

CARLOS EDUARDO PINHEIRO VITORINO
REGINALDO LONGANO JUNIOR
RODRIGO HERCULANO DE OLIVEIRA
VITOR CAÇADOR ALEXANDRE

SARPA

Sistema de Acompanhamento e Registro de Presença Acadêmica

Trabalho de Conclusão de Curso
apresentado à Escola Politécnica
da Universidade de São Paulo,
como parte dos requisitos
necessários para a obtenção do
título de Engenheiro de
Computação.

São Paulo
2006

CARLOS EDUARDO PINHEIRO VITORINO
REGINALDO LONGANO JUNIOR
RODRIGO HERCULANO DE OLIVEIRA
VITOR CAÇADOR ALEXANDRE

SARPA

Sistema de Acompanhamento e Registro de Presença Acadêmica

Trabalho de Conclusão de Curso
apresentado à Escola Politécnica
da Universidade de São Paulo,
como parte dos requisitos
necessários para a obtenção do
título de Engenheiro de
Computação, sob orientação da
Prof.^a Dra. Selma Shin Shimizu
Melnikoff.

São Paulo
2006

Aos nossos pais, que além de dar-nos a vida, bem mais precioso de todos, nos deram a base emocional e educacional para que nos tornássemos, hoje, Engenheiros.

AGRADECIMENTOS

Aos nosso pais que nos deram a oportunidade de nos tornamos Engenheiros, hoje. Seremos sempre gratos por todo o apoio moral, afetivo e financeiro dado até esse ponto da nossa vida.

A nossa família e amigos, que torceram por nós ao longo dessa jornada de 5 anos na Escola Politécnica.

A nossa professora e orientadora Selma Shin Shimizu Melnikoff, que nos transmitiu seus valiosos conhecimentos e nos orientou com excelência na elaboração de mais este projeto.

E aos professores da USP, com os quais tivemos a oportunidade não só de aprender muito do que sabemos, mas também de aprender a aprender coisas novas, para lidar com os desafios dessa próxima fase de nossas vidas.

*"Quando nada parece dar certo, vou
ver o cortador de pedras martelando
sua rocha talvez 100 vezes, sem que
uma única rachadura apareça. Mas na
centésima primeira martelada a pedra
se abre em duas, e eu sei que não foi
aquela que conseguiu isso, mas todas
as que vieram antes."*

Jacob August Riis

RESUMO

O objetivo deste projeto é desenvolver um sistema computacional, com acesso via web, para automatizar o registro e acompanhamento de presença acadêmica na Escola Politécnica da Universidade de São Paulo através da carteira USP, utilizando técnicas de elicitação de requisitos, modelagem UML e metodologia MVC de desenvolvimento. Surgiu da necessidade de automatização do processo atual, que não permite acompanhamento durante o período letivo para os alunos e não fornece estatísticas de *feedback* para os professores, além de ser um processo burocrático e altamente exposto a fraudes.

Palavras-chave: Engenharia de Requisitos. Elicitação de Requisitos. Leitora de Tarja Magnética. J2EE, MVC, UML.

ABSTRACT

The objective of this work is to develop a computational system, with web access, to automatize the registry and accompaniment of academic presence in the Politécnica School of the São Paulo University through the USP card, utilizing elicitation techniques, UML modeling and MVC development methodology. It has emerged from the need of automation of the current process, which does not allow accompaniment during the school periods for the students and does not supply statistical feedback for the teachers, beyond being a bureaucratic and highly exposed to frauds process.

Key words: Requirements Engineering. Requirements Elicitation. Magnetic Stripe Reader. J2EE. MVC. UML.

LISTA DE TABELAS

TABELA 1 – TÉCNICAS DE ELICITAÇÃO X QUESITOS ELICITÁVEIS	28
TABELA 2 – ANÁLISE DE SENSIBILIDADE DE PRESENÇA	43

LISTA DE ILUSTRAÇÕES

FIGURA 1 – FRENTE E VERSO DA CARTEIRA USP	11
FIGURA 2 – LEITORA DE CARTÕES MINIMAG	12
FIGURA 3 - EXEMPLO DE DIAGRAMA DE CASOS DE USO	16
FIGURA 4 – EXEMPLO DE DIAGRAMA DE CLASSE	17
FIGURA 5 – EXEMPLO DE DIAGRAMA DE SEQÜÊNCIA	18
FIGURA 6 – ARQUITETURA <i>MODEL VIEW CONTROLLER</i>	19
FIGURA 7 – ARQUITETURA MODEL 1	20
FIGURA 8 – ARQUITETURA MODEL 2	21
FIGURA 9 – ELABORAÇÃO DA GRADE HORÁRIA DE UM CURSO	23
FIGURA 10 – REPRESENTAÇÃO GRÁFICA ILUSTRANDO PRESENÇA DO ALUNO	45
FIGURA 11 – REPRESENTAÇÃO DOS DETALHES SOBRE A PRESENÇA DO ALUNO	46
FIGURA 12 – EXEMPLO DE VISUALIZAÇÃO DE FREQUÊNCIA EM UMA DISCIPLINA PELO PROFESSOR	47
FIGURA 13 – EXEMPLO DE TABELA DE CONTEÚDO E FREQUÊNCIA	47
FIGURA 14 – EXEMPLO DE ANÁLISE DE FREQUÊNCIA DE UMA TURMA	48
FIGURA 15 – EXEMPLO DE ANÁLISE COMPARATIVA ENTRE TURMAS	49
FIGURA 16 – EXEMPLO DE COMPARAÇÃO COM MÉDIA HISTÓRICA	50
FIGURA 17 – MODELO DE CASOS DE USO	51
FIGURA 18 – DIAGRAMA DE CLASSES	52
FIGURA 19 – DIAGRAMA ENTIDADE RELACIONAMENTO	53
FIGURA 20 – ARQUITETURA DE HARDWARE DO SISTEMA	54
FIGURA 21 – PRINCIPAIS MÓDULOS DE SOFTWARE DO SISTEMA	56

LISTA DE GRÁFICOS

GRÁFICO 1 – TEMPO QUE PROFESSORES LECIONAM NA EP	31
GRÁFICO 2 – TEMPO QUE PROFESSORES LECIONAM	32
GRÁFICO 3 – FREQUÊNCIA COM QUE OS ALUNOS COSTUMAM ASSISTIR ÀS AULAS	32
GRÁFICO 4 – PROFESSORES CONCORDAM COM A PRESENÇA OBRIGATÓRIA?	33
GRÁFICO 5 – MOTIVOS POR QUE PROFESSORES CONCORDAM COM A PRESENÇA OBRIGATÓRIA	33
GRÁFICO 6 - ALUNOS CONCORDAM COM A PRESENÇA OBRIGATÓRIA?	34
GRÁFICO 7 – MOTIVOS POR QUE ALUNOS CONCORDAM COM A PRESENÇA OBRIGATÓRIA	34
GRÁFICO 8 – MOTIVOS POR QUE ALUNOS NÃO CONCORDAM COM A PRESENÇA OBRIGATÓRIA	34
GRÁFICO 9 – MOTIVOS POR QUE ALUNOS CONCORDAM EM PARTE COM A PRESENÇA OBRIGATÓRIA	35
GRÁFICO 10 – O SISTEMA PROPOSTO, SEGUNDO PROFESSORES, TRARIA MELHORIA EM RELAÇÃO AO PROCESSO ATUAL?	36
GRÁFICO 11 – MOTIVOS POR QUE PROFESSORES ACREDITAM QUE O NOVO PROCESSO TRARIA MELHORIA	36
GRÁFICO 12 – VANTAGENS DO SISTEMA, SEGUNDO OS ALUNOS	36
GRÁFICO 13 – DESVANTAGENS DO SISTEMA, SEGUNDO OS ALUNOS	37
GRÁFICO 14 – O SISTEMA PROPOSTO DIFICULTA A FALSIFICAÇÃO DA PRESENÇA?	37
GRÁFICO 15 – REGRAS CONSIDERADAS VÁLIDAS POR PROFESSORES	38
GRÁFICO 16 – REGRAS CONSIDERADAS VÁLIDAS POR ALUNOS	38
GRÁFICO 17 – COMO ALUNOS PODEM FAZER MAU USO DO SISTEMA, SEGUNDO PROFESSORES	39
GRÁFICO 18 – COMO PROFESSORES PODEM FAZER MAU USO DO SISTEMA, SEGUNDO ALUNOS	39
GRÁFICO 19 – ONDE DEVERIA FICAR A LEITORA MAGNÉTICA SEGUNDO OS PROFESSORES	

LISTA DE ABREVIATÖES E SIGLAS

BD	Banco de Dados
CVS	Concurrent Version System
EJB	Enterprise JavaBeans
EP	Escola Polit�cnica
HTML	HyperText Markup Language
IIS	Internet Information Server
ISO	International Organization for Standardization
J2EE	Java 2 Enterprise Edition
JDBC	Java Database Connectivity
JDK	Java Development Kit
JSP	JavaServer Pages
JVM	Java Virtual Machine
MVC	Model-View-Controller
ODBC	Open Database Connectivity
OLE DB	Object Link Embedding Database
OMG	Object Management Group
OMT	Object Modeling Technique
OOSE	Object-Oriented Software Engineering
PVC	Policloreto de vinila
RFID	Radio-Frequency Identification
SARPA	Sistema de Acompanhamento e Registro de Presen�a Acad�mica
SGBD	Sistema Gerenciador de Banco de Dados
SGML	Standard Generalized Markup Language
SQL	Structured Query Language
UML	Unified Modeling Language
USB	Universal Serial Bus
USP	Universidade de S�o Paulo
XML	eXtensible Markup Language
XSLT	eXtensible Stylesheet Language Transformations

VITORINO, Carlos Eduardo Pinheiro. **SARPA: sistema de acompanhamento e registro de presença acadêmica** / C.E.P. Vitorino, R. Longano Junior, R.H. de Oliveira, V.C. Alexandre. São Paulo. 2006. (Trabalho de Formatura) Escola Politécnica, Universidade de São Paulo, São Paulo, 2006.

ERRATA

PÁGINA	LINHA	ONDE SE LÊ	LEIA-SE
FICHA CATALOGRÁFICA	6ª. Linha	185 p.	170 p.
1	6ª. Linha	que é do a sigla	que é a sigla
42	1ª. Linha	Projeto e Implementação	Análise, Projeto e Implementação

SUMÁRIO

1	INTRODUÇÃO	1
1.1	OBJETIVO DO PROJETO DE FORMATURA	1
1.2	MOTIVAÇÃO E JUSTIFICATIVA	2
1.3	ESTRUTURA DO DOCUMENTO	2
2	CONCEITOS E TECNOLOGIAS	4
2.1	CONCEITOS	4
2.1.1	CONTROLE DE PRESENÇA	4
2.1.2	CARTÕES MAGNÉTICOS	4
2.1.3	INTERFACE USB	4
2.1.4	PROCESSO DE DESENVOLVIMENTO DE SOFTWARE	5
2.1.5	MODELAGEM DE PROCESSO DE NEGÓCIO	5
2.1.6	TÉCNICAS DE ELICITAÇÃO DE REQUISITOS	5
2.1.7	ARQUITETURA DE SOFTWARE	6
2.1.8	ARQUITETURA CLIENTE SERVIDOR	6
2.2	TECNOLOGIAS	7
2.2.1	J2EE - JAVA 2 ENTERPRISE EDITION	7
2.2.2	STRUTS	8
2.2.3	HIBERNATE	8
2.2.4	JUNIT	8
2.2.5	XML - EXTENSIBLE MARKUP LANGUAGE	8
2.2.6	CVS - CONCURRENT VERSION SYSTEM	9
2.2.7	TOMCAT	9
2.2.8	MYSQL	10
2.2.9	ECLIPSE	10
2.2.10	XSTREAM	10
2.2.11	CARTEIRA DE IDENTIFICAÇÃO USP	11
2.2.12	LEITORA DE CARTÃO MAGNÉTICO MINIMAG	11
2.3	IMPORTÂNCIA	13
3	METODOLOGIA DO PROJETO	14
3.1	ETAPA DE ELICITAÇÃO DE REQUISITOS	14
3.2	ETAPA DE ESPECIFICAÇÃO DE REQUISITOS	14
3.2.1	UML - UNIFIED MODELING LANGUAGE	15
3.2.2	DIAGRAMA DE CASOS DE USO	15
3.2.3	DIAGRAMA DE CLASSES	16
3.2.4	DIAGRAMA DE SEQUÊNCIA	17
3.3	ETAPA DE PROJETO E IMPLEMENTAÇÃO	18
3.3.1	ESTRUTURA MODEL VIEW CONTROLLER	18
3.3.2	MODEL	19
3.3.3	VIEW	19
3.3.4	CONTROLLER	20
3.3.5	ARQUITETURAS MODEL 1 E MODEL 2	20
4	MODELAGEM DO PROCESSO DE NEGÓCIO	22
4.1	PROCESSO ATUAL	22
4.1.1	ELABORAÇÃO DA GRADE HORÁRIA	22
4.1.2	MATRÍCULA	23

4.1.3	EMIÇÃO DAS LISTAS	23
4.1.4	CONTABILIZAÇÃO DAS FREQUÊNCIAS	24
4.1.5	LANÇAMENTO DAS FREQUÊNCIAS	24
4.2	PROCESSO PROPOSTO	24
4.2.1	ELABORAÇÃO DA GRADE HORÁRIA E MATRÍCULA	24
4.2.2	REGISTRO DE PRESENÇA	25
4.2.3	CONTABILIZAÇÃO DAS FREQUÊNCIAS	25
4.2.4	LANÇAMENTO DAS FREQUÊNCIAS	25
4.2.5	ACOMPANHAMENTO DA PRESENÇA	25
4.2.6	FEEDBACK ESTATÍSTICO	26
5	ELICITAÇÃO DE REQUISITOS	27
5.1	INTRODUÇÃO	27
5.2	ENGENHARIA DE REQUISITOS E TÉCNICAS DE ELICITAÇÃO DE REQUISITOS	27
5.3	ESCOLHA DA TÉCNICA DE ELICITAÇÃO	29
5.4	TÉCNICAS DE ELABORAÇÃO DE QUESTIONÁRIOS	29
5.5	ELABORAÇÃO E APLICAÇÃO DO QUESTIONÁRIO	30
5.6	ANÁLISE DOS RESULTADOS OBTIDOS	31
5.6.1	PERFIL DOS ALUNOS E PROFESSORES	31
5.6.2	OPINIÕES SOBRE A PRESENÇA OBRIGATÓRIA	32
5.6.3	VANTAGENS E DESVANTAGENS DO SISTEMA	35
5.6.4	INTEGRIDADE DA PRESENÇA	37
5.6.5	REGRAS VALIDADORAS DE PRESENÇA	37
5.6.6	MAU USO DO SISTEMA	38
5.6.7	POSIÇÃO DAS LEITORAS MAGNÉTICAS	39
5.6.8	OUTRAS OPINIÕES RELEVANTES	40
5.6.9	DECISÕES TOMADAS A PARTIR DOS RESULTADOS OBTIDOS	41
6	PROJETO E IMPLEMENTAÇÃO	42
6.1	ANÁLISE DE REQUISITOS	42
6.1.1	CRITÉRIOS DE PRESENÇA	42
6.1.2	EXCEÇÕES	43
6.1.3	ALUNO FALTOU À AULA	43
6.1.4	A AULA MUDOU DE DIA	43
6.1.5	A AULA SOFREU ATRASO	44
6.1.6	A AULA FOI CANCELADA	44
6.1.7	INTERFACES DE ACESSO	44
6.1.8	ANÁLISES ESTATÍSTICAS	44
6.1.9	MÓDULO ALUNO	45
6.1.10	MÓDULO PROFESSOR	46
6.1.11	ANÁLISE DE UMA DISCIPLINA ESPECÍFICA	46
6.1.12	ANÁLISE DE OUTRAS DISCIPLINAS DO MESMO CURSO E SEMESTRE	48
6.1.13	ANÁLISE DE TURMAS DE UMA MESMA DISCIPLINA	48
6.1.14	ANÁLISE DE DISCIPLINA CONTRA A MÉDIA HISTÓRICA	49
6.1.15	MÓDULO ADMINISTRADOR	50
6.1.16	MODELAGEM	50
6.1.17	MODELO DE CASOS DE USOS	51
6.1.18	DIAGRAMA DE CLASSES	52
6.1.19	DIAGRAMA DE ENTIDADE-RELACIONAMENTO	52
6.1.20	DIAGRAMAS DE SEQUÊNCIA	54
6.2	ARQUITETURA DE HARDWARE	54
6.2.1	BLOCO DE LEITURA	54
6.2.2	BLOCO SERVIDOR DE REGISTROS	55
6.2.3	BLOCO SERVIDOR WEB	55
6.3	ARQUITETURA DE SOFTWARE	56
6.3.1	GERENCIAMENTO DE REGISTRO E PRESENÇA	56

6.3.2	CONTROLLER	57
6.3.3	GERENCIADOR DE PERSISTÊNCIA	57
6.3.4	TESTES	57
6.3.5	MODELAGEM E CONFIGURAÇÃO DO BANCO DE DADOS	57
6.3.6	MODEL	57
6.3.7	VIEW	57
6.3.8	ACTION	58
6.3.9	FORM	58
6.4	MÓDULOS E SERVIÇOS AUXILIARES DA ARQUITETURA DE SOFTWARE	58
6.4.1	SERVIÇO DE LEITURA DE CARTÕES	58
6.4.2	CONVERSÃO DE OBJETO EM ARQUIVO XML	59
6.4.3	ENVIO DO ARQUIVO XML AO SERVIDOR DE BD	59
6.4.4	MÓDULO DE RECEPÇÃO DE REGISTROS	59
6.4.5	MÓDULO DE ACOMPANHAMENTO DA PRESENÇA ACADÊMICA	60
6.4.6	ACOMPANHAMENTO PELO ALUNO	60
6.4.7	ACOMPANHAMENTO PELO PROFESSOR	60
6.5	INTEGRAÇÃO E TESTES	61
6.5.1	TESTES UNITÁRIOS	61
6.5.2	TESTES DE INTEGRAÇÃO	61
7	CONSIDERAÇÕES FINAIS	62
7.1	CONCLUSÃO	62
7.2	RESULTADOS OBTIDOS	62
7.3	DIFICULDADES ENCONTRADAS	62
7.4	APRENDIZADO COM O PROJETO	63
7.5	CONTRIBUIÇÃO	63
7.6	TRABALHOS FUTUROS	64
7.7	VIABILIDADE DE IMPLANTAÇÃO	64
	BIBLIOGRAFIA E REFERÊNCIAS	65
	APÊNDICE A – MODELOS DE QUESTIONÁRIOS UTILIZADOS	67
	APÊNDICE B – ESPECIFICAÇÃO DOS REQUISITOS	71
	APÊNDICE C – ESPECIFICAÇÃO DA ARQUITETURA	88
	APÊNDICE D – ESPECIFICAÇÃO DOS CASOS DE USO	104
	APÊNDICE E – DIAGRAMAS DE SEQUÊNCIA	138
	APÊNDICE F – PLANO DE TESTES	144
	ANEXO A – EXEMPLOS DE RESPOSTAS OBTIDAS AOS QUESTIONÁRIOS	164
	ANEXO B – MODELO DE LISTA DE PRESENÇA ATUAL	170

1 INTRODUÇÃO

1.1 Objetivo do Projeto de Formatura

O objetivo deste trabalho é projetar um sistema capaz de automatizar o processo de controle de presença às aulas dos alunos da Escola Politécnica e desenvolver uma versão inicial do sistema que visa evidenciar seus benefícios em relação ao processo atual e sua viabilidade de implementação. O nome adotado para o sistema é SARPA, que é do a sigla para Sistema de Acompanhamento e Registro de Presença Acadêmica. As principais funções do SARPA são:

- Registro da presença, através da passagem da carteira USP por leitoras de cartão magnético nas salas de aula da EP, o aluno registra sua entrada e saída da sala de aula e consegue a presença se seu tempo de permanência em sala atende a critérios de porcentagem de frequência requerida em cada aula, estabelecidos pelo professor para cada disciplina. Desse modo, o processo de registro de presença é automatizado.
- Consulta de histórico estatístico de presenças, permitindo comparações como, por exemplo, entre dois períodos de uma disciplina ministrada pelo mesmo professor, entre duas disciplinas diferentes de um mesmo professor ou entre a mesma disciplina ministrada por professores diferentes.
- Apoio para análise da programação das disciplinas: os professores podem associar, nas disciplinas que lecionam, o conteúdo planejado aula a aula para auxiliar na sua análise ao fim da disciplina e, ainda, servir de referência para os alunos.
- Apresentação do horário atual e da porcentagem de frequência requerida para as disciplinas oferecidas no momento.
- Acesso às presenças de todos alunos, pelo professor, de disciplinas que este ministra e acesso, pelo aluno, de informações de frequência das disciplinas que esse está matriculado.
- Pedido da revisão de presença: o aluno pode requerer, mediante justificativa, a revisão de sua presença em determinada aula.
- Alteração de parâmetros da aula, devido a mudanças em salas de aula ou variações nos horários de início e término da mesma. No caso de uma alteração posterior à data da aula, o professor poderá pedir o recálculo das presenças pelo sistema.
- Impressão de listas pré-formatadas, no caso de exceções do sistema – como, por exemplo, uma leitora magnética defeituosa – é possível o uso do sistema antigo como contingência.

Para alcançar o objetivo proposto foram feitas, em um momento inicial, diversas reuniões para discutir os requisitos do projeto e, em um segundo momento, os requisitos mais

centrais do projeto foram obtidos através da aplicação de técnicas de Engenharia de Requisitos envolvendo os usuários chave do sistema.

1.2 Motivação e Justificativa

A motivação do projeto surgiu da busca pela automatização de processos ineficientes e burocráticos dentro do contexto da USP. Logo, o sistema atual de controle de presença, enquadrou-se perfeitamente nessa busca. Além de ser um sistema altamente suscetível a fraudes, não há nenhum acompanhamento automatizado da frequência do aluno na disciplina e os docentes não tem ferramenta de análise da aderência dos alunos às disciplinas que ministram. Além disso, a questão da demora no cálculo e computação da frequência final dos alunos também serviu como motivação para o sistema, bem como a questão ambiental da impressão desnecessária das listas em papel.

Além desses fatores, o projeto abrange vários aspectos que se mostraram como novidade para os integrantes do grupo como a integração entre *hardware* e *software* e as técnicas de Engenharia de Requisitos utilizadas na especificação. Tudo isso aliado às diversas tecnologias utilizadas no desenvolvimento do *software*, tornaram grandioso o desafio do projeto.

É importante comentar que no início de 2006, o Governo Federal começou a implantar, nas escolas públicas do país, um sistema chamado Safe (Sistema Nacional de Acompanhamento da Frequência Escolar) que tem objetivos muito similares ao deste projeto. Os objetivos desse projeto do governo – intitulado Projeto Presença – são: garantir a permanência dos alunos na escola; oferecer dados precisos e atualizados para subsidiar a implementação e o monitoramento das políticas públicas educacionais; fornecer mecanismos para uma gestão escolar mais eficiente; otimizar a distribuição dos recursos públicos federais, de acordo com o número de matrículas nas escolas de cada município e possibilitar a integração com os programas sociais (MINISTÉRIO DA EDUCAÇÃO, 2006).

Essa iniciativa do Governo evidencia a importância que deve ser dada à frequência em cursos presenciais, que é atribuída como responsabilidade do poder público segundo o artigo 208 da Constituição Brasileira. Além disso, é evidente a importância de se ter ferramentas para análise e controle da gestão escolar.

Por todos esses motivos decidiu-se desenvolver este projeto que pode proporcionar uma grande melhoria em relação ao processo atual.

1.3 Estrutura do documento

O Capítulo 2 apresenta os conceitos técnicos e de negócio considerados relevantes e que são mencionados ao longo da monografia. O Capítulo 3 dá uma visão sobre a metodologia de modelagem e desenvolvimento, baseada na UML. O Capítulo 4 descreve o processo antigo do sistema de controle de presença e, então, o processo automatizado proposto por esse projeto. No Capítulo 5 são explicados mais detalhadamente os conceitos de Engenharia de Requisitos utilizados no trabalho e como estes foram aplicados na especificação de requisitos chave

do projeto. No Capítulo 6 são feitas a análise e a especificação dos requisitos funcionais do sistema, bem como a descrição da implementação do projeto e a arquitetura de *hardware* proposta e de *software* implementada, além dos testes feitos. No Capítulo 7 são apresentadas as considerações finais relacionadas a esse trabalho. Em seguida, encontram-se as referências bibliográficas, os apêndices e os anexos que deram apoio à realização deste projeto.

2 CONCEITOS E TECNOLOGIAS

Esse capítulo tem o objetivo de apresentar conceitos e tecnologias que foram de grande importância para realização desse projeto. Para um melhor aproveitamento da leitura dessa monografia, é recomendado que esses conceitos sejam compreendidos.

2.1 Conceitos

2.1.1 Controle de Presença

O curso de graduação na Universidade de São Paulo é predominantemente presencial, significando que deve haver um controle de presença para garantir que os alunos estão atendendo a um número mínimo de aulas para que sua frequência na disciplina em questão seja considerada aceitável. O sistema SARPA propõe um controle automático dessa presença pela implantação de leitoras de cartão magnético para leitura da carteira USP e um sistema *web* utilizado por alunos para controle de suas respectivas presenças e por professores para controle das presenças de suas turmas.

2.1.2 Cartões magnéticos

Um cartão magnético é um tipo de cartão que possui a capacidade de armazenar dados, modificando o magnetismo de partículas minúsculas em uma faixa de material magnético presente no cartão. A faixa magnética é lida através do contato físico entre uma leitora e o cartão, após este passar inteiramente pela cabeça de leitura.

Este tipo de cartão é usado geralmente em cartões de crédito, cartões de identidade, bilhetes do transporte e vales-refeição.

De modo geral, é possível armazenar dados em três trilhas magnéticas, na mesma fita magnética presente no cartão. A trilha 1 pode gravar até 79 caracteres, a trilha 2, 40 e a trilha 3 até 107 caracteres (PISTONE,2006).

2.1.3 Interface USB

O USB foi concebido na óptica do conceito de *Plug and Play*, minimizando o esforço de concepção de periféricos, no que diz respeito ao suporte por parte dos sistemas operacionais e *hardware*.

O sistema foi projetado de maneira que possam ser ligados vários periféricos pelo mesmo canal (i.e., porta USB). Assim, mediante uma topologia em árvore, é possível ligar até 127 dispositivos a uma única porta do computador, utilizando, para a derivação, *hubs* especialmente concebidos. Se, por exemplo, as impressoras ou outros periféricos existentes hoje tivessem uma

entrada e saída USB, poder-se-ia ligar estes como uma corrente de até 127 (INTEL, 2006) dispositivos, interligados, os quais o computador gerenciaria sem nenhum problema, levando em conta o tráfego requerido e velocidade das informações solicitadas pelo sistema.

Estes *hubs* são também dispositivos USB, com classe específica, e responsáveis pela gestão da sua sub-árvore e cooperação com os nós acima (o computador ou outros *hubs*). Esta funcionalidade foi adaptada da vasta experiência em redes de BUS, como a Ethernet. Nesse caso, o computador apenas encaminhará os pacotes USB (unidade de comunicação do protocolo, ou URB, do inglês *Uniform Request Block*) para uma das portas, e o pacote transitará pelo barramento até ao destino, encaminhado pelos *hubs* intermediários.

2.1.4 Processo de Desenvolvimento de Software

O processo de desenvolvimento de *software* engloba desde a fase de planeamento do sistema, até a aceitação do sistema pelo cliente, passando por especificação, arquitetura, codificação, implementação, testes, homologação, implantação dentre outras (PRESSMAN, 2002). Há diversos modelos de processo de *software* com diferentes abordagens e diferentes resultados. É uma decisão que cabe à equipe de desenvolvimento, qual modelo adotar.

2.1.5 Modelagem de Processo de Negócio

A análise e modelagem de um processo de negócio devem permitir a introdução segura de regras, tempos, rotas e papéis funcionais no ambiente organizacional. Quer o processo já exista de forma desestruturada, quer seja um novo processo necessário para suportar um novo negócio, sua análise e modelagem garantem o compartilhamento do conhecimento na sua implantação e a gerência desse mesmo conhecimento na operacionalização.

A modelagem de processo de negócio também tem a função de responder a questões como:

- Quais informações direcionam o processo de negócio.
- Quais informações são geradas.
- Quem gera tais informações.
- Quem consome tais informações.
- Onde essas informações são processadas.
- Que informação é gerada ao final do processo.

2.1.6 Técnicas de Elicitação de Requisitos

O papel da Engenharia de Requisitos aplicada à Engenharia de *Software* é ajudar a identificar, produzir uma documentação adequada, dar manutenção e gerir os requisitos funcionais de um sistema. Por essa razão, o uso de técnicas de levantamento de requisitos na fase de especificação de um *software* complexo é essencial.

"Uma questão básica da Engenharia de Requisitos é como descobrir o que usuários realmente necessitam" (GOGUEN, 1997, p.1, tradução dos autores). Nessa mesma linha de pensamento, Goguen (1997) ainda ressalta que muitos grandes projetos podem fracassar devido ao levantamento inadequado de requisitos e que, muitas vezes, definições inadequadas de necessidades do sistema por parte dos desenvolvedores ou clientes, levam ao fracasso no levantamento de requisitos.

Portanto, sistemas complexos devem contar com ferramentas adequadas para o levantamento de requisitos, tentando maximizar a chance de sucesso de um projeto.

