

ANTONIO ELIZEU DA ROCHA NETO

**APLICAÇÃO DE TÉCNICAS DE ENGENHARIA DE SOFTWARE
PARA O DESENVOLVIMENTO DE UM SISTEMA DE CONTROLE DE
HORÁRIO**

Monografia apresentada ao Departamento de
Ciência da Computação da Universidade Federal
de Lavras como parte das exigências para
conclusão do curso de Ciência da Computação
para obtenção do título de Bacharel..

Orientador
Prof.^a Ana Cristina Rouiller

LAVRAS
MINAS GERAIS – BRASIL
2005

ANTONIO ELIZEU DA ROCHA NETO

**APLICAÇÃO DE TÉCNICAS DE ENGENHARIA DE SOFTWARE
PARA O DESENVOLVIMENTO DE UM SISTEMA DE CONTROLE DE
HORÁRIO**

Monografia apresentada ao Departamento de
Ciência da Computação da Universidade Federal
de Lavras como parte das exigências para
conclusão do curso de Ciência da Computação
para obtenção do título de Bacharel.

Aprovada em ____ de Janeiro de 2005.

Adler Diniz de Souza
(Co-orientador)

Geovane Nogueira Lima
(Co-orientador)

Prof.^a Ana Cristina Rouiller
UFLA
(Orientador)

LAVRAS
MINAS GERAIS – BRASIL
2005
2

À Ana pela orientação, a Geovane e Adler pela ajuda.

Agradecimentos

À Deus pelas oportunidades que tem me dado.

À minha mãe que com muita humildade e garra tem enfrentado os problemas da vida, sem deixar que isto atrapalhe sua alegria e sempre dando conselhos para a minha vida.

Ao meu pai(em memórias) que muito me aconselhou nos rumos que eu tinha que tomar durante minha vida, obrigado por ser meu amigo meu conselheiro e acima de tudo a pessoa que com suas ajudas me fez crescer e aprender as maldades da vida.

Aos meus irmãos que muito me apóiam e em especial ao Alexandre que teve que se tornar o homem da casa, pessoa por quem me orgulho bastante.

À Poliana que além de namorada foi amiga, companheira e bengala onde eu sempre me escorava.

À minha prima Fernanda que foi peça chave em minha caminhada para a faculdade, obrigado pela ajuda e insistência.

Ao meu padrinho pelos conselhos e pela perseverança, não me deixando parar de estudar.

À minha madrinha Lú pela ajuda e pelos bolinhos que sempre foram muito gostosos.

À tia diva pela alegria que contagiava agente.

À minha madrinha Madá pela lição de vida.

Aos meus tios e tias que acreditaram em mim e claro por comprarem minhas rifas.

Aos meus primos e primas que muito me apoiaram em minhas decisões.

Ao meu amigo Leonardo, que sempre estava de prontidão para escutar meus problemas.

Aos meus irmãos do 110 mesmo que de criação, ao Peixe(Wagner) que muito me ajudou em varias noites conversando, ao Natanas que mesmo com as chatices é gente boa, ao Whasley pelas aulas de filosofia e teologia, ao Cabeção(Carlos) por me mostrar que nem tudo no ultimo período é estudos e ao Alessandro que foi a única pessoa no apartamento que me seguiu com a queda de cabelo, salve ao peixe que já tinha começado, a vocês ficam boas lembranças que mesmo com nossas diferenças soubemos nos dar muito bem nesses quatro anos, obrigado a todos por me tolerarem.

Ao grupo dos 8 (Ana, Antonio, Giuliano, Heron, Lauro, Michelet, Rafael, Weslei) que com muito amor ao que faziam tornou o UFLATEC em um local onde podemos ao mesmo tempo aprender e ensinar, Valew moçada.

Aos companheiros e irmãos da SWQuality em especial meu chefe querido Geovane pelos puxões de orelha.

Resumo

O presente trabalho objetiva aplicar e validar algumas das técnicas proposta pela Engenharia de Software para auxiliar o desenvolvimento do sistema de controle de horário da UFLA (Universidade Federal de Lavras).

Palavras Chaves: Engenharia de Requisito, Modelagem, Teste, Validação , Controle de Horário Escolar

Abstract

The present objective work to apply and to validate some of the techniques proposal for the Engineering of Software to assist the development of the system of control of schedule of UFLA (Universidade Federal de Lavras).

Keys Word: Requirements Engineering, Modeling, Test, Validation, Control to School Schedule

Sumário

1	Introdução	6
1.1	Contextualização e Motivação	6
1.2	Objetivo e Justificativa	6
1.3	Metodologia	7
1.4	Organização do Trabalho.....	7
2	Engenharia de Software	9
2.1	Histórico da Engenharia de Software	9
2.2	A Engenharia de Software	10
2.3	Subáreas da Engenharia de Software	11
2.4	Falhas em Softwares	13
3	O Desenvolvimento do Sistema Siscoh	15
3.1	Visão geral	15
3.2	Arquitetura	16
3.3	Elicitação de Requisitos	18
3.4	Modelagem	19
3.5	A Implementação	25
3.6	Teste e Validação do Sistema Siscoh.....	30
3.6.6	Testes realizados no sistema Siscoh	34
4	O Sistema Siscoh	35
4.1	Horário	38
4.2	Turma de Aluno	43
4.3	Professor	44
4.4	Espaço Físico	48
4.5	Disciplina	51
4.6	Turma de Disciplina.....	53

4.7 Curso.....	56
4.8 Departamento.....	57
4.9 Grupo	58
5 Conclusão	60
5.1 Considerações Finais e Projetos Futuros	60
6 Referências	62

Lista de Figuras

Figura 2.1 - Relacionamento entre engenharia de processo, gerenciamento de projeto e engenharia do produto.	12
Figura 3.1 – Arquitetura de Três Camadas do Sistema Siscoh.....	17
Figura 3.2 – Plataforma Cliente/Servidor	18
Figura 3.3 – Exemplo de Diagrama de Caso de Uso do Cadastro de um Curso.	21
Figura 3.4 – Exemplo de diagrama de classe do Sistema Siscoh	22
Figura 3.5 – Exemplo de Diagrama de Seqüência do Sistema Siscoh.....	24
Figura 3.5 – Exemplo de Diagrama de Colaboração do Sistema Siscoh.....	25
Figura 4.1 – Tela Inicial do Sistema Siscoh	35
Figura 4.2 – Tela Principal do Sistema Siscoh	36
Figura 4.3 – Tela de Criação de um Novo Horário do Sistema Siscoh	39
Figura 4.4 – Tela de Realização do Tombo Escolar	40
Figura 4.5 – Tela de Acesso a um Período Letivo.....	42
Figura 4.6 – Tela de Criação do Disquete para o DRCA.....	43
Figura 4.7 – Tela de Visualização da Turma de Aluno	44
Figura 4.8 – Tela de Cadastramento de um Professor	45
Figura 4.9 – Tela de Visualização de um Professor.....	46
Figura 4.10 – Tela de Alteração de Dados de um Professor.....	47
Figura 4.11 – Tela de Restrições de Horários de um Professor.....	48
Figura 4.12 – Tela de Cadastro de um Espaço Físico.....	49
Figura 4.13 – Tela da Taxa de Utilização dos Espaços Físicos em Forma de Pizza.....	50
Figura 4.14 – Tela da Taxa de Utilização dos Espaços Físicos em Forma de Histograma.....	51
Figura 4.15 – Tela de Cadastro de uma Disciplina.....	52
Figura 4.16 – Tela de Visualização de uma Turma de Disciplina	53

Figura 4.17 – Tela de Alteração dos Dados de uma Turma de Disciplina	54
Figura 4.18 – Tela de Cadastro do Horário Padrão da Turma de Disciplina	55
Figura 4.19 – Tela de Cadastro de um Curso.....	56
Figura 4.20 – Tela de Cadastro de um Departamento	58
Figura 4.21 – Tela de Cadastro de um Grupo.....	59
Figura 5.1 – Ciclo de Desenvolvimento de um Sistema	61

Lista de Tabelas

Tabela 3.1 – Comparativa entre MySQL e PostgreSQL.....	28
Tabela 4.1 – Tabela de Inversão de Horário.....	41
Tabela 4.2 – Tabela de Inversão de Turno.....	41

1 Introdução

Segundo [Sommerville, 1998] a Engenharia de Software é uma disciplina da engenharia que se ocupa de todos os aspectos da produção de software, desde os estágios iniciais de especificação do sistema até a manutenção desse sistema, depois que ele entrou em operação.

[Pressman, 2002] define que Engenharia de Software é a criação e a utilização de sólidos princípios de engenharia a fim de obter software de maneira econômica, que seja confiável e que trabalhe eficientemente em máquinas reais.

O presente trabalho se enquadra em uma subárea da Engenharia de Software que estuda técnicas para o desenvolvimento de software: Engenharia do Produto.

Portanto aplicaremos os conceitos de algumas destas técnicas para a construção de um sistema que controle os horários dos alunos, professores e locais utilizados para ministrar atividades da UFLA (Universidade Federal de Lavras).

1.1 Contextualização e Motivação

[Pressman, 2002] afirma que a utilização dos princípios sólidos de Engenharia de Software obtém uma maneira econômica e confiável para a construção de um sistema.

Foi utilizados neste trabalho estes princípios a fim de analisar o quanto a Engenharia de Software pode ajudar ou não no desenvolvimento de um sistema.

1.2 Objetivo e Justificativa

O Objetivo deste trabalho é aplicar e avaliar as seguintes técnicas de Engenharia de Software para o desenvolvimento de um sistema: elicitação de requisitos, modelagem do sistema, implementação, teste e validação. Estas técnicas de

Engenharia de Software foram aplicadas no desenvolvimento de um sistema de controle de horário dos alunos, professores e locais para ministrar atividades da UFLA.

A PRG (Pró Reitoria de Graduação) é um setor da UFLA que é responsável pelo gerenciamento dos horários, portanto, foi o cliente.

