionários registrados, as consultas podem ser feitas também por usuários cadastrados.

Os alunos terão acesso ao sistema online através de senhas criptografadas fornecidas na efetuação do cadastro.

Contabilização - O sistema deve ser capaz de registrar os eventos relevantes à segurança em um arquivo de *log* que possibilite verificar todas essas ações. Além disso, o sistema deve permitir a

escolha dos eventos a serem registrados para minimizar o tamanho dos arquivos com o objetivo de garantir sua eficiência. Os *logs* devem ser protegidos contra alteração ou destruição não autorizadas.

### 3.3.7.2 Desempenho

Tempo de resposta operação de matrícula – O intervalo de tempo entre a requisição de inserção de novas turmas em uma determinada matrícula e a confirmação da matrícula não podem ultrapassar 2 segundos.

Tempo de resposta para operação de cancelamento de matrícula - O intervalo de tempo entre a requisição de cancelamento de matrícula e a confirmação do cancelamento não pode ultrapassar 1 segundo).

Tempo de resposta para pesquisa no BD - O tempo para realização de qualquer operação de pesquisa de turmas, alunos, docentes, disciplinas, cursos, notas, horários, matrículas, não pode ser maior do que 10 segundos.

## 3.4 PROJETO DA ARQUITETURA DO AMBIENTE

Para a implementação da ferramenta que estará disponível na secretaria, foram utilizadas as seguintes opções:

Tab21. Requisitos do módulo Intranet.

Restrição	Descrição
Sistema operacional	O Sistema operacional compatível (Windows 9X) para a linguagem escolhida.
Ambiente	O sistema deverá executar em um Pentium 166 MHz com 16MB ou superior, com impressora de tecnologia laser ou de jato de tinta, a ser usada para impressão de todos os relatórios.
Servidor de banco de dados	O servidor no qual ficará o BD será o LINUX por questões de segurança.
Linguagem	VB 6.0 foi escolhida pelos clientes.

Para o módulo disponível na Internet foram escolhidas as seguintes opções:

Tab22. Requisitos do módulo Internet.

Número de ordem	Restrição	Descrição
1	Camada de interface com usuário	A linguagem HTML foi escolhida pelo fato de ser interpretada por qualquer browser em qualquer plataforma.
2	Camada <i>middleware</i>	A linguagem PHP foi escolhida porque suporta várias plataformas e uma variedade de servidores <i>Web</i> , além de permitir conexões persistentes do BD.
3	Camada de BD	O MySQL foi escolhido pelas seguintes razões: <ul style="list-style-type: none"> <li>• BD relacional Open Source;</li> </ul> Altamente escalonável e fácil de administrar;
4	Segurança	O produto deverá restringir o acesso através de senhas individuais para cada cliente e usuário.
5	Servidor <i>Web</i>	O Apache foi escolhido pelo fato de ser Open Source.
6	Sistema Operacional LINUX	O LINUX foi escolhido por questões de segurança.

O esboço de uma arquitetura é imprescindível para que um projeto se apresente de forma clara ao leitor. Segue abaixo a representação genérica da arquitetura do software.

O projeto arquitetural visa decompor o sistema em subsistemas menores para reduzir a complexidade do problema original. São identificadas e definidas camadas (subsistemas), módulos (partes de subsistemas) e interdependências.

### 3.4.1 Modelo de implementação

Segue abaixo o diagrama de implementação que modela os aspecto físico do sistema.

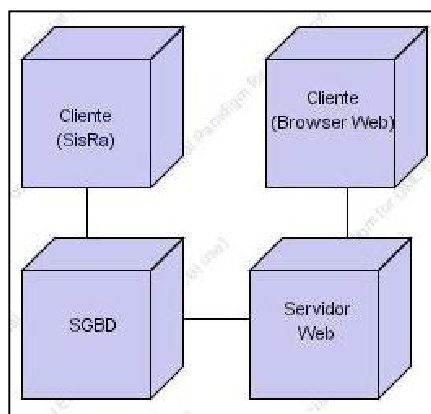


Fig24. Diagrama de implantação

### 3.4.1.1 Diagrama de colaboração - SisRA

O diagrama de colaboração abaixo determina as responsabilidades de cada objeto para enviar e receber mensagens.

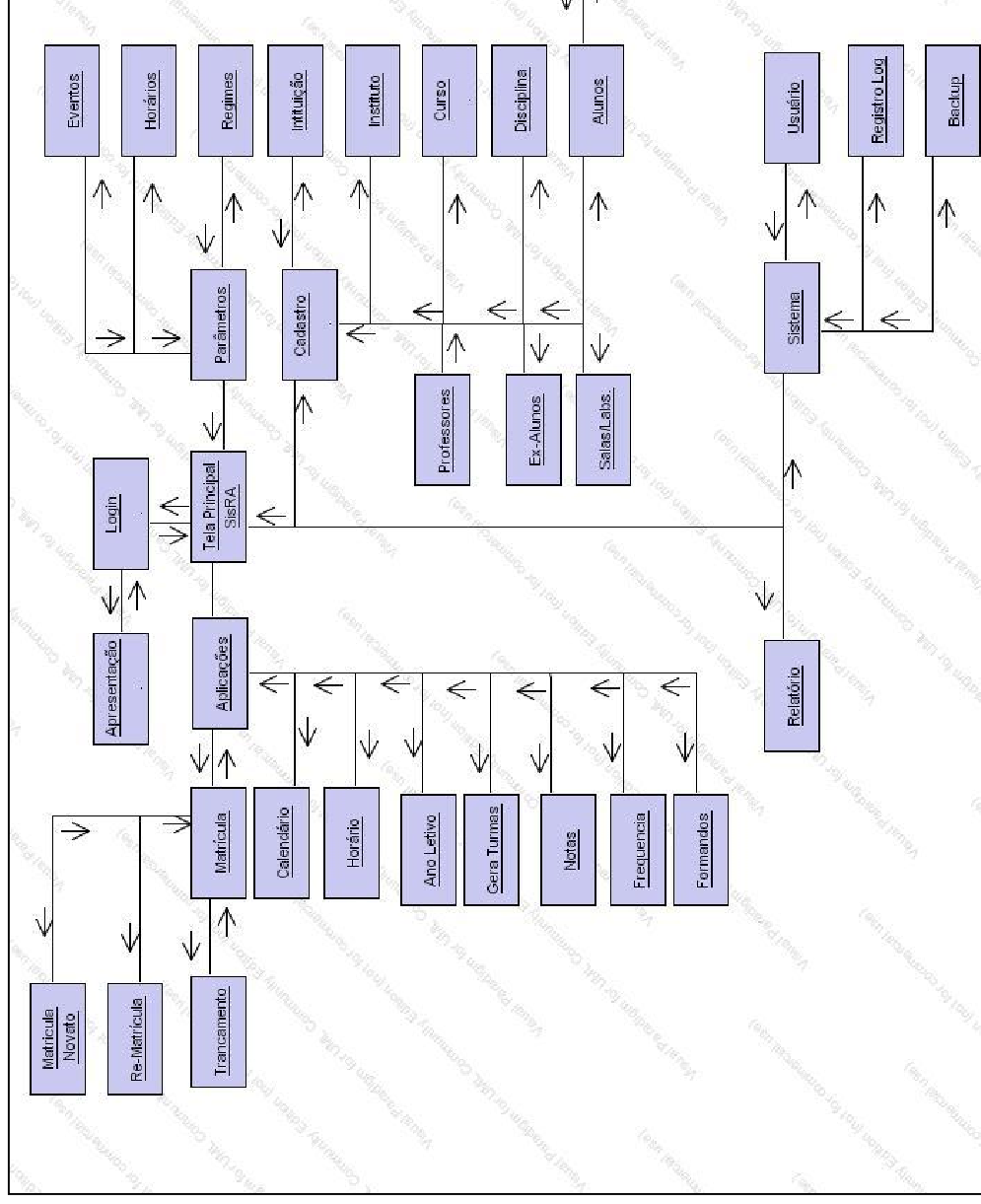


Fig25. Diagrama de colaboração – Módulo Intranet.

### 3.4.2 Subsistemas e componentes

Os subsistemas abaixo representam os sistemas envolvidos no projeto, e são baseados nos pacotes de arquitetura e análise.

Os subsistemas serão posteriormente detalhados em componentes individuais ou conjuntos de componentes.