2.1.7 Arquitetura de Software

Os fatores que podem causar o insucesso de sistemas de *software* e não são percebidos pelas pessoas envolvidas durante o desenvolvimento são muitos e identificá-los não é trivial. Especificações e implementações baseadas unicamente em requisitos funcionais não são garantia de um bom andamento do projeto e nem de robustez do sistema, em especial quando for considerado o dinamismo das definições. A preocupação em atender todos os requisitos necessários, funcionais e não funcionais, dos componentes deste sistema deve ser constante e fornecerá base para antecipar importantes tomadas de decisão, minimizando riscos. Chama-se esta definição, arquitetura de *software* (VAROTO, 2002).

"A arquitetura de um *software* pode ser observada sob diversas perspectivas ou visões, cada qual ressaltando aspectos específicos. As visões podem ser utilizadas como base para se definir a arquitetura de um *software*, considerando sua abrangência no processo de desenvolvimento de sistemas, seu foco de atuação e seu impacto nos fatores de qualidade requeridos para a solução do problema." (VAROTO, 2002, p.4).

2.1.8 Arquitetura Cliente Servidor

A arquitetura cliente servidor é uma infra-estrutura versátil e modular que visa melhorar usabilidade, flexibilidade, interoperabilidade e escalabilidade comparado com computação centralizada ou *mainframes* (SOFTWARE ENGINEERING INSTITUTE, 2006). Cada instância do cliente pode enviar requisições para o servidor ou servidor de aplicação. Existem muitos tipos diferentes de servidores, desde servidores de arquivo, passando por servidores de e-mail até servidores de terminais.

Embora a idéia citada anteriormente possa ser aplicada de muitas maneiras diferentes, o exemplo mais simples para se visualizar é o uso de uma página web na internet. Por exemplo, se se estiver lendo um site de notícias, o computador de acesso pessoal e *web browser* podem ser considerados clientes, e os computadores, banco de dados e aplicações responsáveis por manter tal *site* no ar são o servidor. Quando nosso navegador web faz uma requisição ao *site* de notícias, o servidor pesquisa toda a informação requisitada nos bancos de dados do *site*, monta a página web e a envia de volta ao cliente para que possamos consultá-la.

Principais características de um servidor:

- Passivo.
- Aguarda requisições.
- Após receber requisições, realiza seu processamento e responde ao cliente.

Principais características de um cliente:

- Ativo.
- Envia Requisições.
- Aguarda a resposta do servidor.

2.2 Tecnologias

Essa seção apresenta as seguintes tecnologias utilizadas neste projeto:

- J2EE - Java 2 Enterprise Edition
- Struts
- Hibernate
- JUnit
- XML - eXtensible Markup Language
- CVS - Concurrent Version System
- Tomcat
- MySQL
- Eclipse
- XStream
- Carteira de Identificação Usp
- Leitora De Cartão Magnético Minimag

2.2.1 J2EE - Java 2 Enterprise Edition

J2EE (*Java 2 Platform, Enterprise Edition*) (SUN, 2006) é uma plataforma Java projetada para grandes empresas que necessitam de soluções de computação de larga escala. A Sun Microsystems e a IBM projetaram o J2EE para simplificar o desenvolvimento de aplicações em ambientes envolvendo clientes. J2EE simplifica o desenvolvimento de aplicações e reduz a quantidade de programação necessária e treinamento do programando através de utilização de módulos padronizados. Ele também inclui muitos componentes da plataforma Java 2, *Standard Edition*, como:

- *The Java Development Kit* (JDK).
- Tecnologia *Write Once Run Anywhere* permitindo grande portabilidade.
- *Java Database Connectivity* 2.0 (JDBC), equivalente ao *Open Database Connectivity* (ODBC) oferecendo suporte para interface de banco de dados Java.

- Modelo de segurança que oferece proteção aos dados de aplicações locais e remotas via web.

2.2.2 Struts

Struts é um *framework open-source* que auxilia na construção de aplicações para a web. Ele é construído em Java, e seu coração consiste numa camada de controle flexível baseada nas tecnologias Java *Servlets*, *JavaBeans*, *ResourceBundles* e XML.

O Struts favorece o desenvolvimento de aplicações seguindo o paradigma MVC (*Model-View-Controller*). Além disso, fornece um componente *Controller* e se integra a outras tecnologias para oferecer suporte aos componentes *Model* (como JDBC, EJB's, etc.), e *View* (como JSP e XSLT) (APACHE SOFTWARE FOUNDATION, 2006).

2.2.3 Hibernate

Hibernate é uma ferramenta *open-source* que facilita a comunicação entre aplicação Java e banco de dados, como manipulação de dados, gerenciamento de transações, *pooling* de conexões, permite herança, associação um-para-muitos ou um-para-um e composição.

Toda a configuração do Hibernate é feita através de arquivos XML, os quais contêm mapeamentos de tabelas/Java e detalhes de *pooling* de conexões. Combinados, estes arquivos fornecem total configurabilidade à aplicação (HIBERNATE, 2006).

2.2.4 JUnit

O JUnit é um framework que facilita o desenvolvimento e execução de testes unitários em código Java.

Os principais motivos que favorecem o seu uso são (JUNIT, 2006):

- JUnit pode verificar se cada unidade de código funciona da forma esperada.
- Facilita a criação, execução automática de testes e a apresentação dos resultados.
- É Orientado a Objeto.

2.2.5 XML - eXtensible Markup Language

XML é uma linguagem de marcação estruturada com suporte a informações do tipo texto e figuras entre outros criando um meio para que o computador gere e leia dados de forma não ambígua (W3C, 2006). Uma das grandes vantagens de se usar XML sobre outros formatos

binários é que, se necessário, o próprio desenvolvedor pode interpretar o documento, dado que se trata de um modelo textual.

O desenvolvimento de XML iniciou-se em 1996 e faz parte de uma Recomendação W3C desde 1998. Como se pode ver não é uma tecnologia muito recente, principalmente dado que antes de XML já existia SGML, linguagem que foi desenvolvida na década de 80 e faz parte do padrão ISO desde 1986 sendo largamente utilizada em grandes projetos de documentação. Já o desenvolvimento no padrão HTML começou somente nos anos 90. Os projetistas da XML aproveitaram as melhores partes da SGML, guiados pela experiência acumulada com HTML, e produziram algo que não é em nada menos poderoso que SGML, e amplamente mais regular e simples de usar. Contudo, às vezes é difícil distinguir algumas evoluções de revoluções. E deve ser dito que, enquanto SGML é usada principalmente para documentação técnica e muito menos para outros tipos de dados, com XML ocorre exatamente o oposto.

Optou-se pela utilização de XML devido à experiência que era possuída com a linguagem e a facilidade de se obter suporte, dado que é uma tecnologia de licença livre e bastante difundida.

2.2.6 CVS - Concurrent Version System

Como ferramenta de controle de versão para suporte ao desenvolvimento do sistema foi utilizado o sistema CVS (*Concurrent Version System*), fazendo papel do repositório central do sistema. O sistema oferece suporte a facilidades como comparação e compartilhamento de códigos, e já havia sido utilizado na ocasião de outra disciplina, mostrando-se confiável e permitindo que vários programadores gerassem e comparassem seus códigos em poucos minutos.

Outro fator levado em consideração foi o suporte nativo oferecido pelo Eclipse para o CVS. Dentro da própria interface de desenvolvimento é possível acessar várias funções do CVS como *UPDATE* e *COMMIT*. Tal facilidade reduz o tempo gasto para se compartilhar as mudanças efetuadas durante o desenvolvimento, que em poucos segundos podem ser observadas pelos demais desenvolvedores (CVS, 2006).

2.2.7 Tomcat

Tomcat é um servidor web gratuito que oferece suporte à linguagem Java. Implementando as especificações de *servlets* e JSPs em *servlets*, Tomcat é o resultado de um trabalho colaborativo de desenvolvedores de *software* livre, sendo disponível, portanto, não somente seu código executável, mas todo o seu código fonte. Pode ser usado como um servidor isolado ou trabalhar em conjunto com outros *Web Servers*, incluindo Apache, Netscape Enterprise Server e Microsoft Internet Information Server (IIS), e trabalha em conjunto com a JVM versão 1.1 ou superior.

Optou-se pelo Tomcat por ser uma das mais naturais escolhas em termos de ferramentas, uma vez que já se havia trabalhado com o produto em uma disciplina anterior e obtido resultados muito bons. O Tomcat também mostrou atender aos requisitos de segurança, confiabilidade e rapidez da qual se necessitava (APACHE TOMCAT, 2006).

2.2.8 MySQL

MySQL é um *software* livre de SGBD (Sistema Gerenciador de Banco de Dados) baseado na linguagem SQL para processar dados em um banco de dados. Oferece API's para diversas linguagens, incluindo C, C++, Eiffel, Java, Perl e PHP. Também oferece conexões OLE DB e ODBC para ambientes Microsoft.

MySQL é normalmente usado por aplicações *web* e se tornou uma alternativa popular a bancos de dados proprietários devido à sua rapidez e confiabilidade.

A opção pelo MySQL se fundamenta principalmente na sua confiabilidade, segurança e gratuidade. Chegou-se a considerar usar outras versões de gerenciadores de banco de dados gratuitos, mas nenhum deles possuía um número tão grande de usuários e conseqüentemente facilidade de suporte para solução dos problemas que ocorreram durante o desenvolvimento (MYSQL, 2006).

2.2.9 Eclipse

O ambiente de desenvolvimento selecionado foi o Eclipse, *software* gratuito de grande popularidade para desenvolvimento de aplicações em Java. O *software* é oferecido sob licença GNU, ou seja, livre de qualquer custo, desde que não se faça uso comercial do mesmo.

Tal ferramenta foi selecionada baseado na grande gama de recursos disponibilizados, dentre os quais se destacam: (ECLIPSE, 2006):

- Suporte multi-formato: A plataforma de desenvolvimento Eclipse oferece suporte aos mais variados tipos de arquivo, sejam eles aplicações J2EE, JSP ou arquivos XML e HTML.
- Interface organizada e completa: A interface de trabalho disponibilizada pelo Eclipse é uma das mais organizadas dentre os *softwares* de desenvolvimento para Java, situando-se entre as mais conceituadas pela facilidade de uso e disponibilidade de recursos para o desenvolvimento de aplicações.
- Suporte a desenvolvimento em equipe: O Eclipse oferece suporte ao sistema de controle de versão CVS, permitindo aos membros da equipe de desenvolvimento trabalharem em diferentes partes do sistema em paralelo e facilmente obterem a versão mais atualizada do sistema sem a necessidade de se enviar o sistema todo ou separar as partes que sofreram alteração.

2.2.10 XStream

A ferramenta XStream fornece suporte a serialização de objetos XML para classes Java e vice-versa. Permite que objetos Java sejam convertidos em XML para que outros dispositivos possam recuperar essa informação e interpretá-la. Trabalha nos dois sentidos, ou seja, é capaz de

gerar o arquivo XML a partir de um objeto e de recriar o objeto através da leitura do XML (XSTREAM, 2006).

2.2.11 Carteira de Identificação USP

A carteira de identificação USP baseia em dois mecanismos de identificação principais, a tarja magnética e o código de barras. Diferentemente do código de barras que só guarda a identificação do aluno (número USP), a tarja magnética oferece não só a identificação, mas também o nome do aluno e acima disso um nível de segurança maior, uma vez que é mais difícil copiar um cartão magnético em comparação com um código de barras, pois é necessário um equipamento especial para copiá-lo, enquanto que um código de barras pode ser copiado com um simples xérox.

A opção no projeto pela carteira USP se baseia no fato de que não seria necessária a compra de nenhum outro equipamento que as leitoras magnéticas. Não seria necessário trocar o atual modelo de carteira USP tampouco fazer cadastro de dados como impressões digitais ou outros identificadores biométricos, uma vez que todos os dados necessários já estão armazenados na mesma.

A carteira de identificação USP é confeccionada em PVC, com foto digitalizada, código de barras e tarja magnética, os quais possibilitam que em qualquer ponto de acesso devidamente controlado, seja possível identificar a pessoa a partir de seu número USP (Figura 1).



Figura 1 – Frente e verso da carteira USP

A sua trilha magnética apresenta três trilhas: na trilha 1 está o nome do aluno; na trilha 2 está o número USP e na trilha 3, caracteres de controle.

2.2.12 Leitora de Cartão Magnético MINIMAG

A leitora de tarja magnética MiniMag™ da ID-TECH pode ler 1, 2, ou 3 trilhas de informações da tarja magnética. Quando conectado com um computador o MiniMag é totalmente compatível com o Windows. Para o ele, os dados decodificados aparecem como se fossem colocados manualmente por um operador através do teclado.

Os dados enviados podem ser formatados com prefixo/sufixos ou caracteres terminais, para estar em conformidade com o protocolo desejado. Essa formatação é feita através de um *software* fornecido pelo fabricante.

Fácil de ser usado e extremamente durável (vida útil de no mínimo 500.000 passagens) ele possui um indicador LED e *beep* confirmando se a leitura foi bem sucedida (IDTECH, 2000). Na Figura 2, vemos uma leitora de cartão Minimag.



Figura 2 – Leitora de Cartões Minimag

Fonte: MAC-POS Hardware, 2006

2.3 Importância

Como ressaltado no início deste capítulo, sua importância é entender conceitos e tecnologias de grande relevância para este projeto de formatura. Além de esclarecer conceitos importantes como os de controle de presença e de técnicas utilizadas no projeto, esse capítulo evidencia as características de tecnologias utilizadas para que o entendimento do restante desse trabalho tenha a base teórica apropriado.

3 METODOLOGIA DO PROJETO

O objetivo de se adotar uma metodologia para o desenvolvimento de *software* recai primordialmente na necessidade de entregar o *software*, seguindo as especificações do cliente, no prazo determinado, ao custo projetado e oferecendo qualidade tanto no que se refere aos requisitos funcionais quanto aos requisitos não funcionais. Para que essas especificações sejam atendidas, é comum adotar modelos de *software*.

Ao observar-se a definição da palavra "modelo" encontra-se algo semelhante à definição encontrada em (Houaiss, 2006): "coisa ou pessoa que serve de imagem, forma ou padrão a ser imitado, ou como fonte de inspiração". Ou seja, adota-se uma série de passos e atividades a serem seguidas, que são práticas tomadas como indicadores de que a o sucesso do *software* em desenvolvimento será provavelmente maior.

Como metodologia de desenvolvimento do projeto optou-se por uma divisão em três grandes blocos: elicitação de requisitos, especificação de requisitos e projeto e implantação. O grupo optou por esse modelo devido aos excelentes resultados obtidos nas disciplinas de Engenharia de *Software I e II*, no qual foram desenvolvidos projetos semelhantes, partindo desde a especificação até a apresentação do produto final ao cliente.

3.1 Etapa de Elicitação de Requisitos

A etapa de análise de requisitos consiste da definição da forma como os requisitos do *software* em desenvolvimento serão levantados, seguida da análise dos dados e finalmente da definição dos requisitos que serão atendidos pelo sistema. Essa etapa, em seus requisitos mais complexos, colocou os desenvolvedores em contato com a Engenharia de Requisitos e Técnicas de Elicitação. Esses conceitos são abordados com maior detalhe no Capítulo 5. O restante dos requisitos, os mais triviais, foram elaborados por consenso entre os desenvolvedores do projeto.

3.2 Etapa de Especificação de Requisitos

A segunda etapa consiste da especificação dos requisitos. Nessa fase são especificadas as características do sistema, refletindo o levantamento dos requisitos funcionais, não funcionais e das facilidades que serão disponibilizadas aos usuários.

Para efetuar a modelagem optou-se pela *Unified Modeling Language (UML)* por uma série de fatores dentre as quais destaca-se a flexibilidade muito maior oferecida pela UML na adaptação de toda a documentação gerada a realidade do *software* que está sendo desenvolvido.

3.2.1 UML – Unified Modeling Language

Após o surgimento do conceito de orientação a objeto na Noruega através do trabalho de Dahl e Nygaard em 1967, houve rápida difusão do conceito e, nasceu a necessidade de uma metodologia a ser seguida para comportar sistemas cada vez maiores e mais complexos.

Na década de 80 várias metodologias voltadas para projetos orientados a objeto foram difundidas, saindo de pouco menos de 10 em 1989 a mais de 50 em 1994. Apesar da grande diversidade de metodologias, era difícil encontrar uma que conseguisse satisfazer todas as necessidades de modelagem, gerando o que ficou conhecida como a guerra de metodologias. (BOOCH, 2005)

Algumas metodologias de desenvolvimento naturalmente se destacaram entre as demais, notoriamente as de Booch e Jacobson com a OOSE (Object-Oriented Software Engineering) e Rumbaugh com a OMT (Object Modeling Technique). Cada uma das metodologias era suficiente individualmente, mas era possível identificar claramente seus pontos fortes e fracos.

Com o passar do tempo, cada um dos três criadores passou a adotar em seu modelo as características de outras metodologias que julgavam interessantes impulsionando suas metodologias para um nível de exposição mundial no que se referia a metodologias orientadas a objeto.

Naturalmente Booch, Jacobson e Rumbaugh notaram que mesmo de forma independente suas metodologias estavam convergindo para um modelo comum, e a partir desse fato decidiram unificar as metodologias com três principais objetivos: Reduzir a confusão de metodologias para os usuários e criar um modelo estável e mais aceito por parte tanto de usuários e desenvolvedores, e finalmente procuravam compartilhar as experiências de cada um deles.

Em Outubro de 1995 foi lançado o primeiro rascunho que foi continuamente melhorado até por volta de 2003 quando foi então lançada a versão 2.0 já sob o controle da OMG (Object Management Group), uma organização sem fins lucrativos que trabalha no modelo de força tarefa para padronização de diversas tecnologias. (BOOCH, 2005)

Resumidamente, a linguagem UML é usada para visualização, pois ajuda a comunicar o que deve ser feito de forma visual, é usada para especificação, pois garante modelos precisos e livres de ambigüidades, é usada para construção já que permite a geração de código de forma direta a partir dos diagramas e é usada para documentação já que produz documentos desde a fase de requisitos até os testes realizados para validar o sistema.

3.2.2 Diagrama de Casos de Uso

O diagrama de casos de uso se propõe a ajudar a reconhecer as funções do sistema do ponto de vista dos usuários. Para isso são identificados os atores do *software* em desenvolvimento com as funções do seu interesse. Desta forma, é possível se ter uma visão global do sistema, e da sua funcionalidade. Quatro elementos fazem parte do diagrama de casos de uso: atores, o sistema, os casos de uso e as relações existentes entre os elementos.

Na Figura 3, observa-se um exemplo de diagrama de casos de uso. Pode-se notar a existência de atores (representados por bonecos), dos casos de uso (representados pelas figuras ovais) e as relações entre eles (representadas pelas linhas pretas cheias).

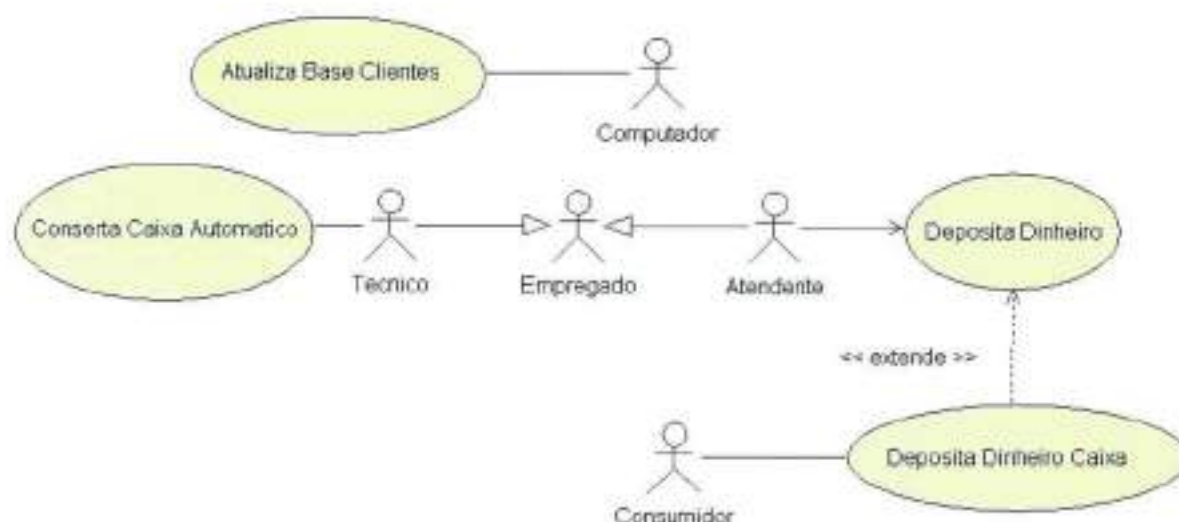


Figura 3 - Exemplo de diagrama de casos de uso

Fonte: Adaptado de SUN, 2006c, tradução dos autores

3.2.3 Diagrama de Classes

A função principal do diagrama de classes é mostrar as classes que fazem parte do sistema e seus relacionamentos. Através da confecção do diagrama de classes é possível ter uma visão geral da forma como as classes estão interagindo e seus papéis.

Através do diagrama de classes é possível inferir o nome da classe em questão, seus atributos e métodos. Também é possível observar qual é a multiplicidade da associação e se a associação é obrigatória ou opcional. Finalmente através do diagrama de classes é possível observar a generalização entre classes, ou seja, uma classe geral que contém atributos e métodos comuns e classes especializadas que, além dos métodos e atributos da classe genérica possuem também seus próprios atributos e métodos (BELL, 2006b).

Na Figura 4, pode-se identificar todas as informações citadas anteriormente. Nota-se a relação 1 para N entre o Pedido e Cliente, sendo que toda ordem deve estar obrigatoriamente relacionada a um consumidor. Nota-se que a ordem possui os atributos de data de recebimento, se está paga (estaPago), número e seu preço, e pode-se também identificar seus tipos. É possível observar também os métodos que promovem a interação com essas classes, sendo que para a classe Pedido tem-se os métodos despachar enviar e fechar.

Já na classe Cliente, tem-se um exemplo de generalização, em que existem os atributos comuns a clientes corporativos e a clientes individuais. Os atributos e métodos que dizem respeito especificamente a cada tipo de consumidor ficam em suas respectivas classes.

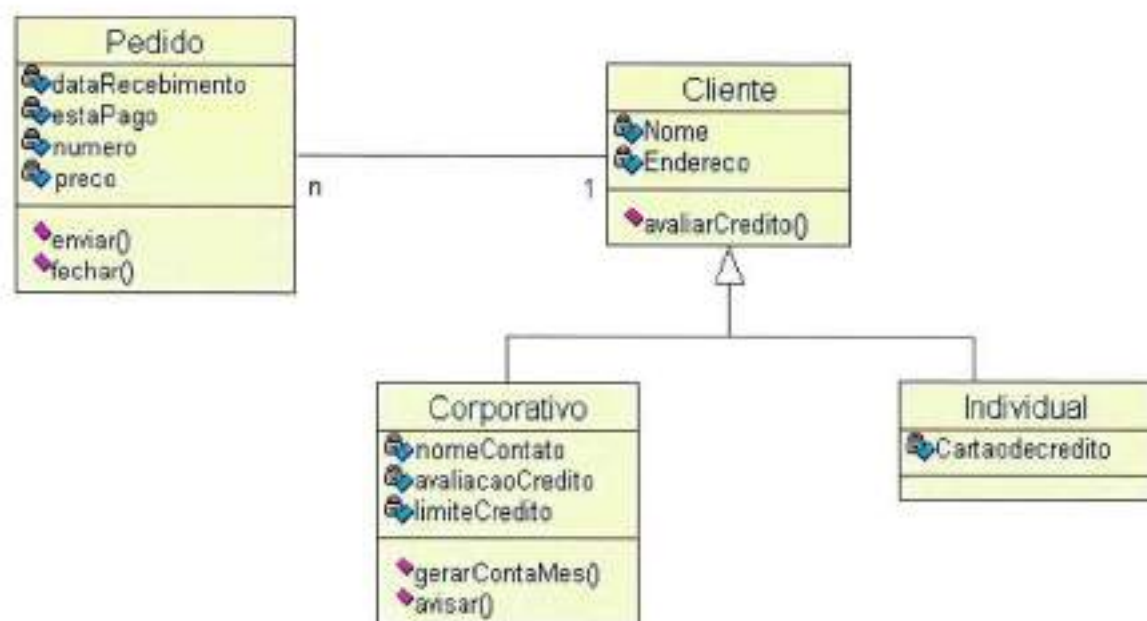


Figura 4 – exemplo de diagrama de classe

Fonte: Adaptado de UML Tutorial, 2006 , tradução dos autores

3.2.4 Diagrama de Seqüência

A finalidade do diagrama de seqüência é mostrar a ordem em que os objetos interagem no sistema. Através de sua confecção é possível entender de que forma se dará a interação entre os objetos, quais métodos serão necessários para essa interação e quais informações são passadas. Até o corpo técnico de um projeto pode fazer uso de tais diagramas já que eles descrevem qual é a seqüência de acontecimentos que torna a execução de tarefas. (BELL, 2006c)

Na Figura 5, pode-se observar a interação que ocorre entre os objetos analista, sistema e segSistema. Na vertical tem-se a linha do tempo, enquanto na horizontal representa-se o fluxo de mensagens entre os objetos.

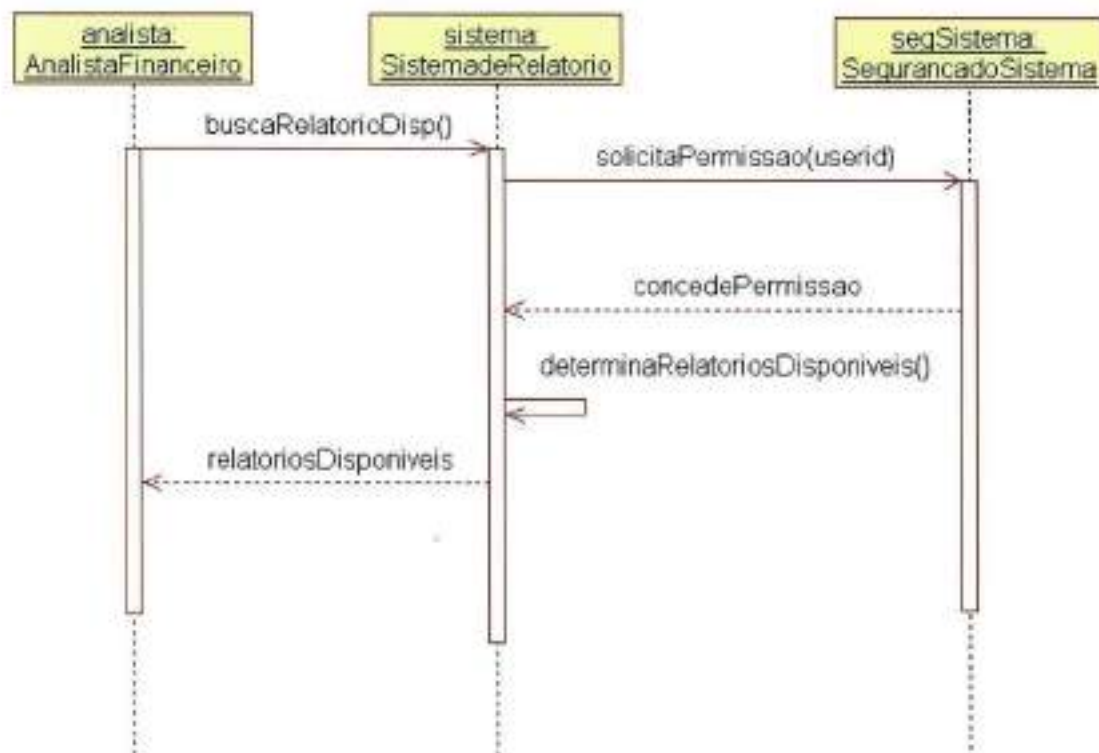


Figura 5 – Exemplo de diagrama de seqüência

Fonte: Adaptado de BELL, 2006c, tradução dos autores

3.3 Etapa de Projeto e Implementação

A terceira fase engloba o projeto e a implementação do sistema. Como produtos dessa fase temos, por exemplo, elementos de banco de dados, captação de registros, os módulos operacionais de presença. Nessa fase também é feita a integração dos módulos e os testes.

Para implementação dos módulos operacionais, que envolvem a captação e recepção de registros e o módulo de acompanhamento de presença, a escolha foi a utilização da estrutura MVC (*Model View Controller*). Os motivos para a escolha desse padrão foram que ele promove o desenvolvimento do sistema em camadas, proporciona um maior desacoplamento e possibilitando, portanto, o desenvolvimento modular do sistema.

3.3.1 Estrutura Model View Controller

A estrutura de desenvolvimento *Model View Controller* (ou *MVC pattern*) (SUN, 2006b) consiste na separação do *software* em três camadas: lógica do sistema (*model*), de seu controle (*controller*) e sua visualização (*view*). A origem dessa arquitetura é creditada a linguagem *Smalltalk*, que era usada para mapear processos de entrada, saída e processamento na interface gráfica e visava uma abordagem mais simples para desenvolvimento de *software*. O que se notou foi que tais conceitos são muito úteis e facilmente transferidos para aplicações multi-usuário, sendo

possível através de um mesmo *model* atender a demandas diferentes através da customização da *view* e *controller*.

O objetivo principal dessa divisão é permitir que diversos usuários compartilhem através de suas visualizações a mesma lógica (*model*), simplificando o desenvolvimento, já que a mesma lógica é utilizada por todas as visualizações o que reduz a quantidade de testes e conseqüentemente o custo de desenvolvimento. A estrutura MVC é exibida na Figura 6.