1.3 Metodologia

Nesta seção será descrito as técnicas utilizadas para o desenvolvimento do sistema para a PRG:

1. Estudo de sistemas específicos para as necessidades da PRG;
2. Elicitação de Requisitos;
3. Definição de uma arquitetura;
4. Modelagem;
5. Implementação
 - a. Linguagem fonte e de web;
 - b. Banco de dados;
 - c. Ambiente de desenvolvimento;
 - d. Servidor de paginas.
6. Teste e Validação.

1.4 Organização do Trabalho

Além deste capítulo introdutório, o presente trabalho está organizado em mais cinco capítulos e um CD em anexo. O capítulo 2 deste trabalho serão apresentadas às técnicas da Engenharia de Software para o desenvolvimento de sistemas.

O capítulo 3 demonstrará as etapas realizadas para o desenvolvimento do sistema Siscoh.

O capítulo 4 a algumas das funcionalidades do sistema Siscoh.

O capítulo 5 descreve a conclusão.

O capítulo 6 apresenta o referencial bibliográfico.

O CD em anexo, apresenta a modelagem, o banco de dados, o documento de requisito e os códigos fontes do sistema Siscoh.

2 Engenharia de Software

O modo mais provável do mundo ser destruído, como concorda a maioria dos especialistas, é através de um acidente. É aí que nós entramos. Somos profissionais da computação. Nós causamos acidentes [Nathaniel Borenstein, 1991].

Segundo [Sommerville, 1998] a Engenharia de Software é uma disciplina da engenharia que se ocupa de todos os aspectos da produção de software, desde os estágios iniciais de especificação do sistema até a manutenção desse sistema, depois que ele entrou em operação.

[Pressman, 2002] define engenharia de software como a criação e a utilização de sólidos princípios de engenharia a fim de obter software de maneira econômica, que seja confiável e que trabalhe eficientemente em máquinas reais.

2.1 Histórico da Engenharia de Software

[Pressman, 1995] afirma que em 1969, o desenvolvimento de software está fora de controle, nesta época surge o termo Engenharia de Software, a partir de então são estabelecidos os princípios de “boas praticas de programação” e a introdução de novas linguagens de programação. Entre os anos de 1972 e 1977, surge a programação estruturada, cresce a consciência sobre o ciclo de vida como um todo, começam a medir a confiabilidade de um sistema. A partir de 1978 começam os desenvolvimentos utilizando ferramentas CASE (*Computer-Aided Software Engineering*), estas ferramentas são utilizadas para o auxílio do desenvolvimento de sistemas. A partir década de 90 surge a aplicação de Inteligência Artificial à Engenharia de Software, bem como o uso da tecnologia Orientada a Objeto e começam também a dar um foco maior no processo de desenvolvimento de software.

[Sommerville, 1998] diz que a Engenharia de Software surgiu em meados dos anos 70 numa tentativa de contornar a crise do software e dar um tratamento de engenharia (mais sistemático e controlado) ao desenvolvimento de sistemas de software complexos.

Um sistema de software complexo se caracteriza por um conjunto de componentes abstratos de software (estruturas de dados e algoritmos) encapsulados na forma de procedimentos, funções, módulos, objetos ou agentes e interconectados entre si, compondo a arquitetura do software, que deverão ser executados em sistemas computacionais [Sommerville, 1998].

[Pressman, 1995] afirma que os fundamentos científicos para a Engenharia de Software envolvem o uso de modelos abstratos e precisos que permitem ao engenheiro especificar, projetar, implementar e manter sistemas de software, avaliando e garantido suas qualidades. Além disto, a engenharia de software é um mecanismo para se planejar e gerenciar o processo de desenvolvimento.

2. 2 A Engenharia de Software

A Engenharia de Software tem por finalidade auxiliar na construção de software da melhor maneira possível [Pressman, 1995].

Desde os anos 60's, quando a frase "*the software crisis*" foi pronunciada, muitos problemas desta área foram identificados, e muitos deles ainda persistem, tais como [Gibbs, 1994]:

1. Previsão pobre – desenvolvedores não prevêm adequadamente quanto tempo e esforço serão necessários para produzir um sistema de software que satisfaça as necessidades (requisitos) dos clientes/usuários. Sistemas de software são geralmente entregues muito tempo depois do que fora planejado;

2. Programas de baixa qualidade – programas de software não executam o que o cliente deseja conseqüência talvez da pressa para se entregar o produto.

Os requisitos originais podem não ter sido completamente especificados, ou podem apresentar contradições, e isto pode ser descoberto muito tarde durante o processo de desenvolvimento;

3. Alto custo para manutenção – a manutenção pode ser corretiva, quando ocorrem enganos (erros, falhas) no sistema já entregue, ou evolutiva, quando novas características são adicionadas ao sistema de software. Ambas são caras quando o sistema original foi construído sem uma arquitetura clara e visível;

4. Duplicação de esforços – é difícil compartilhar soluções ou reusar código, em função da característica de algumas linguagens adotadas, por falta de confiança no código feito por outra pessoa e até mesmo pela ausência/deficiência de documentação das rotinas e dos procedimentos já construídos.

De forma geral, podemos dividir as funções de uma empresa desenvolvedora de softwares em três grupos principais [Garg, 1996]:

1. Definir, analisar, simular, medir e melhorar os processos da organização;
2. Construir o produto de software;
3. Medir, controlar, modificar e gerenciar os projetos de software.

Estes três grupos são abordados, respectivamente, pela *engenharia do processo*, pela *engenharia do produto* e pelo *gerenciamento de projeto* [Garg, 1996]. O relacionamento entre estes grupos é mostrado na Figura 2.1.

2.3 Subáreas da Engenharia de Software

A Engenharia do Produto é encarregada do desenvolvimento e manutenção dos produtos e serviços de software. A principal Figura da engenharia do produto é a metodologia de desenvolvimento, que engloba uma linguagem de representação,

um modelo de ciclo de vida e um conjunto de técnicas. Os ambientes tradicionais de desenvolvimento de software têm se preocupado essencialmente com a engenharia do produto, assumindo um processo implícito e tendo como foco o produto. Todavia, a engenharia do produto por si só é insuficiente para suprir as necessidades da empresa desenvolvedora de software e torná-la mais produtiva e adequada às exigências do mercado [Rouiller, 2001].

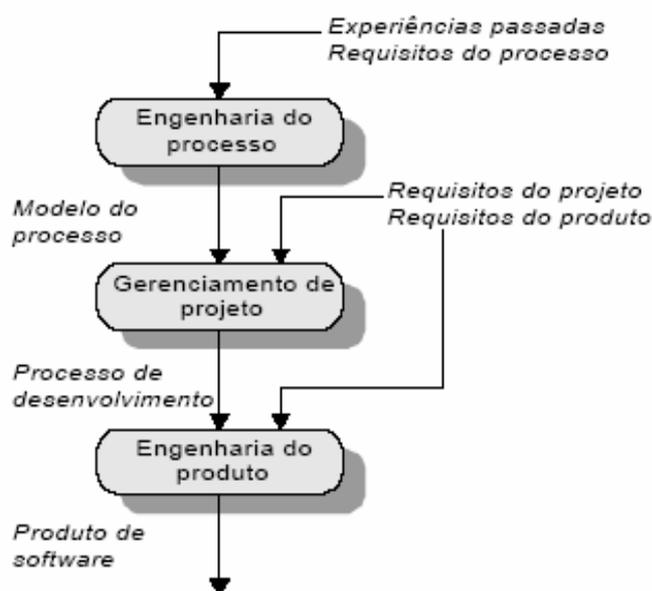


Figura 2.1 - Relacionamento entre engenharia de processo, gerenciamento de projeto e engenharia do produto.

[Rouiller, 2001] afirma que a *Engenharia de Processo* tem como metas a definição e a manutenção dos processos da Empresa Desenvolvedora de Sistemas. Ela deve ser capaz de facilitar a definição, a análise, e a simulação de um processo. Assim como estar apta a implantar, avaliar, medir e melhorar o processo definido pela Empresa Desenvolvedora de Sistemas.

A *Engenharia de Processo* pode ser vista como uma extensão da função tradicional da garantia de qualidade, e trata os processos de software de uma

forma sistemática com um ciclo de vida bem definido. Contudo, as Empresas Desenvolvedoras de Sistemas ainda sentem muita dificuldade em entender, definir e gerenciar estes processos [Rouiller, 2001].

[Rouiller, 2001] afirma que o Gerenciamento de Projeto objetiva assegurar que processos particulares sejam seguidos, coordenando e monitorando as atividades da engenharia do produto. Um processo de Gerenciamento de Projeto deve identificar estabelecer, coordenar e monitorar as atividades, as tarefas e os recursos necessários para um projeto produzir um produto e/ou serviço de acordo com seus requisitos. Todavia, gerenciar projetos de software é uma atividade complexa devido a uma série de fatores, tais como: dinamicidade do processo, grande número de variáveis envolvidas, exigência de adaptabilidade ao ambiente de desenvolvimento e constantes alterações no que foi planejado. Estes fatores dificultam o trabalho das equipes de desenvolvimento na medição do desempenho dos projetos, na verificação de pontos falhos, no registro de problemas, na realização de estimativas e planejamentos adequados.

Engenheiros de software devem adotar uma abordagem sistemática e organizada para seu trabalho e usar ferramentas e técnicas apropriadas dependendo do problema a ser solucionado, das restrições de desenvolvimento e dos recursos disponíveis [Rouiller, 2001].

A Engenharia de software não deve ser uma atividade ou tarefa única, pois deve lidar com todos os aspectos do desenvolvimento de software, desde a concepção e desenvolvimento, até a implementação, avaliação de qualidade e manutenção contínua [Rouiller, 2001].

2.4 Falhas em Softwares

Alguns acidentes podem ocorrer por softwares maus projetados e/ou maus planejados, vejamos a seguir alguns relatos:

Em [Sunnyday, 2004] consta que máquina de radioterapia com software controlador possuía um dispositivo responsável por sincronizar o hardware, quando este foi removido descobriram que o software não possuía sincronismo, causando assim diversas mortes em decorrência de queimaduras, chegou à conclusão de que o programador não possuía experiência em programação concorrente;

Segundo detalhes de [FDA, 2004] 28 pacientes Sapir expostos a radiação, 8 morreram 3 dos quais em decorrência direta do fato, espera-se que 3/4 dos 20 sobreviventes desenvolvam ‘sérias complicações que em alguns casos podem vir a ser fatais’. Especialistas encontraram equipamentos radioterápicos ‘em perfeito funcionamento’; os dados coletados pelo sistema não indicavam nenhuma situação de emergência, assim o FDA (*Food and Drug Administration*) concluiu que um dos fatores causadores da falha foi a ‘interpretação, por parte do software, dos dados coletados dos blocos dos feixes de radiação’.