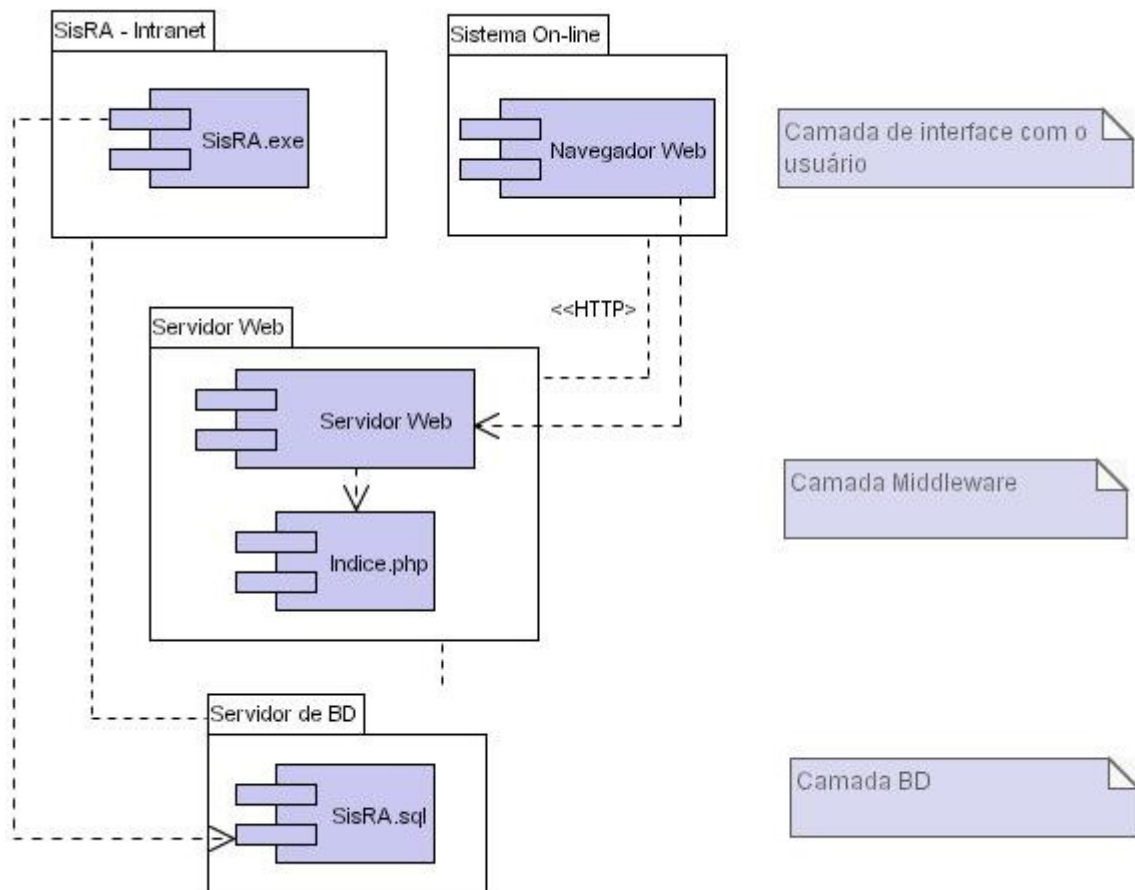


Fig26. Camadas do sistema.



### 3.4.2.1 Diagrama de interface –Intranet

O diagrama de interface define as classes, atributos e serviços de cada tela do módulo Intranet.

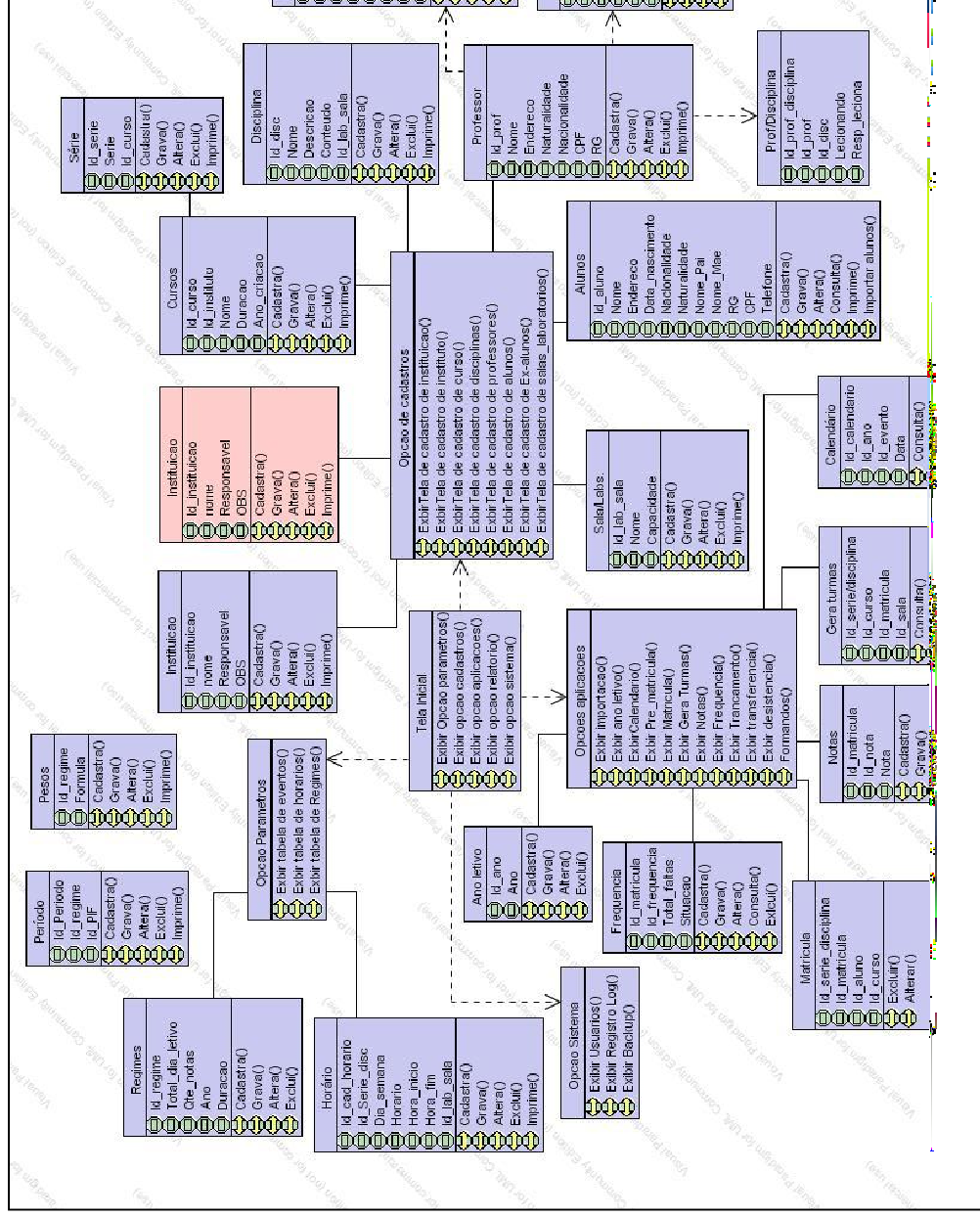


Fig27. Diagrama de interface - Intranet.

### 3.4.2.2 Diagrama de interface -Internet

O diagrama de interface define as classes, atributos e serviços de cada tela do módulo Intranet.