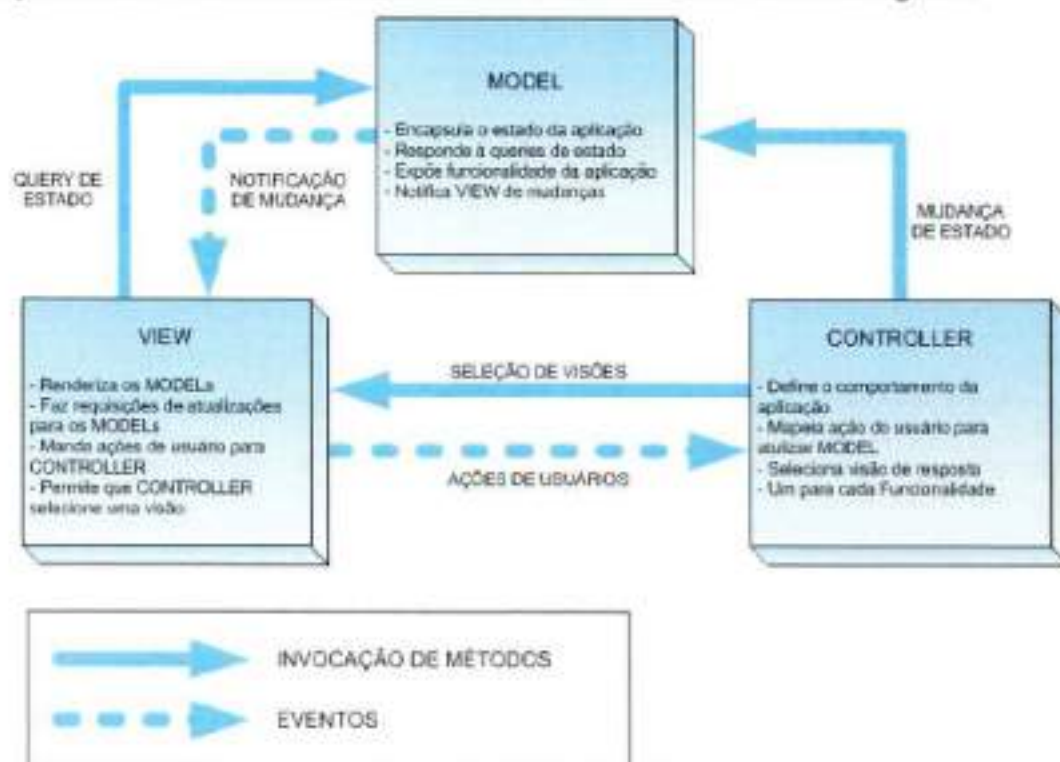


Figura 6 – Arquitetura Model View Controller

Fonte: Adaptado de SUN, 2006b, tradução dos autores

3.3.2 Model

A camada *model* define as regras de negócio, sem se preocupar com o modo como o usuário vai interagir com essa lógica ou mesmo como os dados resultantes de tais interações deverão ser representados. Isso provê uma grande capacidade de reutilização de código e componentes dado que um mesmo *model* é usado por diversos usuários e representado de diversas maneiras, que inclusive podem ser adicionadas e modificadas sem que haja qualquer prejuízo para a lógica de negócio.

3.3.3 View

O papel da camada *view* é apresentar os dados que acessa através da camada lógica e manter consistência quando aos dados exibidos. Isso significa que se houver qualquer mudança nas informações armazenadas no *model*, a camada *view*, deve então consultar quais

informações forma alteradas e atualizá-las. Para tanto a camada *view* registra-se como uma ouvinte da *controller*.

3.3.4 Controller

A camada *controller* é responsável por transferir as ações e realizar a iteração entre o usuário e as camadas *view* e *model*. Tais ações podem consistir desde ativação de um processo do negócio até a mudança do estado da *model*.

Dado o *pattern* MVC (SUN, 2006b), é relativamente simples modificar a *controller* para que novas formas de interação sejam adicionadas incorrendo em apenas uma fração do custo total de se modificar um *software* que trabalhe com as três camadas mencionadas de forma não independente.

3.3.5 Arquiteturas Model 1 e Model 2

Há duas arquiteturas predominantes no desenvolvimento de aplicações web: *model* 1 (Figura 7) e *model* 2 (Figura 8). A diferença principal entre as arquiteturas se refere ao modo como as requisições são tratadas. Enquanto na arquitetura *model* 1 o mesmo elemento é responsável por receber as requisições e exibir os dados para o usuário (*view+controller*), na estrutura *model* 2, que implementa o MVC *pattern* (SUN, 2006b), há uma entidade específica para tratar das requisições do usuário (*controller*) e para tratar da exibição dos dados (*view*) (ORACLE, 2006).

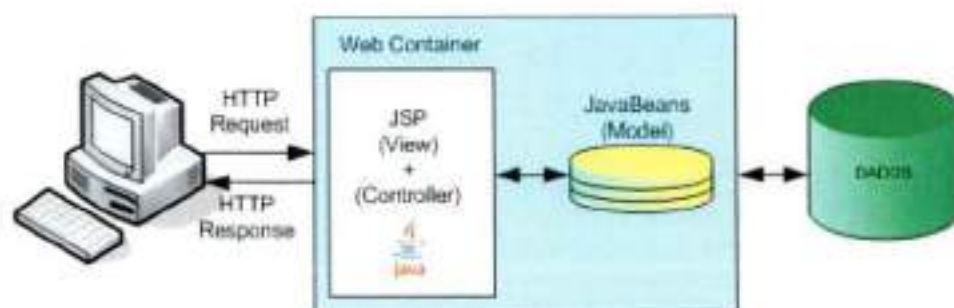


Figura 7 – Arquitetura model 1

Fonte: Adaptado de ORACLE, 2006, tradução dos autores

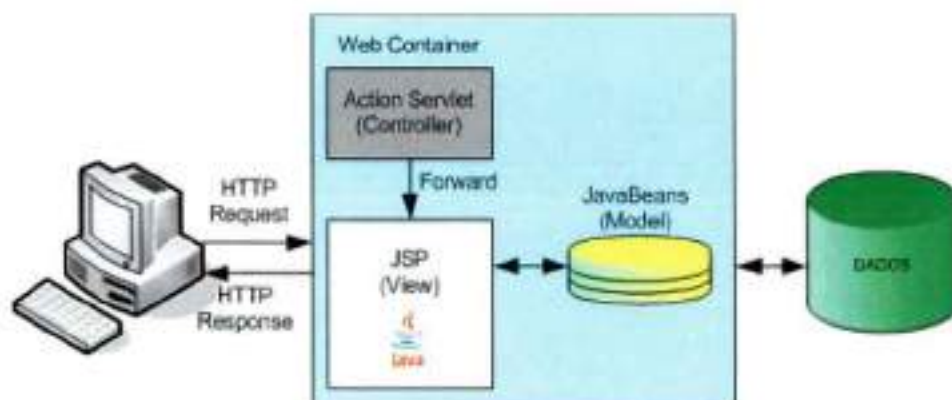


Figura 8 – Arquitetura model 2

Fonte: Adaptado de ORACLE, 2006, tradução dos autores

Optou-se por usar a arquitetura *model 2* devido ao fato de que embora pareça mais complicado à primeira vista por adicionar uma camada extra ao processo, na verdade ocorre uma simplificação já que é possível desenvolver camadas com menos funções, resultando em menor complexidade e manutenibilidade.

4 MODELAGEM DO PROCESSO DE NEGÓCIO

Esse capítulo apresenta o processo atual de acompanhamento e registro da presença acadêmica e, depois, apresenta o processo novo com a introdução do sistema SARPA.

4.1 Processo Atual

O processo atual, apesar de apresentar alguns processos já automatizados, tem seu núcleo – registro de presença – ainda num processo muito burocrático como ficará evidenciado a seguir. O processo atual é dividido em, basicamente, cinco fases principais: Elaboração da Grade Horária, Matrícula, Emissão das Listas, Contabilização das Frequências e Lançamento das Frequências no Sistema. Para que fosse entendido o processo atual, consultou-se o Prof. Dr. Paulo Sérgio Cugnasca, coordenador de graduação do Departamento de Engenharia de Computação e Sistemas Digitais da EP.

4.1.1 Elaboração da Grade Horária

A grade horária de um determinado período – semestre ou quadrimestre – é elaborada com antecedência de dois ou três meses e então é lançada no sistema Júpiter para todas ênfases. Somente o curso cooperativo tem suas disciplinas lançadas na *intranet* da EP. O sistema Júpiter tem nele cadastradas todas as disciplinas que já foram, estão sendo e que deverão ser cursadas de todos os alunos do USP. Nesse sistema também se encontram as notas e as frequências dos alunos em cada uma das disciplinas e seu status (como aprovado, recuperação ou reprovado) referente àquela disciplina. Os alunos cadastrados podem acessar essas informações através da internet e, toda vez que uma nota/frequência é cadastrada, o aluno é notificado via e-mail.

As grades são elaboradas conforme a disponibilidade de salas, professores e do número de alunos em cada turma. Essa dinâmica ocorre através da interação entre os diferentes departamentos que ministram aulas para um determinado curso e a coordenação desse curso. A Figura 9 ilustra esse processo para a grade de Engenharia da Computação:

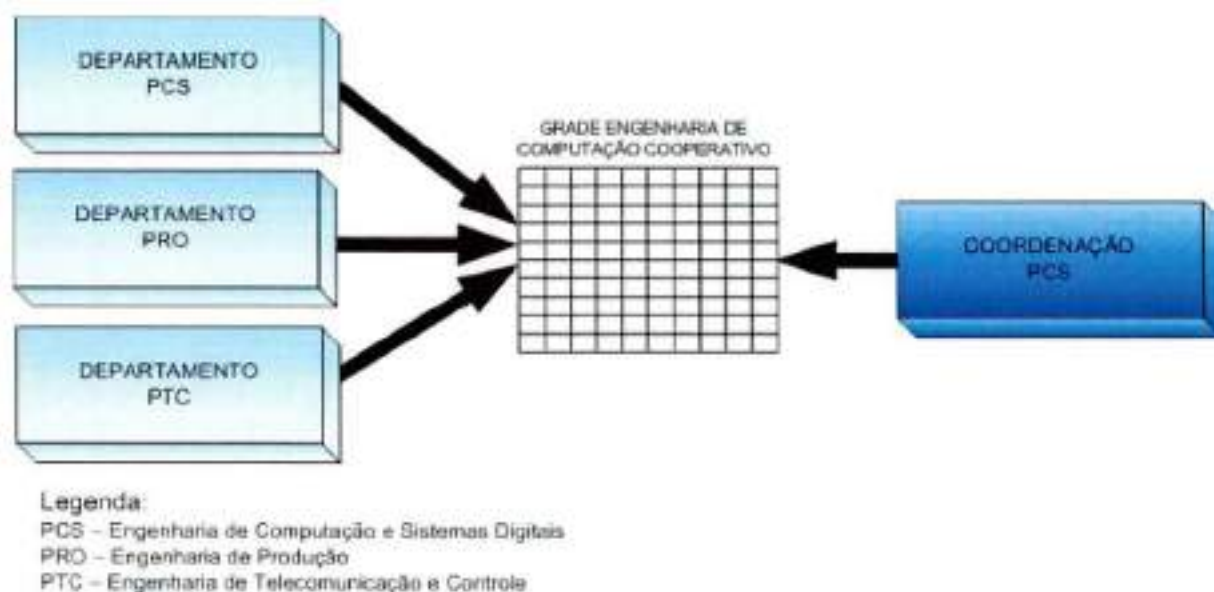


Figura 9 – Elaboração da Grade Horária de um curso

4.1.2 Matrícula

Após as grades se tornarem disponíveis, os alunos as verificam e escolhem a ordem de preferência pelas diferentes grades disponíveis. Ao término do período de matrícula, essas informações são compiladas e transmitidas ao sistema Júpiter, que faz então uma verificação dos pré-requisitos necessários para cursar as matérias nas quais o aluno se matriculou e faz também uma verificação de conflitos de horário. As exceções de pré-requisitos são resolvidas, geralmente, nos departamentos e as de conflitos na grade são resolvidos na coordenação do curso.

4.1.3 Emissão das Listas

Após os alunos estarem efetivamente matriculados em suas turmas, começa o processo de emissão das listas de chamada, que tem sua distribuição feita de duas maneiras distintas: a secretaria do departamento imprime as listas e as entrega aos professores antes de suas aulas ou os próprios professores podem imprimi-las através do sistema web do Júpiter. Um exemplo de como é a lista de presença no processo atual se encontra no Anexo B – Modelo de Lista de Presença Atual.

Após o professor estar com a lista impressa ele a leva, em todas as aulas que ministra e pode fazer o registro da presença de duas maneiras: fazendo chamada oral ou passando a lista para os alunos assinarem. Esse segundo processo, que é o mais comum, apresenta um nível de confiança muito baixo, devido à alta taxa de alunos que assinam para os colegas ausentes e da falta de tempo dos professores de conferir a autenticidade das assinaturas nas listas.

4.1.4 Contabilização das Frequências

No final do período, as frequências devem ser contabilizadas para o cálculo da frequência global que, para ser obtida a aprovação nos cursos da USP, é de 70%. Isso ocorre também de duas maneiras: o professor contabiliza as frequências por um controle próprio, ou então entrega suas listas – semanalmente ou ao fim do curso – para a secretaria contabilizá-las. Esse segundo processo normalmente ocorre quando o professor tem múltiplas turmas ou turmas muito grandes e seu trabalho de contabilização fica muito custoso.

4.1.5 Lançamento das Frequências

Após a contabilização há duas maneiras da frequência ser lançada, em conjunto com as notas, no sistema Júpiter. A secretaria as lança através de seu sistema administrativo ou o próprio professor através do sistema web do Júpiter.

4.2 Processo Proposto

Esse projeto propõe alterações significativas em duas fases do processo atual: Emissão de Listas e Contabilização de Frequências. As outras fases do processo atual não serão alteradas pelo sistema SARPA, visto que o sistema tem seus próprios módulos de cadastro de disciplinas, para elaboração da grade horária. Seria desejável uma integração futura do SARPA com o sistema Júpiter, porém isso não faz parte do escopo desse projeto.

Além das alterações em duas fases do processo atual, o sistema SARPA prevê funcionalidades não presentes atualmente, como: acompanhamento aula a aula da frequência por alunos e professores e *feedback* estatístico da frequência para o professor.

Sendo assim, no processo proposto tem-se, basicamente, quatro fases principais: Elaboração da Grade Horária e Matrícula, Registro de Presença, Contabilização das Frequências, Lançamento das Frequências. Além disso, apresenta dois recursos que não existem no processo atual: Acompanhamento da Presença e *Feedback* Estatístico.

4.2.1 Elaboração da Grade Horária e Matrícula

Esse processo é um espelho do cadastro de turmas do Júpiter. O sistema SARPA oferece interface para o cadastramento de alunos, disciplinas, turmas, salas e professores, para que possa ser feita a alocação dinâmica desses elementos conforme necessitado pelas diferentes grades de cada ênfase. Em uma versão futura, esse processo, idealmente, poderia ser integrado ao sistema Júpiter, porém como foi dito anteriormente, isso não faz parte do escopo deste projeto.

4.2.2 Registro de Presença

O registro de presença é feito pelo aluno em todas as aulas através de uma leitora de cartões magnéticos encontrada do lado de dentro de cada sala de aula, ao lado da porta. Ao início de cada aula, com uma tolerância de 10 minutos antes do horário até o fim de cada aula, com uma tolerância de 10 minutos após, o aluno deverá passar o cartão para que possa ser contabilizada uma porcentagem de tempo em que ele assistiu à determinada aula. Ele receberá presença nessa aula se essa porcentagem for igual ou superior à uma porcentagem determinada pelo professor de cada disciplina, variando entre 50% e 85%.

Esse é o processo que configura uma das maiores mudanças em relação ao processo atual, pois além da automatização, oferece uma presença com um grau de confiança maior, pois, a dificuldade de se burlar a presença em um sistema com verificação via carteira USP é maior do que numa lista escrita. Inclusive, essa é a opinião da maioria dos próprios alunos, conforme resultado de uma pesquisa feita durante a elicitação de requisitos desse projeto, conforme é exibido no Capítulo 5.

Além disso, o sistema permite recálculo de presenças para alunos específicos ou em datas específicas. Isso foi feito, pensando-se nas mudanças – horário, dia, sala – que normalmente ocorrem em cursos da EP ou quaisquer outros cursos universitários.

4.2.3 Contabilização das Frequências

A contabilização das frequências é feita automaticamente pelo sistema e pode ser consultada a qualquer momento ao decorrer da disciplina. Essa automatização otimiza a tarefa de contabilização feita no sistema atual pelos professores ou pela secretaria de cada departamento. É importante ressaltar que com esse processo automatizado, o lançamento de frequências ao fim do período será mais rápido, tornando possível a disponibilização das notas e frequências mais cedo para os alunos.

4.2.4 Lançamento das Frequências

O lançamento pode ser feito no sistema Júpiter a partir desses resultados contabilizados no sistemas SARPA. Tanto a secretaria como o professor pode lançar as frequências no sistema Júpiter. Novamente, nesse ponto, o ideal seria uma integração com o sistema Júpiter, mas essa integração não está prevista nesse projeto.

4.2.5 Acompanhamento da Presença

O acompanhamento de presenças também evidencia uma das diferenças essenciais entre o processo atual e o proposto, pois no processo atual isso só ocorre em casos raríssimos em que o professor contabiliza a frequência periodicamente e disponibiliza para a sala. Mesmo nesses casos, ainda há um problema de privacidade dos alunos, que é violada quando é

divulgada a lista para toda sala. No processo proposto cada aluno só terá acesso à sua porcentagem de presença em cada disciplina.

Com esse acompanhamento, o aluno tem maior controle sobre a sua frequência e se ela está dentro ou não do mínimo requerido para aprovação (70%). Além disso, o aluno pode acompanhar a quais aulas faltou e que conteúdo perdeu, aula a aula no sistema. Já o professor tem uma visão geral das frequências da sala e pode observar quais alunos estão com uma frequência abaixo da mínima necessária e acompanhar esses casos mais de perto.

4.2.6 Feedback Estatístico

Essa é outra grande vantagem em relação ao processo atual. O professor tem um feedback da frequência em suas aulas para tentar, assim, descobrir qual o comportamento das suas turmas frente às aulas ministradas e tendo a possibilidade de comparar graficamente as frequências com a de outros professores, matérias ou períodos, configurando, assim, uma ferramenta de avaliação nunca antes tida pelos professores da EP.

5 ELICITAÇÃO DE REQUISITOS

5.1 Introdução

Devido à particularidade do papel que construtor de regras validadoras de presença exerce sobre o sistema, inicialmente existem múltiplas possibilidades para o desenvolvimento desse módulo. A sua importância é evidente, pois ele manipula o atributo chave do sistema, a presença, e, portanto, desenvolvê-lo sem uma melhor especificação de seus requisitos, com participação de seus principais usuários – os professores e os alunos – seria imprudente, pois poderia comprometer o sistema como um todo.

Portanto, foi decidido que se adotasse uma abordagem diferente, através da utilização de ferramentas que permitissem um levantamento consistente dos requisitos desse módulo junto a seus principais usuários. Na busca de técnicas para alcançar tal objetivo, deparou-se com o conceito de Engenharia de Requisitos e de Técnicas de Elicitação de Requisitos.

5.2 Engenharia de Requisitos e Técnicas de Elicitação de Requisitos

Como visto no Capítulo 2, Item 2.1.6, o uso de técnicas de levantamento de requisitos na fase de especificação de um sistema complexo de *software* é essencial e, portanto, nessa fase decidiu-se utilizar algumas das práticas da Engenharia de Requisitos.

Há um conjunto de práticas que podem ser aplicadas denominadas técnicas de elicitação. A Tabela 1, a seguir, mostra um conjunto dessas técnicas e qual delas produz, de maneira mais eficiente ou não, cada quesito elicítavel. Quanto mais escuro o sombreamento, mais eficaz é a técnica para produzir o quesito correspondente.

TABELA 1 – Técnicas de Elicitação x Quesitos Elicitáveis

	Quesitos Elicitáveis	Trabalho Atual	Problemas Atuais	Objetivos e Assuntos Chave	Idéias de Sistemas Futuros	Possibilidades Realistas	Consequências e Riscos	Comprometimento	Resolução de Conflitos	Requisitos	Prioridades	Integridade
Técnicas												
Análise de Usuários												
Entrevista em Grupo												
Observação												
Demonstração de Tarefas												
Estudo de Documentação												
Questionários												
Brainstorm												
Grupos de foco												
Workshops de domínio												
Workshops de design												
Prototipação												
Experimentos Piloto												
Empresas Similares												
Perguntas aos Fornecedores												
Negociação												
Análise de Risco												
Custo/Benefício												
Análise de objetivo-domínio												
Análise de requisito-domínio												

Fonte: Adaptado de LAVESEN, 2002, p. 338, tradução dos autores.

Cruzando esses quesitos com as técnicas é possível fazer uma análise de quais técnicas são mais adequadas. Porém, logicamente, é necessário fazer uma análise de quais técnicas são viáveis para determinado levantamento. É importante observar, também, que muitas técnicas que aparentam ser adequadas para o quesito, são técnicas aplicadas a sistemas já existentes, o que pode inviabilizar seu uso em levantamento de sistemas que ainda não se encontram implementados.

5.3 Escolha da Técnica de Elicitação

Com base na Tabela 1, inicialmente pode-se pensar que o quesito "Objetivos e Assuntos Chave", atenderia à necessidade, porém o objetivo do construtor de regras e o seu papel no sistema já estão bem definidos. Então, foram rapidamente identificados quais quesitos atendem à necessidade do projeto, que é especificar e desenvolver corretamente o construtor de regras. Esses quesitos são "Idéias de Sistemas Futuros" e "Requisitos".

Do cruzamento entre quesitos e técnicas foi possível identificar que as técnicas que melhor atendem à necessidade são as de "Entrevista" e "Questionário". Foi possível, então, traçar um comparativo entre essas técnicas para definir qual delas era a mais adequada para alcançar o objetivo desejado.

Em linhas gerais, um questionário pode se tornar equivalente a uma entrevista, com a desvantagem de não ter a dinâmica que se tem numa entrevista e a possibilidade de melhor direcioná-la para obter os resultados desejados. Por esse motivo, as perguntas devem ser muito bem elaboradas, de forma a deixar o entrevistado com o mínimo de dúvidas sobre uma ou mais perguntas.

A vantagem de ser feito um questionário é que o número de pessoas que contribuem com respostas geralmente é maior, pois é mais simples enviar questionários e compilar respostas do que organizar entrevistas, individuais ou em grupos, onde há sempre o problema de agendas conflitantes.

Desse modo chegou-se à conclusão de que o questionário era a técnica de elicitação que mais se adequava ao levantamento.

5.4 Técnicas de Elaboração de Questionários

Existem diversas técnicas e alguns pontos de decisão na elaboração do questionário. O primeiro é a definição do(s) grupo(s) sobre qual(is) será aplicado o questionário. Idealmente, é preciso "entrevistar alguns membros de cada grupo de usuários" (LAVESSEN, 2002, p. 338, tradução dos autores).

Antes da aplicação do questionário, é necessário que haja uma sucinta, porém coesa, descrição do sistema e enunciação dos objetivos do questionário, pois "dizer para os usuários: 'Diga-me o que você quer' é pouco provável de resultar em informação útil. Pessoas acham muito mais fácil falar sobre um contexto definido do que em termos gerais". (KOTONYA, 1998, p.63, tradução dos autores).

Quanto às questões, elas devem ser de fácil e claro entendimento e podem ser fechadas (alternativa) ou abertas (dissertativa). A vantagem de questões fechadas é que se torna mais fácil compilar e contabilizar os resultados, porém elas podem deixar de coletar outras visões e opiniões importantes de usuários, não previstas pelo questionário inicial.

Outra técnica bastante utilizada é de começar o questionário com questões simples, que não necessariamente tem impacto sobre o que se espera extrair do questionário, mas com o intuito de familiarizar a pessoa que o está respondendo e deixá-la à vontade.

É altamente indicado também que se faça, antes de enviar o questionário à um número grande de pessoas, um teste de conformidade com um número reduzido de pessoas, para avaliar se as respostas e a compreensão das questões foi dada dentro do esperado.

5.5 Elaboração e Aplicação do Questionário

Tem-se três grupos principais que são usuários-chave do sistema: professores, alunos e funcionários. Os diretamente afetados pelo construtor são os professores e os alunos, portanto o foco é sobre eles.

Logicamente, o ponto de vista de alunos e professores sobre o sistema pode diferir em diversos pontos e, por isso, foi decidido fazer um questionário para cada grupo. Esses questionários encontram-se no Apêndice A – Modelos de Questionários Utilizados.

Para tentar obter uma resposta melhor direcionada das pessoas que responderam ao questionário foi preparado um breve texto, antecedendo o questionário explicando o objetivo do sistema e do questionário.

Foi decidido inicialmente que as questões seriam abertas, pois apesar de o trabalho na compilação dos resultados ser maior nesse tipo de questionário, achou-se que era importante dar espaço para comentários mais amplos, para que nenhum aspecto – negativo ou positivo – do sistema, na percepção de alunos e professores, ficasse sem espaço para ser comentado. Seguindo essa mesma linha, deixou-se um espaço aberto para comentários sobre quaisquer aspectos, relacionados ao sistema, que fossem considerados relevantes.

Foi utilizada técnica de usar questões simples no início do questionário, para deixar alunos e professores à vontade. Também foram utilizadas perguntas polêmicas – no caso, sobre a concordância com a presença obrigatória – para despertar o interesse em responder o questionário.

Por fim, foi aplicado o questionário a apenas um aluno e um professor, escolhidos a dedo, a fim de avaliar o entendimento do questionário e o conteúdo das respostas. A partir das respostas dessa fase, foi observado que algumas questões tinham domínio muito bem definido e poderiam, sem prejudicar o resultado, ser transformadas em questões fechadas. Foi feita uma revisão também do cabeçalho do questionário, que explicava os objetivos do questionário e do sistema, pois foi detectada uma certa dificuldade de interpretação desse cabeçalho.

No final, obteve-se um questionário muito simples e rápido de responder – cerca de 5 minutos – e que atendeu os objetivos almejados. O questionário final foi enviado a diversos alunos e professores. Observou-se, inicialmente, que não foram obtidas respostas para muitos dos questionários enviados. Esse fato já era esperado e algumas mensagens reforçando o pedido foram necessárias para que o questionário fosse respondido. Foram recebidas as respostas de 20 alunos 5º

anistas da Engenharia de Computação Cooperativo e 10 professores de variados departamentos da EP.

5.6 Análise dos Resultados Obtidos

A seguir são apresenta-se, através da análise gráfica e estatística, os resultados obtidos com o questionário feito com alunos e professores. Alguns exemplos de questionários respondidos se encontram no Anexo A – Exemplos de Respostas Obtidas aos Questionários.

É importante observar que algumas respostas, devido ao caráter aberto das questões, podem ter influência em duas porcentagens diferentes. Por exemplo, quando perguntado qual a vantagem do sistema, um mesmo indivíduo pode responder que é evitar fraudes e auxiliar o professor com estatísticas sobre sua disciplina. Foi considerado como se fossem duas respostas diferentes que influenciam no resultado estatístico adquirido.

5.6.1 Perfil dos Alunos e Professores

Essas são respostas às perguntas feitas para descontrair os indivíduos em relação ao questionário. Elas não têm influência direta sobre o resultado, mas mostram um dado interessante sobre o perfil desses indivíduos.

Aos professores, perguntou-se há quanto tempo atuava como professor na EP. O resultado se apresenta no Gráfico 1.

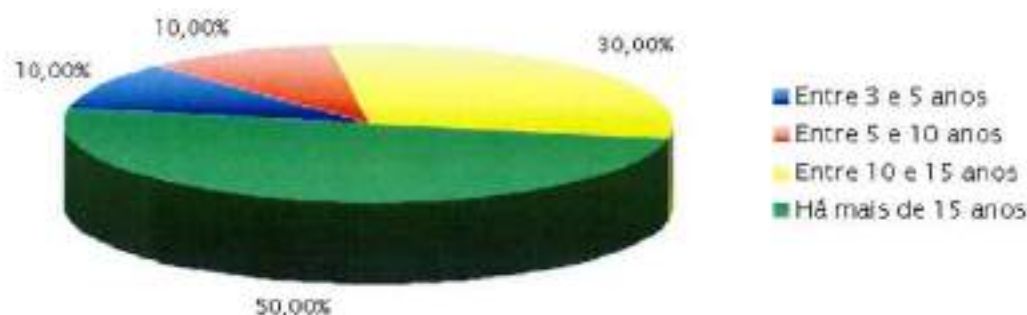


Gráfico 1 – Tempo que professores lecionam na EP

Foi também perguntado, quanto tempo lecionava dentro ou fora da EP. O resultado se apresenta no Gráfico 2.

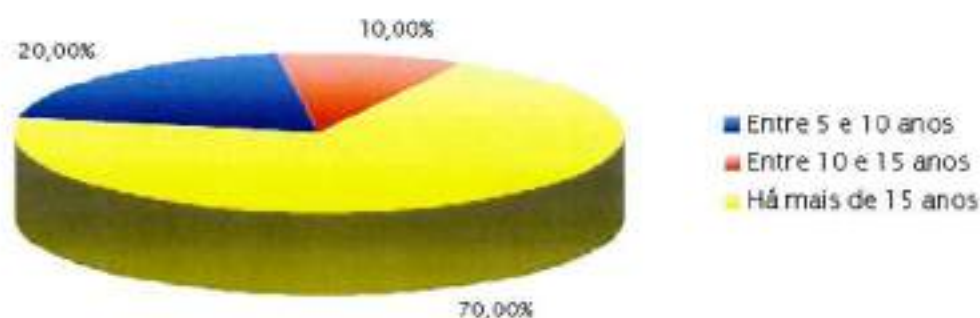


Gráfico 2 – Tempo que professores lecionam

Pode-se ver pelos resultados que os professores que responderam ao questionário são em geral muito experientes, tendo em sua grande maioria (80%) ao menos 10 anos de experiência.