Para evitar que fatos como esses ocorram, algumas atitudes foram tomadas como, por exemplo: *Praxis Cdis System 1993*, pratica uma medida chamada de *bugs/kloc* – *bugs* por cada mil linhas de código *loc* (*lines of code*) [Universia Brasil, 2004].

Estas medidas são especificadas na entrega do sistema, algumas conclusões podem ser tomadas como, por exemplo: um software dito de qualidade gira em torno de *10bugs/kloc* e um software de alta qualidade esta razão ficam em torno de 1 ou menos *bugs/kloc* [Universia Brasil, 2004].

3 O Desenvolvimento do Sistema Siscoh

Neste capítulo serão abordadas as etapas do desenvolvimento do sistema Siscoh (Sistema de Controle de Horário), uma visão mais detalhada do mesmo encontra-se no capítulo 4.

As etapas descritas neste capítulo serão:

1. Escolha da arquitetura
2. Elicitação de Requisitos
3. Modelagem
4. Implementação
5. Teste e Validação

3.1 Visão geral

O Siscoh objetiva realizar de forma geral, o controle dos horários semestrais das Turmas de Disciplinas¹ de graduação da UFLA, além dos seguintes cadastros: professores, locais onde serão ministradas as atividades, turmas de alunos, departamentos e cursos.

Informações como restrições de horário de realização de atividade de uma disciplina, restrições de utilização de espaços físicos e restrições de horário de professores também são cadastrados no sistema.

Para que se possa fazer um melhor controle da utilização desses espaços, facilitando a reserva de locais e a alocação desses para as turmas de disciplinas. No sistema Siscoh pode-se realizar o cadastramento de informações relevantes acerca dos espaços físicos como o tipo de acesso (escada, rampa, degrau, entre outros), recursos disponíveis no local (retro projetor, *data show*, microcomputador). A partir de relatórios e mensagens geradas pelo sistema, as

¹ Turma de Disciplina é um conjunto de várias Turmas de Alunos

pessoas responsáveis pelo cadastramento do horário na PRG poderão fazer as alterações deixando o consistente.

Informações gerenciais de grande importância para a gestão acadêmica pode ser geradas pelo sistema, tais como:

- Quantidade de vagas nas turmas de disciplina por período, por curso e por departamento;
- Tamanho médio das turmas de disciplinas;
- Histórico das disciplinas ligadas a um determinado departamento;
- Histórico das disciplinas ministradas por um determinado professor;
- Taxa de utilização dos espaços físicos por setor, por departamento e por curso;
- Taxa de ociosidade dos espaços físicos por setor, por departamento e por curso;
- Evolução da taxa de utilização dos espaços físicos durante os semestres;
- Taxa de ociosidade de um determinado professor por semestre;
- Variação do número de turmas de disciplinas durante os semestres;
- Recursos disponíveis aos alunos, tais como computadores, televisores, retro projetores.

Outro importante recurso do sistema é o de agendar reuniões através do controle de horário dos professores, ou seja, o sistema siscoh é capaz de agendar uma reunião levando em consideração os horários vagos que os professores têm em comum.

3.2 Arquitetura

Segundo [A.Silva Filho, 2002] a arquitetura de software é uma estrutura que serve para o entendimento de componentes de um sistema e seus inter-relacionamentos.

Existem várias formas de arquitetura, como por exemplo:

1. Duas camadas (cliente/servidor);
2. Três camadas (usuário, regras de negócio e a base de dados);
3. n Camadas (Definidas pelo projetista).

Para o sistema Siscoh foi escolhida a arquitetura de três camadas.

3.2.1 Arquitetura de Três Camadas

[Budi Kurniawan, 2003] afirma que uma arquitetura três camadas é dividida em apresentação, ou seja, onde é mostrado ao usuário, negócio ou lógica, nela encontram-se as regras que devem ser seguidas o sistema, a terceira, é a de acesso aos dados.

O sistema Siscoh foi desenvolvido utilizando este tipo de arquitetura como podemos observar na Figura 3.1, onde a camada de negócio está localizada em um servidor e a camada de dados está localizada em outro.

Como o sistema Siscoh é de utilização via internet, ele foi desenvolvido utilizando a plataforma cliente/servidor, como visto na figura 3.2.



Figura 3.1 – Arquitetura de Três Camadas do Sistema Siscoh

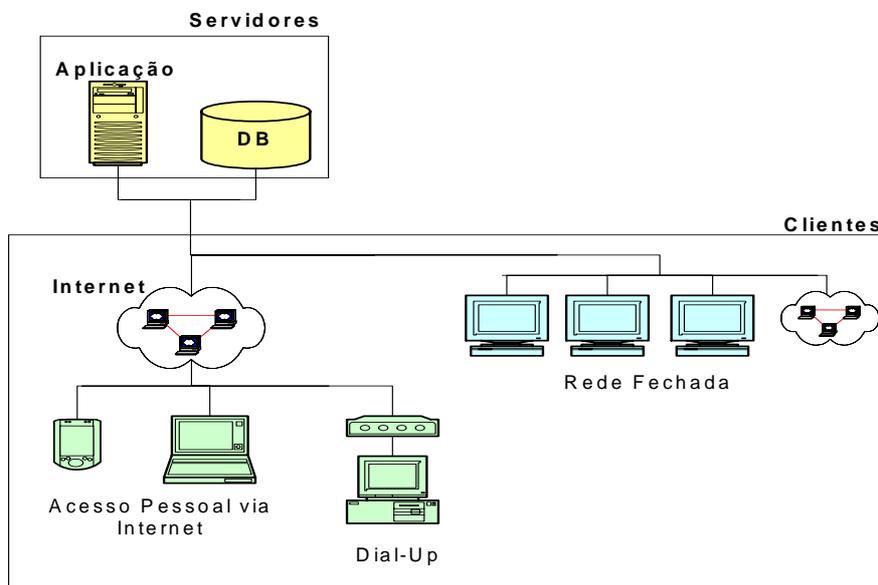


Figura 3.2 – Plataforma Cliente/Servidor

3.3 Elicitação de Requisitos

[Vasconcelos, 2004] afirma que na etapa de elicitação de requisitos, o engenheiro de software procura capturar os requisitos do sistema, buscando obter um conhecimento do domínio do problema. Usualmente, são feitas entrevistas com o cliente e/ou é consultado material existente descrevendo objetivos e desejos da organização.

Segundo [Sommerville, 1998] existem várias formas de elicitar requisitos tais como: prototipação, etnografia, análise e leituras de documentos, entre outros.

[Vasconcelos, 2004] destaca que o uso apenas de entrevista não é suficiente para obter todas as informações necessárias para o sistema em desenvolvimento e o engenheiro de software precisa se envolver com o trabalho do cliente, se misturar com os funcionários, observar, aprender e questionar.

Um dos objetivos da etapa de elicitação deve ser o entendimento da razão pela qual os procedimentos atuais do cliente são feitos da forma que são. Trata-se, portanto, de uma atividade de aprendizagem [Vasconcelos, 2004]:

- Do comportamento de sistemas existentes, incluindo:
 - Procedimentos manuais;
 - Engenharia reversa de softwares existentes;
 - Interfaces.
- Do conhecimento do domínio de aplicação, ou seja, da parte específica do domínio que está relacionada com o sistema de software a ser implementado;
- Dos objetivos e limitações dos usuários, incluindo:
 - Limitações funcionais;
 - Limitações organizacionais.

A etapa de elicitação de requisito para o sistema Siscoh teve uma duração de 3 meses e foi realizada por 3 pessoas, utilizou-se das seguintes técnicas:

- Entrevista com o cliente;
- Estudo do comportamento do sistema existente;
- Leitura de documentos.

Nesta etapa foi gerado o documento de requisito que se encontra no CD em anexo.

3.4 Modelagem

[Pressman, 1995] afirma que os resultados da etapa de elicitação são documentados em forma de modelos conceituais, que descrevem, estática e dinamicamente, aspectos do problema e do domínio de aplicação.

Contudo o modelo conceitual seleciona propriedades do domínio de aplicação que são de interesse do cliente, mas ignora detalhes e relações que podem ser importantes para outros propósitos [Pressman, 1995].

Para a modelagem e criação de um modelo conceitual, foi utilizado a UML (*Unified Modeling Language*).

A UML - é a linguagem para se modelar a estrutura da aplicação, comportamento, e arquitetura, mas também processo do negócio e estrutura de dado, segundo [UML, 2004].

A UML segue as recomendações da OMG (Open Management Group) [OMG, 2004], ela é composta por nove diagramas [UML, 2004]:

- Diagrama de caso de uso;
- Diagrama de classes;
- Diagrama de objetos;
- Diagrama de componentes;
- Diagrama de distribuição;
- Diagrama de seqüência;
- Diagrama de colaboração;
- Diagrama de estados;e
- Diagrama de atividade.

Para a modelagem dos diagramas citados acima, foi utilizado a ferramenta *Rational Rose*, esta é uma ferramenta proprietária, porém a UFLATEC possui licença. Esta ferramenta é segundo uma das mais utilizadas do mercado.

Nas próximas seções descrevermos sucintamente os seguintes diagramas que foram utilizados para a modelagem do sistema Siscoh: caso de uso, classes, seqüência e colaboração.

3.4.1 Diagrama de Caso de Uso

A modelagem de um diagrama de caso de uso é uma técnica usada para descrever e definir os requisitos funcionais de um sistema.

Os requisitos são escritos em termos de atores externos, caso de uso e o sistema modelado. Os atores representam o papel de uma entidade externa ao sistema como um usuário, um hardware, ou outro sistema que interage com o sistema modelado.

A Figura 3.3 apresenta o caso de uso do Cadastro de um Curso no sistema Siscoh. Os outros casos de uso podem ser vistos no CD em anexo.