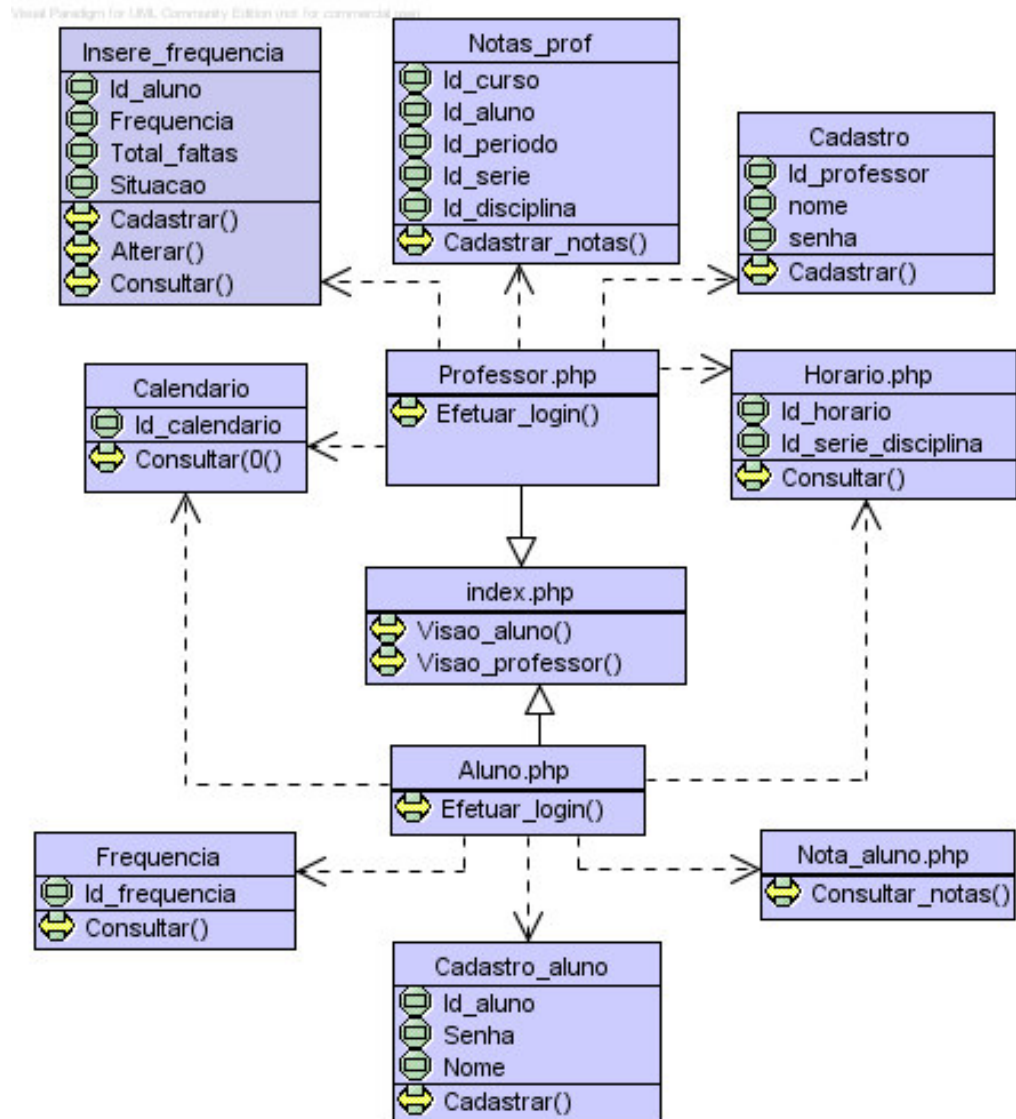


Fig28. Diagrama de Interface – Sistema Online.

### 3.4.2.3 Diagrama de classe de persistência

O diagrama de persistência define as classes pertencentes ao BD que estará disponível no servidor.

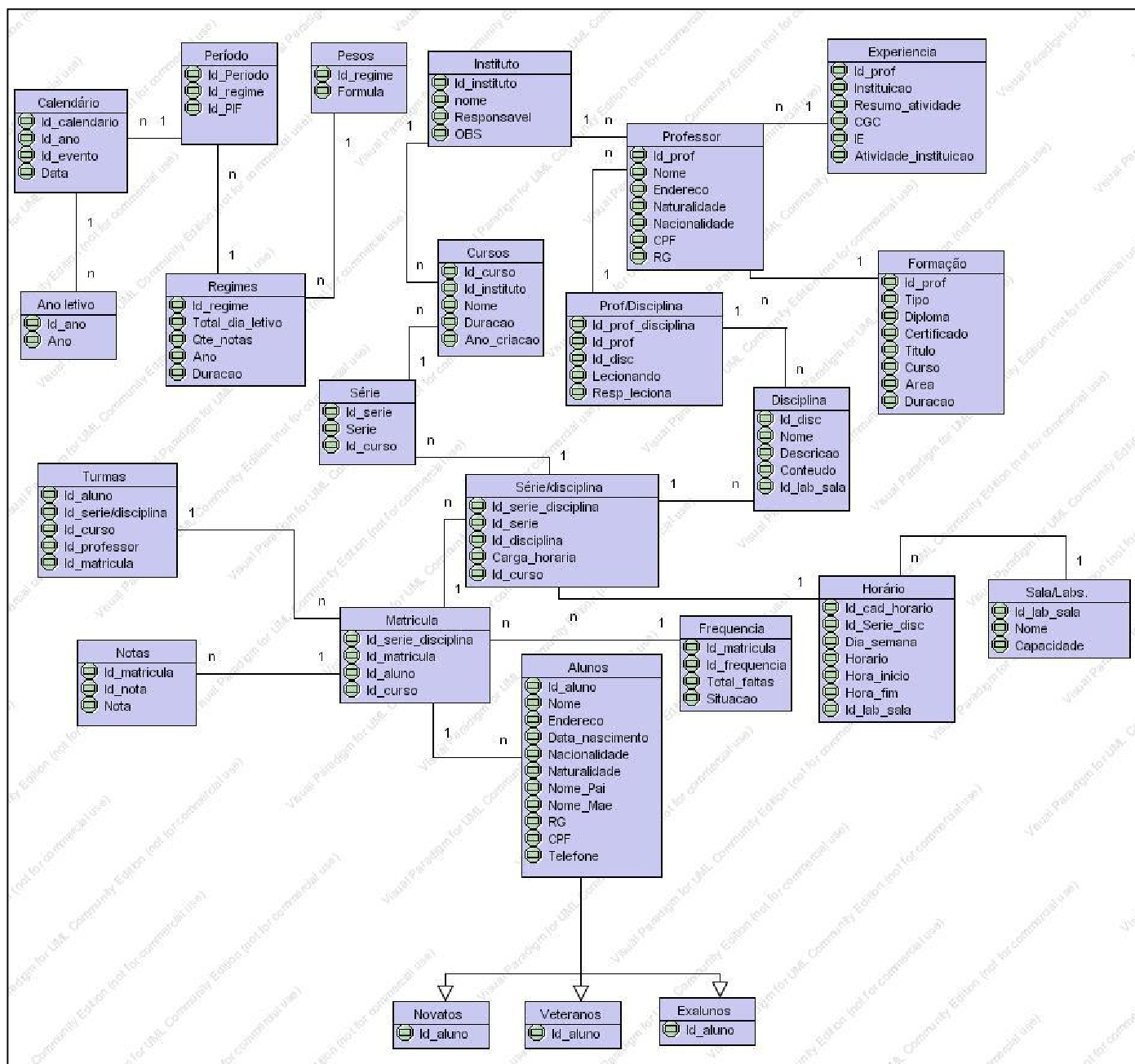


Fig29. Diagrama de classe persistente

### 3.4.3 Diagrama de componentes

Mostra a organização e a dependência de um conjunto de componentes, representando a visão sistema. É relacionado ao diagrama de classes, já que um componente mapea uma ou mais classes, interfaces