Nessa mesma linha de perguntas para descontrair, foi perguntado aos alunos com qual frequência eles costumam assistir às aulas. O resultado se apresenta no Gráfico 3.



Gráfico 3 – Frequência com que os alunos costumam assistir às aulas

Já por esse gráfico pode-se observar que grande parte dos alunos costuma assistir às aulas frequentemente.

Então, apesar dessas respostas não terem influência direta sobre o resultado esperado do questionário, observa-se que ambos grupos que participaram da pesquisa são grupo engajados e comprometidos com as aulas lecionadas na EP.

5.6.2 Opiniões Sobre a Presença Obrigatória

Essa pergunta segue a linha da pergunta polêmica para despertar o interesse dos grupos em responder o questionário.

Foi feita a mesma pergunta para alunos e professores, sobre a concordância com a presença obrigatória e os motivos que o levam a essa opinião. Primeiramente, pode-se observar os resultados dos professores nos Gráficos 4 e 5, respectivamente.



Gráfico 4 – Professores concordam com a presença obrigatória?

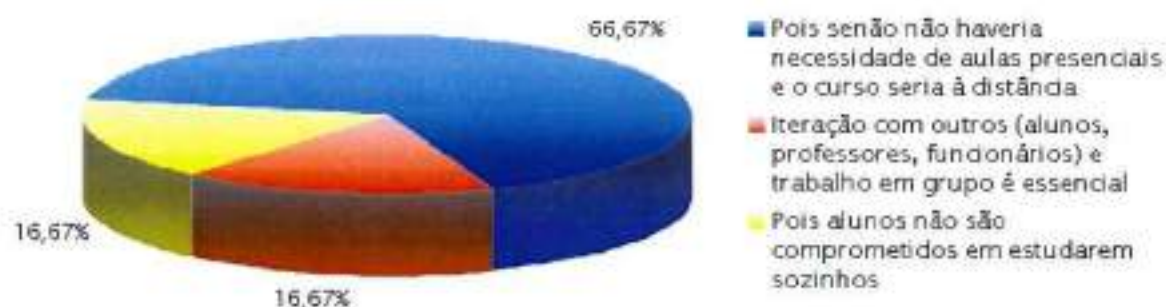


Gráfico 5 – Motivos por que professores concordam com a presença obrigatória

Só 10% dos professores não concordam com a presença, pois acredita que a presença à aula deve ser incentivada com uma aula interessante e não obrigatória e outros 10% concordam em partes pelo mesmo motivo, porém têm consciência que esse processo de conscientização da importância das aulas não é algo que pode ser resolvido facilmente, no presente. Observa-se também que a maioria dos professores concorda com a obrigatoriedade da presença, pois acredita que se o curso é presencial, o aluno tem que comparecer.

Observa-se, agora, os resultados dos alunos à mesma pergunta no Gráfico 6.

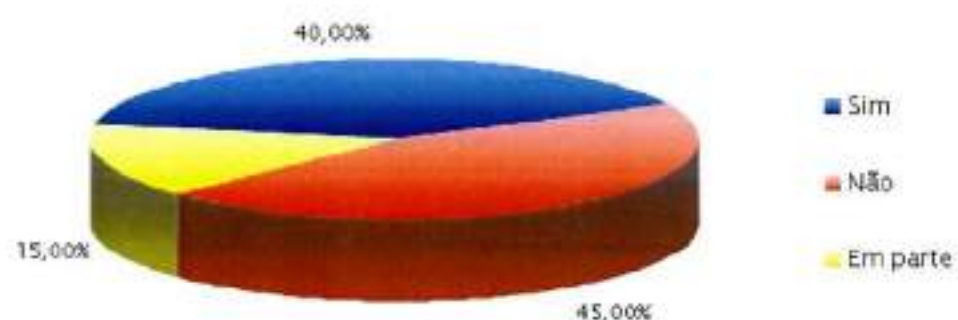


Gráfico 6 - Alunos concordam com a presença obrigatória?

Como era esperado, há um equilíbrio maior entre as opiniões dos alunos, com uma ligeira vantagem da não concordância com a presença obrigatória. Pode-se observar o porquê nos Gráficos 7 a 9.



Gráfico 7 - Motivos por que alunos concordam com a presença obrigatória

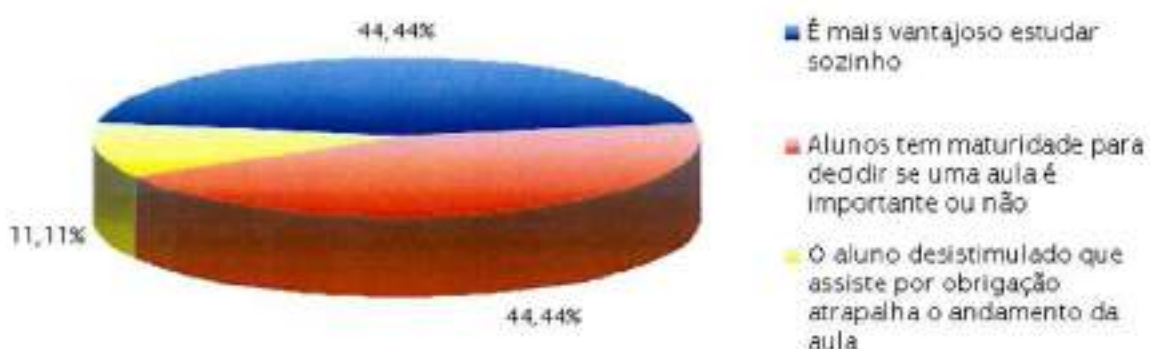


Gráfico 8 - Motivos por que alunos não concordam com a presença obrigatória

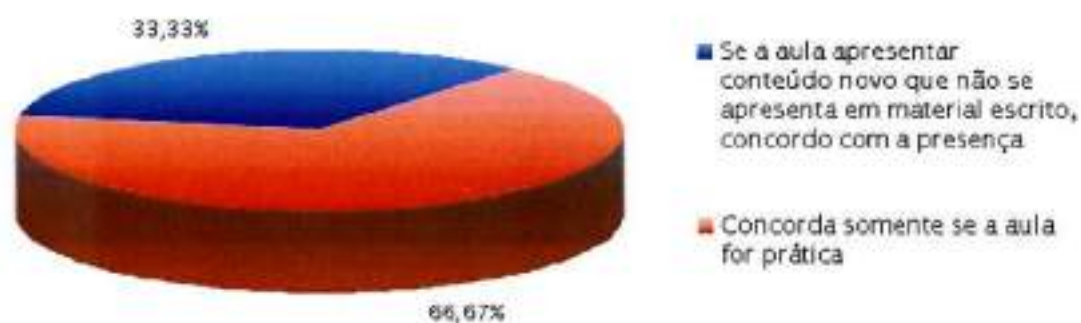


Gráfico 9 – Motivos por que alunos concordam em parte com a presença obrigatória

O que se observa desses resultados é uma oposição de idéias entre os alunos sobre o mesmo tema. Por um lado, os alunos que concordam com a presença obrigatória – mesmo em parte – acreditam que a aula tem algo a mais para acrescentar do que os livros, tanto refletido na experiência dos professores, como em aulas práticas, que têm dificuldade de serem reproduzidas à distância. Já por outro lado, alunos que discordam da presença obrigatória, acreditam ser mais vantajoso estudar sozinho ou selecionar as aulas que assiste ou não, devido justamente ao conteúdo pouco interessante ou que pode ser encontrado em fontes escritas com o mesmo aproveitamento.

O SARPA, no entanto, não entrará no mérito do conteúdo das aulas. Ele somente controla a presença e dá resultados estatísticos para apoio ao professor baseado nela. Era importante então, nesse sentido, entender se os grupos acreditam que a presença obrigatória é válida.

5.6.3 Vantagens e Desvantagens do Sistema

Foi perguntado a ambos os grupos, com uma sintaxe ligeiramente diferente, quais seriam as vantagens e desvantagens enxergadas por eles que o sistema apresentaria em relação ao processo atual. Primeiramente, pode-se observar o resultado à pergunta para os professores sobre a concordância ou não em que o processo proposto traria melhoria em relação ao processo atual, nos Gráficos 10 e 11.



Gráfico 10 – O sistema proposto, segundo professores, traria melhoria em relação ao processo atual?

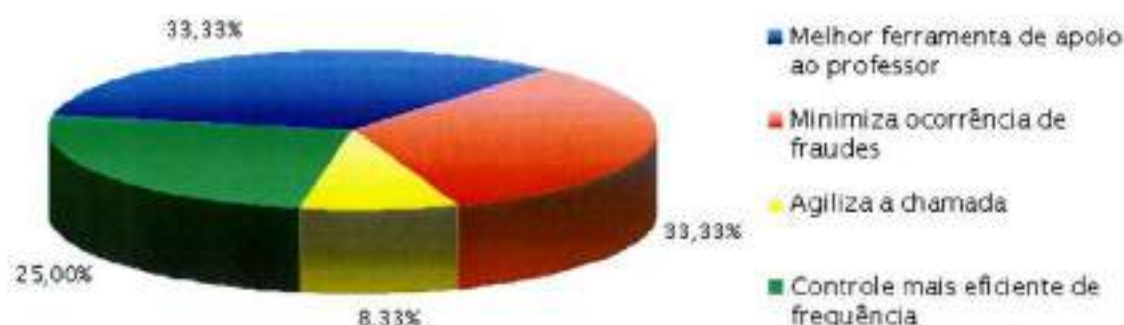


Gráfico 11 – Motivos por que professores acreditam que o novo processo traria melhoria

Ainda foi possível observar dos 20% de professores que não acreditam que o sistema traria melhoria, metade acredita que a chamada é mais eficiente e a outra metade simplesmente discorda da presença obrigatória, então a implementação do sistema seria um retrocesso no sentido de promover a consciência de assistir às aulas simplesmente por elas serem interessantes.

Para os alunos foi perguntado, mais diretamente, quais eram as vantagens e desvantagens enxergados por eles do sistema proposto em relação ao processo atual. Os resultados são observados nos Gráficos 12 e 13.

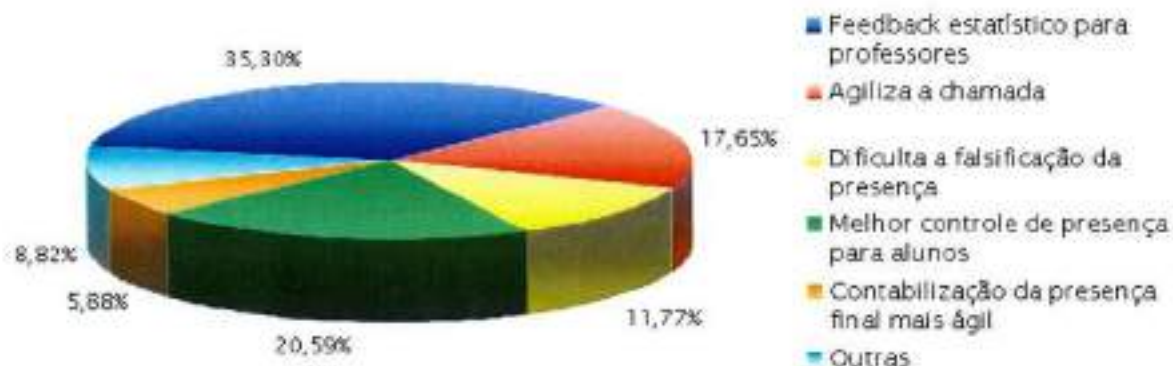


Gráfico 12 – Vantagens do sistema, segundo os alunos



Gráfico 13 – Desvantagens do sistema, segundo os alunos

Em ambos os grupos foi possível observar, então, que o fator mais indicado como vantagem do sistema, foi que esse seria uma ferramenta de auxílio estatístico ao professor. As desvantagens apontadas não afetam diretamente a especificação do sistema, porém foram levadas em consideração.

5.6.4 Integridade da Presença

Foi perguntado somente aos alunos – como potenciais burladores da integridade da presença – se o sistema novo dificultaria a falsificação da presença. Observa-se o resultado no Gráfico 14.



Gráfico 14 – O sistema proposto dificulta a falsificação da presença?

Apesar de aproximadamente apenas 12% das respostas à pergunta das vantagens do sistema terem sido "diminuir as falsificações da presença", pode-se observar que, quando perguntado diretamente, os alunos, na grande maioria – 75% – acreditam que a dificuldade de burlar a presença no sistema proposto é maior.

5.6.5 Regras Validadoras de Presença

Foi perguntado, para ambos os grupos, quais as regras que achavam válidas para validar a presença em uma aula. Observa-se o resultado de alunos e professores nos Gráficos 15 e 16.

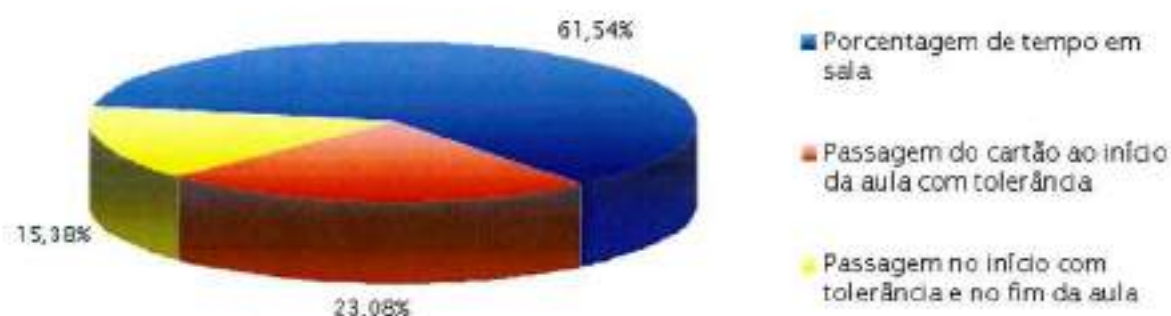


Gráfico 15 – Regras consideradas válidas por professores

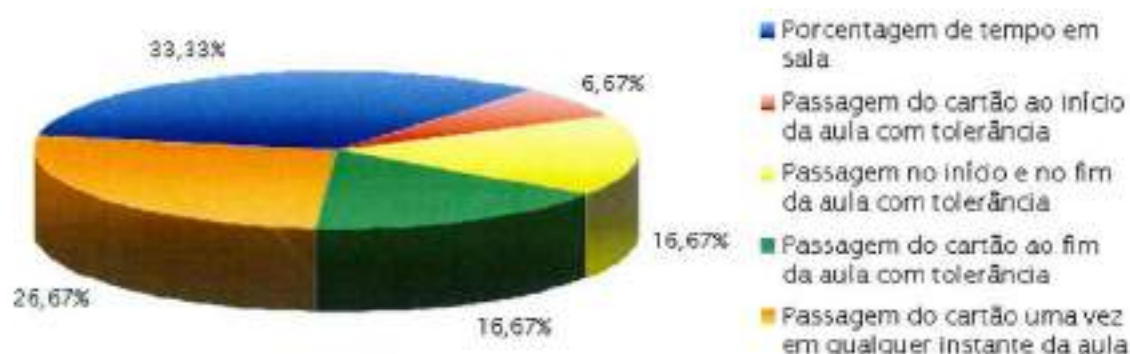


Gráfico 16 – Regras consideradas válidas por alunos

Pode-se observar que ambos os grupos apontam com mais frequência a porcentagem de tempo em sala de aula como método válido. Outros métodos como a passagem do cartão no início e no fim da aula com tolerância são, em essência, equivalentes à porcentagem de tempo em sala de aula.

5.6.6 Mau Uso do Sistema

Perguntou-se à professores como eles enxergavam que alunos podiam fazer mau uso do sistema e, por outro lado, à alunos como eles enxergavam um mau uso por parte dos professores. Observa-se o resultado de ambos grupos nos Gráficos 17 e 18.



Gráfico 17 – Como alunos podem fazer mau uso do sistema, segundo professores

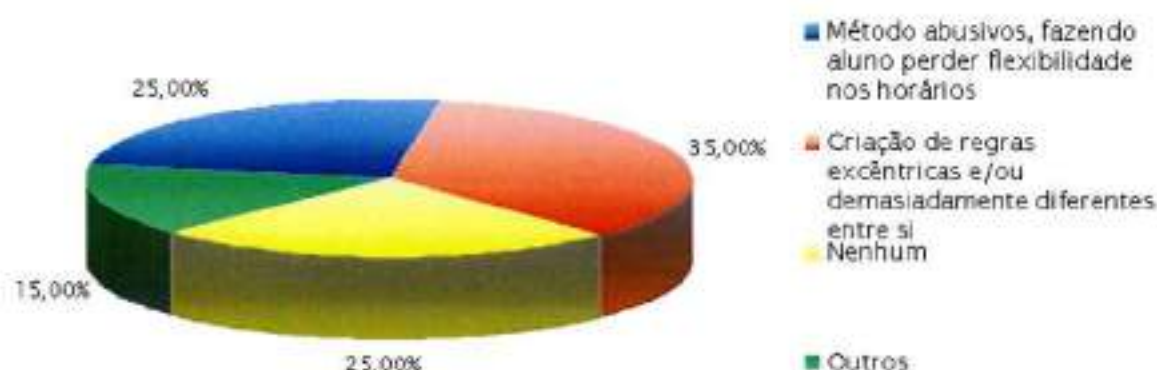


Gráfico 18 – Como professores podem fazer mau uso do sistema, segundo alunos

Os professores, logicamente, enxergam que os alunos podem burlar a presença passando a carteira para outros, porém os próprios alunos vêem dificuldade em falsificar a presença no sistema proposto, podendo aumentar ou diminuir o grau dessa dificuldade baseado no posicionamento da leitora de cartões.

Já a questão maior entre os alunos é a liberdade excessiva dada à criação de regras aos professores que pode dar margem à criação de regras excêntricas ou por demasiado diferentes entre si, tornando o processo de validação de presença mais complicado do que deveria ser. Ainda há a preocupação com métodos abusivos – como, por exemplo, exigir 100% de presença – que façam com que os alunos percam o mínimo de flexibilidade desejável.

5.6.7 Posição das Leitoras Magnéticas

Evidentemente, a posição das leitoras magnéticas influencia no andamento da aula e no grau de dificuldade de burlar a presença. Por isso, foi perguntado para ambos os grupos qual seria o lugar ideal das leitoras magnéticas. Observa-se o resultado de ambos os grupos nos Gráficos 19 e 20.

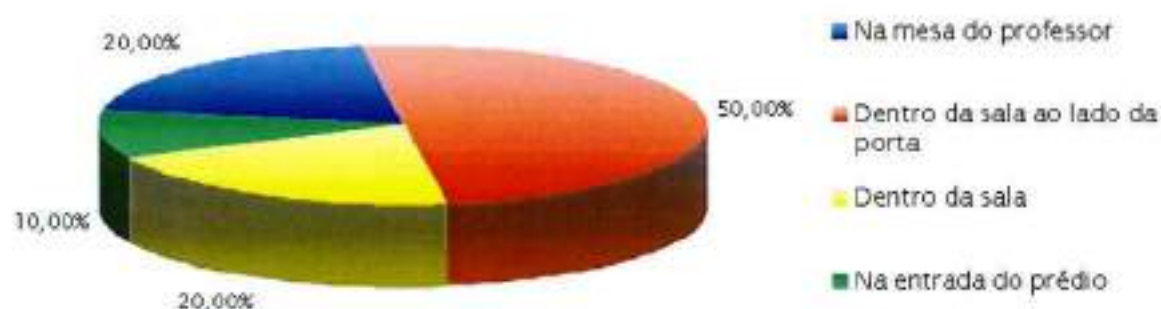


Gráfico 19 – Onde deveria ficar a leitora magnética segundo os professores



Gráfico 20 – Onde deveria ficar a leitora magnética segundo os alunos

Em ambos os grupos, a maioria absoluta acredita que deva ficar dentro da sala ao lado da porta, tanto por dificultar a falsificação da presença como para evitar depredações da leitora.

5.6.8 Outras Opiniões Relevantes

Em ambos questionários, foi deixado um espaço aberto para comentários relevantes em relação ao sistema.

Entre os professores foi comentado por 20% que a solução a ser adotada deveria ser biométrica ao invés da leitora magnética, para diminuir as fraudes, porém devido ao maior custo, complexidade de implementação e ao tempo que seria gasto na criação de uma base com os dados biométricos dos estudantes para a validação, inviabilizam a implementação de uma solução desse caráter no projeto.

Também foi comentada uma solução RFID ao invés da carteira magnética para diminuição de fraudes, porém, a não ser que o aluno tenha o RFID fisicamente preso a si, essa solução é equivalente à da leitora magnética em termos de dificuldade na falsificação da presença.

Mencionou-se também sobre a flexibilidade dos métodos validadores de presença, e que esta deveria ser adequada.

Pelo lado dos alunos, foi mencionado utilizar uma catraca eletrônica ao invés de simplesmente uma leitora, porém acreditamos ser um dispositivo que tem um impacto psicológico muito negativo sobre a noção de liberdade de espaço e, portanto, na visão macro, não constituiria uma melhoria ao sistema.

Também se mencionou a possibilidade do professor iniciar e fechar a aula passando uma carteira sua, porém o sistema perderia muito robustez, sem contar o fato que o professor estaria dependente de um cartão, que não necessita estar em sua posse no processo atual.

Ainda, mencionou-se que em caso de pane do sistema, deve-se poder voltar ao método convencional. Esse fato já estava previsto na especificação do sistema.

Por fim, foi mencionado que o sistema apesar de dificultar a falsificação da presença, não resolve o problema, porém esse não é o foco principal do sistema e sim a automatização e controle da presença.

5.6.9 Decisões Tomadas a Partir dos Resultados Obtidos

Com base nesses resultados consolidados, chegou-se a um consenso sobre as duas principais decisões tomadas no projeto.

A primeira decisão tomada foi a de permitir um único método validador de presença: a porcentagem em tempo de aula, que deve variar entre 50 e 85%. O principal fator que levou à essa decisão foi o fato de que a grande maioria dos participantes da pesquisa, tanto alunos quanto professores, apontaram a porcentagem do tempo em aula como método para a validação da presença. Além disso, foi decisivo o fato de alunos terem apontado como um dos principais fatores de mau uso do sistema, pelo professor, a criação de métodos excêntricos e em demasia diferentes entre si para a validação de presença.

Utilizando somente a porcentagem de tempo em aula, os métodos não são confusos e os alunos têm certa flexibilidade nos horários das disciplinas. Um professor menos rigoroso pode aceitar apenas 50% de tempo em aula e o mais rigoroso de todos, ainda deixará uma janela para eventuais atrasos ou saídas mais cedo.

Uma aula convencional da EP tem 100 minutos. Desse modo o professor que adotar o método mais rigoroso estará concedendo, no máximo, 15 minutos de ausência da sua aula ao aluno.

Além disso, a principal vantagem do sistema indicada pelos grupos foi ser uma ferramenta de apoio estatístico ao professor, portanto esse será um grande foco do sistema, em que o módulo de estatísticas permitirá o professor comparar o desempenho de sua disciplina com a de turmas passadas, outras disciplinas ministradas por ele, ou até da mesma disciplina ministrada por outros professores. Obviamente, até essas comparações terem o efeito desejado, deve-se construir uma base histórica de pelo menos um ciclo da disciplina.

6 PROJETO E IMPLEMENTAÇÃO

Este capítulo é dividido em quatro grandes blocos. No primeiro, é feita a análise dos requisitos funcionais do sistema. No segundo, é apresentada a arquitetura de *hardware*. No terceiro, é descrita a implementação da arquitetura de *software*. Finalmente, no quarto, são apresentados os testes feitos.

6.1 Análise de Requisitos

6.1.1 Critérios de Presença

Conforme se estabeleceu pela aplicação das técnicas de elicitação mencionadas no capítulo anterior, definiu-se que o critério de presença a ser adotado pelo sistema será uma porcentagem do tempo total de duração de uma aula que, no caso da EP, geralmente é de 100 minutos.

Ressalta-se aqui que apenas um modelo de verificação de presença foi adotado também como resultado da pesquisa feita no Capítulo 5. Havia um receio por parte dos alunos que uma grande gama de regras poderia vir a causar confusão e ser difícil de ser assimilada pelos alunos.

Do ponto de vista do grupo, acredita-se que adicionar um grau de complexidade muito grande a um ponto tão fundamental do sistema poderia fazer com que em pouco tempo ele caísse em desuso, dado que traria mais problemas do que soluções com relação ao processo vigente, o que claramente não é o objetivo do SARPA.

Embora tenha sido necessário impor essa restrição, é importante ressaltar que ainda sim há grande flexibilidade na variação da taxa de presença, que poderá variar entre 50% e 85% do período total da aula, sendo útil tanto para professores que desejam ser mais rígidos e para os que não vêem tanta necessidade de manter o aluno na aula por tanto tempo.

O sistema divide igualmente a janela de ausência (ausência = 100% - presença requerida) igualmente entre o início da aula e o final da aula de modo que haja margem para que o aluno se atrase alguns minutos e tenha a flexibilidade de sair um pouco mais cedo.

Para que o aluno obtenha presença na aula, ele precisa identificar-se no início e no final da aula passando seu cartão na leitora magnética dentro do limite estabelecido no sistema. Esse sistema de dupla passagem evita que o aluno passe o cartão, e em seguida, recebendo presença por uma aula que na realidade não foi assistida, fato que pode ser muito comum com a lista de presença assinada.

Finalmente, é permitido que o aluno registre sua presença 10 minutos antes do início da aula e até 10 minutos depois do final da aula, para evitar que todos os alunos façam fila após o início da aula para registrarem sua presença ou que haja uma corrida ao final da aula para que todos passem seus cartões antes do término oficial da aula. Essa característica também cobre pequenos atrasos no final de uma aula.

Qualquer outra exceção será tratada individualmente pelo professor e é descrita no item de operação do sistema.

A Tabela 2 ilustra uma análise de sensibilidade considerando-se taxa de presença variando de 50% até o limite de 85% para uma aula que se inicia às 14:00h.

TABELA 2 – Análise de sensibilidade de presença

Tempo mínimo de presença (%)	50%	60%	70%	80%	85%
Tempo da aula	100m	100m	100m	100m	100m
Início da aula	14:00h	14:00h	14:00h	14:00h	14:00h
Termino da aula	15:40h	15:40h	15:40h	15:40h	15:40h
Tolerância máxima total	50m	40m	30m	20m	15m
Tolerância máxima na entrada	25m	20m	15m	10m	7m
Tolerância máxima na saída	25m	20m	15m	10m	7m
Presença aberta em:	13:50h	13:50h	13:50h	13:50h	13:50h
Horário limite para chegada	14:25h	14:20h	14:15h	14:10h	14:07h
Horário mínimo para saída	15:15h	15:20h	15:25h	15:30h	15:33h
Presença fechada em:	15:50h	15:50h	15:50h	15:50h	15:50h

6.1.2 Exceções

Um ponto importante do SARPA é a flexibilidade quanto a exceções que podem ocorrer por motivos diversos, como mudança de salas de aula ou atrasos. Para tanto foi desenvolvido um módulo de controle no qual o professor pode, através de sua conta no sistema, ajustar a presença dos alunos para eventuais fatores não modelados como operacionais pelo sistema.

Abaixo descrevem-se alguns cenários comuns que podem levar o SARPA a calcular incorretamente a presença, e requerer ajuste manual do professor. Embora só estejam listados alguns cenários, ressalta-se fortemente que o sistema tem flexibilidade de corrigir quaisquer exceções que possam vir a ocorrer.

6.1.3 Aluno Faltou à Aula

Caso um aluno tenha faltado à determinada aula por motivo justificável, o professor tem a possibilidade de atribuir presença individual ao aluno. Tal requisição é feita através do próprio sistema SARPA, evitando que seja necessária a utilização de e-mail externo.

6.1.4 A Aula Mudou de Dia

Caso a aula tenha mudado de dia o professor tem três opções:

- Antes que a aula ocorra, o professor pode configurar o sistema informado que a aula foi transferida para uma outra sala, permitindo assim que o sistema monitore o terminal correto;

- Se a mudança foi feita de última hora, o professor após a aula, pode solicitar que o sistema recupere os registros atribuídos à sala em que a aula foi realmente ministrada sejam aproveitados e a presença corrigida;
- Em último caso o professor pode atribuir presença individual para os alunos que estavam na aula ou atribuir presença coletiva para a turma.

6.1.5 A Aula Sofreu Atraso

Caso a aula tenha sofrido atraso, o professor tem a possibilidade de registrar um atraso de N minutos no início da aula e toda a programação é deslocada conforme esse atraso. Os registros de presença são então recalculados e registrados de forma correta no banco de dados.

6.1.6 A Aula Foi Cancelada

Caso a aula tenha sido cancelada o professor tem a possibilidade de:

- Remarcar a aula para uma data futura à sua escolha e cadastrá-la manualmente no sistema, ou ainda solicitar a um funcionário do departamento que o faça;
- Invalidar a aula fazendo com que o sistema a desconsidere-a na conta da presença, caso não haja reposição.

6.1.7 Interfaces de Acesso

A interface de acesso escolhida foi a *web*, na qual o usuário a partir de qualquer *browser* acessa a página do sistema e então acessa as informações e funcionalidades. Optou-se pela interface *web* em função da portabilidade e da não necessidade de instalação de qualquer *software* para acesso ao sistema. Como resultado o sistema pode ser acessado por diversos sistemas operacionais e em qualquer máquina com acesso *web* a Internet. O *layout* da interface é customizado para o tipo de usuário que acessa o sistema. Existem quatro *layouts* distintos, que diferem no número de opções de funcionalidade disponíveis: aluno, professor, funcionário e administrador.