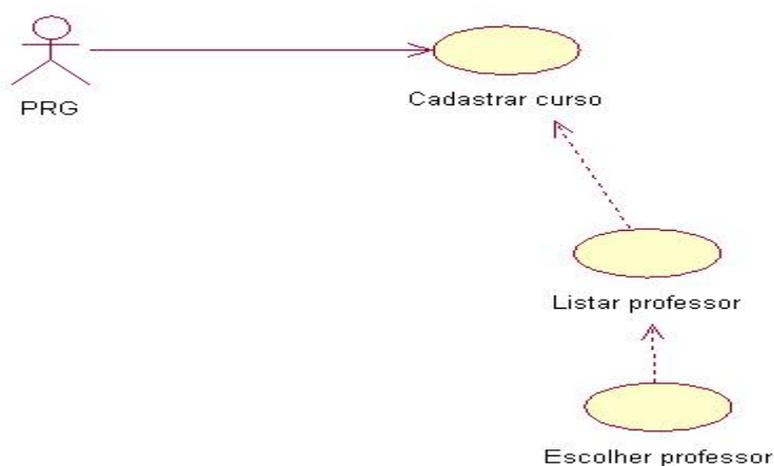


Figura 3.3 – Exemplo de Diagrama de Caso de Uso do Cadastro de um Curso.

Neste caso de uso está ocorrendo a seguinte seqüência:

- Listar os Professores;
- Escolher um professor;
- Cadastrar os dados do curso;

3.4.2 Diagrama de Classes

O diagrama de classes demonstra a estrutura das classes de um sistema onde estas representam as "coisas" que são gerenciadas pela aplicação modelada.

Classes podem se relacionar com outras através de diversas maneiras: associação (conectadas entre si), dependência (uma classe depende ou usa outra), especialização (uma classe é uma especialização de outra), ou em pacotes (classes agrupadas por características similares).

Todos estes relacionamentos são mostrados no diagrama de classes juntamente com as suas estruturas internas, que são os atributos e operações.

A Figura 3.4 apresenta a modelagem de algumas classes das quais é composta o sistema Siscoh. A modelagem completa do sistema Siscoh se encontra no CD em anexo.

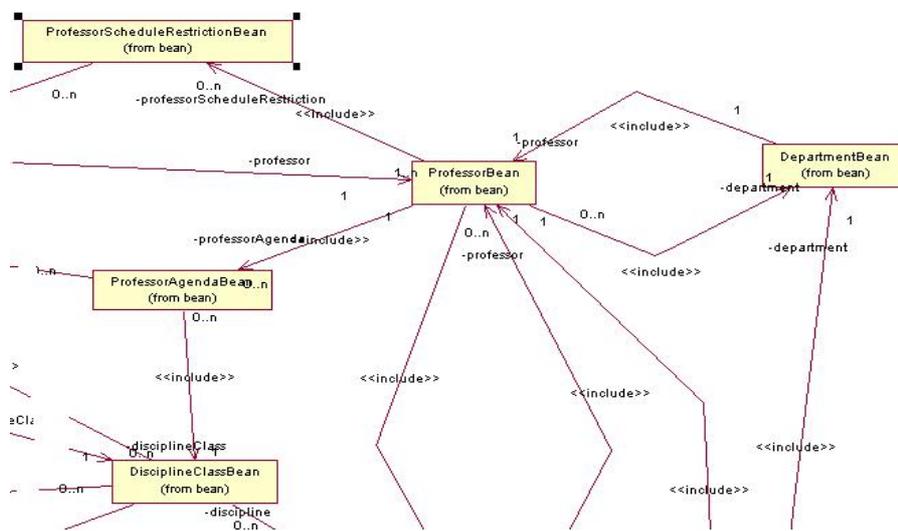


Figura 3.4 – Exemplo de diagrama de classe do Sistema Siscoh

Este diagrama de classe mostra o relacionamento entre algumas entidades do sistema Siscoh, como por exemplo:

A entidade *ProfessorBean*, ela esta relacionada com o *DepartmentBean*, *ProfessorAgendaBean* e *ProfessorScheduleBean*, podemos interpretar estes relacionamentos da seguinte forma:

ProfessorBean com *DepartmentBean*, um *ProfessorBean* pertence a um *DepartmentBean* e um *DepartmentBean* possui vários professores.

ProfessorBean com *ProfessorAgendaBean*, um professor possui ou não *ProfessorAgendaBean* e um *ProfessorAgendaBean* pertence apenas um *ProfessorBean*.

ProfessorBean com *ProfessorScheduleBean*, um professor possui ou não *ProfessorScheduleBean* e um *ProfessorScheduleBean* pertence apenas um *ProfessorBean*.

3.4.3 Diagrama de Seqüência

Um diagrama de seqüência mostra a colaboração dinâmica entre os vários objetos de um sistema. O aspecto importante deste diagrama é que a partir dele percebe-se a seqüência de mensagens enviadas entre os objetos.

Este diagrama mostra a interação entre os objetos, bem como os fluxos de execução do sistema Siscoh.

A Figura 3.5 apresenta o diagrama de seqüência de uma das funcionalidades que é composta o sistema Siscoh. Os outros diagramas do sistema Siscoh estão no CD em anexo.

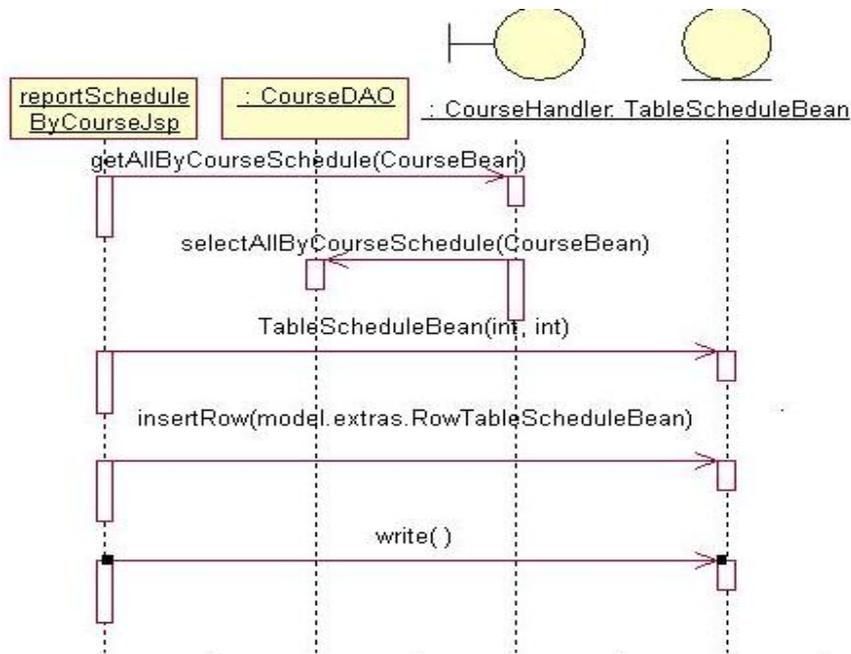


Figura 3.5 – Exemplo de Diagrama de Seqüência do Sistema Siscoh

A descrição deste diagrama seria da seguinte forma:

1. Selecione todos os cursos;
2. Selecione um determinado horário do curso escolhido;
3. Crie uma Tabela de horário para o curso escolhido;
4. Inserir as linhas necessárias para o horário do curso;
5. Escreva o horário na tela.

3.4.4 Diagrama de Colaboração

Um diagrama de colaboração mostra de maneira semelhante ao diagrama de seqüência, a colaboração dinâmica entre os objetos. Por serem semelhantes pode-se escolher entre utilizar o diagrama de colaboração ou o diagrama de seqüência.

No diagrama de colaboração, além de mostrar a troca de mensagens entre os objetos, percebem-se também os objetos com os seus relacionamentos.

A Figura 3.5 apresenta o diagrama de colaboração de uma das funcionalidades que é composta o sistema Siscoh. Os outros diagramas do sistema Siscoh estão no CD em anexo.

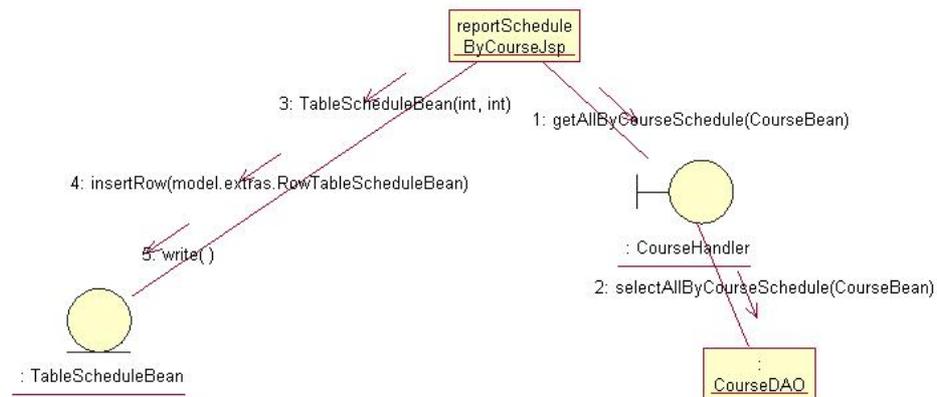


Figura 3.5 – Exemplo de Diagrama de Colaboração do Sistema Siscoh

A descrição deste diagrama seria da seguinte forma:

1. Selecione todos os cursos;
2. Selecione um determinado horário do curso escolhido;
3. Crie uma Tabela de horário para o curso escolhido;
4. Inserir as linhas necessárias para o horário do curso;
5. Escreva o horário na tela.

3.5 A Implementação

Segundo [Pressman, 1995] durante esta etapa, o projeto do software é implementado como um conjunto de unidades de uma linguagem de programação (ex: programas ou classes).

Essa etapa baseia-se totalmente no uso de ferramentas e/ou ambientes de apoio à programação (ex: compiladores, depuradores de código e editores sintáticos) [Pressman, 1995].