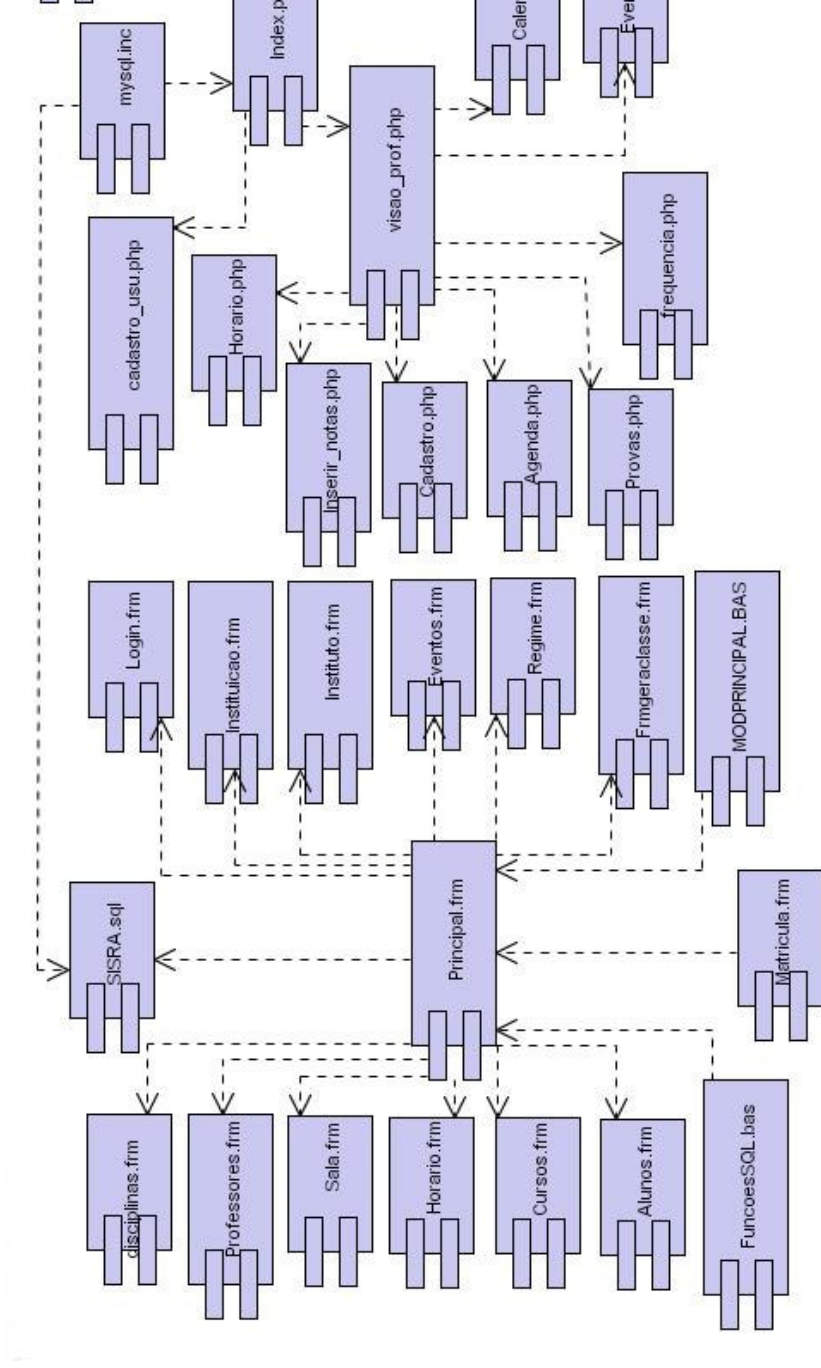


Fig30. Diagrama de componentes.

### 3.4.3.1 Tratamento de erros

O tratamento dos erros cometidos pelo usuário e erros do sistema é feito através de mensagens explicativas. Alguns destes erros podem ser corrigidos em tempo de execução e não impedem que o usuário prossiga com o seu trabalho. Erros que não podem ser corrigidos pela aplicação (como falhas de sistema operacional) exigem que o sistema seja reiniciado. Toda ação potencialmente destrutiva exige confirmação do usuário.

### 3.4.3.2 Tratamento da ajuda ao usuário

Será disponibilizada ajuda online para o usuário. O acesso à ajuda online pode ser feito a qualquer momento durante a execução do sistema, através da tecla F1. A ajuda é dependente de contexto, isto é, serão mostradas explicações de acordo com a situação de quando ela foi solicitada, permitindo-se navegação para tópicos relacionados ou não. A ajuda online descreverá as funções de todos os elementos da interface e o fluxo de execução das funções do produto.

Tab23. Tratamento de erros.

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

### 3.4.3.3 Segurança

Ao se cadastrar o usuário deverá fornecer uma senha, que dependerá sua entrada no sistema. As opções disponíveis durante a execução do sistema dependem do tipo de usuário que está conectado.

### 3.4.3.4 Plano de testes

O principal objetivo do teste realizado foi testar a funcionalidade da ferramenta SisRA implementada comparando com as especificações de requisitos. Segue abaixo a lista de requisitos que foram realizados os testes.

Tab24. Teste dos use cases.

Use case	Funcionou	Não Funcionou
Disponibilizar eventos	✓	
Controlar horários	✓	
Gerar turmas	✓	
Estipular de regime	✓	
Controlar de cursos	✓	
Controlar disciplinas	✓	
Controlar docentes	✓	
Controlar instituição	✓	
Controlar institutos	✓	
Controlar de salas e laboratórios.	✓	
Emitir relatórios e documentos	✓	
Controlar notas	✓	
Controlar de frequência	✓	
Gerar <i>Backup</i>	✓	
Autenticar funcionários	✓	
Controlar alunos	✓	

Abaixo segue o detalhamento dos testes realizados com os use cases relacionados à realização de matrícula.

Tab25. Procedimento de teste.

Número	Procedimento de teste
1	Realizar matrícula
2	Matricular novatos.
3	Efetuar re-matrícula.
4	Trancar matrícula.

Tab26. Critérios de completudeza e sucesso.

<b>Número</b>	<b>Critérios</b>
<b>1</b>	A tela de matrícula funcionou corretamente ao selecionar as opções existentes.
<b>2</b>	Depois de seleciona a opção Matrícula novatos, todo o procedimento de cadastro da matrícula foi realizado com sucesso. as inscrições pelo nome do candidato.
<b>3</b>	A opção Efetuar re-matrícula funcionou corretamente.
<b>4</b>	A opção trancamento de matrícula funcionou corretamente.

### Procedimento de teste Use case realizar matrícula

Tab27. Procedimento de teste – Realizar Matrícula

<b>Identificação</b>	<b>Use case realizar matrícula</b>
<b>Objetivo</b>	Verificar se as opções de Matricular novatos, efetuar re-matricula e trancar matrícula foi realizada corretamente.
<b>Requisitos especiais</b>	Nenhum.
<b>Fluxo</b>	<p>Abrir interface Tela Matrícula.            Escolher opção            Escolher <b>Matrícula</b></p> <ul style="list-style-type: none"> <li>Selecionar o número de matrícula.</li> <li>Selecionar o curso.</li> <li>Selecionar a série.</li> <li>Selecionar o botão Ok.</li> </ul> <p>Abrir interface Tela Matrícula.            Escolher opção            Escolher <b>Matrícula</b></p> <ul style="list-style-type: none"> <li>Selecionar o número de matrícula.</li> <li>Selecionar o curso.</li> <li>Selecionar a série.</li> <li>Selecionar o botão Ok.</li> </ul> <p>Escolher <b>Rematrícula</b></p> <ul style="list-style-type: none"> <li>Selecionar o número de matrícula.</li> <li>Selecionar o curso.</li> <li>Selecionar a série.</li> <li>Escolher as disciplinas que serão cursadas.</li> <li>Selecionar o botão Ok.</li> </ul> <p>Escolher <b>Trancamento</b></p> <ul style="list-style-type: none"> <li>Selecionar o número de matrícula.</li> <li>Selecionar o curso.</li> <li>Selecionar a série.</li> <li>Escolher as disciplinas que será trancada.</li> <li>Selecionar o botão Ok.</li> </ul>

## Capítulo quarto

### 4 AVALIAÇÃO DA FERRAMENTA DE REGISTRO ACADÊMICO

#### 4.1 ANÁLISE DOS DADOS

As próximas seções têm como objetivo descrever o funcionamento dos módulos Internet e Intranet do SisRA, onde será apresentada uma análise relativa aos testes realizados com alunos, funcionários e docentes.

Até a data de conclusão deste documento, o período de matrículas ainda não havia iniciado, ou seja, o resultado real da utilização da ferramenta será conhecido apenas no início do ano letivo de 2003. Porém, com a finalidade de testar o funcionamento da ferramenta, foi simulado todo o processo de matrícula onde os dois módulos foram avaliados.

#### Público-Alvo

O público-alvo da ferramenta disponível na Internet pode ser visto na figura abaixo, que está dividido entre docentes e alunos entre faixas etárias variadas.

A ferramenta será desenvolvida principalmente para atender as necessidades dos alunos, pois como já visto no capítulo 2, estes clientes, que são do tipo "cliente-produto", geralmente têm grande influência no processo de tomada de decisões. O módulo Intranet foi desenvolvido para atender as necessidades do funcionário do DRA.

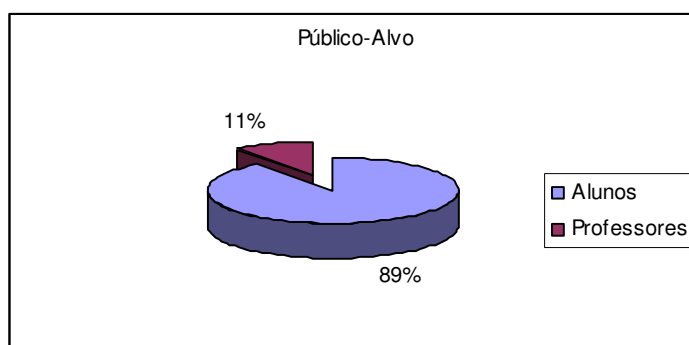


Fig31. Público-alvo do sistema online.



Para a realização da análise dos resultados, foram selecionados alguns funcionários do DRA, professores dos cursos de Computação, Engenharia civil e Nutrição e alunos do 1º, 2º e 3º ano dos referentes cursos.

Os professores inseriram suas disposições de horários e os funcionários do DRA realizaram a montagem de horários, que se tornaram disponíveis para os alunos na Internet. A inserção de dados necessária para testar o sistema foi inserida pelos funcionários.

Os professores utilizaram o sistema online para inserir as notas e frequências, e como esperado, através da Internet os alunos puderam rapidamente consultá-las.

Com o objetivo de analisar o sucesso de todas as fases necessárias para a construção da ferramenta, procurou-se trabalhar com os dois ambientes em paralelo.