6.1.8 Análises Estatísticas

A funcionalidade de análise estatística é um dos grandes diferenciais quando se compara a utilização do SARPA com o sistema atual. Através do sistema é possível acrescentar a possibilidade de utilização de todo o banco de dados de presença gerado na criação de estatísticas que permitam aos alunos acompanhar sua presença e principalmente aos professores, ter um *feedback* completo de como suas aulas estão sendo frequentadas, comparar com outros professores e ainda durante o transcorrer do curso poder trabalhar para aumentá-la.

6.1.9 Módulo Aluno

Através da análise de requisitos com relação ao aluno, notou-se a necessidade que o mesmo pudesse controlar melhor sua presença durante o semestre, evitando ser reprovado por um número pequeno de faltas que poderiam ter sido evitadas caso o aluno pudesse fazer um acompanhamento aula a aula da sua presença.

No módulo aluno, é possível observar a programação aula a aula de cada matéria, indicando qual é sua presença nessa matéria, o número de faltas até o momento e o número de faltas possíveis para que se atinja o mínimo de 70% de presença.

As Figuras 10 e 11 ilustram, respectivamente, o gráfico de presença e a tabela que indica o conteúdo de cada aula, destacando as que foram perdidas pelo aluno para ele tenha a possibilidade de cobrir a matéria perdida.

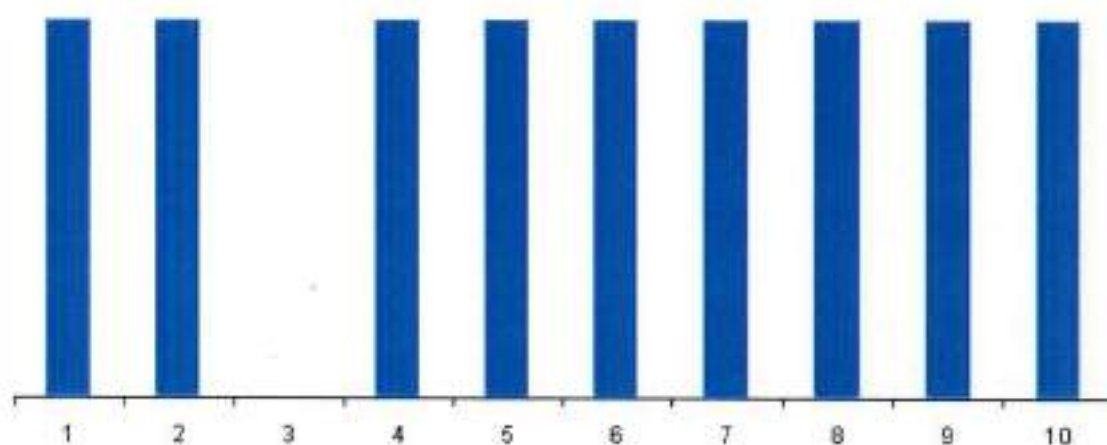


Figura 10 – Representação gráfica ilustrando presença do aluno

Aula	Conteúdo	Presença
1	Matéria A	Sim
2	Matéria B	Sim
3	Matéria C	Não
4	Matéria D	Sim
5	Matéria E	Sim
6	Matéria F	Sim
7	Matéria G	Sim
8	Matéria H	Sim
9	Matéria I	Sim
10	Primeira prova	Sim
11	Matéria J	-
12	Matéria L	-
13	Matéria M	-
14	Matéria N	-
15	Matéria O	-
16	Matéria P	-
17	Matéria Q	-
18	Matéria R	-
19	Matéria S	-
20	Segunda prova	-
Presença até o momento		90%
Faltas		1
Faltas possíveis		6
Faltas restantes		5

Figura 11 – Representação dos detalhes sobre a presença do aluno

6.1.10 Módulo Professor

Através da análise de requisitos com relação ao professor, notou-se a necessidade do mesmo de acompanhar de forma mais próxima a frequência de sua aula e o modo como esta se comporta durante período de aulas. Isso dá ao professor um *feedback* aula a aula, permitindo que mudanças sejam feitas no decorrer do período letivo.

O módulo professor permite que diversas análises estatísticas sejam geradas sobre a frequência dos alunos que, em sua instância final, pode ajudar o professor a detectar os resultados do seu estilo de aula, se estes estão adequados ou não e, eventualmente, realizar alguma mudança ainda no transcorrer do período letivo.

6.1.11 Análise de Uma Disciplina Específica

Essa é a função mais básica oferecida, em que o professor pode observar a frequência em sua disciplina separadamente, mostrando o histórico de presença aliado à tabela que indica qual foi a matéria dada em cada uma das aulas.

Nas Figuras 12 e 13, vê-se uma análise de frequência de uma matéria e turma qualquer, na qual um professor ministra suas aulas. É possível notar que houve uma presença

constante até duas aulas antes da prova, mas que nessas duas aulas a presença cai consideravelmente, o que pode significar, por exemplo, que a turma está sobrecarregada de trabalhos e provas e não têm tempo para assistir as aulas ou que o conteúdo dado em aula não está apropriado para o que a turma acredita que serão os temas abordados na prova.

Facilmente, o professor pode usar outro modo de comparação para saber se essa queda na presença é algo que afetou só sua turma ou todas as turmas, ou mesmo turmas dos anos anteriores (média histórica).

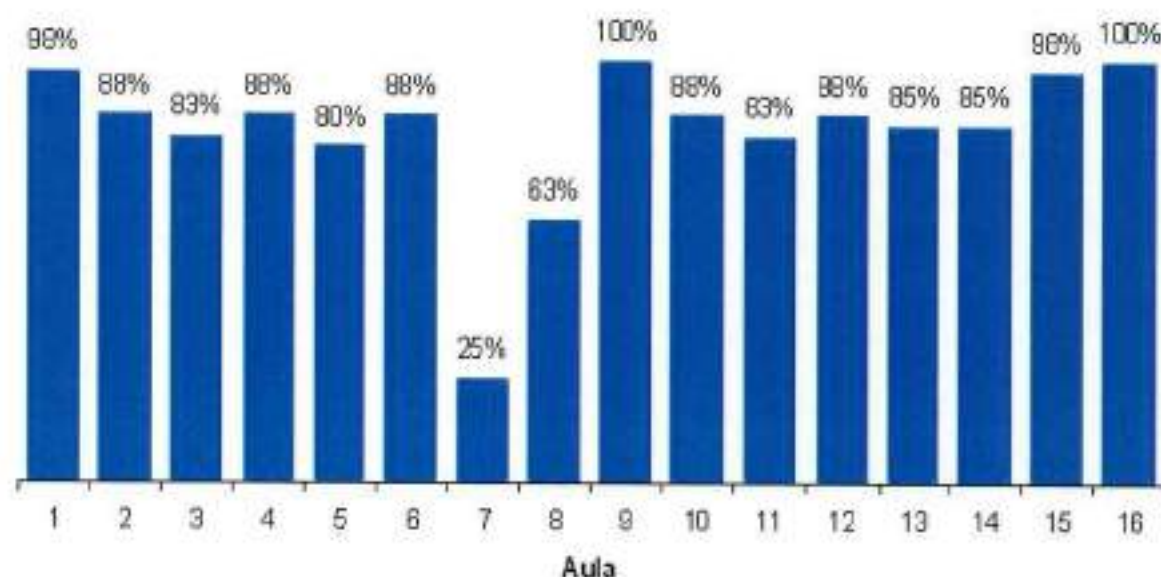


Figura 12 – Exemplo de visualização de frequência em uma disciplina pelo professor

Data	Horário	Aula	Alunos presentes	Conteúdo da aula	Presença
3/1/2006	14:00h	1	39	Matéria A	98%
3/3/2006	8:20h	2	35	Matéria B	88%
3/8/2006	14:00h	3	33	Matéria C	83%
3/10/2006	8:20h	4	35	Matéria D	88%
3/15/2006	14:00h	5	32	Matéria E	80%
3/17/2006	8:20h	6	35	Matéria F	88%
3/22/2006	14:00h	7	10	Matéria G	25%
3/24/2006	8:20h	8	25	Matéria H	63%
3/29/2006	14:00h	9	40	Primeira Prova	100%
3/31/2006	8:20h	10	35	Matéria J	88%
4/5/2006	14:00h	11	33	Matéria K	83%
4/7/2006	8:20h	12	35	Matéria L	88%
4/12/2006	14:00h	13	34	Matéria M	85%
4/14/2006	8:20h	14	34	Matéria N	85%
4/19/2006	14:00h	15	39	Matéria O	98%
4/21/2006	8:20h	16	40	Segunda Prova	100%

Figura 13 – Exemplo de tabela de conteúdo e frequência

6.1.12 *Análise de Outras Disciplinas do Mesmo Curso e Semestre*

Neste caso, o professor pode observar se a frequência em sua aula está em linha, abaixo ou acima da frequência de outras matérias da turma. Ter essa possibilidade permite, por exemplo, identificar se um nível de presença baixo é característica da própria turma ou se isso é um problema da matéria do professor especificamente.

Na Figura 14, tem-se um exemplo da comparação da frequência em uma disciplina com a média das frequências das outras disciplinas que uma turma esteja cursando, e pode-se observar que o nível de presença, embora baixo, está em linha com as demais disciplinas, o que provavelmente é uma característica da própria turma.

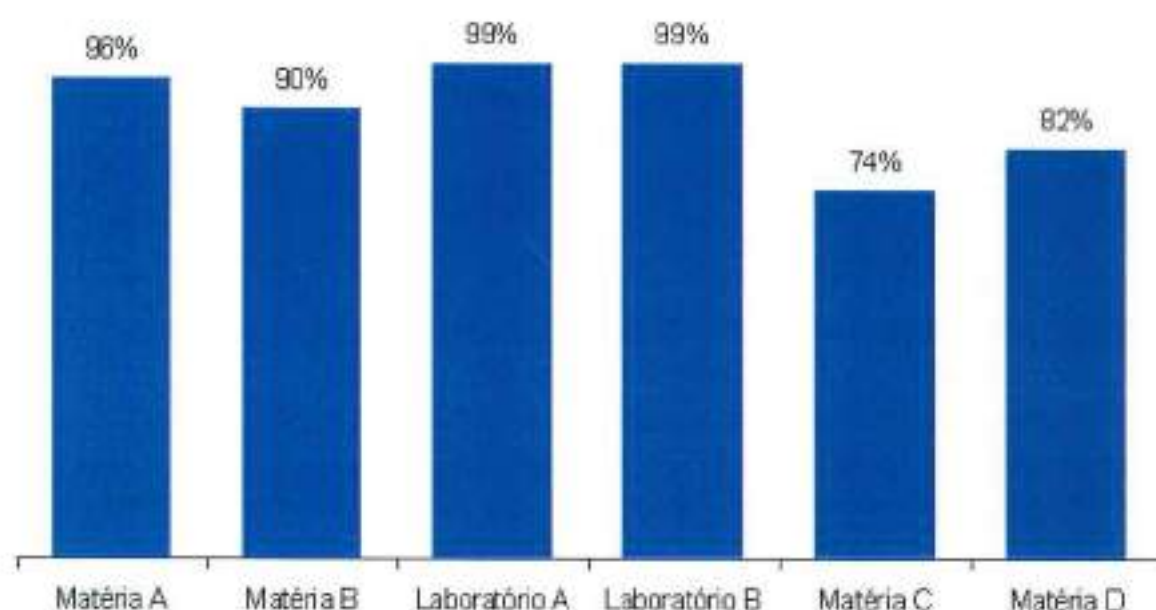


Figura 14 – Exemplo de análise de frequência de uma turma

6.1.13 *Análise de Turmas de Uma Mesma Disciplina*

Nesse modo de comparação, especialmente útil ao ciclo fundamental da EP (Biênio), o professor pode, dentro da mesma disciplina, comparar a frequência de sua turma com a de outros professores. Esse módulo é especialmente útil, pois nos primeiros anos existe uma tendência dos alunos preferirem assistir aulas com a turma que tem um professor mais didático ou mesmo que teve na prova uma média muito acima das demais.

Na Figura 15, exibe-se a comparação da frequência de turmas numa matéria hipotética. Nota-se que há turmas com uma frequência substancialmente mais alta que as demais, ou seja, pode estar havendo migração de alunos entre turmas. É importante ressaltar que o sistema não prevê a contabilização de presença para alunos que assistem a aula em uma turma que não seja a que ele está matriculado.

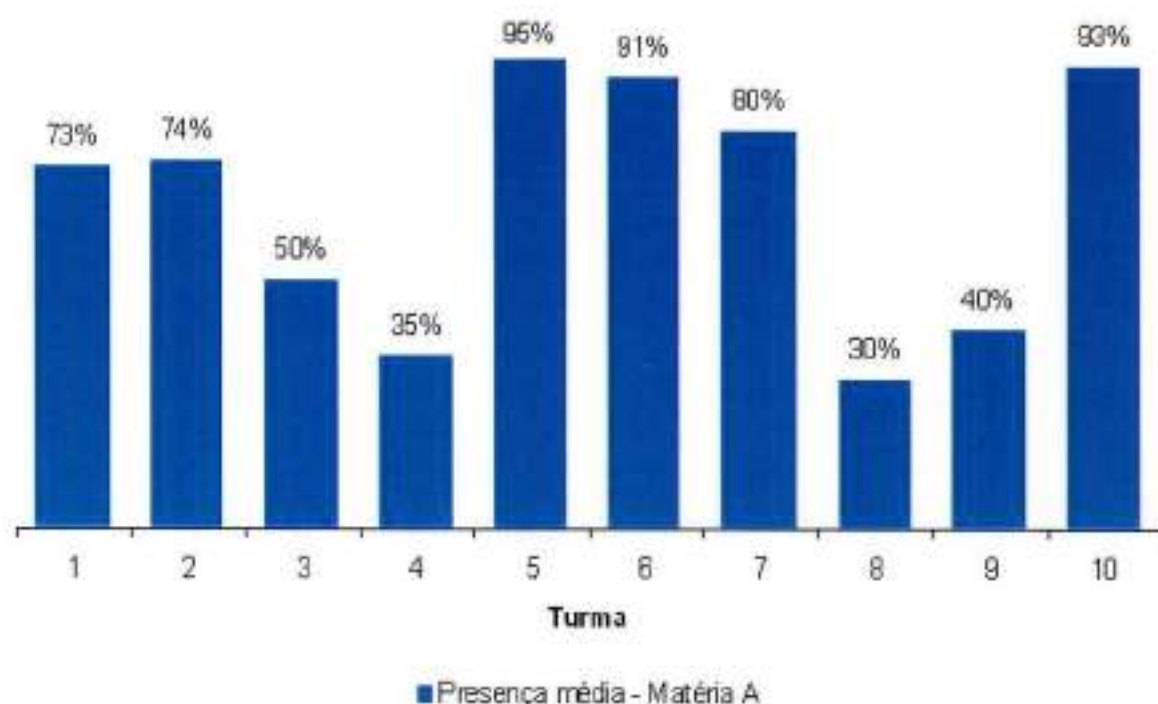


Figura 15 – Exemplo de análise comparativa entre turmas

6.1.14 *Análise de Disciplina Contra a Média Histórica*

Esse módulo permite ao professor comparar a frequência de sua turma com as turmas dos anos anteriores. Caso não tenha havido uma mudança grande de grade curricular, é possível observar a evolução da frequência em comparação com a frequência histórica da disciplina. É particularmente útil para professores assumindo a disciplina, pois permite identificar sua aula está sendo considerada útil pelos alunos. A Figura 16 ilustra essa comparação.

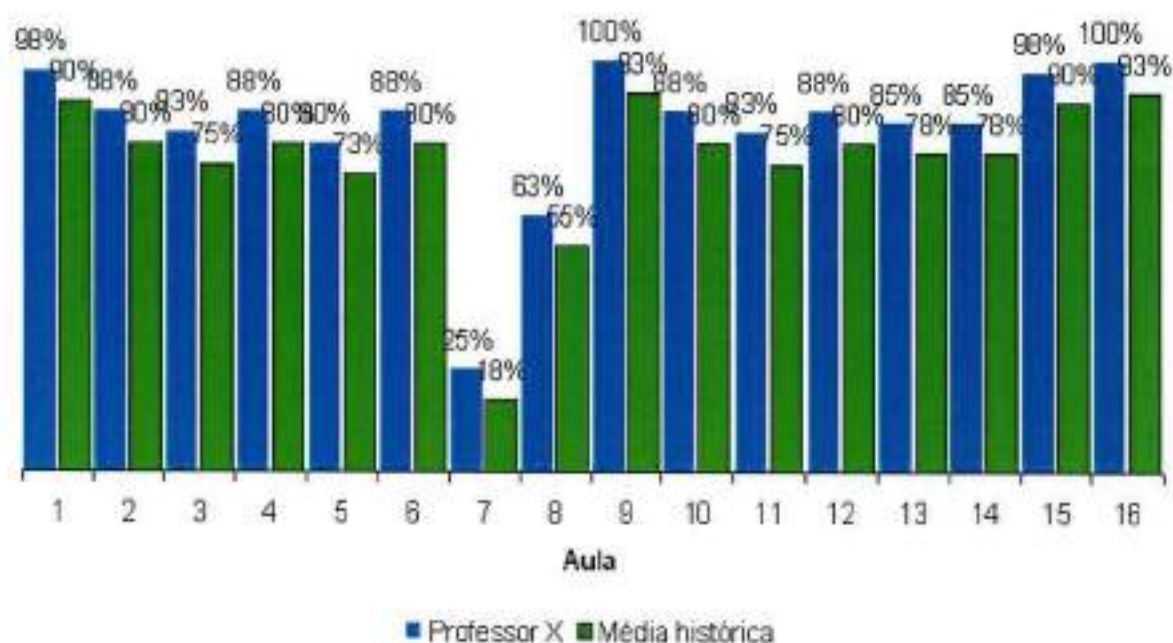


Figura 16 – Exemplo de comparação com média histórica

6.1.15 Módulo administrador

É de entendimento do grupo que o administrador deva ter acesso a todo o conteúdo de estatísticas a que o professor tem, com a vantagem de simular a operação como um usuário específico de um professor, o que daria a ele a capacidade de observar exatamente qual é o problema descrito pelo professor e corrigi-lo.

A mesma característica é aplicada quando o administrador faz manutenção no acesso de alunos, podendo observar as estatísticas simulando o acesso de um aluno específico.

6.1.16 Modelagem

Utilizando UML, uma série de documentos foi gerada para garantir que os requisitos sejam atingidos e para garantir que haja documentação suficiente, caso alguma expansão ou modificação seja necessária no sistema. É importante ressaltar a relevância desse último aspecto, já que caso não haja documentação, o contratante do sistema muito provavelmente irá ficar dependente do desenvolvedor do sistema, pois este será o único que realmente sabe como o sistema funciona. No caso de haver documentação, é possível contratar um outro desenvolvedor para implementar as mudanças.

6.1.17 Modelo de Casos de Usos

Seguindo o modelo UML, uma das práticas é elaborar o diagrama de casos de uso, para que se possa observar se os requisitos do cliente estão sendo atendidos. A Figura 16 ilustra o modelo de casos de uso e a descrição detalhada dos casos de uso encontra-se no Apêndice D – Especificação dos Casos de Uso. Na Figura 17, pode-se observar 4 atores:

- Administrador, que é responsável pela manutenção do sistema, e que possui acesso a qualquer função disponível;
- Funcionário, responsável por todo o cadastramento de informações no sistema, bem como mantê-las atualizadas;
- Professor, responsável por ajuste de presença que possa eventualmente ser necessário, possuindo acesso a funções estatísticas;
- Aluno, registra presença passando a carteira USP e tem acesso somente a sua frequência nas matérias do período letivo.

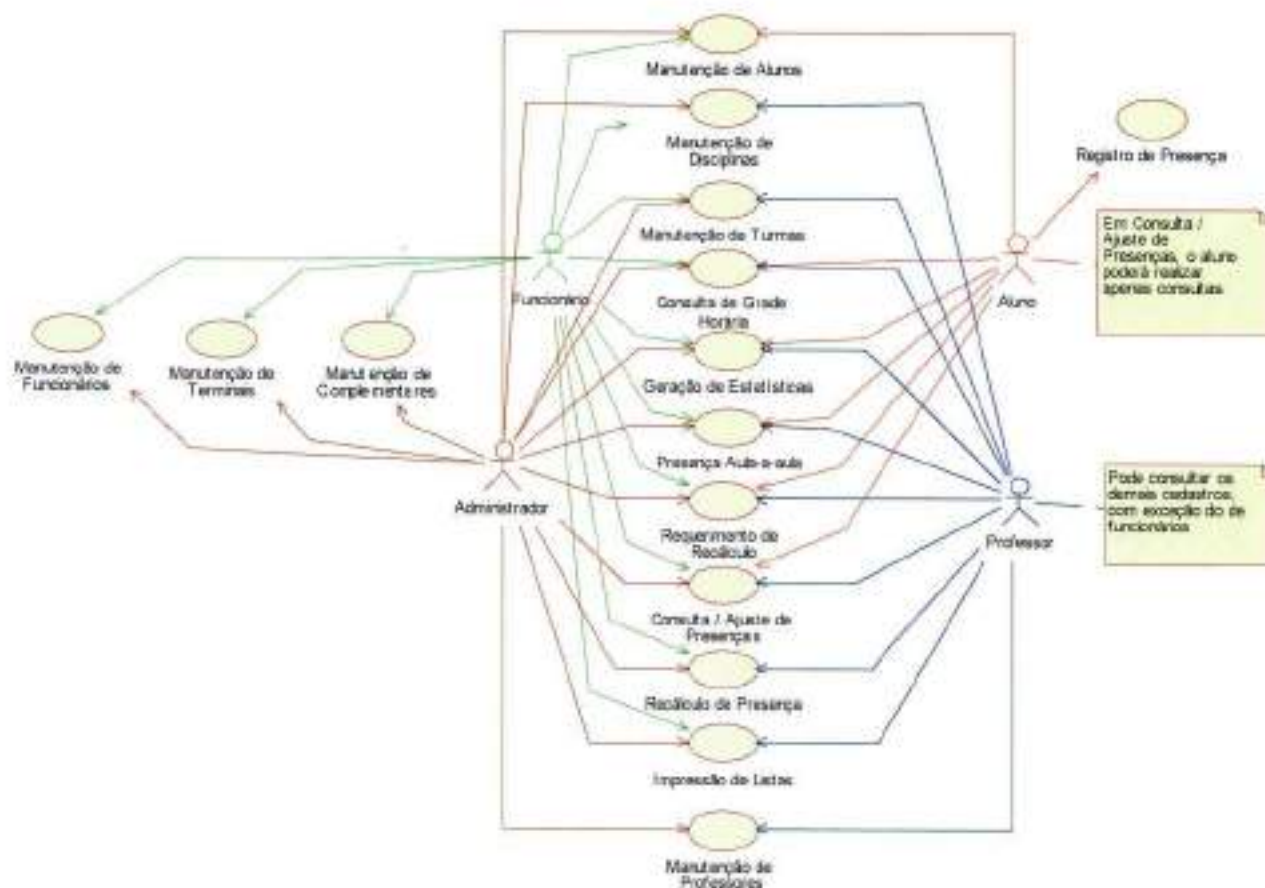


Figura 17 – Modelo de casos de uso

guia rápido de referência do modo como as tabelas se relacionam, incluindo a cardinalidade do relacionamento.

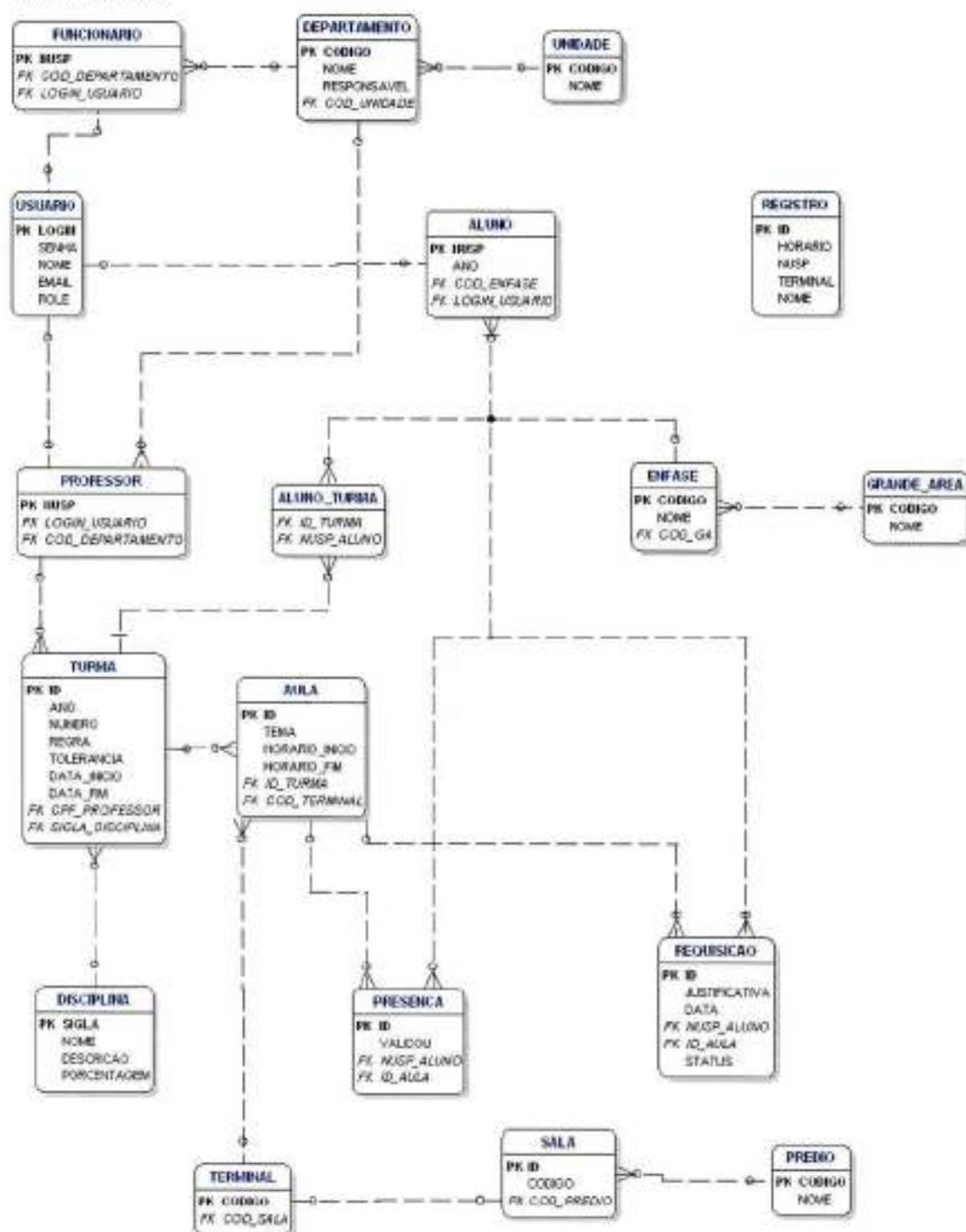


Figura 19 – Diagrama Entidade Relacionamento

6.1.20 Diagramas de Seqüência

Os diagramas de seqüência ajudaram a definir a ordem em que os objetos interagem no sistema. Os principais diagramas do sistema se encontram no Apêndice E – Diagramas de Seqüência.

6.2 Arquitetura de Hardware

A arquitetura de *hardware* do sistema SARPA pode ser dividida em três blocos principais observados na Figura 20: Bloco de Leitura, Bloco Servidor de Registros e Bloco Servidor Web.

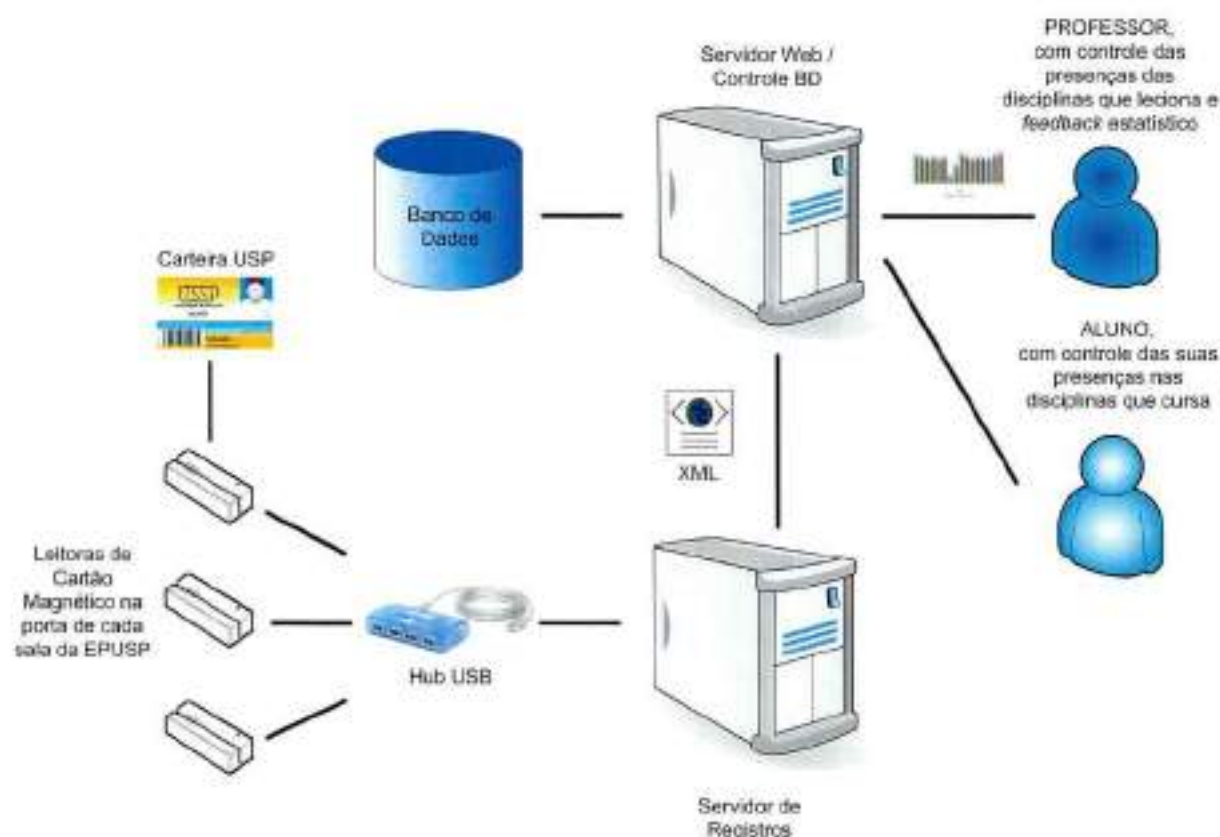


Figura 20 – Arquitetura de Hardware do sistema

6.2.1 Bloco de Leitura

Conforme citado na seção 2.2 de tecnologias utilizadas, optou-se pela tecnologia USB nas leitoras por uma série de razões dentre as quais destaca-se a facilidade de concatenar as leitoras através de cascata ou *hub* e a transparência quanto à tecnologia de registro da presença, podendo variar entre a própria leitura magnética, RFID ou mesmo leitores biométricos. Além disso, no padrão USB é possível concatenar até 127 dispositivos seja através de *hub* ou através de conexão em cascata. Esse fator foi um ponto chave de decisão quanto à tecnologia utilizada, já que havia a opção de utilização de leitoras seriais, mas que demandaria o desenvolvimento de um *hub* serial, o que foge do escopo do projeto.