[Vasconcelos, 2004] afirma que, o paradigma da linguagem de programação utilizada nessa etapa está diretamente relacionado ao paradigma utilizado no projeto do sistema. Portanto ao se fazer um projeto orientado a objetos, por exemplo, é recomendado implementá-lo em uma linguagem que siga o mesmo paradigma:

Nesta seção será descrito as tecnologias utilizadas para o desenvolvimento do sistema Siscoh. Uma descrição mais detalhada do sistema Siscoh encontra-se no capítulo 4.

3.5.1 Linguagem Fonte

Segundo [Pressman, 1995] linguagem fonte é um conjunto de regras sintáticas e semânticas usadas para definir um programa de computador. Uma linguagem permite que um programador especifique precisamente sobre quais dados um computador vai atuar, como estes dados serão armazenados ou transmitidos e quais ações devem ser tomadas sob várias circunstâncias.

Como pode ser visto na figura 3.2 o sistema Siscoh segue a plataforma cliente/servidor, sendo que o servidor seria acessado através da internet e este não estava determinado qual seria o sistema operacional, por isto necessitava de uma linguagem fonte que possuísse as seguintes características:

1. Independente de plataforma, o código gerado pode ser compilado ou interpretado em qualquer sistema operacional desde que exista um compilador ou interpretador para a linguagem do código fonte;
2. Gratuita, assim não implicaria em ônus para a UFLA.

3. Possuir suporte a internet, ou seja, deveria conseguir desenvolver com esta linguagem um sistema que fosse acessado via *web browser*;

4. Possuir o paradigma de Orientação a Objeto.

Para a escolha da linguagem foram consideradas as seguintes linguagens: C++ e Java.

Optou-se por Java, pois ela conseguia atender todas as necessidades exigidas pelo sistema Siscoh, além das características, a equipe que iria desenvolver o sistema já estava familiarizada com a mesma.

3.5.2 Linguagem para Web

Segundo [Pressman, 1995] linguagem para web é um conjunto de regras sintáticas e semânticas usadas para definir um programa de computador que possa ser acessado via *web browser*.

Para a codificação das páginas web foram consideradas as seguintes linguagens: ASP (*Active Server Pages*), PHP (*Personal Home Pages*), JSP (*Java Server Pages*). Optou-se por JSP pois esta linguagem possuía as seguintes características:

1. Para a codificação das páginas em JSP é utilizado Java (Linguagem fonte escolhida para o desenvolvimento);
2. Produz conteúdos dinâmicos que possam ser reutilizados.

3.5.3 Banco de Dados

Segundo [FIRB, 2004] banco de dados é uma coleção de dados relacionados a um tópico ou propósito em particular (específico).

Como estávamos desenvolvendo um sistema utilizando tecnologias gratuitas, esta deveria ser uma característica do banco de dados a ser escolhido.

Devido ao fado do sistema Siscoh ter que lidar com um grande volume de informação, o Banco de Dados escolhido deverá possuir as seguintes características:

1. Ser gratuito;
2. Possuir integridade referencial;
3. Ter *subselects*;
4. Ser possível a definição de gatilhos (*triggers*).

Foram analisados os bancos de dados *MySql* e *PostgreSQL*. A Tabela 3.1 apresenta o comparativo entres os Bancos de Dados em questão.

Tabela 3.1 – Comparativa entre MySQL e PostgreSQL

Característica analisada	MySQL (v4. 23)	PostgreSQL (v7.2)
Licença	Gratuito	Gratuito
Integridade dos Dados	Não possui	Possui
Suporte a SubSelect	Não suporta	Suporta
Suporte a Tiggers	Não suporta	Suporta

Levando em consideração a Tabela 3.1, optou-se por escolher o *PostgreSQL*, pois este possuía as características que estávamos precisando para o sistema Siscoh.

3.5.4 Servidor de Página Web

O servidor de paginas é uma ferramenta que gerencia as páginas do sistema, bem como as requisições feitas por um usuário.

O servidor deveria possuir as seguintes características:

1. Ser gratuito;
2. Possuir suporte a linguagem Java.

Comparou os seguintes servidores para aplicação *JBoss* [JBOSS, 2004] e *Tomcat* [JAKARTA,2004]. Optou-se por *Tomcat*, pois era o único gratuito.

3.5.5 Ambiente de Desenvolvimento

Ambiente de Desenvolvimento é uma ferramenta para os programadores. Ela permite escrever, compilar e *debugar* [Netbeans, 2004].

O Ambiente de Desenvolvimento deveria possuir as seguintes características:

1. Ser gratuito;
2. Desse suporte à linguagem Java;
3. Pudessem ser utilizados em computadores que possuíssem pouca memória.

Foram analisados os seguintes Ambientes de Desenvolvimento JBuilder [Borland, 2004] e *NetBeans* [NetBeans, 2004], optou-se por *NetBeans*, pois era o único gratuito.

3.5.6 Controle de Versão

Durante o processo de desenvolvimento do sistema Siscoh, ocorreram alguns problemas como, por exemplo, não estávamos conseguindo saber quais eram os arquivos que estavam sendo utilizados na versão do sistema Siscoh que estávamos trabalhando.

Como estávamos tendo que refazer algumas funcionalidades, isto estava gerando muita perda de tempo. Optamos por usar um programa que realizasse o controle de versão de documentos.

A UFLATEC ² realizava o controle de seus documentos utilizando uma ferramenta chamada Freevcs (*Free Version Control System*, www.freevcs.org). Optou-se, então, por utilizá-la no desenvolvimento do sistema Siscoh.

3.6 Teste e Validação do Sistema Siscoh

Existe grande possibilidade de injeção de falhas no processo de desenvolvimento de software. Assim, os custos associados às falhas de software justificam uma atividade de teste cuidadosa e bem planejada, como parte dos esforços, no sentido de garantir a qualidade do software [Vasconcelos, 2004].

3.6.1 Introdução

Os testes representam a última oportunidade de detectar erros antes do software ser entregue aos usuários. A atividade de testes pode ser feita de forma manual e/ou automática e tem por objetivos [Vasconcelos, 2004]:

- Produzir casos de teste que tenham elevada probabilidade de revelar um erro ainda não descoberto, com uma quantidade mínima de tempo e esforço;
- Comparar o resultado dos testes com os resultados esperados, a fim de produzir uma indicação da qualidade e da confiabilidade do software. Quando há diferenças, inicia-se um processo de depuração para descobrir a causa.

Segundo [Vasconcelos, 2004] a realização de testes não consegue mostrar ou provar que um software ou programa está correto. O máximo que os testes de um software conseguem provar é que ele contém defeitos ou erros. Quando os testes realizados com um determinado software não encontram erros, haverá sempre duas possibilidades:

- A qualidade do software é aceitável;
- Os testes foram inadequados.

² Centro Tecnológico da UFLA, local onde foi desenvolvido o sistema Siscoh.

3.6.2 Estágios de Teste

Existem diferentes estágios de teste associados ao desenvolvimento de um produto de software [Vasconcelos, 2004]:

- Teste de unidade: visa testar individualmente cada um dos componentes (programas ou módulos), procurando garantir que funcionam adequadamente;
- Teste de integração: visa testar o relacionamento entre as diversas unidades integradas. Em outras palavras, garantir que a interface entre os módulos funcione adequadamente, pois não há garantias de que unidades testadas em separado funcionarão em conjunto;
- Teste de sistema: conjunto de testes cujo objetivo primordial é colocar completamente à prova todo o sistema, baseado em um computador. Em outras palavras, testa a integração do software com o ambiente operacional - hardware, pessoas e dados reais;
- Teste de aceitação (homologação): são testes realizados pelo cliente/usuário com o objetivo de validar o sistema a ser implantado. A motivação maior para esses testes é o fato do desenvolvedor nunca conseguir prever como o usuário realmente usará um software numa situação real.

3.6.3 Abordagens de Teste

[Vasconcelos, 2004] afirma que existem basicamente duas abordagens que podem ser aplicadas aos diferentes tipos de teste:

- Abordagem funcional (caixa-preta): concentra-se nas interfaces do software e visa mostrar que: as entradas são aceitas; as saídas são as esperadas; a integridade dos dados é mantida. Aplica-se, principalmente,

aos testes de validação, sistemas e aceitação, mas pode ser também usada com os testes unitários;

- Abordagem estrutural (caixa-branca): visa mostrar que os componentes internos do software (programas) realmente funcionam. Em princípio, os testes de caixa branca pretendem garantir que todas as estruturas dos programas e todos os possíveis casos sejam testados. Aplica-se, principalmente, aos testes unitários e de integração. Na prática, mesmo para pequenos programas, é geralmente impossível testar todas as possibilidades.

3.6.4 Tipos de Teste

Existem vários tipos de teste que podem ser executados nos diversos estágios de teste e utilizando as diferentes abordagens existentes [Vasconcelos, 2004]:

- Teste de funcionalidade: testa a funcionalidade geral do sistema, em termos de regras de negócio (fluxo de trabalho), considerando-se tanto as condições válidas como as inválidas;
- Teste de recuperação de falhas: seu objetivo é forçar o software a falhar de diversas maneiras e verificar se a recuperação é adequada;
- Teste de segurança de acesso: tenta certificar-se de que todos os mecanismos de proteção embutidos no software, de fato, o protegerão dos acessos indevidos;
- Teste de carga: tenta confrontar o software ou os programas com situações anormais. Ele executa o software de uma forma que exige recursos em quantidade, frequência e volume bem maiores do que o uso normal;
- Teste de desempenho: são testes que visam verificar o desempenho ou performance do software. São, muitas vezes,

combinados ou feitos juntamente com os testes de estresse. São comuns em software de tempo real;

- Teste de portabilidade: são testes que verificam o grau de portabilidade do produto de software em diferentes ambientes de hardware/software.

A execução de todos esses tipos de teste pode ser feita de forma combinada. Ou seja, nenhum desses testes exclui o outro.

3.6.5 Responsabilidades pelos testes

Segundo [Vasconcelos, 2004] os testes normalmente, são feitos pela equipe de desenvolvimento, i.e., por engenheiros de software. No entanto, é ideal que pelo menos os testes de sistemas sejam feitos por uma equipe de testes independente. É importante que os grandes sistemas e programas também sejam testados por outras pessoas que não os seus desenvolvedores, e que tais pessoas sejam bastante críticas. Assim, poderíamos ter a seguinte distribuição de responsabilidades:

Equipe de desenvolvimento:

- Testes de unidades;
- Testes de integração.