As conclusões em relação aos *workflows* são descritas abaixo:

- Levantamento de requisitos - As entrevistas realizadas com os usuários do sistema proposto foram essenciais para a captura dos requisitos e criação do documento “Plano de desenvolvimento do sistema”. A ferramenta SmartWorks 2.0 Personal Edition, auxiliou na construção do cronograma e diagrama de Grantt.
- Análise de requisitos - Através da utilização da UML, todos os requisitos puderam ser especificados de maneira concisa, permitindo, desta forma, um melhor entendimento de como o sistema será desenvolvido.
- Projeto - Seguindo o conceito de UML, os diagramas de classe, implantação e componentes auxiliaram na definição dos subsistemas e estruturação do código-fonte.
- Implementação - As linguagens VB, PHP, HTML escolhidas, permitiram que o sistema fosse desenvolvido num período relativamente curto devido ao fato de haver um nível de conhecimento elevado destas. O editor de HTML Dreamweaver MX foi bastante útil na construção das páginas.

O manual do usuário possibilitou a diminuição do tempo e esforço necessários para desempenhar atividades necessárias para que o software pudesse ser testado.

## 4.2 RESULTADOS DA AVALIAÇÃO

### 4.2.1 Opinião dos usuários em relação ao módulo Internet

Para a avaliação da qualidade do software Web, foi definido um conjunto de atributos de qualidade organizados da seguinte forma.

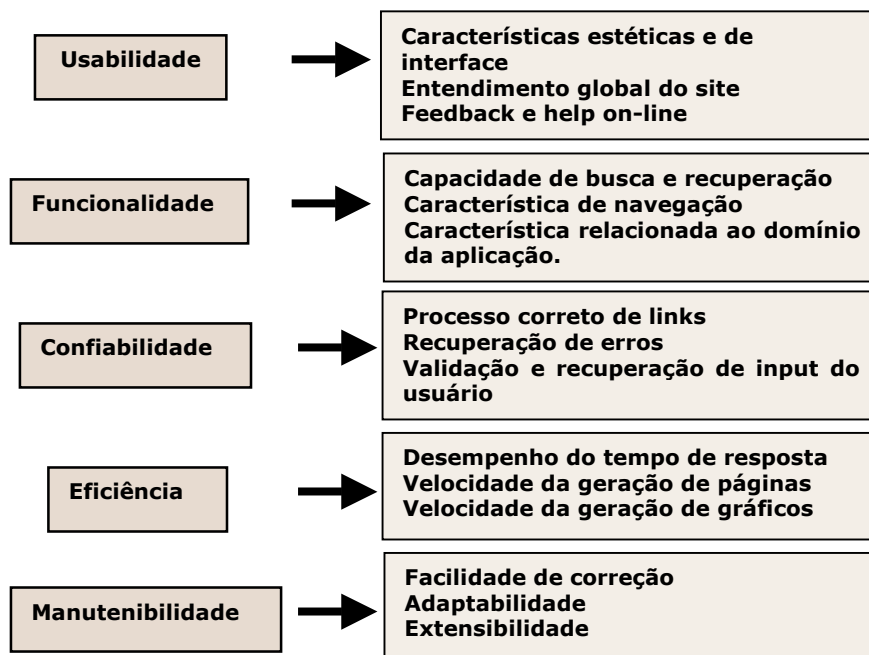


Fig32. Qualidade de aplicações Web.

O processo de avaliação do módulo Internet foi realizado através do modelo de formulário descrito abaixo:

### FORMULÁRIO PARA AVALIAÇÃO DE SITES

<b>Título do site:</b> SisRA on-line <b>Entidade:</b> UNIVERSITAS
<b>Conteúdo</b> <b>1. O título do site é significativo e está de acordo com o seu conteúdo?</b> <input type="checkbox"/> 0-Não <input type="checkbox"/> 1-Sim <b>2. As informações estão organizadas de forma lógica?</b> <input type="checkbox"/> 0-Não <input type="checkbox"/> 1-Sim <b>3. As informações estão organizadas de forma Hierárquica?</b> <input type="checkbox"/> 0-Não <input type="checkbox"/> 1-Sim <b>4. A homepage apresenta um sumário de informações que o site contém?</b> <input type="checkbox"/> 0-Não <input type="checkbox"/> 1-Sim <b>5. A homepage possui uma quantidade adequada de informações, o que a faz não ser confusa?</b> <input type="checkbox"/> 0-Não <input type="checkbox"/> 1-Sim

<p><b>5. A homepage informa a data de última atualização do site?</b>  <input type="checkbox"/> 0-Não <input type="checkbox"/> 1-Sim</p>
<p><b>Facilidade de utilização</b></p> <p><b>1. O site possui um mapa com a organização do seu conteúdo?</b>  <input type="checkbox"/> 0-Não <input type="checkbox"/> 1-Sim</p> <p><b>2. É preciso percorrer muitas páginas para chegar a informação desejada?</b>  <input type="checkbox"/> 0-Não <input type="checkbox"/> 1-Sim</p> <p><b>3. Os títulos dos botões ajudam a navegação?</b>  <input type="checkbox"/> 0-Não <input type="checkbox"/> 1-Sim</p> <p><b>4. Os botões de navegações fazem o que esperamos?</b>  <input type="checkbox"/> 0-Não <input type="checkbox"/> 1-Sim</p> <p><b>5. As informações podem ser impressas facilmente?</b>  <input type="checkbox"/> 0-Não <input type="checkbox"/> 1-Sim</p>
<p><b>6 As fontes utilizadas são legíveis?</b>  <input type="checkbox"/> 0-Não <input type="checkbox"/> 1-Sim</p> <p><b>7 A formatação do texto utilizada facilita a leitura?</b>  <input type="checkbox"/> 0-Não <input type="checkbox"/> 1-Sim</p> <p><b>8 O tempo de espera para a página ser carregada é aceitável?</b>  <input type="checkbox"/> 0-Não <input type="checkbox"/> 1-Sim</p> <p><b>8 O tempo de espera para a consulta e envio de informações é aceitável?</b>  <input type="checkbox"/> 0-Não <input type="checkbox"/> 1-Sim</p>
<p><b>Comentários</b></p>

No que se refere à organização das informações de maneira lógica no site, podemos concluir que alunos e docentes não encontraram dificuldades nos acesso às informações. O gráfico abaixo mostra as opiniões obtidas através do formulário de avaliação:

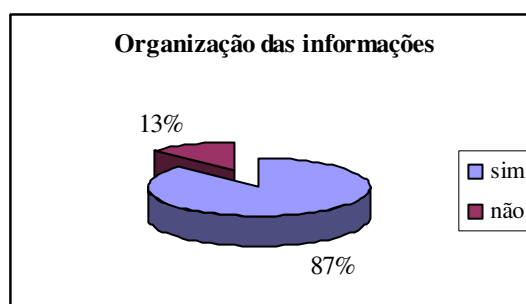


Fig33. Requisitos relacionado ao conteúdo do site.

Uma outra característica que deve ser apontada está relacionada com o tempo de espera para consultar e enviar informações ao banco de dados. Os resultados obtidos com este requisito

demonstraram que a linguagem PHP é uma das melhores opções para o desenvolvimento de páginas dinâmicas.

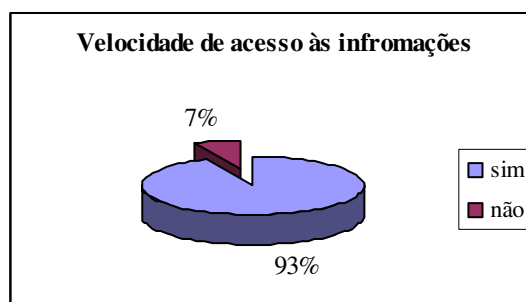


Fig34. Requisitos relacionado à velocidade de acesso as informações.

#### 4.2.2 Opinião dos usuários em relação ao módulo Intranet

A principal conclusão que pode ser obtida é que o uso da ferramenta melhorou significativamente a potencialidade da Instituição trabalhar dados e produzir informações que sirvam de apoio à tomada de decisão, com ganho de competitividade, onde o acesso mais rápido à informação armazenada em BD produz novas possibilidades de atender os alunos.

Com a utilização do questionário mostrado na tabela abaixo, foi possível obter a opinião geral dos funcionários em relação a ferramenta.

Tab28. Questionário de análise de resultados.