O *hub* USB por si só fornece alimentação para os dispositivos, pois, normalmente, possui sua própria fonte de alimentação, evitando que outros cabos tenham que ser utilizados, aumentando os custos de implantação do sistema.

6.2.2 Bloco Servidor de Registros

Como segundo bloco tem-se o servidor de registros. O servidor de registros é responsável pela monitoração de todas as leitoras USB ligadas a ele, gerando um arquivo XML contendo as informações dos cartões para que sejam importados pelo servidor de dados. Esse arquivo XML fica armazenado temporariamente no próprio servidor de registro que, de tempos em tempos, irá enviá-los ao servidor *web* para que possam ser inseridos no banco de dados.

Todo o trabalho de monitoração, criação e envio do arquivo XML ao servidor de banco de dados é feito através de um serviço desenvolvido pelo grupo, que trabalha de forma independente do sistema SARPA. Isso traz grande flexibilidade o sistema, já que se, em uma versão futura, houver a necessidade de implantação de outro método de identificação, como biométrico, apenas o serviço deverá ser alterado ou expandido, restando apenas ajustes mínimos ao SARPA.

Idealmente, os servidores – de registro e *web* – deveriam estar separados, devido ao fato de que monitorar a porta USB constantemente exige capacidade de processamento considerável, que em simulações feitas, com a leitora de cartões Minimag adquirida pelo grupo, chegou a passar de 80% de utilização do processador em uma máquina Intel Pentium III. Caso o servidor *web* estivesse rodando na mesma máquina, poderia haver lentidão no servidor e, em casos extremos, arquivos XML poderiam deixar de ser gerados.

Outro fator determinante para essa separação origina-se do fato de que o servidor *web* é muito mais exposto a qualquer tentativa de ataque externo se comparado a um servidor interno. Para evitar que um ataque ao servidor *web* derrube todo o módulo de registro de presença, a separação mostrou-se como a melhor escolha.

Nessa arquitetura, mesmo que o servidor *web* saia do ar, os registros continuam sendo armazenados no servidor de registros através dos arquivos XML para que quando o servidor *web* voltar a estar *online*, esses arquivos sejam transferidos para ele.

Porém, é importante ressaltar que devido à restrições de custo e para fins de apenas ter um protótipo funcional do sistema, a arquitetura SARPA foi implementada em um único servidor.

6.2.3 Bloco servidor web

O terceiro bloco compreende o banco de dados e o servidor de acesso *web*, pelo qual os usuários acessam o sistema.

O servidor *web*, conforme descrito anteriormente, monitora uma pasta de entrada buscando arquivos XML que contém dados dos registros de presença que são, então, inseridos em um banco de dados temporário.

Caracterizou-se esse banco de dados temporário pelo fato de que os registros ficam armazenados nele por pelo menos um ano, para que caso haja algum pedido de revisão de presença, os registros possam ser importados novamente.

Além do banco de dados, o servidor web mantém o servidor HTTP responsável por permitir o acesso web dos usuários. Esse serviço é disponibilizado pela ferramenta Tomcat, conforme descrito no Capítulo 2.

6.3 Arquitetura de Software

A arquitetura de software do sistema implementa o padrão model 2, variação do modelo MVC (Model View Controller) através da ferramenta Struts, separando o controle, a visualização e a lógica de negócio conforme ilustrado, anteriormente, na Figura 6 do Capítulo 3. Na Figura 21, são vistos os principais módulos do sistema.

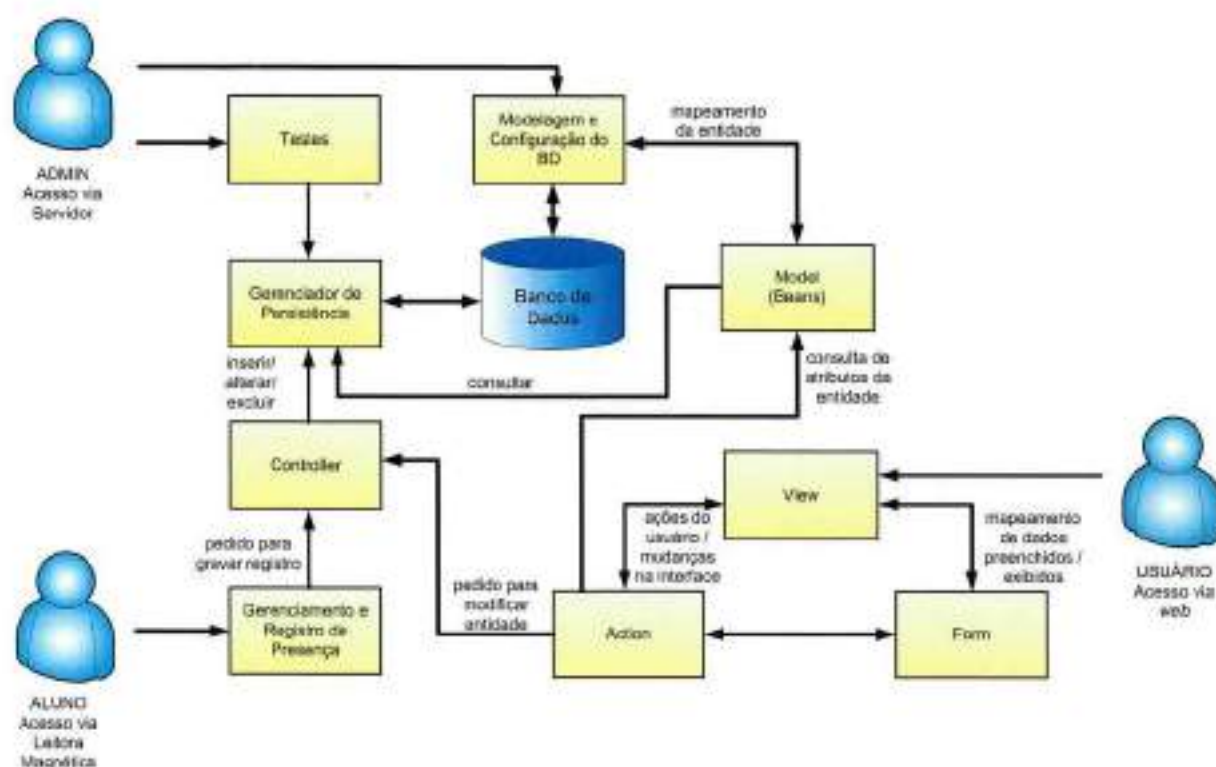


Figura 21 – Principais Módulos de Software do Sistema

6.3.1 Gerenciamento de Registro e Presença

Esse módulo é responsável por receber as informações sobre as carteiras USP que estão sendo enviadas pelas diversas leitoras USB. Este módulo faz uso da ferramenta XStream para gerar o arquivo XML referente a cada uma um dos registros (cada vez que a carteira é lida) e

também é responsável por recuperar os arquivos e reconvertê-los em objeto para fazer o envio ao módulo *Controller* juntamente com o pedido (ação) de persistência.

6.3.2 *Controller*

O módulo *Controller* é responsável por converter os diversos pedidos de ação em pedidos de persistência. Frequentemente, um pedido de ação implica em mais de um pedido de persistência no banco de dados, ou seja, pode ser necessário inserir mais de um objeto por pedido. Após determinar quais são os objetos que devem ser persistidos, eles são então enviados para o Gerenciador de Persistência que os armazenará no banco de dados.

6.3.3 *Gerenciador de Persistência*

O gerenciamento de persistência faz uso da ferramenta Hibernate para converter os objetos recebidos e inserir os registros necessários no banco de dados, realizando a consistência de atributos necessária.

6.3.4 *Testes*

O módulo de testes possui a função de, através do módulo de gerenciamento de persistência, popular o banco de dados com as informações que serão utilizadas para o teste do sistema. É particularmente útil pois, sempre que necessário, é possível retornar o banco de dados em um estado inicial conhecido.

Outra função é testar todos os modos de persistência e recuperação de dados garantindo que o módulo de gerenciamento de persistência esteja funcionando corretamente para cada *bean*.

6.3.5 *Modelagem e Configuração do Banco de Dados*

É responsável por receber o mapeamento dos diferentes *beans* capturando os *tags* inseridos para então modelar e gerar automaticamente a estrutura do banco de dados.

6.3.6 *Model*

O módulo *Model* consiste nas classes que representam as entidades do sistema. Cada classe possui métodos para interação com o banco de dados e métodos de validação de persistência.

6.3.7 View

É responsável por fazer a interface entre o usuário e o sistema, recebendo os comandos do usuário e enviando-os para o módulo *Action*. Também é responsável por todas as informações exibidas na tela para o usuário, além de monitorar o módulo *Form* para qualquer mudança seja refletida na interface.

6.3.8 Action

Sua função é receber e executar as ações comandadas pelo usuário na interface (*View*). As informações utilizadas pelo módulo *Action* são obtidas através do módulo *Form* e não da própria interface *View* como se pode pensar em um primeiro momento.

6.3.9 Form

O módulo *Form* é responsável por capturar e alimentar as informações que são exibidas na interface do usuário (*View*). Sua construção é bastante simples e consiste principalmente de métodos *get* e *set*, que fazem a comunicação com a interface.

6.4 Módulos e Serviços Auxiliares da Arquitetura de Software

Os módulos e serviços descritos nesta seção, apesar de não serem módulos principais, têm relevante importância para o sistema e são detalhados a seguir.

6.4.1 Serviço de Leitura de Cartões

O funcionamento da leitora USB é similar ao de um teclado USB, isto é, as informações que são obtidas dos cartões magnéticos são digitadas no servidor. Prevendo que o servidor de registros, provavelmente, será utilizado por outras aplicações (embora isso não seja recomendado) foi utilizado o recurso de formatação de dados disponível na leitora.

Conforme descrito anteriormente, o serviço é executado em segundo plano no servidor de registros e monitora as entradas de dados advindas das portas USB em busca de informações. Quando qualquer informação é recebida via esse teclado virtual criado pela leitora, sua consistência é testada em busca do cabeçalho identificador para ser processada.

Grande parte dos modelos de leitoras magnéticas permite que elas sejam configuradas para adicionar informações desejadas aos dados que são lidos dos cartões. Tal possibilidade foi explorada para identificar a leitora, com um código de identificação, e formatar a informação do cartão com o cabeçalho correspondente. Essa formatação é necessária pois, como a informação de passagem de cartão é obtida através de um teclado virtual, o que é obtido através do teclado real do servidor deverá ser ignorado, e os cabeçalhos são utilizados com esse propósito.

Como em todo cartão magnético, as informações da carteira USP estão dispostas no formato de trilhas, sendo que a trilha 1 guarda o nome do proprietário, a trilha 2 o número USP e a trilha 3 caracteres de controle. Através da programação da leitora, o grupo configurou os cabeçalhos de forma que a mensagem efetivamente recebida pelo serviço que monitora a porta USB tem o seguinte formato:

```
#USP%COD_TERMINAL_C/_3_DIGITOS%NOME_DO_PROPRIETARIO  
%NUSP_C/_10_DIGITOS  
%USP#&
```

Nota-se que o cabeçalho escolhido foi #USP, seguido do código do terminal, o nome do proprietário, seu número USP e o cabeçalho de encerramento USP#&. Optou-se por utilizar o símbolo % como separador de informações para efeito de organização.

6.4.2 Conversão de Objeto em Arquivo XML

Após a recepção da informação obtida pela leitora, o servidor de registros separa as informações e cria um objeto trilha que reflete os dados do proprietário do cartão com um atributo adicional que armazena a hora em que o registro foi recebido. Para que essa informação seja recuperada posteriormente pelo servidor de banco de dados é necessário que seja criado um arquivo XML com os dados da passagem do cartão mais a data/hora em que ele foi gerado. Para tal conversão optou-se pela ferramenta XStream, de uso gratuito que transforma um objeto Java em um arquivo XML. Para efeito de organização, o nome do arquivo XML gerado foi definido como uma combinação da data/hora da leitura combinada com o número USP do proprietário do cartão.

6.4.3 Envio do Arquivo XML ao Servidor de BD

Todos os arquivos XML são armazenados em uma pasta temporária no servidor de registros e a cada N minutos são copiados pelo serviço da leitora para o servidor de BD, onde serão lidos e persistidos no banco de dados. Caso o servidor de banco de dados esteja indisponível, os arquivos XML vão sendo armazenados no servidor de registros (na arquitetura proposta) e tão logo o servidor web esteja disponível novamente serão copiados para serem incluídos no BD.

6.4.4 Módulo de Recepção de Registros

De forma análoga ao servidor de registros, o servidor de banco de dados varre constantemente sua pasta de entrada em busca de arquivos XML recebidos do servidor de registros para inseri-los no banco de dados, apagando os arquivos já lidos, resultando em economia de espaço.

O módulo de recepção de registros se utiliza da mesma ferramenta XStream citada anteriormente para recuperar os arquivos XML e gerar objetos Java que possam ser manipulados e inseridos no banco de dados.

6.4.5 Módulo de Acompanhamento da Presença Acadêmica

O módulo de acompanhamento de presença acadêmica é responsável por disponibilizar todos os serviços de estatística e manipulação através do banco de dados. Seus principais usuários são os alunos e professores.

6.4.6 Acompanhamento pelo Aluno

As funções disponíveis para o aluno envolvem o acompanhamento de sua presença durante o período de aulas, por matéria, bem como permite que ele tenha a chance de saber qual foi o assunto da aula que ele perdeu. O sistema calcula automaticamente o número de faltas do aluno até o momento, o número de faltas possíveis e o número de faltas restantes para que ainda seja possível a aprovação no curso.

Adicionalmente, o sistema permite que o aluno possa enviar seus pedidos de ajuste ao professor através do SARPA, sem a necessidade de utilização de e-mail externo. Isso é particularmente útil para economia de tempo, já que assim que o professor recebe o pedido, caso ele concorde, com apenas alguns cliques a presença do aluno é ajustada.

6.4.7 Acompanhamento pelo Professor

O módulo do professor tem a função de permitir o acompanhamento e a realização de ajustes nas presenças dos alunos quando conveniente.

A primeira parte do módulo do professor se refere às manutenções e ajustes que possam ser necessários durante o curso da utilização do sistema. Dentro dos problemas previstos como possíveis, o sistema foi desenvolvido para permitir que o próprio professor possa fazer os ajustes referentes à presença sem ajuda de outro funcionário ou administrador. Tomou-se o cuidado de garantir que tais ajustes, quando necessários, não tomem muito tempo do professor, o que poderia tornar o sistema burocrático e desinteressante.

O módulo de manutenção permite, ao professor, ajustar a presença dos alunos em sua disciplina. Essa ferramenta tem a função de assegurar que mesmo em caso de exceção o sistema possa se adaptar e ainda assim refletir a situação real da folha de presença. Os dois problemas mais comuns de serem tratados aqui são pedidos de revisão de presença por parte dos alunos e ajustes quanto às características da aula (data, horário ou sala).

Essas alterações dos parâmetros devem ser inseridas no sistema, para que o cálculo das presenças seja correto.

O segundo módulo a que o professor tem acesso refere-se às análises estatísticas. Esse módulo conforme pode ser visto nesse mesmo Capítulo no Item 6.1.10, permite ao

professor diversas análises que podem ser usadas para acompanhamento do desempenho dos alunos durante os períodos de aula. Acredita-se que esses dados, quando analisados, só são obtidos ao final do curso, o que impossibilita que o professor possa tomar alguma atitude ainda no decorrer do curso, após notar, por exemplo, que sua média de presença é substancialmente menor do que a dos outros professores da mesma matéria.

Procurou-se oferecer análises que fizessem sentido aos professores e que extraísse a maior quantidade de informações dos dados disponíveis no sistema.

6.5 Integração e Testes

No curso do desenvolvimento do sistema, foi necessária a execução de diversos procedimentos de testes para garantir a operação do sistema de forma desejada. O grupo considera que a fase de testes possui um papel extremamente relevante no desenvolvimento de *software*. Através dele é possível simular a execução de tarefas desde cada *bean* (classe que modela a entidade) até os testes de integração, em que o sistema é testado como um todo. Optou-se por separar os testes em dois níveis: unitários e de integração. Abaixo descrevem-se as atividades envolvidas em cada um dos testes. No Apêndice F – Roteiro de Testes, é possível observar este o roteiro de testes previstos.

6.5.1 Testes Unitários

Os testes unitários são executados de forma automática através da ferramenta Junit, que popula o banco de dados e realiza testes nos *beans*. Esse teste unitário é executado de forma automática pela ferramenta e assegura que todos os métodos implementados estão sendo executados de forma adequada e que a persistência das informações está correta. Com isso é possível assegurar-se de que eles foram implementados da forma correta e não estão gerando erros de operação.

6.5.2 Testes de Integração

Diferentemente dos testes unitários, os testes de integração contemplam o funcionamento global do sistema.

Foi desenvolvido um roteiro de testes integrados que na visão do grupo é suficiente para garantir que o funcionamento do sistema está ocorrendo da forma prevista. Os testes desenvolvidos compreendem tanto o funcionamento normal quando o tratamento de exceções, incluindo atrasos, cancelamentos e mudança da data das aulas.

7 CONSIDERAÇÕES FINAIS

7.1 Conclusão

O sistema apresentou os resultados esperados, conseguindo atender aos seus requisitos chave propostos por este projeto. Foram feitos testes unitários ao longo do desenvolvimento do projeto, os chamados testes de caixa branca e ao final do desenvolvimento foram feitos testes caixa preta para avaliar a integração entre os módulos de *software* do sistema e a integração destes módulos com o *hardware*.

7.2 Resultados Obtidos

Ao final do desenvolvimento do projeto, obtiveram-se todas as classes previstas na modelagem inicial, além das classes de integração com o Struts e o Hibernate, funcionando da maneira esperada. Além disso, a integração do *hardware* das leitoras magnéticas deu-se como esperada, e a leitura dos registros de diferentes carteiras USP, simulando testes de registro de presença foram bem sucedidos.

Seria interessante que fossem feitos os testes de carga do sistema, com várias leitoras magnéticas conectadas à rede USB, para validar o modelo de arquitetura adotado. Esses testes não foram possíveis devido a uma restrição orçamentária do projeto que somente permitiu trabalhar com uma única leitora magnética.

As funcionalidades chave propostas pelo sistema foram atendidas, dando especial destaque ao acompanhamento de presença acadêmica pelo aluno e o *feedback* estatístico dado aos professores das disciplinas que estes lecionam.

Além disso, é interessante ressaltar que para validar o sistema, os usuários chave – professores, alunos e funcionários da EP – deveriam testar as funcionalidades e usabilidade do sistema.

7.3 Dificuldades Encontradas

Inicialmente na fase de definição do escopo e requisitos do projeto, houve algumas discussões devido à complexidade do sistema e chegou-se à conclusão que era necessário o uso de ferramentas que auxiliassem no levantamento dos requisitos. Então foi decidido o uso de técnicas de elicitación de requisitos, mais especificamente, os questionários direcionados a alunos e professores.

Houve certa dificuldade na aderência desses grupos ao questionário e houve a necessidade de um acompanhamento mais próximo, através de pedidos via e-mail para que os questionários fossem respondidos dentro do prazo previsto para o levantamento de requisitos.

Após a fase de modelagem, a principal dificuldade técnica encontrada foi a configuração e uso do Hibernate. Apesar do grupo já ter utilizado esta ferramenta em outros trabalhos acadêmicos, surgiram problemas ao configurá-la e adaptá-la para o sistema SARPA, já que foi escolhido fazer o mapeamento automático da persistência dos *beans* através de tarefas configuradas no Eclipse, ao invés de gerar manualmente os arquivos XML de configuração para tal.

Outro problema encontrado foi a integração das diversas tecnologias e ferramentas utilizadas no projeto. Por se tratar de uma combinação específica para o sistema SARPA, a quantidade de informações disponíveis na internet e em livros nem sempre atendia a necessidade do grupo, ocasionando maior tempo gasto para ajustar o bom funcionamento dos módulos integrados.

Além dos problemas técnicos, enfrentou-se certa dificuldade durante a implementação para controlar e planejar ações para cada problema e exceção que o sistema poderia gerar, além de seu uso incorreto pelos usuários. Assim, procurou-se programar o sistema de forma que esses problemas não ocasionem danos à integridade das informações que o mesmo guarda e que, em situações que fogem de situações normais das atividades acadêmicas (como atrasos de aula, mudanças de sala e outros), possam ser tomadas ações para que o sistema se adapte e funcione corretamente.

7.4 Aprendizado com o Projeto

Com esse projeto, além da elaboração de um trabalho escrito no nível de uma monografia de conclusão de curso, aprendeu-se a especificar e desenvolver um projeto real de porte médio, o que não havia ocorrido nas disciplinas de Laboratório de *Software* ministradas na EP. Essa experiência evidenciou a dificuldade que se têm em levantar e lidar com requisitos reais, mostrando que simplificações adotadas nos Laboratórios de *Software*, não eram viáveis no caso de um sistema real.

Devido à complexidade do projeto, na busca de levantamento por requisitos, foram aprendidas técnicas novas de levantamento de requisitos que não foram vistas nas disciplinas da EP e que abrem uma nova perspectiva para especificação de projetos de *software*, não tida antes pelo grupo.

Além disso, foi possível realizar a integração de um projeto de *software* com elementos de *hardware* e simulação de uma arquitetura física, o que, também, não havia ocorrido nas disciplinas da EP. Esse fato fez o grupo ter uma visão mais ampla de requisitos não funcionais do projeto como disponibilidade e confiança dos dados coletados nas leitoras magnéticas e fez o grupo tomar decisões no sentido de maximizar a qualidade desses requisitos.

7.5 Contribuição

Uma das principais contribuições desse projeto foi a infra-estrutura para aplicações *web* de médio e grande porte. Essa infra-estrutura envolve todo um conjunto de classes

Java de integração com o Struts e o Hibernate que podem ser reaproveitadas por outros projetos sem grandes alterações.

Além dessas classes de infra-estrutura, foram desenvolvidas classes de obtenção de dados da leitora magnética, via porta USB. Os dados são obtidos da leitora e escritos em arquivos XML, somente com o uso dessas classes, constituindo, assim, uma solução eficiente e barata.

Também, a pesquisa feita na elicitação de requisitos do projeto mostra um pouco a visão de alunos e professores quanto ao processo atual e necessidade de se ter um novo processo, com mais recursos, como os propostos pelo SARPA.

Ainda vale ressaltar que o projeto foi desenvolvido utilizando somente ferramentas abertas e livres, comprovando a eficácia e eficiência de *softwares* com essa filosofia, tornando a solução mais barata.

7.6 Trabalhos Futuros

Há duas linhas de trabalhos futuros para a implantação do sistema. Para a implantação na EP e em outras unidades da USP, é fundamental que o cadastro de grades horárias e alunos seja integrado com o sistema Júpiter. Essa integração removeria grande parte dos cadastros feitos no SARPA, facilitando a tarefa do administrador e dos funcionários que operam o sistema, desburocratizando o processo e maximizando a viabilidade do uso.

A segunda linha seria para viabilizar a implantação em outras escolas/universidades, adaptando os módulos de cadastro para se adequar a essas outras instituições. Em termos de grade horária, o sistema já é flexível o suficiente para seguir os modelos particulares de outras instituições.

7.7 Viabilidade de Implantação

O sistema SARPA teve todas funcionalidades previstas atendidas, porém antes da sua implantação seriam necessários um bom planejamento e testes de carga sobre a arquitetura de *hardware* proposta com *hubs* USB.

Após a validação da arquitetura proposta, o ideal seria fazer um projeto piloto, envolvendo uma ou duas turmas da EP para que esse grupo reduzido faça uma avaliação do sistema, propondo melhorias e fazendo sugestões que deveriam ser analisadas caso a caso, para uma posterior implantação com todas as turmas.

Ainda é importante ressaltar que, para uma implantação em grande porte, não é necessário, porém é altamente recomendável que o sistema SARPA já esteja integrado ao sistema Júpiter e que dados de presença histórica de anos anteriores sejam inseridos no sistema para constituir uma base estatística em que os professores, logo no primeiro período de uso do sistema, já tivessem um *feedback* estatístico adequado.

BIBLIOGRAFIA E REFERÊNCIAS

APACHE SOFTWARE FOUNDATION, **Site Oficial do Jakarta Struts Framework**. Disponível em: <<http://struts.apache.org>>. Acesso em: 30 out. 2006.

APACHE TOMCAT, **Site Oficial do Apache Tomcat**. Disponível <<http://tomcat.apache.org>>. Acesso em: 30 out. 2006.

BELL, D. **An Introduction to the Unified Modeling Language**. 2006a. Disponível em <www-128.ibm.com/developerworks/rational/library/769.html> Acesso em: 31 out. 2006.

BELL, D. **UML Basics - The class diagram**. 2006b. Disponível em <<http://www-306.ibm.com/software/rational/uml/>> Acesso em: 01 nov. 2006.

BELL, D. **UML's Sequence Diagram**. 2006c. Disponível em: <http://www-128.ibm.com/developerworks/rational/library/3101.html> Acesso em: 02: nov. 2006.

BOOCH, G.; RUMBAUGH, J; JACOBSON, I. **The Unified Modeling Language User Guide – Second Edition**. Pearson Education, United States:Massachusetts, 2005.

CVS, **Site Oficial do CVS**. Disponível em: <<http://www.nongnu.org/cvs/>>. Acesso em: 30 out. 2006.

HOUAISS, **Dicionário eletrônico**. Disponível em: <<http://noticias.uol.com.br/educacao/>>. Verbetes: Modelo. Acesso em: 31 out. 2006

ECLIPSE, **Site Oficial do projeto Eclipse**. Disponível em: <<http://www.eclipse.org>>. Acesso em: 30 out. 2006.

GOGUEN, J.A.; LINDE, C. **Techniques for Requirements Elicitation**. Disponível em: <<http://www.cs.brown.edu/courses/cs190/2006/assignments/goguen%5B1%5D.pdf>>. Acesso em: 10 jul. 2006.

HIBERNATE, **Site Oficial do Projeto Hibernate**. Disponível em: <<http://www.hibernate.org>>. Acesso em: 30 out. 2006.

IDTECH, **Manual do MINIMAG – Leitora de Cartão Magnético**. 2000. 33p.

INTEL, **Universal Serial Bus (USB*)**. Disponível em: <<http://www.intel.com/technology/usb>>. Acesso em: 27 out. 2006.

JUNIT, **Site Oficial do JUnit Testing Framework**. Disponível em: <<http://www.junit.org>>. Acesso em: 30 out. 2006.

KOTONYA, G.; SOMMERVILLE, I. **Requirements Engineering**. Chichester, Inglaterra:Wiley, 1998. 282 p.

LAVESSEN, S. **Software Requirements Styles And Techniques**. Londres, Inglaterra: Addison-Wesley, 2002. 575 p.

MAC POS HARDWARE. **Point of Sale Hardware And Software** Disponível em: <http://www.macpos-hardware.com/shop/products_pictures/minimag.jpg>. Acesso em: 30 out. 2006.

MINISTÉRIO DA EDUCAÇÃO. **Portal do Projeto Presença, BRASIL**. Disponível em: <<http://portal.mec.gov.br/projetopresenca/index.php?option=content&task=category§ionid=18&id=127&Itemid=372>>. Acesso em: 26 out. 2006.

MYSQL AB, **Site Oficial do Banco de Dados MySQL**. Disponível em: <<http://www.mysql.com>>. Acesso em: 30 out. 2006.

ORACLE, About the Model 1 and Model 2 Architecture – Disponível em:

<http://www.oracle.com/webapps/online-help/jdeveloper/10.1.3?topic=adf_aboutmvc2_html>. Acesso em: 31 out. 2006.

PISTONE, E. Serviços Tecnológicos e TIB. Bate-Papo programado do IPT. Disponível em:

<<http://www.ipt.br/atividades/servicos/chat/?ARQ=99>>. Acesso em: 30 out. 2006.

PRESSMAN, R. S. **Engenharia de Software**. Rio de Janeiro, Brasil: McGraw-Hill, 2002. 843 p.

SOFTWARE ENGINEERING INSTITUTE, Client/Server Software Architectures--An Overview.