Equipe independente de testes:

- Testes de sistema.

Usuário:

- Teste de aceitação.

3.6.6 Testes realizados no sistema Siscoh

Mesmo sabendo da importância de se implementar os testes em um sistema, não conseguimos aplicar todos, pois a equipe era pequena para alocar pessoas apenas para fazer os testes.

Os testes implantados foram:

1. Teste de unidade;
2. Teste de integração.

4 O Sistema Siscoh

O sistema Siscoh permite fazer o controle dos horários semestrais das turmas de disciplinas de graduação da UFLA (Universidade Federal de Lavras). O sistema faz também o cadastramento dos professores, dos locais onde serão ministradas as aulas, das turmas de alunos, dos departamentos e dos cursos.

A partir de relatórios, e mensagens geradas pelo sistema, os responsáveis pelo cadastro do horário na PRG, poderão fazer alterações para que não haja erros ou inconsistências no horário, bem como decidir sobre a melhor forma de utilizar os espaços físicos da UFLA.

Neste capítulo serão descritos algumas das funcionalidades do sistema Siscoh, a título de ilustração do trabalho. O sistema Siscoh completo encontra-se no CD em anexo.



Figura 4.1 – Tela Inicial do Sistema Siscoh

A Figura 4.1 mostra a tela inicial do sistema, nela o usuário deve entrar com um usuário e uma senha.

O sistema Siscoh possuirá vários perfis e cada usuário tem um perfil no qual está associado a uma ou mais funcionalidades dentro do sistema.

A Figura 4.2 Apresenta a tela que será mostrada ao usuário após a entrada no sistema Siscoh.



Figura 4.2 – Tela Principal do Sistema Siscoh

A Figura 4.2 contém uma breve descrição sobre cada uma das funcionalidades que podem ser acessadas a partir do menu lateral. As descrições são as seguintes:

*Horário*³: o usuário poderá acessar funcionalidades como criar um novo horário, realizar tombo⁴, arquivar horário e acessar versões anteriores de horário.

³ Horário, é utilizado para expressar o período letivo do alunos.

*Turma de Aluno*⁵: o usuário poderá manipular as diversas turmas de alunos dos cursos, encontrando funcionalidades como cadastrar, alterar, visualizar disciplinas e relatórios de horários.

Professor: é possível alocar professores para ministrar aulas para determinada turma de disciplina, cadastrar professor, cadastrar restrições de horário entre outras funcionalidades.

Espaço Físico: o usuário poderá cadastrar, alterar ou remover recursos, como computadores ou retro projetores. Pode-se ainda alocar o espaço físico para determinada disciplinas, além cadastrar o tipo de acesso ao espaço físico.

Disciplina: o usuário poderá cadastrar, alterar e remover determinada disciplina, além de obter diversos relatórios dessas disciplinas.

Turma de Disciplina: o usuário poderá cadastrar diversas turmas de disciplinas para uma determinada disciplina permitindo assim a identificação dos professores para cada turma.

Curso: o usuário poderá visualizar os professores pertencentes ao curso, poderá também cadastrar diversos currículos para um só curso, além de relatórios de horário e de disciplinas.

Departamento: o usuário terá acesso às diversas funcionalidades, entre elas relatórios de professores, turma de disciplinas, horário e ainda o histórico das disciplinas durante os semestres.

Grupo: pode-se agendar reuniões com grupos de pessoas cadastradas, obtendo um melhor horário entre as pessoas do grupo.

⁴ Tombo é uma forma de expressar que um horário que está, por exemplo, na segunda feira na parte da manha, vai para a quarta feira na parte da tarde, as tabela 4.1 e 4.2 expressão esta mudança de turno e dia.

⁵ Turma de Alunos são os alunos que entraram no mesmo período, ano e curso.

As seções a seguir mostra de uma forma mais detalhada algumas das funcionalidades do sistema Siscoh, o sistema completo encontra-se no CD em anexo.

4.1 Horário

Algumas das funcionalidades implementadas no sistema Siscoh para o Horário serão descritas a seguir:

4.1.1 Criar um Novo Horário

A Figura 4.3 mostra a tela para criação de um novo horário.



Figura 4.3 – Tela de Criação de um Novo Horário do Sistema Siscoh

Para a criação de um novo horário o usuário deverá entrar com uma data inicial, uma final e enviar os dados.

O novo horário criado não terá nenhuma informação.

4.1.2 Realizar Tombo

A Figura 4.4 apresenta a funcionalidade referente à realização do tombo.

Os períodos na UFLA são invertidos, ou seja, se em um semestre a disciplina é ofertada aos alunos na parte da manhã, no período seguinte são ofertadas na parte da tarde e dois dias depois como pode ser visto na tabela 4.1 e 4.2. O sistema então permite ao usuário fazer estas inversões.



Figura 4.4 – Tela de Realização do Tombo Escolar

Para a realização do tombo o usuário deverá entrar com a data inicial, final do novo horário, decidir se é com ou sem inversão de turno e enviar os dados.

O sistema Siscoh irá copiar todas as informações dos horários das turmas de alunos, turmas de disciplinas, salas de aulas e professores da UFLA, depois de copiados ele irá analisar se o usuário entrou com inversão de turno ou não.

O sistema fará uma alteração dos horários segundo a Tabela 4.1.

Tabela 4.1 – Tabela de Inversão de Horário.

Dia da semana que esta sendo a atividade	Dia da semana que será a atividade no próximo semestre.
segunda-feira	quarta-feira
terça-feira	quinta-feira
quarta-feira	sexta-feira
quinta-feira	segunda-feira
sexta-feira	terça-feira
Sábado	Sábado

Tabela 4.2 – Tabela de Inversão de Turno.

Horário de aula atual	Horário de aula do próximo semestre
7:00	13:00
8:00	14:00
9:00	15:00
10:00	16:00
11:00	17:00

Após fazer os cálculos para saber qual será o dia de cada atividade, o sistema faz uma verificação para saber se a pessoa escolheu com ou sem mudança de turno.

Caso a opção for com mudança de turno, o sistema Siscoh irá alterar os horários conforme a Tabela 4.2, mas se a opção for sem mudança de turno não acontecerá esta mudança.

Acabando o processo de alteração dos horários o sistema Siscoh irá gravar os dados do novo horário no banco de dados.

Acessar Período Letivo

A Figura 4.5 mostra a tela para acessar um período letivo.



Figura 4.5 – Tela de Acesso a um Período Letivo

Ao escolher uma versão do horário, o sistema Siscoh irá mostrar todas as informações desta versão. Exemplo de uma versão de horário é 2005-1, primeiro semestre do ano de 2005.

4.1.4 Gerar Disquete para o DRCA

A Figura 4.6 mostra a tela para gerar um disquete para a DRCA.



Figura 4.6 – Tela de Criação do Disquete para o DRCA.

Disquete para o DRCA, é um disquete com as configurações necessárias para gerar os horários dos alunos da UFLA, neste disquete contém informações de locais, horas e aulas que serão dadas para o período que esta sendo gerado o disquete.

4.2 Turma de Aluno

Turma de Aluno são os alunos que se matricularam no mesmo ano, semestre e curso.

Algumas das funcionalidades implementadas no sistema Siscoh para a Turma de Aluno serão descritas a seguir:

4.2.1 Visualizar Turma de Aluno

A Figura 4.7 mostra a tela de visualização da turma de aluno.



Figura 4.7 – Tela de Visualização da Turma de Aluno

Na Figura 4.7 o usuário poderá ter acesso a algumas funcionalidades tais como: *alterar dados* o usuário poderá alterar dados, *relatório de horário* poderá visualizar o horário e em *disciplina* o usuário poderá visualizar as disciplinas, estas funcionalidades serão mostradas para a turma de aluno visualizada.

4.3 Professor

Algumas das funcionalidades implementadas no sistema Siscoh para o Professor serão descritas a seguir:

4.3.1 Cadastrar Professor

A Figura 4.8 mostra a tela de cadastramento de um professor.

SISTEMA DE CONTROLE DE HORÁRIOS

SISCOH 2005-1

Principal Voltar Sair

Horário Turma de Aluno Professor Espaço Físico Disciplina Turma de Disciplina Curso Departamento Grupo

Principal >> Professor >> Cadastrar Professor

Cadastrar Professor

Nome

SIAPE

Departamento do professor DEPT: ADMINISTRAÇÃO ECONOMIA

Situação Ativo

Escolha a(s) Disciplina(s) que o professor pode ministrar:

Departamento

Figura 4.8 – Tela de Cadastramento de um Professor

Para cadastrar um professor deverá informar os seguintes dados:

- Nome;
- Número do SIAPE⁶;
- Departamento que o professor é vinculado;
- A situação, sendo esta ativo ou inativo;
- As disciplinas que este professor pode ministrar.

Após os dados informados o usuário deverá confirmar o cadastro.

4.3.2 Visualizar Professor

A Figura 4.9 mostra a tela de visualização de um professor.

⁶ SIAPE é o número que identifica um professor no Governo.



Figura 4.9 – Tela de Visualização de um Professor

A partir da Figura 4.9 o usuário poderá acessar algumas funcionalidades do sistema Siscoh tais como:

4.3.3 Alterar Dados

A Figura 4.10 mostra a tela de alteração de dados de um professor.



Figura 4.10 – Tela de Alteração de Dados de um Professor

O usuário poderá alterar os seguintes dados do professor:

- Nome;
- Departamento;
- Situação;
- Disciplinas.

Após entrar com os dados novos, o usuário deverá confirmar a alteração.

4.3.4 Restrições de Horários

A Figura 4.11 mostra a tela de restrições de horários de um professor.



Figura 4.11 – Tela de Restrições de Horários de um Professor

Na Figura 4.11 o usuário poderá adicionar quais os horários que o professor não poderá ministrar as aulas ou outras atividades.

Após adicionar os horários restritos o usuário deverá confirmar o envio dos dados.

4.4 Espaço Físico

Algumas das funcionalidades implementadas no sistema Siscoh para o Espaço Físico serão descritas a seguir:

4.4.1 Cadastrar Espaço Físico

A Figura 4.12 mostra a tela de cadastramento do espaço físico.