### RESULTADOS OBTIDOS COM A APLICAÇÃO DO QUESTIONÁRIO AOS FUNCIONÁRIOS

**Total de funcionários envolvidos na pesquisa: 15**

*Qual sua opinião em relação ao sistema considerando os requisitos abaixo.*

<b>Facilidade para utilização do software</b>	<b>Razoável</b>	<b>Bom</b>	<b>Ótimo</b>
Facilidade de entendimento	3	3	9
Facilidade de aprendizado	2	4	9
Satisfação do usuário	2	2	11
Facilidade de operar e controlar	1	1	13
<b>Funcionalidade</b>	<b>Razoável</b>	<b>Bom</b>	<b>Ótimo</b>
Adequação	1	2	12
Acurácia	1	5	9
Interação com outros sistemas	1	4	10
De acordo com normas e regulamentações	2	8	5
Segurança de acesso	5	4	6
<b>Facilidade de correções e alterações</b>	<b>Razoável</b>	<b>Bom</b>	<b>Ótimo</b>

Facilidade de diagnosticar falhas	4	2	9
Facilidade na remoção de defeitos	2	4	9
Estabilidade	4	1	10
Facilidade de testes	1	2	12
<b>Confiabilidade</b>	<b>Razoável</b>	<b>Bom</b>	<b>Ótimo</b>
Maturidade	1	5	9
Tolerância a falhas	-	3	12
Capacidade de se reestabelecer	3	4	8

Os gráficos abaixo mostram os resultados obtidos através do questionário respondido pelos funcionários. Foram levantadas questões relacionadas aos requisitos não-funcionais que são os principais fatores relacionados com a qualidade do software.

No que se refere às funções que atendem as necessidades explícitas e implícitas para a finalidade a que se destina o produto, 86% dos funcionários citaram que a facilidade para operar o sistema atendeu as expectativas.

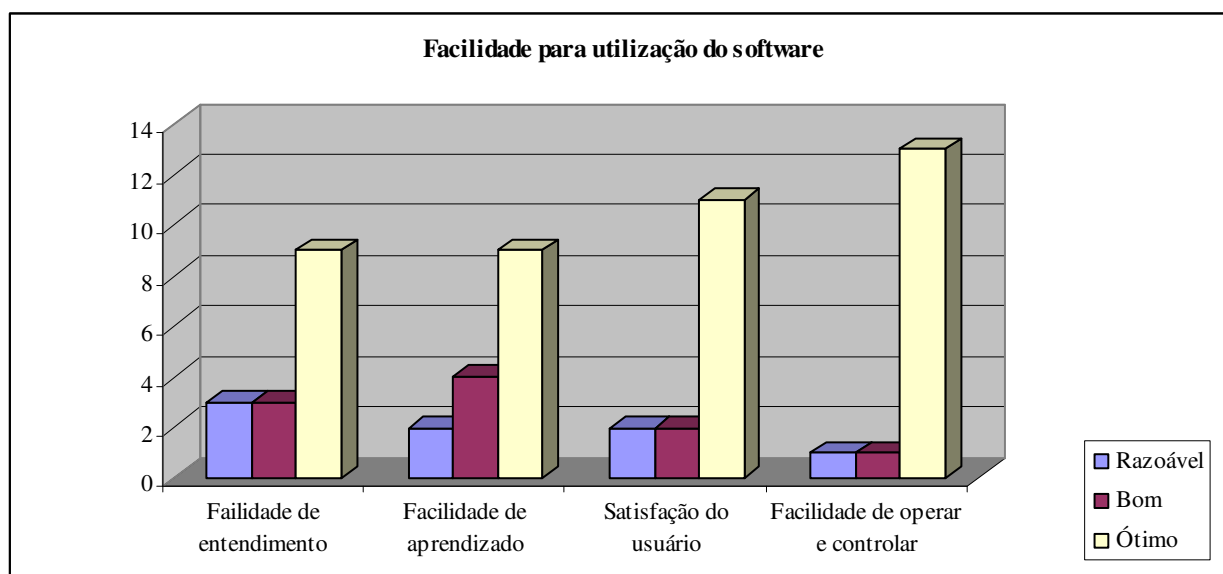


Fig35. Requisitos de usabilidade.

Em relação à funcionalidade do sistema pode-se concluir que a capacidade de interagir com o sistema de vestibular atingiu a meta desejada dos funcionários.

Cerca de 54% citaram que o sistema está de acordo com as normas e regulamentações do sistema.

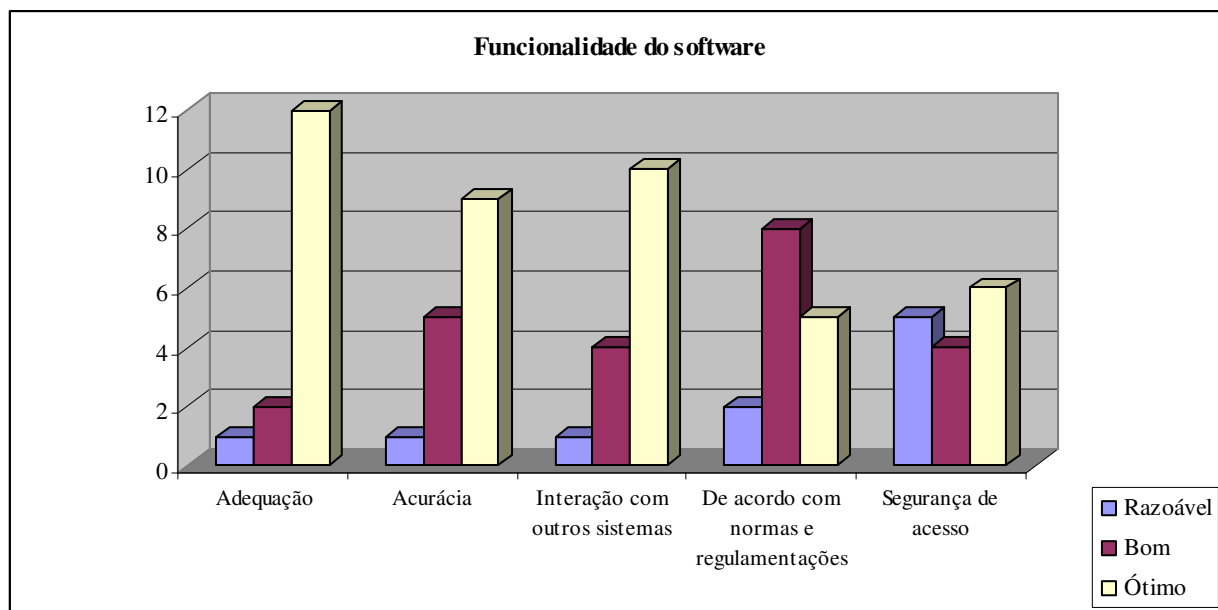


Fig36. Requisitos de funcionalidade.

Em relação à facilidade de alterações e correções do sistema pode-se concluir que a fase de teste obteve o sucesso e esperado. Com a realização dos teste foi possível notar a ausência de riscos e defeitos inesperados.

Cerca de 54% citaram que o sistema está de acordo com as normas e regulamentações do sistema.

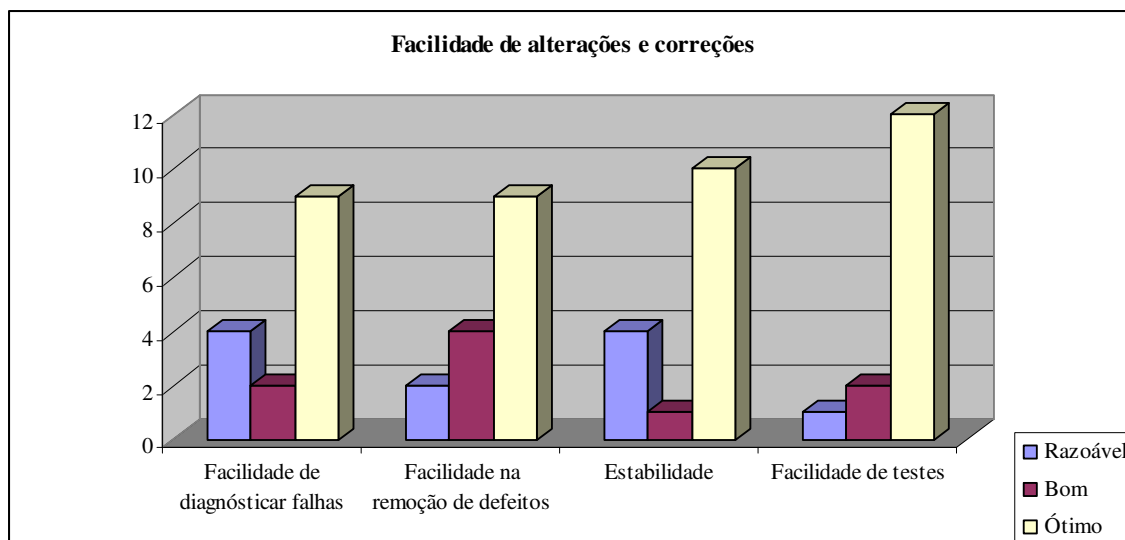


Fig37. Requisitos de manutenibilidade.