Disponível em:<http://www.sei.cmu.edu/str/descriptions/clientserver_body.html> Acesso em: 21 nov. 2006.

SUN MICROSYSTEMS, The J2EE 1.4 Tutorial. 2006a. Disponível em:

<<http://java.sun.com/j2ee/1.4/docs/tutorial/doc>>. Acesso em: 27 out. 2006.

SUN MICROSYSTEMS, Java Blueprints – Model View Controller. 2006b. Disponível

em:<<http://java.sun.com/blueprints/patterns/MVC-detailed.html>>. Acesso em: 31 out. 2006.

SUN MICROSYSTEMS, UML Modeling: Creating Use Case Diagrams. 2006c Disponível

em:<http://developers.sun.com/prodtech/javatools/jsenterprise/learning/tutorials/jse8/uml_use_case.html>. Acesso em: 01 nov. 2006.

UML TUTORIAL, Class Diagram. Disponível em:

<http://pigseye.kennesaw.edu/~dbraun/csis4650/A&D/UML_tutorial/class.htm> Acesso em: 1 out. 2006.

VAROTO, A.C. Visões em arquitetura de Software. Disponível em:

<<http://www.ime.usp.br/dcc/posgrad/teses/ane.pdf>>. Acesso em: 30 out. 2006.

XSTREAM, About XStream. Disponível em <<http://xstream.codehaus.org/index.htm>> Acesso em: 19 nov. 2006.

W3C, XML in 10 points. Disponível em: <<http://www.w3.org/XML/1999/XML-in-10-points.html>>.

Acesso em: 27 out. 2006.

APÊNDICE A – MODELOS DE QUESTIONÁRIOS UTILIZADOS

Sistema SARPA (Sistema de Acompanhamento e Registro de Presença Acadêmica) – Cabeçalho dos Questionários

Objetivo do Sistema:

O objetivo do sistema é permitir o registro e acompanhamento, pelos alunos e professores, da presença em aula durante período acadêmico, procurando melhorar o controle e disponibilidade das informações referentes à mesma, para ambas as partes. Serão apresentados, além da presença aula-a-aula, gráficos e estatísticas das presenças ao longo das disciplinas para auxiliar o professor a gerenciar melhor o conteúdo de suas aulas baseado em evidência estatística.

Isso será feito através de validação da presença aula-a-aula, com a passagem da carteirinha USP em uma leitora magnética encontrada em todas as salas dos prédios da Escola Politécnica, respeitando regras que validam a presença.

Objetivo do Questionário:

Levantar junto aos usuários-chave os requisitos do módulo de Regras Validadoras de Presença e o escopo dessas regras.

Essas regras, cadastradas pelos professores, serão as diretrizes para validar ou não a presença do aluno.

A princípio elas não estão definidas, mas como exemplos de regras poderíamos pensar em:

- Porcentagem de tempo da aula acima de 80%;
- Passagem da carteirinha uma vez durante qualquer momento da aula;
- Passagem da carteirinha nos primeiros 20 minutos de aula, etc.

Por isso elaboramos o seguinte questionário, para melhor definir esse módulo de Regras. A maioria das perguntas são abertas propositalmente, para estimular a diversidade nas respostas.

QUESTIONÁRIO DO PROFESSOR

- 1- Há quantos anos leciona na Escola Politécnica?
☐ Até 3 anos
☐ Entre 3 e 5 anos
☐ Entre 5 e 10 anos
☐ Entre 10 e 15 anos
☐ Há mais de 15 anos

- 2- Há quantos anos é professor?
☐ Até 3 anos
☐ Entre 3 e 5 anos
☐ Entre 5 e 10 anos
☐ Entre 10 e 15 anos
☐ Há mais de 15 anos

- 3- Concorda com a presença obrigatória? Porquê?

- 4- Acha que um controle automatizado da presença poderia trazer melhoria em relação ao processo atual? Porquê?

- 5- Que regras para validar a presença você, como professor, acha adequado aplicar?

- 6- De que modo você enxerga que os alunos possam fazer mau uso do sistema, tirando vantagem do mesmo?

- 7- Onde você acha que seria ideal que a leitora de cartões ficasse? Dentro ou fora da sala? Ao lado da porta? Em cima da mesa do professor?

- 8- Espaço livre para comentários:

QUESTIONÁRIO DO ALUNO

- 1- Com que frequência você diria que costuma assistir às aulas?
☐ Sempre
☐ Frequentemente
☐ Em torno de metade
☐ Raramente
☐ Nunca
- 2- Concorda com a presença obrigatória? Porquê?
- 3- Quais vantagens e desvantagens você enxerga no sistema proposto em relação ao atual?
- 4- Você acha que o sistema dificulta a falsificação da presença?
- 5- Que métodos para validar a presença você, como aluno, acha adequado aplicar (exemplo: porcentagem de tempo da aula, passagem da carteirinha uma vez durante a aula, passagem da carteirinha nos primeiros 20 minutos de aula)?
- 6- De que modo você enxerga que o professor pode prejudicar o aluno com o sistema (exemplo: métodos abusivos de validação de presença ou vários métodos distintos que possam confundir o aluno)?
- 7- Onde você acha que seria ideal que a leitora ficasse? Dentro ou fora da sala? Ao lado da porta? Em cima da mesa do professor?
- 8- Espaço livre para comentários:

APÊNDICE B – ESPECIFICAÇÃO DOS REQUISITOS

Sistema SARPA

Especificação de Requisitos de Software

Autores: <i>Carlos Eduardo Pinheiro Vitorino</i> <i>Reginaldo Longano Junior</i> <i>Rodrigo Herculano de Oliveira</i> <i>Vitor Caçador Alexandre</i>	Data de emissão: 21/02/2006
Revisor: <i>Reginaldo Longano Junior</i>	Data de revisão 05/11/2006
Orientadora: <i>Selma Shin Shimizu Melnikoff</i>	



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

Avenida Professor Luciano Gualberto, travessa 3 nº 158 CEP 05508-900 São Paulo SP
Telefone: (11) 3091-5583 Fax (11) 3091-5294

Departamento de Engenharia de Computação e Sistemas Digitais

FOLHA DE CONTROLE DE REVISÕES

Número da versão	Data de emissão	Registro de modificações
00	21/02/2006	Versão Inicial do Documento
01	22/03/2006	Revisão do requisito de validação de presença.
02	05/11/2006	Revisão Final do Documento
03		
04		
05		
06		
07		
08		
09		
10		



Índice

1	Objetivo do Documento	3
2	Objetivo do Sistema.....	4
2.1	Nome do Sistema	4
2.2	Escopo.....	4
3	Descrição Geral	5
3.1	Perspectivas do Produto	5
3.1.1	Interfaces de Usuário	5
3.1.2	Interfaces de Hardware	5
3.1.3	Interfaces com Software.....	6
3.1.4	Operação.....	6
3.2	Funções do Software.....	6
3.3	Restrições.....	7
3.4	Hipóteses e Dependências.....	7
4	Requisitos Específicos.....	9
4.1	Modelo de Casos de Uso	9
4.2	Modelo de Classes	9
4.3	Modelo de Sequência.....	9
4.4	Modelo de Bancos de Dados	9
5	Requisitos Não Funcionais.....	11
6	Padrões de Interface	12
6.1	Estilo da interface.....	12
6.2	Identificação dos principais objetos.....	12
6.3	Padrões Visuais.....	13
6.4	Tipos de dados.....	14
7	Crêrios de Aceitação.....	15



1 Objetivo do Documento

O objetivo deste documento é identificar de forma completa e clara todos os requisitos a serem atendidos pelo sistema a ser desenvolvido. As seções a seguir dão uma visão geral do sistema e enumeram seus requisitos.



2 Objetivo do Sistema

O objetivo do sistema é permitir o registro e acompanhamento, pelos alunos e professores, da presença em aula durante período acadêmico, procurando melhorar o controle e disponibilidade das informações referentes à mesma, para ambas as partes.

2.1 Nome do Sistema

O nome adotado para o sistema é SARPA, sigla para Sistema de Acompanhamento e Registro de Presença Acadêmica.

2.2 Escopo

A motivação do sistema foi a busca pela automatização de processos ineficientes e burocráticos dentro do contexto da universidade. Logo, o sistema tem como meta, implementar uma maneira de automatizar a lista de presença manuscrita, que é o modo atual pela qual a frequência dos alunos em sala de aula é controlada.

Isso será alcançado com leitoras magnéticas alocadas nas salas de aulas e/ou laboratórios, onde devem ser lidas as carteiras de identificação USP dos alunos (lembrando que a carteira de identificação USP é um recurso comum entre o corpo discente). Então, através de regras específicas de cada disciplina para validação de presença, esta será ou não aceita. As leitoras serão integradas em um sistema WEB que permitirá acompanhamento dessas presenças pelos alunos e o gerenciamento de suas disciplinas pelos professores.

Assim, o sistema busca trazer como principais benefícios em relação ao sistema atual um controle mais preciso das presenças, um monitoramento e gerenciamento mais claro das mesmas. Alguns outros benefícios podem ser mencionados como: o acúmulo de dados estatísticos de presença em períodos anteriores, buscando fornecer ao professor um componente importante que é a presença em sala, como um indicador de qualidade das aulas ou nível de interesse da turma; a economia de papel com o fim das listas de chamada impressas; a diminuição das "fraudes" de presença, que ocorrem no método atual, quando um aluno acaba forjando a assinatura de outro.



3 Descrição Geral

Esse sistema integra hardware (computadores, leitores magnéticos e rede) e software (sistema Web). Essas duas partes são totalmente dependentes e, portanto, tem suas especificações com igual importância. O nível de detalhamento adequado é dado nas seções a seguir.

3.1 Perspectivas do Produto

Em primeira fase esse produto é, por completo, um sistema. Ou seja, não é parte de um sistema maior. Porém, em trabalhos futuros, pode-se fazer estudos da sua integração com um sistema global de atribuição de notas e frequências acadêmicas que, no caso da Universidade de São Paulo, é o Júpiter. É importante deixar claro que não é o escopo desse projeto a integração com o sistema Júpiter.

3.1.1 Interfaces de Usuário

A interface operacional com o usuário do sistema será o leitor de cartão magnético, presente nas salas de aula e/ou laboratórios.

A interface de acompanhamento será a típica interface web, com componentes básicos desse tipo de interface, em computadores com acesso a internet e um dos navegadores suportados. Os padrões da interface são estabelecidos no capítulo 6.

3.1.2 Interfaces de Hardware

O sistema fará interface com as leitoras magnéticas, através de uma rede que é melhor descrita no documento de especificação de arquitetura. As informações também ficarão armazenadas num servidor que deve ter capacidade de processamento para suportar a carga de registros a serem inseridos nas tabelas, especialmente durante os picos de início e fim de cada aula.



3.1.3 Interfaces com Software

Como mencionado anteriormente, apesar de ser interessante a integração com o Sistema Júpiter, esse não é o escopo do sistema em sua fase atual.

3.1.4 Operação

O sistema realizará backup automático diariamente, durante a madrugada. E manterá histórico das passagens de cartão por 2 períodos letivos, de modo a englobar um semestre com uma certa tolerância, para que o cálculo das presenças possa ser reavaliado à pedido de alunos e professores, se preciso. O histórico de presenças por sua vez, será mantido para consultas estatísticas.

3.2 Funções do Software

As principais funções do software estão sucintamente definidas a seguir:

- *Cadastro de terminais:* Os terminais terão uma identificação única para serem cadastrados no sistema e serão associados a uma e somente uma sala do prédio.

- *Cadastro de Disciplinas:* O sistema deverá permitir que, antes do início de cada período letivo sejam cadastradas as disciplinas referentes à grade horária das diferentes turmas associada a uma sala. Associado à esse cadastro deve ser associado o parâmetro de porcentagem global de presença, usada como critério de aprovação ou não na disciplina.

- *Cadastro de professores:* Os professores da unidade serão cadastrados para que possa ser feita a associação com as disciplinas que o mesmo leciona, e permitir acesso ao sistema.

- *Cadastro de alunos:* Cada aluno será cadastrado de modo a estar associado às aulas que está cursando naquele período letivo.

- *Registro da presença:* através da passagem da carteira USP por leitoras de cartão magnético nas salas de aula da EP, o aluno registra sua entrada e saída da sala de aula e consegue a presença se seu tempo de permanência em sala atende a critérios de porcentagem de frequência requerida em cada aula, estabelecidos pelo professor para cada disciplina. Desse modo, o processo de registro de presença é automatizado.

- *Consulta de histórico estatístico de presenças:* permitindo comparações como, por exemplo, entre dois períodos de uma disciplina ministrada pelo mesmo professor, entre



duas disciplinas diferentes de um mesmo professor ou entre a mesma disciplina ministrada por professores diferentes.

- *Apoio para análise da programação das disciplinas:* os professores podem associar, nas disciplinas que lecionam, o conteúdo planejado aula a aula para auxiliar na sua análise ao fim da disciplina e, ainda, servir de referência para os alunos.

- *Apresentação do horário atual e da porcentagem de frequência requerida para as disciplinas oferecidas no momento.*

- *Acesso às presenças:* de todos alunos, pelo professor, de disciplinas que este ministra e acesso, pelo aluno, de informações de frequência das disciplinas que esse está matriculado.

- *Pedido da revisão de presença:* o aluno pode requerer, mediante justificativa, a revisão de sua presença em determinada aula.

- *Alteração de parâmetros da aula:* devido a mudanças em salas de aula ou variações nos horários de início e término da mesma. No caso de uma alteração posterior à data da aula, o professor poderá pedir o recálculo das presenças pelo sistema.

- *Impressão de listas pré-formatadas:* no caso de exceções do sistema – como, por exemplo, uma leitora magnética defeituosa – é possível o uso do sistema antigo como contingência.

3.3 Restrições

O sistema apresenta poucas restrições e as que foram levantadas até o momento são elencadas a seguir:

- *Salas sem Terminal:* As salas que não possuem a leitora de cartão não poderão ser salas válidas para alocar disciplinas pertencentes à grade horária. Nestes casos, teria que ser impressa a lista de chamada e as presenças serem inseridas manualmente no sistema.

- *Tamanho do Banco de Dados:* Com uma conta aproximada vimos que, na Escola Politécnica, no pior caso, o banco receberia 30.000 registros de passagem de cartão e 15.000 registros de presença por dia. Portanto, para armazenar uma carga dessas, seria necessário estudar quais os períodos em que o histórico ficaria armazenado no sistema antes de ser feito um backup, para não ultrapassar a capacidade do servidor de banco de dados.

3.4 Hipóteses e Dependências



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

Avenida Professor Luciano Gualberto, travessa 3 nº 158 CEP 05508-900 São Paulo SP
Telefone: (11) 3091-5583 Fax (11) 3091-5294

Departamento de Engenharia de Computação e Sistemas Digitais

O sistema, para estar implantado e 100% operacional, depende das leitoras magnéticas estarem disponíveis no prazo.



4 Requisitos Específicos

Esta seção descreve os requisitos no nível que permita desenvolver o sistema satisfazendo os requisitos do produto e dando base para desenvolvimento do plano de testes, validando esses requisitos.

4.1 Modelo de Casos de Uso

Ref.: "Sistema SARPA – Especificação de Casos de Uso"

4.2 Modelo de Classes

Ref.: "Sistema SARPA – Diagramas de Classe"

4.3 Modelo de Sequência

Ref.: "Sistema SARPA – Diagramas de Sequência"

4.4 Modelo de Bancos de Dados

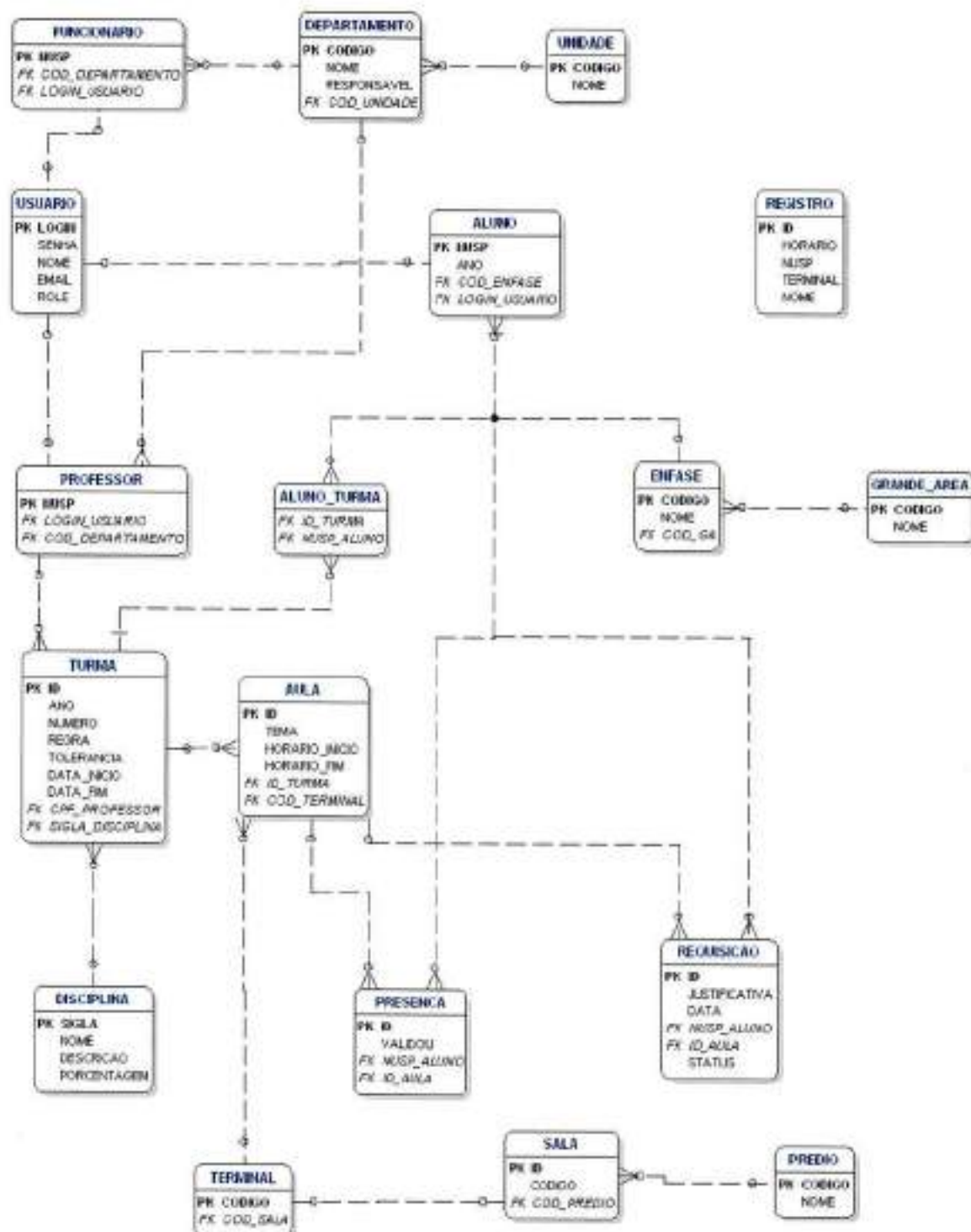
O diagrama entidade-relacionamento, mostrado na à seguir, tem como função ilustrar o relacionamento entre as tabelas de um banco de dados. Oferece ajuda ao programador durante a implementação, pois é um guia rápido de referência do modo como as tabelas conversam entre si, incluindo a cardinalidade do relacionamento.



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

Avenida Professor Luciano Gualberto, travessa 3 nº 158 CEP 05508-900 São Paulo SP
Telefone: (11) 3091-5583 Fax (11) 3091-5294

Departamento de Engenharia de Computação e Sistemas Digitais





5 Requisitos Não Funcionais

Alguns requisitos não funcionais são essenciais para esse sistema e devem ser levados em consideração.

- *Usabilidade*: O sistema deve ser simples de usar e ter uma interface amigável, para realmente estimular o acompanhamento e manutenção das presenças, tanto pelos alunos como pelos professores.

- *Confiabilidade*: A integridade dos dados é importante, para que não seja calculada errada a frequência dos alunos nas disciplinas, pois, no critério atual, a frequência é fator reprovatório das disciplinas.

- *Disponibilidade*: Crítica durante o período das aulas e alta durante outros períodos (acompanhamento e manutenção).

- *Segurança de acesso*: Deve ter perfis de acesso diferente para alunos e professores, e ter senha para acesso, pois o sistema será web e permitirá alterações.

- *Desempenho*: Deverá ter desempenho suficiente para suportar os horários de pico de leitura, ou seja, o início e fim das aulas.



6 Padrões de Interface

6.1 Estilo da interface

A interface do sistema será baseada no estilo web, contendo janelas específicas para cada função a ser requerida.

Para cada funcionalidade, a consulta às informações do sistema ou outras operações de navegação serão realizadas através de manipulação direta.

As operações de cadastro e os inputs de informações relativas a lógica do negócio serão realizados através de formulários, facilitando o procedimento e evitando que ocorra um desvio de padrões.

6.2 Identificação dos principais objetos

Os objetos mais relevantes manipulados na interface serão:

- MENUS:

Haverá um menu localizado em uma barra horizontal abaixo do LOGO do sistema com itens referentes as funcionalidades do sistema. Cada um desses itens dará acesso a um outro menu no formato *dropdown*, com novas opções de acesso.

- BOTÕES:

Os botões são padrão web com fundo cinza e fonte preta.

- FORMULÁRIOS:

Serão utilizados para o preenchimento das informações de cadastro

As cores predominantes no portal serão: AZUL, AMARELO, BRANCO, PRETO e CINZA.



6.3 Padrões Visuais

As telas, de modo geral, seguirão o layout apresentado abaixo:

The screenshot shows the SARPA web application interface. At the top, there is a blue header with the 'sarpa' logo in yellow and the text 'sistema de acompanhamento e registro de presença acadêmica' in blue. Below the header is a navigation bar with links: CADASTROS, GRADE HORÁRIA, DISCIPLINAS, IMPRESSÃO DE LISTAS, and LOGOUT. The main content area is titled 'CADASTRO DE ALUNOS' and contains a form with fields for NUSP, Nome, Grande Área, Ênfase, Ano, and E-mail. There are 'OK' and 'Cancelar' buttons at the bottom of the form.

- O menu superior terá fundo azul e fonte branca.
- Os formulários, inclusive os botões, serão padrão Windows XP.
- O conteúdo das páginas terá fundo cinza e elementos centralizados com fonte preta;



6.4 Tipos de dados

Os tipos de dados inseridos nos campos terão padrões que estarão demonstrados ao lado dos campos que os requerem.

Os tipos críticos (data, hora e numéricos) estão definidos abaixo:

Data – DD/MM/AAAA

Hora – HH:MM



7 Critérios de Aceitação

A aceitação do sistema será por partes, visto que o requisito de desempenho não poderá ser testado, pois é necessária uma massa de testes grandes e uma carga grande de registros de presença simultaneamente ocorrendo. Para tal teste também seria necessário a instalação de diversas leitoras magnéticas e a verba inicial destinada ao projeto não prevê a aquisição dos equipamentos em tal quantidade.

APÊNDICE C – ESPECIFICAÇÃO DA ARQUITETURA

Sistema SARPA

Especificação de Arquitetura do Sistema

Autores: <i>Carlos Eduardo Pinheiro Vitorino</i> <i>Reginaldo Longano Junior</i> <i>Rodrigo Herculano de Oliveira</i> <i>Vitor Caçador Alexandre</i>	Data de emissão: 23/03/2006
Revisor: <i>Carlos Eduardo Pinheiro Vitorino</i>	Data de revisão 05/11/2006
Orientadora: <i>Profa. Dra. Selma Shin Shimizu Melnikoff</i>	
Versão 2.0	



FOLHA DE CONTROLE DE REVISÕES

Número da versão	Data de emissão	Registro de modificações
00	23/03/2006	Versão inicial
01	26/06/2006	Primeira Revisão
02	05/11/2006	Revisão Final
03		
04		
05		
06		
07		
08		
09		
10		



Índice

1	Objetivo do Documento	3
2	Descrição do projeto	4
3	Especificação da plataforma de desenvolvimento	5
4	Descrição da arquitetura de software	6
4.1	Diagrama de Classes	6
4.2	Arquitetura de software.....	7
4.3	Descrição do conteúdo dos packages	8
5	Descrição da arquitetura de hardware.....	10
5.1	Visão Geral da Arquitetura	10
5.2	Diagrama de Implantação.....	Erro! Indicador não definido.
5.3	Descrição dos nós e dos relacionamentos	11
5.4	Modelo abstraído do diagrama	Erro! Indicador não definido.
5.4	11
6	Alocação do software ao hardware.....	Erro! Indicador não definido.
6.1	Descrição.....	Erro! Indicador não definido.
7	Plano de integração.....	13
7.1	Estratégia de implementação, integração e teste	13
7.2	Ambiente de teste.....	14



1 Objetivo do Documento

O objetivo deste documento é descrever a arquitetura do sistema, através da arquitetura de software, arquitetura de hardware e a alocação dos elementos de software na arquitetura de hardware.

A arquitetura de software é representada através de packages e as dependências entre eles. A arquitetura de hardware é representada através do Diagrama de Hardware. A alocação dos elementos de software no hardware é representada através da distribuição dos packages no Diagrama de Alocação.



2 Descrição do projeto

O objetivo do sistema é permitir o registro e acompanhamento, pelos alunos e professores, da presença em aula durante período acadêmico, procurando melhorar o controle e disponibilidade das informações referentes à mesma, para ambas as partes.

A motivação do sistema foi a busca pela automatização de processos ineficientes e burocráticos dentro do contexto da universidade. Logo, o sistema tem como meta, implementar uma maneira de automatizar a lista de presença manuscrita, que é o modo atual pela qual a frequência dos alunos em sala de aula é controlada.

Isso será alcançado com leitoras magnéticas alocadas nas salas de aulas e/ou laboratórios, onde devem ser lidas as carteiras de identificação USP dos alunos (lembrando que a carteira de identificação USP é um recurso comum entre o corpo discente). Então, através de regras de porcentagem de tempo em aula, configurável para cada disciplina para validação de presença, esta será ou não aceita. As leitoras serão integradas em um sistema web que permitirá acompanhamento dessas presenças pelos alunos e o gerenciamento de suas disciplinas pelos professores.

Assim, o sistema busca trazer como principais benefícios em relação ao sistema atual um controle mais preciso das presenças, um monitoramento e gerenciamento mais claro das mesmas. Alguns outros benefícios podem ser mencionados como: o acúmulo de dados estatísticos de presença em períodos anteriores, buscando fornecer ao professor um componente importante que é a presença em sala, como um indicador de qualidade das aulas ou nível de interesse da turma; a economia de papel com o fim das listas de chamada impressas; a diminuição das "fraudes" de presença, que ocorrem no método atual, quando um aluno forja a assinatura de outro.



3 Especificação da plataforma de desenvolvimento

❑ **Linguagem de programação:**

Java, pois se trata de uma plataforma bastante sólida e confiável.

❑ **Ferramenta IDE:**

Eclipse 3.1, por ser uma ferramenta livre, largamente utilizada no mercado e que oferece boa integração com as APIs Java.

❑ **Sistema Operacional:**

Windows / Linux. Por se tratar de um sistema desenvolvido em Java, o sistema SARPA será multiplataforma.

❑ **Controle de Versão:**

CVS (Concurrent Version System), que servirá também como repositório central do sistema. Será utilizado devido ao processo distribuído de programação que será adotado, com alterações simultâneas nos mesmos recursos (códigos e documentos), otimizando assim a fase de implementação.

❑ **Servidor WEB:**

Apache Tomcat 5.5, pois é um servidor HTTP confiável, com suporte a Java.

❑ **Gerenciador de Banco de Dados:**

MySQL Server 4.1.2, por ser um servidor bastante popular e livre para desenvolvimento.