The screenshot shows the SISCOH system interface. At the top left is the SISCOH logo and the year 2005-1. The main header is 'SISTEMA DE CONTROLE DE HORÁRIOS'. Below this is a navigation bar with 'Principal' and 'Voltar' buttons. A secondary navigation bar contains links for 'Horário', 'Turma de Aluno', 'Professor', 'Espaço Físico', 'Disciplina', 'Turma de Disciplina', 'Curso', 'Departamento', and 'Grupo'. The current page is 'Cadastrar Espaço Físico', indicated by the breadcrumb 'Principal >> Espaço Físico >> Cadastrar Espaço Físico'. The form itself is titled 'Cadastrar Espaço Físico' and includes the following fields: 'Código do Novo Espaço Físico' (text input), 'Descrição' (text input), 'Capacidade' (text input), 'Recursos' (text input), and 'Tipos de AcessoDDD'. Under 'Tipos de AcessoDDD', there are two dropdown menus: 'Tipo de Espaço Físico' (with 'Escolha o tipo' selected) and 'Localização' (with 'Escolha a localização' selected). A 'Cadastrar' button is at the bottom of the form. The footer of the page says 'siscoh - insertPlace'. The browser's taskbar at the bottom shows 'Concluído' and 'Internet'.

Figura 4.12 – Tela de Cadastro de um Espaço Físico

Na Figura 4.12 o usuário deverá adicionar os seguintes dados para o cadastramento de um espaço físico:

- Código;
- Descrição;
- Capacidade;
- Recursos disponíveis no local;
- Localização.

Após entrar com os dados acima o usuário deverá confirmar o cadastro.

4.4.2 Relatórios

As Figuras 4.13 e 4.14 mostram a taxa de utilização dos espaços físicos.



Figura 4.13 – Tela da Taxa de Utilização dos Espaços Físicos em Forma de Pizza.

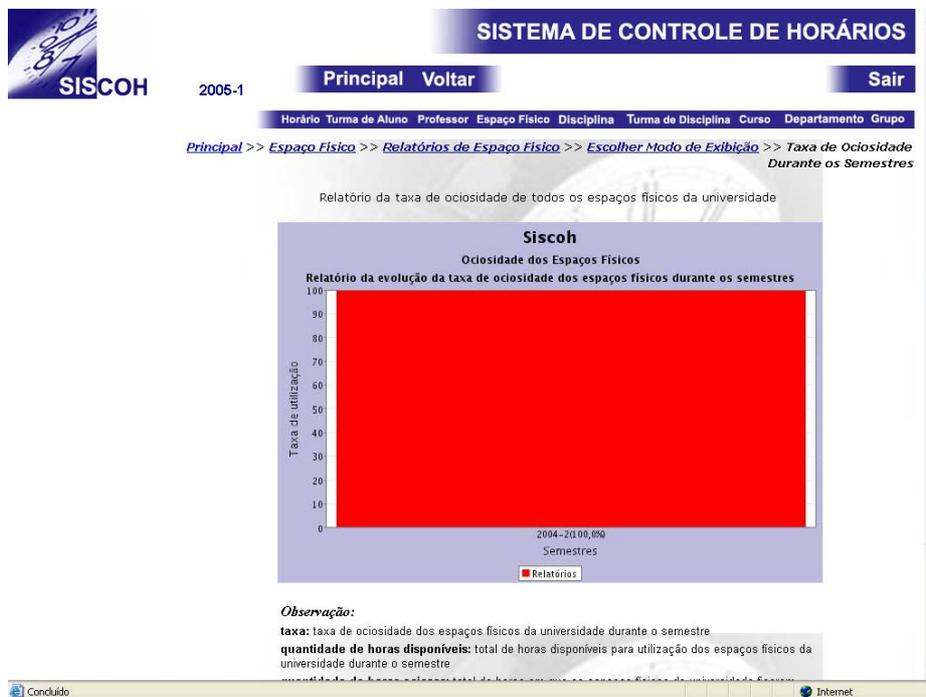


Figura 4.14 – Tela da Taxa de Utilização dos Espaços Físicos em Forma de Histograma.

4.5 Disciplina

O sistema Siscoh prevê algumas funcionalidades para trabalhar com as disciplinas, estas serão descritas a seguir:

4.5.1 Cadastrar Disciplina

A Figura 4.15 mostra a tela para o cadastramento de uma disciplina.

The screenshot shows the SISCOH system interface. At the top left is the SISCOH logo and the year 2005-1. The main title is 'SISTEMA DE CONTROLE DE HORÁRIOS'. Below the title are navigation buttons: 'Principal', 'Voltar', and 'Sair'. A horizontal menu contains: 'Horário', 'Turma de Aluno', 'Professor', 'Espaço Físico', 'Disciplina', 'Turma de Disciplina', 'Curso', 'Departamento', and 'Grupo'. The current page is 'Cadastrar Disciplina', indicated by the breadcrumb 'Principal >> Disciplina >> Cadastrar Disciplina'. The form fields are: 'Código' (text input), 'Nome' (text input), 'Abreviatura' (text input), 'Departamento' (dropdown menu with 'escolha um departamento' selected), 'Carga Horário Teórica' (text input), 'Carga Horário Prática' (text input), and 'Créditos' (text input). A 'Cadastrar' button is at the bottom of the form. The background of the form area features a faint image of a clock face. The footer of the page contains 'siscoh - insertDiscipline' and a taskbar with 'Concluído' and 'Internet' icons.

Figura 4.15 – Tela de Cadastro de uma Disciplina

Na Figura 4.15 o usuário deverá entrar com os seguintes dados:

- Código;
- Nome;
- Abreviatura;
- Departamento;
- Carga horária teórica;
- Carga horária prática;
- Créditos.

Após entrar com os dados o usuário deverá confirmar o cadastro.

4.6 Turma de Disciplina

Turma de Disciplina é o agrupamento de varias Turmas de Alunos que fazem uma determinada disciplina.

Algumas das funcionalidades implementadas no sistema Siscoh para a Turma de Disciplina serão descritas a seguir:

4.6.1 Visualizar Turma de Disciplina

A Figura 4.16 mostra a tela de visualização de uma Turma de Disciplina.

The screenshot shows the SISCOH system interface. At the top left is the SISCOH logo and the year 2005-1. The main header is 'SISTEMA DE CONTROLE DE HORÁRIOS'. Below this is a navigation bar with 'Principal' and 'Voltar' buttons, and a 'Sair' button on the right. A secondary navigation bar contains links: 'Horário', 'Turma de Aluno', 'Professor', 'Espaço Físico', 'Disciplina', 'Turma de Disciplina', 'Curso', 'Departamento', and 'Grupo'. The current page is 'Cadastrar Turma de Disciplina', indicated by the breadcrumb 'Principal >> Turma de Disciplina >> Cadastrar Turma de Disciplina'. The form contains the following fields: 'Disciplina' (text input), 'Tipo da Disciplina' (dropdown menu with 'Teórica' selected), 'Professor' (text input), 'Número da Turma de Disciplina' (text input), 'Número de Vagas da Turma' (text input), 'Escolha o curso para incluir a(s) turma(s) de alunos' (text label), 'Curso da Turma de Alunos' (dropdown menu with 'escolha um curso' selected), and a 'Limpar' button. The background of the form area features a faint clock face. At the bottom right of the form area, the text 'siscoh - insertDisciplineClass' is visible. The browser window title bar at the bottom shows 'Concluído' and 'Internet'.

Figura 4.16 – Tela de Visualização de uma Turma de Disciplina

A Figura 4.16 mostra a composição de uma turma de disciplina, ou seja quais são as turmas de alunos e qual é a disciplina da turma de disciplina visualizada.

4.6.2 Alterar Dados da Turma de Disciplina

A Figura 4.17 mostra a tela de alteração de dados da turma de disciplina.

The screenshot shows the SISCOH system interface for editing discipline class data. The interface includes a navigation menu, a form for entering discipline details, and a table of selected classes.

SISTEMA DE CONTROLE DE HORÁRIOS

SISCOH 2005-1 **Principal** **Voltar** **Sair**

Horário Turma de Aluno Professor Espaço Físico Disciplina Turma de Disciplina Curso Departamento Grupo

[Principal](#) >> [Turma de Disciplina](#) >> [Visualizar Turma de Disciplina](#) >> [Alterar Turma de Disciplina](#)

Alterar Turma de Disciplina

Disciplina: FIT127 - INFORMÁTICA NA AGRICULTURA

Tipo da Disciplina: Teórica

Professor: ADAO

Número da Turma: 1

Número de Vagas: 40

Curso da Turma de Alunos: escolha um curso

Turmas Seleccionadas

Cód. Curso	Cód. Turma de Alunos	Ano	Semestre	Período	Nº Vagas	
G001	01A	2005	1	1	20	Remover
G001	01B	2005	1	1	20	Remover

siscoh - alterDisciplineClass

Figura 4.17 – Tela de Alteração dos Dados de uma Turma de Disciplina

Na Figura 4.17 o usuário poderá alterar os seguintes dados:

- Tipo da disciplina (Prática ou Teórica);
- Qual o professor que ministra a turma;
- Número da turma;
- Quantidade de vagas;
- Turmas de alunos que fazem à disciplina.

Após entrar com os dados o usuário deverá confirmar as alterações.

4.6.3 Horário Padrão

A Figura 4.18 mostra a tela de cadastramento do horário padrão ⁷da Turma de Disciplina.