A análise dos resultados relacionado à confiabilidade evidencia que o conjunto de funções desempenhadas pelo software atendeu a necessidades dos usuários.

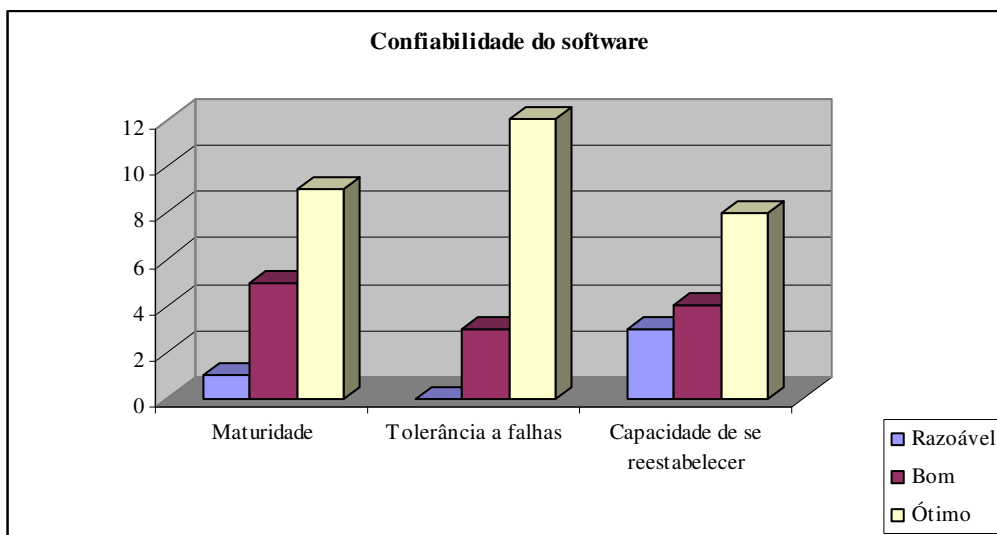


Fig38. Requisitos de confiabilidade.

#### 4.2.3 Opinião dos usuários em relação ao sistema Online

Antes de registrar os resultados obtidos com a ferramenta desenvolvida, é importante destacar que não existia um lugar onde os alunos pudessem consultar informações sobre notas, frequências e horários a não ser no quadro de aviso localizado no corredor da Instituição. Além disso, para a requisição de documentos e declarações, os alunos precisavam aguardar dias até que o pedido pudesse ser entregue.

No ambiente do aluno, disponível na página do UNIVERSITAS, foi disponibilizada uma enquete na qual o aluno pôde dar sua opinião sobre o funcionamento do sistema.

O gráfico abaixo mostra a opinião dos alunos em relação à comodidade para o acesso às diversas informações disponíveis através do sistema Online.

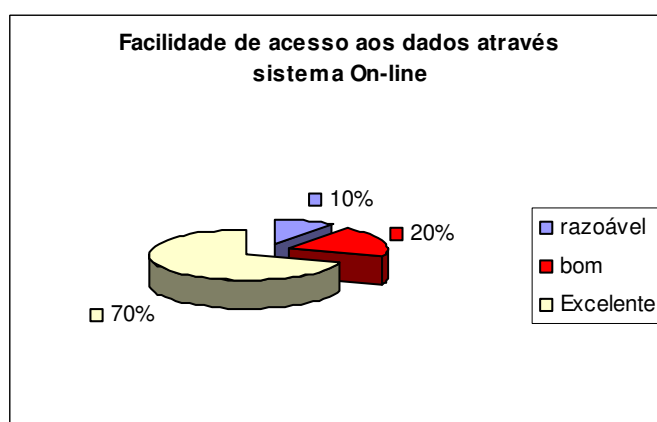


Fig39. Facilidade de acesso aos dados.

O gráfico abaixo registra que 98% dos alunos julgaram totalmente importante o uso da Internet como instrumento de controle da vida acadêmica, apenas 1% dos alunos acharam que o sistema não contribui em nada, pois não possuem acesso à Internet em suas casas, ou seja, teriam que se dirigir à Instituição para realizar a consulta e 1% são indiferentes.

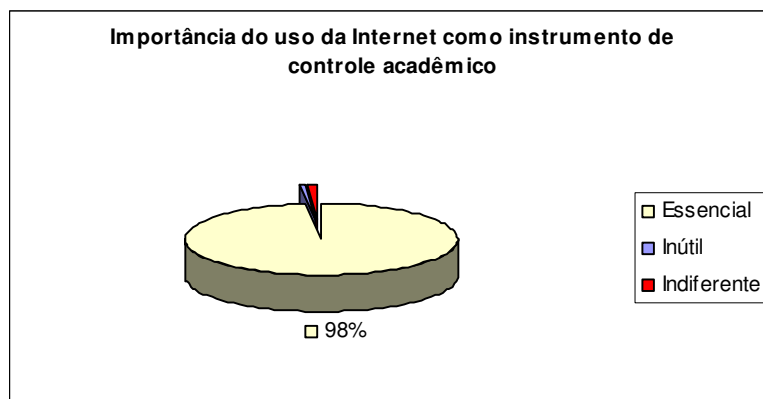


Fig40. Importância da Internet no controle acadêmico.



## *Capítulo quinto*

### **5 CONCLUSÕES**

#### **5.1 CONCLUSÃO**

O objetivo deste capítulo é apresentar as conclusões obtidas com a aplicação dos conceitos relacionados à gestão acadêmica e métodos de engenharia de software nas diversas etapas de desenvolvimento do sistema proposto.

Através dos conceitos abordados no capítulo 2 e do desenvolvimento da ferramenta de controle acadêmico, foi possível compreender a maioria dos fatores relacionados ao assunto, bem como suas dificuldades, obstáculos e perspectivas. Com a implementação da ferramenta é possível afirmar que as considerações feitas abaixo não são apenas fundamentadas na teoria ou na revisão da literatura, mas também nas opiniões dos usuários que puderam comprovar a eficiência do sistema desenvolvido.

Através do levantamento das informações relacionadas ao sistema do DRA, ficou claro a existência de diversos problemas no controle de informações dos alunos. Além disso, as tecnologias antes utilizadas encontravam-se ultrapassadas, impossibilitando a integração com os sistemas implantados em outros setores. Desta forma, o capítulo 3 apresentou a proposta para a construção de um modelo que pudesse atender as necessidades dos alunos, funcionários e dirigentes do UNIVERSITAS.

O capítulo 3 abordou ainda as etapas de desenvolvimento do sistema, comprovando as vantagens de se utilizar as técnicas de engenharia de software. Foram apresentadas as tecnologias que tornaram possível a implementação dos ambientes para o controle acadêmico. Destaca-se nesta fase a utilização dos diagramas da UML que contribuíram para a modelagem OO, eliminando a grande quantidade de métodos existentes.

Com a implantação do novo sistema, os objetivos antes definidos foram alcançados proporcionando aos alunos maior comodidade e facilidade de acesso às notas, aos funcionários uma maneira mais eficiente de administrar as informações relativas ao controle acadêmico e aos dirigentes a oportunidade de acompanhar melhor o processo de gestão acadêmica da instituição.

Com base nas informações coletadas no capítulo 4, a principal conclusão que pode ser obtida é que o uso da ferramenta melhorou significativamente a potencialidade da instituição trabalhar dados e produzir informações que sirvam de apoio à tomada de decisão, com ganho de

competitividade, onde o acesso mais rápido à informação armazenada em BD produz novas possibilidades de atender os alunos.

## 5.2 TRABALHOS FUTUROS

Como perspectiva de atividades futuras dando continuidade ao processo de análise de resultados deste trabalho, pretende-se e adicionar novas funcionalidades ao sistema.

Em termos específicos de extensão dos resultados obtidos nesta pesquisa são propostos os seguintes estudos:

- Viabilidade de implantação de um sistema financeiro integrado aos sistemas de controle acadêmico e processo seletivo.
- Viabilidade de implantação de um sistema integrado de DRH.
- Ampliação do sistema online (realização de matrícula, emissão de boletos bancários).
- Integração de uma área Financeira, por se tratar de assuntos relacionados à economia do instituto.
- Integração de uma área Setor de Contas a Receber, visto tratar-se de uma das áreas que, tipicamente, apresenta filas que certamente poderão vir a ser reduzidas.
- Integração de uma área Seleção e Capacitação de professores, facilitando a interação do DRH com o DRA no que diz respeito à adequação dos professores aos métodos exigidos nas disciplinas que lhe são confiadas.
- Integração de uma área Auxílio Monitoria, melhorando o fluxo de informações entre os setores DRA, DRH e Departamento Financeiro, facilitando a seleção e contratação dos monitores dos diversos laboratórios e disciplinas da instituição, atendendo às exigências feitas para cada seleção.
- Integração de uma área Auxílio a Biblioteca, facilitando a interação dos setores bibliotecário, DRA e departamento financeiro, afim de proporcionar mais agilidade no acesso às informações dos alunos que utilizam dos recursos disponíveis na biblioteca.
- Integração de uma área Pesquisa, facilitando o controle das informações dos alunos e professores que os orientam durante a realização dos trabalhos.
- Integração de uma área Cursos de Extensão, para controlar e melhorar a qualidade dos cursos de extensão oferecidos pela instituição.

## REFERÊNCIAS BIBLIOGRÁFICAS

[ALBERTIN, 1999] ALBERTIN, A.L. *Administração de informática: funções e fatores críticos de sucesso*. 2. ed. São Paulo, Atlas, 1999.

[BALDRIDGE, 1971] BALDRIDGE, J. V. *Power and Conflict in the University: Research in the Sociology of Complex Organizations*. New York: John Wiley & Sons, 1971.

[BIRNBAUM, 1971] BIRNBAUM R. *Organization and Governance in Higher Education*. Massachussets: Ginn Custon Publishing, 1971.

[BODINI, 2000] BODINI, V. L. *Planejamento Estratégico em Universidades*. Dablium - Adminstração em Revista, a.2, n. 4, mar. 2000.

[BOEHM, 1981] BOEHM, B. W. *Software Engineering Economics*. Prentice-Hall, Englewood Cliffs, New Jersey, 1981.

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

[BOEHM, 1988] BOEHM, B.W. *Software risk management*. IEEE Computer Society Press, Washington, 1989.

[BOOCH, 1998] BOOCH G; JACOBSON, I. and RUMBAUGH, J. *Unified Modeling Language 1.3*. White paper, Rational Software Corp., 1998.

[BOOCH, 1994] BOOCH G. *Object-Oriented analysis and Design*. 2 Edition, Benjamin Cummings, 1994.

[BOOCH, 1994] BOOCH G, et al. *UML: Guia do Usuário, O mais avançado tutorial sobre Unified Modeling Language*. Rio de Janeiro. Campus, 2000.

[BROOKS, 1995] BROOKS, F.P. *No silver bullet. In: The Mythical Man-Month. Reading.* Addison-Wesley, 2 Edition, 1995.

[BUARQUE, 1994] BUARQUE, C. *A aventura da universidade.* São Paulo, Ed. da UNESP, 1994.

[CARPES, 1986] CARPES, J. *A Short History of Functions Points and Feature Points Software Productivity Research.* Inc, Beurlington MA, Jun 1986.

[CHUN, 1995] CHUN, L., NIXON, B. A. and YU, E. *Using Non-Functional Requirements to Systematically Support Change - Proceedings of the Second IEEE International Symposium on Requirements Engineering.* IEEE Computer Society Press, 1995.

[COOD, 1991] COOD, P, YORDON, E. *Análise Baseada Em Objetos.* Rio de Janeiro. Campus. 1991.

[DASKALANTONAKIS, 1992] DASKALANTONAKIS, M. K. *A Practical View of Software Measurement and Implementation Experiences Within Motorola.* IEEE Transactions on Software Engineering, v.18, n. 11, November 1992.

[DAVIS e SITARAM, 1994] DAVIS, A.M.; SITARAM, P.A. *Concurrent process model of software development.* ACM SIGSOFT Software Engineering Notes, v. 9, n. 2, April 1994.

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

[DEITEL, 2001] DEITEL, H.M; DEITEL, P.J; *C++ Como programar.* Porto Alegre, Bookman, 2001.

[DELOBEL, 1995] DELOBEL, C. ; LÉCLUSE, C.; RICHARD, P. *Databases: From Relational to Object-Oriented Systems.* International Thomson Computer PRESS, 1995.

[FRANCO, 2000] FRANCO, E. *Marketing educacional*. Anais. Seminário gestão de Instituições de Ensino Superior: da teoria à prática. Fundação Nacional de Desenvolvimento do Ensino Superior Particular. Brasília, FUNADESP, 2000.

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

[HANNA, 1995] HANNA, M., *Farewell to waterfalls?* Software Magazine, May 1995.

[HIRAI, 1998] HIRAI, C., SAEKI, N. *A proposal of an Internet-based software development process model for cots-based systems development*. In: Proceedings of ICSE Workshop: Software Engineering Over the Internet, 1998, IEEECS.

[HUMPHREY, 1995] HUMPHREY, W.S. *A discipline for Software Engineering*. Addison, Wesley, 1995.

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

[JACOBSON, 1992] JACOBSON I.; CHRISTERSON M.; JONSSON P.; OVERGAARD G. *Object-Oriented Software Engineering – A Use Case Driven Approach*. Addison Wesley – ACM Press, 1992.

[JACOBSON, 1994] JACOBSON, I. *Use Cases and Objects. Report on Object Analysis & Design*. New York, v.1, n. 4. December 1994.

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

[JEREMAES, 1994] JEREMAES, P. *Desenvolvimento orientado a objetos: o método Fusion*. Rio de Janeiro, Campus, 1994.

[KARADIMA, 1992] KARADIMA, O. *Administracion y planificacion universitaria: El principio del liderazgo academico*. Santiago de Chile: Seminário – Planificación Estratégica Universitária, 1992.

[KOTONYA, 1998] KOTONYA, G and SOMMERVILLE I. *Requirements Engineering: Processes and Techniques*. John Wiley & Sons, 1998.

[KRUCHTEN, 2000] KRUCHTEN, P. *Rational Unified Process – An introduction*. Reading: Addison-Wesley, 2000.

[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.

[LAWRENCE, 1978] LAWRENCE, H. P. *A General Empirical Solution to the Macro Software Sizing and Estimating Problem*. IEEE Transactions Software Engineering, v.4, n.4, 1978.

[LAWSON, 1998] LAWSON, H. W. *From busyware to stableware*. IEEE Computer, v. 31, n. 10, October 1998.

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

[MACHADO e SILVEIRA, 1998] MACHADO, N.S. e SILVEIRA, A. *Configurações estruturais em organizações universitárias*. Florianópolis, Insular, 1998.

[MARCOVITCH, 1998] MARCOVITCH, J. *A universidade impossível*. São Paulo, Futura, 1998.

[MARTIN & ODELL, 1995] MARTIN, J. e ODELL, J. *Análise e Projeto Orientados a Objeto*. São Paulo, Makron Books, 1995.

[MEYER, 1991] MEYER Jr. V. *Planejamento Estratégico: Uma renovação na gestão das instituições universitárias*. Seminário - A administração universitária rumo ao ano 2000, Brasília, 1991.

[NAKAGAWA, 1994] NAKAGAWA, M. *A gestão da qualidade total na universidade: estratégia de implementação*. Revista IBRAQS, São Paulo, v.1, n. 1, Fevereiro 1994.

[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.

RATIONAL [1998] Rational Unified Process. *Best practices for software development teams*. Rational Software Corporation, 1998.

[ROBERT, 1998] ROBERT, M. *Estratégia*. São Paulo, Negócio Editora, 1998.

[ROYCE, 1998] ROYCE, W. R. *Software Project Management - A Unified Framework*. Addison-Wesley, 1998.

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

[SCHNEIDER, 1998] SCHNEIDER, G.; WINTERS, J. P.; JACOBSON I. *Applying Use Cases: A Pratical Guide*. Addison-Wesley, 1998.

[SILVA, 1998] SILVA, N. P. *Projeto e desenvolvimento de sistemas*. São Paulo, Érica, 1998.

[VON MAURHAUSER, 1990] VON, M. A. *Software Engineering: Methods and Management*. San Diego, Academic Press, 1990.

[WIRFS-BROCK , 1990] WIRFS-BROCK, R.; WILKERSON, B. and WEINER, L. *Designing Object-Oriented Software*. Prentice-Hall, 1990.

[YEH, 1984] YEH, R.T.; ZAVE, P.; CONN, A.P. and COLE Jr., G.E. *Software Requirements: New Directions and Perspectives*. Handbook of Software Engineering, Vick and Ramamoorthy, Van Nostrand Reinhold Co. – 1984.

[ZAVE,1997] ZAVE, P. *Classification of Research Efforts in Requirements Engineering*. ACM Computing Surveys,v. 29, n. 4, 1997.