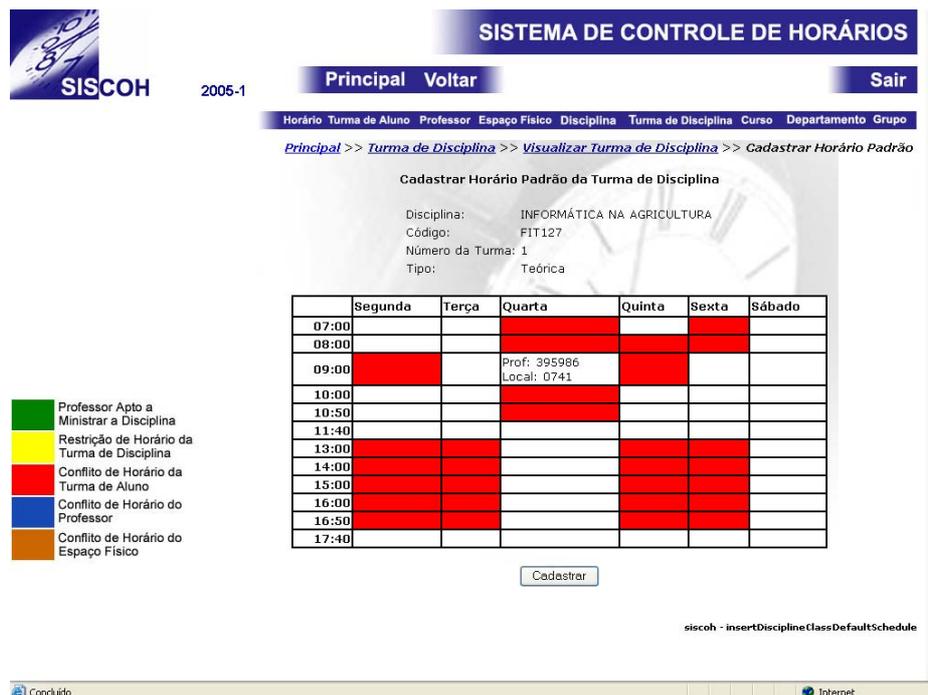


Figura 4.18 – Tela de Cadastro do Horário Padrão da Turma de Disciplina

O usuário deverá entrar com o dia e o horário e qual o professor que ministrará a atividade.

Após entrar com os dados o usuário deverá confirmar o cadastramento do horário padrão, este é o horário de todos os dias do semestre letivo e todos os dias .

⁷ Horário padrão é o horário que será cadastrado para uma turma de disciplina em um determinado período.

O mesmo ocorre em cadastrar horário, neste cadastro além dos dados do cadastro de horário padrão informa-se qual o período que se deseja cadastrar o horário.

4.7 Curso

Algumas das funcionalidades implementadas no sistema Siscoh para o Curso serão descritas a seguir:

4.7.1 Cadastrar Curso

A Figura 4.19 mostra a tela de cadastramento do curso.

SISTEMA DE CONTROLE DE HORÁRIOS

SISCOH 2005-1 Principal Voltar Sair

Horário Turma de Aluno Professor Espaço Físico Disciplina Turma de Disciplina Curso Departamento Grupo

Principal >> Curso >> Cadastrar Curso

Cadastrar Curso

Nome do Curso

Código do Curso

Coordenador do Curso

Quantidade de Períodos

Cadastrar

siscoh - insertCourse

Concluído Internet

Figura 4.19 – Tela de Cadastro de um Curso

Para o cadastramento de um curso o usuário deverá entrar com os seguintes dados:

- Nome;
- Código;
- Coordenador;
- Quantidade de períodos.

Após entrar com os dados o usuário deverá confirmar o cadastramento do curso.

4.8 Departamento

Algumas das funcionalidades implementadas no sistema Siscoh para o Departamento serão descritas a seguir:

4.8.1 Cadastrar Departamento

A Figura 4.20 mostra a tela de cadastramento do departamento.



Figura 4.20 – Tela de Cadastro de um Departamento

Para o cadastramento de um departamento o usuário deverá entrar com os seguintes dados:

- Nome;
- Código;
- Chefe.

Após entrar com os dados o usuário deverá confirmar o cadastramento do departamento.

4.9 Grupo

Um grupo é uma maneira de agrupar professores, para poder marcar uma reunião ou uma outra atividade.

Algumas das funcionalidades implementadas no sistema Siscoh para o Grupo serão descritas a seguir:

4.9.1 Cadastrar Grupo

A Figura 4.21 mostra a tela de cadastramento de um grupo.

SISTEMA DE CONTROLE DE HORÁRIOS

SISCOH 2005-1 Principal Voltar Sair

Horário Turma de Aluno Professor Espaço Físico Disciplina Turma de Disciplina Curso Departamento Grupo

Principal >> Grupo >> Cadastrar Grupo

Cadastrar Grupo

Nome do Grupo

Marque os professores que farão parte do grupo

- ADAO
- ADMILSON BOSCO CHITARRA
- ADRIANA DE SOUZA COUTINHO
- ADRIANA MELLO GARCIA RABE
- AGOSTINHO ROBERTO DE ABRE
- ALBERTO COLOMBO
- ALCIDES MOINO JUNIOR
- ALCIONE DE OLIVEIRA
- ALOISIO RICARDO PEREIRA D
- ALVARO JOAO LACERDA DE AL
- ALVARO RODRIGUES P. JUNIOR
- AMAURI ALVES DE ALVARENGA
- AMERICO IORIO CIOCIOLA
- ANA CRISTINA ROUILLER
- ANA PAULA DUARTE DE OLIVE
- ANA TEREZA DE MEDONCA
- ANDERSON BERNARDO SANTOS
- ANDREIA SCHNEIDER

Concluído Internet

Figura 4.21 – Tela de Cadastro de um Grupo

O usuário deverá entrar com os seguintes dados para o cadastramento de um grupo:

- Nome;
- Professores.

Após isto deverá confirmar o cadastramento.

5 Conclusão

Durante o desenvolvimento do sistema Siscoh foram aplicadas às técnicas de Engenharia de Software descritas no capítulo 2.

Após 12 meses de trabalho observou-se que as técnicas contribuíram durante todo o processo de desenvolvimento. Notou-se que os seguintes fatores são primordiais para tais contribuições, conforme a seguir:

1. Possuir ferramentas CASE (as ferramentas adotadas apóiam as utilizações das técnicas de Engenharia de Software) e os desenvolvedores devem possuir o conhecimento das mesmas;

2. A Equipe de desenvolvimento deve possuir conhecimento das técnicas e estas devem ser manipuladas de forma cautelosa, pois a ferramenta CASE utilizadas, desta forma ajudam as técnicas e quando estas são manuseadas por pessoas inexperientes podem atrapalhar o desenvolvimento;

3. Deve-se formalizar a técnica de teste, ou seja, deve haver uma pessoa ou uma equipe responsável apenas por realizar os testes no sistema como um todo, pois foi observado que esta técnica busca garantir que o sistema desenvolvido esteja conforme o proposto.

5.1 Considerações Finais e Projetos Futuros

Após um ano trabalhando com técnicas de engenharia de software, observou-se que alguns softwares não desenvolvidos dentro dessas técnicas são rejeitados pelos clientes por não refletirem o que realmente almejava.

Pretendo aprofundar na técnica de teste, pois percebi que não é necessário esperar que o próprio cliente aponte erros ou falhas, sendo que esta técnica existe para garantir a conformidade do que foi acordado com o cliente e o produto entregue.

A Figura 5.1, mostra o ciclo de desenvolvimento de um sistema que utiliza a técnica de teste, percebe-se que o teste e validação, seguindo uma ordem, serão realizados antes da entrega e após a implementação.

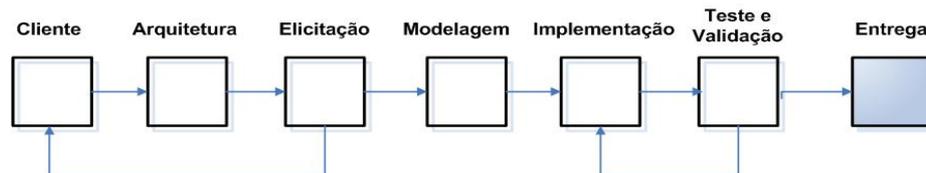


Figura 5.1 – Ciclo de Desenvolvimento de um Sistema

6 Referências

- [A. Silva Filho, 2002] Arquitetura de Software, Editora Campus, 2002.
- [Borland, 2004] Disponível em www.borland.com/jbuilder, acessado 15/11/2004.
- [Budi Kurniawan, 2003] Budi Kurniawan Sistemas multi camadas, 2003.
- [FDA, 2004] Disponível em www.fda.gov/cdrh/ocd/panamaradexp.htm, acessado 12/11/2004.
- [FIRB, 2004] Disponível em www.firb.br/sgbd.rtf, acessado 27/12/2004.
- [Garg, 1996] Garg, P. K. Process-centered Software Engineering Environments. Published by IEEE Computer Society Press, Los Alamitos, CA 90720-1264, 1996.
- [Gibbs, 1994] W. Gibbs. Software's Chronic Crisis. Scientific American Magazine, setembro, 1994.
- [Glen Myres] The Art of Software Testing, Wiley, 1979.
- [JAKARTA, 2004] Disponível em <http://jakarta.apache.org/tomcat>, acessado 16/11/2004.
- [JBoss, 2004] Disponível em www.jboss.org, acessado 16/11/2004.
- [Nathaniel Borenstein] Software Engineering and Other Noble Delusions, Princeton University Press, Princeton, NJ, 1991..
- [Netbeans, 2004] Disponível em www.netbeans.org, acessado 15/11/2004.
- [OMG, 2004] Disponível em www.omg.org, acessado 17/11/2004.
- [Sunnyday, 2004] Disponível em <http://sunnyday.mit.edu/therac-25.html>, acessado 12/11/2004.
- [Pressman, 1995] R. S. Pressman. Software Engineering: A Practitioner's Approach. McGraw-Hill, 3ª ed., 1995.

[Pressman,2002] Roger S. Engenharia de Software. 5. ed. Rio de Janeiro: McGraw-Hill,2002. 843p.

[Rouiller, 2001] Gerenciamento de Projetos de Software para Empresas de Pequeno Porte.

[Sommerville, 1998] Requirements Engineering, Processes and Techniques.

[UML, 2004] Disponível em www.uml.org, acessado 17/12/2004.

[Universia Brasil, 2004] Disponível em www.universiabrasil.net/mit/index.jsp, acessado 12/11/2004.

[Vasconcelos, 2004] Alexandre Marcos Lins de Vasconcelos - Introdução à engenharia de software e aos princípios de qualidade. -- Lavras : UFLA/FAEPE, 2004. 104 p. – Curso de Pós-Graduação “Lato Sensu” (Especialização) a Distância: Melhoria de Processo de Software.