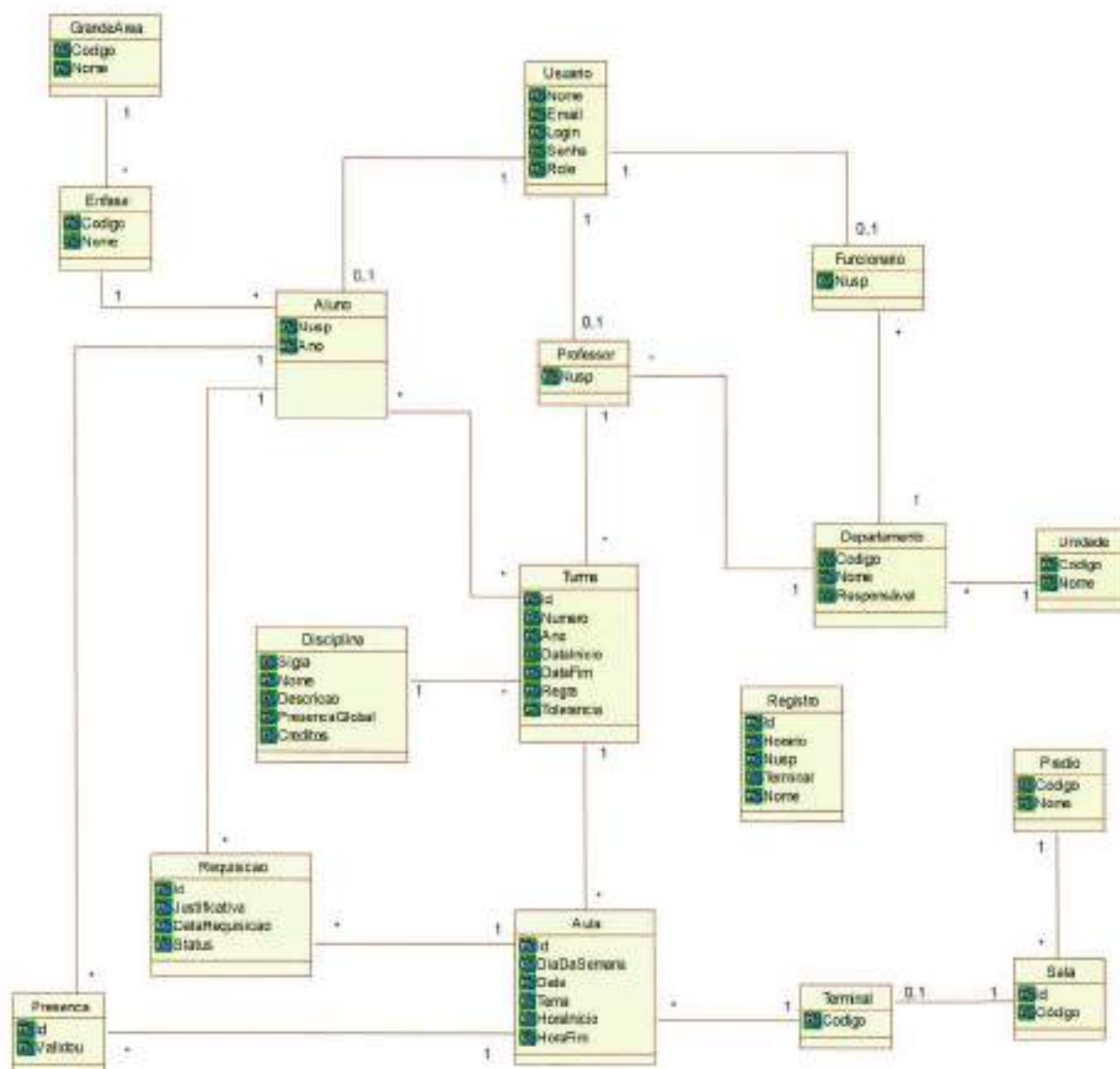
❑ **Ferramentas adicionais:**

- *STRUTS* – Framework utilizado para implementar o desenvolvimento do sistema em camadas.
- *HIBERNATE* - Framework utilizado para realizar mapeamento objeto relacional.
- *JUNIT* – Utilizado para criação de testes automáticos de classes do sistema.



4 Descrição da arquitetura de software

4.1 Diagrama de Classes





4.3 Descrição do conteúdo dos packages

Gerenciamento de Registro e Presença

Esse módulo é responsável por receber as informações sobre as carteiras USP que estão sendo enviadas pelas diversas leitoras USB. Este módulo faz uso da ferramenta XStream para gerar o arquivo XML referente a cada uma um dos registros (cada vez que a carteira é lida) e também é responsável por recuperar os arquivos e reconvertê-los em objeto para fazer o envio ao módulo Controller juntamente com o pedido (ação) de persistência

Controller

O modulo Controller é responsável por converter os diversos pedidos de ação em pedidos de persistência. Frequentemente, um pedido de ação implica em mais de um pedido de persistência no banco de dados, ou seja, pode ser necessário inserir mais de um objeto por pedido. Após determinar quais são os objetos que devem ser persistidos, eles são então enviados para o Gerenciador de Persistência que os armazenará no banco de dados.

Gerenciador de Persistência

O gerenciamento de persistência faz uso da ferramenta Hibernate para converter os objetos recebidos e inserir os registros necessários no banco de dados, realizando a consistência de atributos necessária.

Testes

O módulo de testes possui a função de, através do módulo de gerenciamento de persistência, popular o banco de dados com as informações que serão utilizadas para o teste do sistema. É particularmente útil, pois sempre que necessário é possível retornar o banco de dados em um estado inicial conhecido.

Outra função é testar todos os modos de persistência e recuperação de dados garantindo que o modulo de gerenciamento de persistência está funcionando corretamente para cada bean.

Modelagem e Configuração do Banco de Dados

É responsável por receber o mapeamento dos diferentes beans capturando os tags inseridos para então modelar e gerar automaticamente a estrutura do banco de dados.



Model

O módulo Model consiste nas classes que representam as entidades do sistema. Cada classe possui métodos para interação com o banco de dados e métodos de validação de persistência.

View

É responsável por fazer a interface entre o usuário e o sistema, recebendo os comandos do usuário e enviando-os para o modulo Action. Também é responsável por todas as informações exibidas na tela para o usuário, além de monitorar o módulo Form para qualquer mudança seja refletida na interface.

Action

Sua função é receber e executar as ações comandadas pelo usuário na interface (View). As informações utilizadas pelo modulo Action são obtidas através do modulo Form e não da própria interface View como se pode pensar em um primeiro momento.

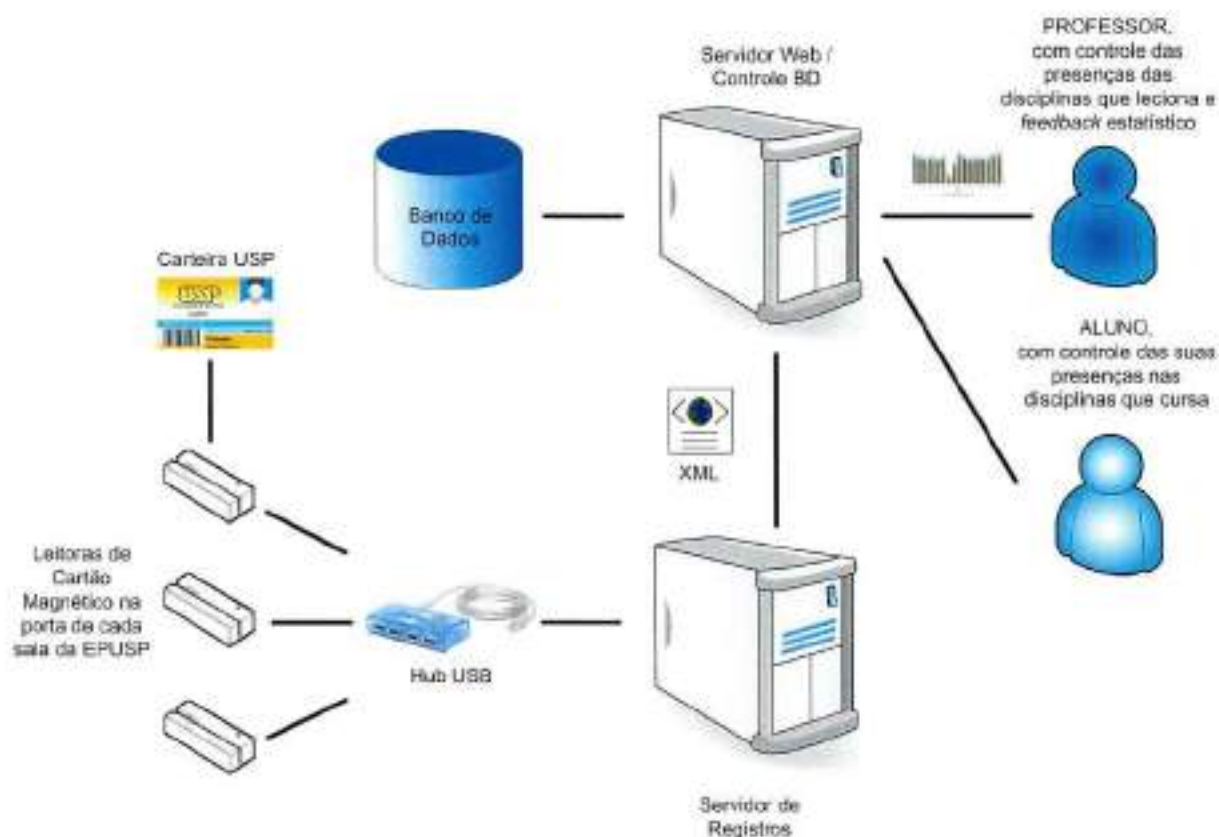
Form

O módulo Form é responsável por capturar e alimentar as informações que são exibidas na interface do usuário (View). Sua construção é bastante simples e consiste principalmente de métodos get e set, que fazem a comunicação com a interface.



5 Descrição da arquitetura de hardware

5.1 Diagrama de Hardware





5.2 Descrição dos nós e dos relacionamentos

Bloco de Leitura

Optou-se pela tecnologia USB nas leitoras por uma série de razões dentre as quais destacamos a facilidade de concatenar as leitoras através de cascata ou hub e a transparência quanto à tecnologia de registro da presença, podendo variar entre a própria leitura magnética, RFID ou mesmo leitores biométricos. Além disso, no padrão USB é possível concatenar até 127 dispositivos seja através de hub ou através de conexão em cascata. Esse fator foi um ponto chave de decisão quanto à tecnologia utilizada, já que havia a opção de utilização de leitoras seriais, mas que demandaria o desenvolvimento de um hub serial, o que foge do escopo do projeto.

O hub USB por si só fornece alimentação para os dispositivos, pois, normalmente, possui sua própria fonte de alimentação, evitando que outros cabos tenham que ser utilizados, aumentando os custos de implantação do sistema.

Um outro fator é o de que o sistema já está preparado para trabalhar com a porta USB do servidor, e qualquer upgrade realizado em uma versão futura muito provavelmente utilizará essa interface.

Bloco Servidor de Registros

Como segundo bloco tem-se o servidor de registros. O servidor de registros é responsável por monitorar todas as leitoras USB ligadas a ele, gerando um arquivo XML contendo as informações dos cartões para que sejam importados pelo servidor de dados. Esse arquivo XML fica armazenado temporariamente no próprio servidor de registro, que de tempos em tempos irá enviá-los ao servidor web para que possam ser inseridos no banco de dados.

Todo o trabalho de monitoração, criação e envio do arquivo XML ao servidor de banco de dados é feito através de um serviço desenvolvido pelo grupo, que trabalha de forma independente do sistema SARPA. Isso traz grande flexibilidade o sistema, já que se, em uma versão futura, houver a necessidade de implantação de outro método de identificação, como biométrico, apenas o serviço deverá ser alterado ou expandido, restando apenas ajustes mínimos ao SARPA.

Idealmente, os servidores – de registro e web – deveriam estar separados, devido ao fato de que monitorar a porta USB constantemente exige capacidade de processamento considerável, que em simulações feitas chegou a passar de 80% de utilização do processador em uma máquina Intel Pentium III. Caso o servidor web estivesse rodando na mesma máquina, poderia haver lentidão no servidor e, em casos extremos, arquivos XML poderiam deixar de ser gerados.



Outro fator determinante para essa separação origina-se do fato de que o servidor web é muito mais exposto a qualquer tentativa de ataque externo se comparado a um servidor interno. Para evitar que um ataque ao servidor web derrubasse todo o módulo de registro de presença, a separação mostrou-se como a melhor escolha.

Nessa arquitetura, mesmo que o servidor web saia do ar, os registros continuam sendo armazenados no servidor de registros através dos arquivos XML para que quando o servidor web voltar a estar online, esses arquivos sejam transferidos para ele.

Porém, é importante ressaltar que devido à restrições de custo e para fins de apenas ter um protótipo funcional do sistema, a arquitetura SARPA foi implementada em um único servidor.

Bloco servidor web

O terceiro bloco compreende o banco de dados e o servidor de acesso web, pelo qual os usuários acessam o sistema.

O servidor web, conforme descrito anteriormente, monitora uma pasta de entrada buscando arquivos XML contendo dados dos registros de presença que são, então, inseridos em um banco de dados temporário.

Caracterizou-se esse banco de dados temporário pelo fato de que os registros ficam armazenados nele por pelo menos um ano, para que caso haja algum pedido de revisão de presença, os registros possam ser importados novamente.

Além do banco de dados, o servidor web mantém o serviço HTTP responsável por permitir o acesso web dos usuários. Esse serviço é disponibilizado pela ferramenta Tomcat



6 Plano de integração

6.1 *Estratégia de implementação, integração e teste*

No início da implementação do sistema, cada integrante da equipe SARPA se responsabilizará por um conjunto de classes. A divisão de tarefa será inicialmente baseada em classes e não em casos de uso, para evitar a concorrência na implementação de classes comuns entre dois ou mais casos de uso, em especial nas classes persistentes.

Após a implementação das classes mais básicas do sistema, partiremos para a implementação das classes de negócio que, por sua vez, estão associadas a um caso de uso. Os casos de uso com maior prioridade na implementação serão aqueles que tratam da parte de negócio do sistema, como por exemplo, o registro de presença.

Já a parte de interface do sistema será abordada por último, quando o sistema já estiver com a parte funcional pronta.

Para realizar a integração dos diversos módulos do sistema, o grupo realizará reuniões voltadas para esse propósito. A estratégia é realizar pequenas integrações juntamente com as outras tarefas de desenvolvimento, para evitar-se chegar à fase final do projeto com cada parte do sistema implementada de uma forma diferente, atrapalhando a conclusão do projeto e gerando re-trabalho.

Os testes serão divididos em duas etapas principais: testes unitários e testes de integração.



6.2 Ambiente de teste

Por se tratar de um projeto com parte do sistema baseada em uma aplicação web, ele deverá estar apto a ser executado nos navegadores mais conhecidos do mercado. Os testes serão realizados nos navegadores Firefox (Mozilla) e Internet Explorer (Microsoft), porém o sistema será desenvolvido para ser melhor visualizado no Firefox.

Durante o desenvolvimento serão realizados dois tipos de testes, cada um com suas peculiaridades:

Testes unitários

São testes executados automaticamente à medida que as classes do sistema são compiladas. Esses testes são criados em classes especiais que testam diversas funcionalidades das classes do sistema. Para isto utilizaremos o framework JUnit.

Testes de integração

Esses testes serão realizados pela equipe após cada evento de integração do sistema. O objetivo desses testes é verificar a consistência do sistema em relação à integração e as funcionalidades.

APÊNDICE D – ESPECIFICAÇÃO DOS CASOS DE USO

Sistema SARPA

Especificação de Casos de Uso

Relatório Técnico

Autores: <i>Carlos Eduardo Pinheiro Vitorino</i> <i>Reginaldo Longano Junior</i> <i>Rodrigo Herculano de Oliveira</i> <i>Vitor Caçador Alexandre</i>	Data de emissão: 21/02/2006
Revisor: <i>Vitor Caçador Alexandre</i>	Data de revisão 05/11/2006

FOLHA DE CONTROLE DE REVISÕES



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

Avenida Professor Luciano Gualberto, travessa 3 nº 156 CEP 05508-900 São Paulo SP
Telefone: (011) 3091-5583 Fax (011) 3091-5294

Departamento de Engenharia de Computação e Sistemas Digitais

Número da versão	Data de emissão	Registro de modificações
00	21/02/2006	Criação do Documento
01	02/03/2006	Revisão do Documento
02	14/10/2006	Revisão do Documento
03	05/11/2006	Revisão Final
04		
05		
06		
07		
08		
09		
10		



Índice

1	Objetivo do Documento	3
2	Diagrama de Casos de Uso.....	4
3	Descrição dos Casos de Uso	5
3.1	Cadastros Gerais.....	5
3.1.1	Cadastro de aluno	5
3.1.2	Cadastro de professor.....	6
3.1.3	Cadastro de funcionário	8
3.1.4	Cadastro de disciplina	10
3.1.5	Cadastro de turma.....	12
3.1.6	Cadastro de terminal	13
3.2	Cadastros Complementares.....	15
3.2.1	Cadastro de grande área.....	15
3.2.2	Cadastro de ênfase	17
3.2.3	Cadastro de unidade	18
3.2.4	Cadastro de departamento.....	20
3.2.5	Cadastro de prédio	22
3.2.6	Cadastro de sala	23
3.3	Grade Horária.....	25
3.3.1	Caso de uso: Consulta de grade horária.....	25
3.4	Disciplinas	26
3.4.1	Caso de uso: Associação de alunos em uma disciplina – por ênfase.....	26
3.4.2	Caso de uso: Associação de alunos em uma disciplina – um-a-um	26
3.4.3	Caso de uso: Associação de alunos em uma disciplina – consultar	27
3.4.4	Caso de uso: Recálculo de presença.....	27
3.4.5	Caso de uso: Programação aula-a-aula	28
3.4.6	Caso de uso: Consulta / Ajuste de presença (professor).....	28
3.4.7	Caso de uso: Consulta de presença (aluno).....	28
3.4.8	Caso de uso: Requerimento de recálculo de presença.....	29
3.5	Estatísticas	29
3.5.1	Caso de uso: Geração de estatística	29
3.6	Listas	30
3.6.1	Caso de uso: Impressão de lista	30
3.7	Registro de presença	30
3.7.1	Caso de uso: Registro de presença	30
4	Descrição dos Atores.....	32



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

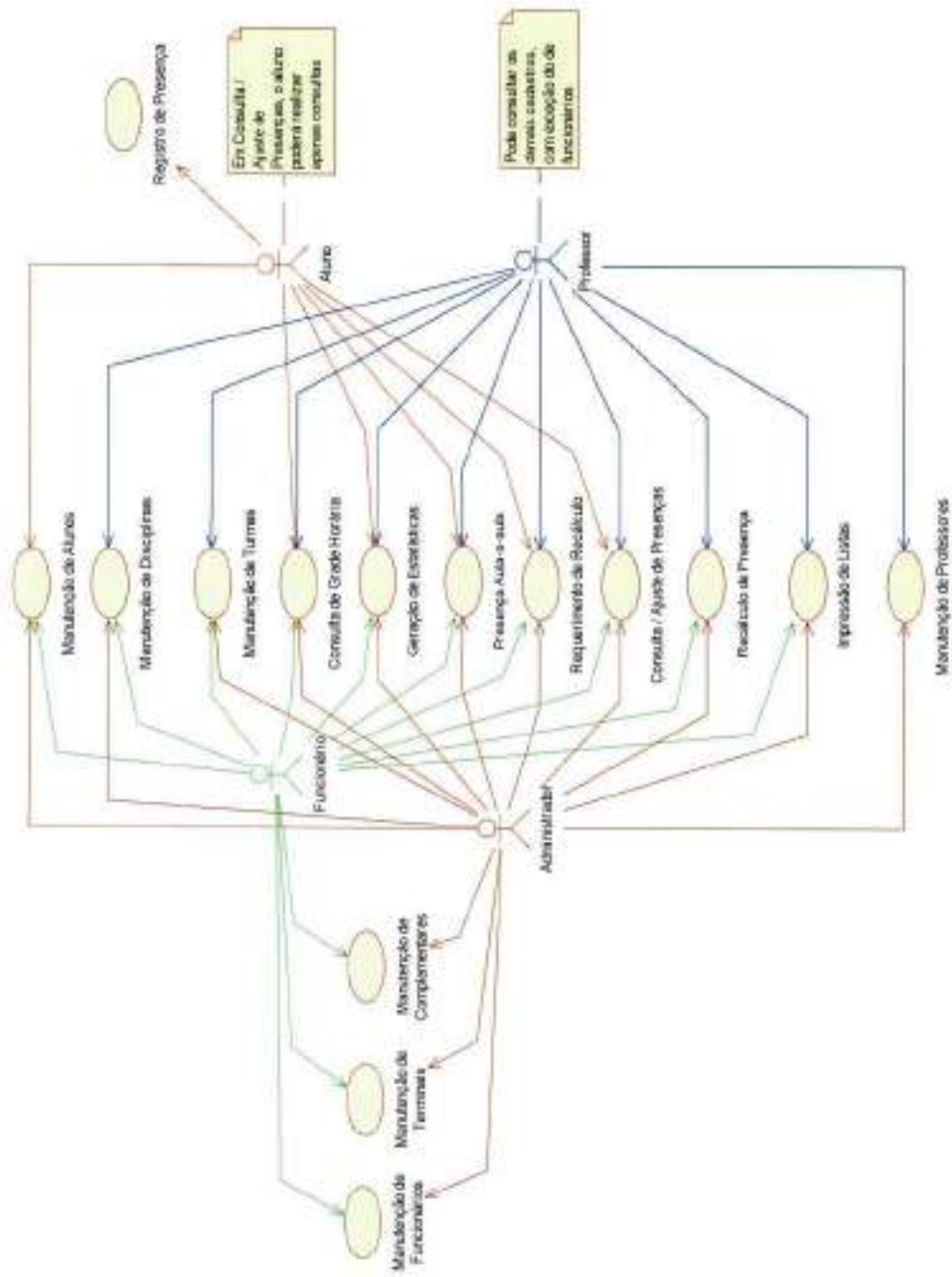
Avenida Professor Luciano Gualberto, travessa 3 nº 158 CEP 05508-900 São Paulo SP
Telefone: (011) 3091-5593 Fax (011) 3091-5294

Departamento de Engenharia de Computação e Sistemas Digitais

1 Objetivo do Documento

Este documento tem como objetivo fornecer uma descrição detalhada dos Casos de Uso do sistema SARPA – Sistema de Acompanhamento e Registro de Presença Acadêmica.

2 Diagrama de Casos de Uso





3 Descrição dos Casos de Uso

3.1 Cadastros Gerais

O módulo *Cadastros Gerais* deverá permitir a manipulação dos cadastros básicos do sistema: Alunos, Professores, Disciplinas, Turmas e Terminais.

3.1.1 Cadastro de aluno

3.1.1.1 Caso de uso: Inserção de aluno

Descrição: Este caso de uso descreve a inserção de um novo aluno

Evento iniciador: solicitação de inserção de aluno

Atores: administrador, funcionário

Pré-condição: usuário autenticado como administrador ou funcionário

Seqüência de eventos:

1. Usuário solicita operação de cadastro de aluno no menu de cadastro
2. Sistema solicita os dados preliminares relativos ao novo aluno
3. Usuário solicita inserção do aluno
4. Usuário insere os dados do novo aluno
5. Sistema insere o novo cadastro e exibe mensagem de sucesso
6. Usuário finaliza a operação

Pós-condição: novo aluno inserido

Extensões:

1. Aluno já existe: sistema apresenta mensagem ao usuário (passo 4)
2. Dados fornecidos do novo aluno não consistentes (passo 4)

Inclusão: busca aluno (passo 4)

3.1.1.2 Caso de uso: Remoção de aluno

Descrição: Este caso de uso descreve a remoção de um aluno

Evento iniciador: solicitação de remoção de aluno

Atores: administrador, funcionário

Pré-condição: usuário autenticado como administrador ou funcionário

Seqüência de eventos:

1. Usuário solicita operação de cadastro de aluno
2. Sistema solicita a identificação do aluno a ser manipulado
3. Usuário insere a identificação do aluno (nusp / nome)
4. Sistema exibe o aluno desejado
5. Usuário seleciona a opção de remoção de aluno
6. Sistema confirma a remoção de aluno
7. Usuário finaliza a operação

Pós-condição: aluno removido



Extensão: não existe

Inclusão: busca aluno (passo 4)

3.1.1.3 Caso de uso: Alteração de aluno

Descrição: Este caso de uso descreve a alteração de um aluno

Evento iniciador: solicitação de alteração de aluno

Atores: administrador, funcionário, aluno

Pré-condição: usuário autenticado como administrador, funcionário ou aluno

Seqüência de eventos:

1. Usuário solicita operação de cadastro de aluno
2. Sistema solicita a identificação do aluno a ser alterado
3. Usuário insere a identificação do aluno (nusp / nome)
4. Sistema exibe o aluno desejado.
5. Usuário seleciona a opção de alteração de aluno
6. Usuário faz as alterações desejadas
7. Usuário finaliza a operação

Pós-condição: aluno alterado

Extensão: dados fornecidos do aluno não consistentes (passo 6): sistema apresenta mensagem ao usuário

Inclusão: busca aluno (passos 4)

3.1.1.4 Caso de uso: Consulta de aluno

Descrição: Este caso de uso descreve a consulta de um aluno

Evento iniciador: solicitação de consulta de aluno

Atores: funcionário, administrador, professor e aluno

Pré-condição: usuário autenticado como funcionário, administrador, professor ou aluno

Seqüência de eventos:

1. Usuário solicita operação de cadastro de aluno
2. Sistema solicita a identificação do aluno a ser consultado
3. Usuário insere a identificação do aluno (nusp / nome)
4. Sistema apresenta os dados do aluno
5. Usuário finaliza a operação

Pós-condição: dados do aluno apresentados

Extensão: não existe

Inclusão: busca aluno (passo 4)

3.1.2 Cadastro de professor



3.1.2.1 Caso de uso: Inserção de professor

Descrição: Este caso de uso descreve a inserção de um novo professor

Evento iniciador: solicitação de inserção de professor

Atores: administrador

Pré-condição: usuário autenticado como administrador

Seqüência de eventos:

1. Usuário solicita operação de cadastro de professor no menu de cadastro
2. Sistema solicita os dados preliminares relativos ao novo professor
3. Usuário solicita inserção do professor
4. Usuário insere os dados do novo professor
5. Sistema insere o novo cadastro e exibe mensagem de sucesso
6. Usuário finaliza a operação

Pós-condição: novo professor inserido

Extensões:

1. Professor já existe: sistema apresenta mensagem ao usuário (passo 4)
2. Dados fornecidos do novo professor não consistentes (passo 4)

Inclusão: busca professor (passo 4)

3.1.2.2 Caso de uso: Remoção de professor

Descrição: Este caso de uso descreve a remoção de um professor

Evento iniciador: solicitação de remoção de professor

Atores: administrador

Pré-condição: usuário autenticado como administrador

Seqüência de eventos:

1. Usuário solicita operação de cadastro de professor
2. Sistema solicita a identificação do professor a ser manipulado
3. Usuário insere a identificação do professor (nusp / nome)
4. Sistema exibe o professor desejado
5. Usuário seleciona a opção de remoção de professor
6. Sistema confirma a remoção de professor
7. Usuário finaliza a operação

Pós-condição: professor removido

Extensão: não existe

Inclusão: busca professor (passo 4)

3.1.2.3 Caso de uso: Alteração de professor

Descrição: Este caso de uso descreve a alteração de um professor

Evento iniciador: solicitação de alteração de professor

Atores: administrador, professor

Pré-condição: usuário autenticado como administrador ou professor

Seqüência de eventos:

1. Usuário solicita operação de cadastro de professor
2. Sistema solicita a identificação do professor a ser alterado
3. Usuário insere a identificação do professor (nusp / nome)



4. Sistema exibe o professor desejado.
5. Usuário seleciona a opção de alteração de professor
6. Usuário faz as alterações desejadas
7. Usuário finaliza a operação

Pós-condição: professor alterado

Extensão: dados fornecidos do professor não consistentes (passo 6): sistema apresenta mensagem ao usuário

Inclusão: busca professor (passos 4)

3.1.2.4 Caso de uso: Consulta de professor

Descrição: Este caso de uso descreve a consulta de um professor

Evento iniciador: solicitação de consulta de professor

Atores: administrador, professor

Pré-condição: usuário autenticado como administrador ou professor

Seqüência de eventos:

1. Usuário solicita operação de cadastro de professor
2. Sistema solicita a identificação do professor a ser consultado
3. Usuário insere a identificação do professor (nusp / nome)
4. Sistema apresenta os dados do professor
5. Usuário finaliza a operação

Pós-condição: dados do professor apresentados

Extensão: não existe

Inclusão: busca professor (passo 4)

3.1.3 Cadastro de funcionário

3.1.3.1 Caso de uso: Inserção de funcionário

Descrição: Este caso de uso descreve a inserção de um novo funcionário

Evento iniciador: solicitação de inserção de funcionário

Atores: administrador

Pré-condição: usuário autenticado como administrador

Seqüência de eventos:

1. Usuário solicita operação de cadastro de funcionário no menu de cadastro
2. Sistema solicita os dados preliminares relativos ao novo funcionário
3. Usuário solicita inserção do funcionário
4. Usuário insere os dados do novo funcionário
5. Sistema insere o novo cadastro e exibe mensagem de sucesso
6. Usuário finaliza a operação

Pós-condição: novo funcionário inserido

Extensões:

1. Funcionário já existe: sistema apresenta mensagem ao usuário (passo 4)



2. Dados fornecidos do novo funcionário não consistentes (passo 4)

Inclusão: busca funcionário (passo 4)

3.1.3.2 Caso de uso: Remoção de funcionário

Descrição: Este caso de uso descreve a remoção de um funcionário

Evento iniciador: solicitação de remoção de funcionário

Atores: administrador

Pré-condição: usuário autenticado como administrador

Seqüência de eventos:

1. Usuário solicita operação de cadastro de funcionário
2. Sistema solicita a identificação do funcionário a ser manipulado
3. Usuário insere a identificação do funcionário (nusp / nome)
4. Sistema exibe o funcionário desejado
5. Usuário seleciona a opção de remoção de funcionário
6. Sistema confirma a remoção de funcionário
7. Usuário finaliza a operação

Pós-condição: funcionário removido

Extensão: não existe

Inclusão: busca funcionário (passo 4)

3.1.3.3 Caso de uso: Alteração de funcionário

Descrição: Este caso de uso descreve a alteração de um funcionário

Evento iniciador: solicitação de alteração de funcionário

Atores: administrador, funcionário

Pré-condição: usuário autenticado como administrador ou funcionário

Seqüência de eventos:

1. Usuário solicita operação de cadastro de funcionário
2. Sistema solicita a identificação do funcionário a ser alterado
3. Usuário insere a identificação do funcionário (nusp / nome)
4. Sistema exibe o funcionário desejado.
5. Usuário seleciona a opção de alteração de funcionário
6. Usuário faz as alterações desejadas
7. Usuário finaliza a operação

Pós-condição: funcionário alterado

Extensão: dados fornecidos do funcionário não consistentes (passo 6): sistema apresenta mensagem ao usuário

Inclusão: busca funcionário (passos 4)

3.1.3.4 Caso de uso: Consulta de funcionário

Descrição: Este caso de uso descreve a consulta de um funcionário

Evento iniciador: solicitação de consulta de funcionário

Atores: funcionário, administrador



Pré-condição: usuário autenticado como funcionário ou administrador

Seqüência de eventos:

1. Usuário solicita operação de cadastro de funcionário
2. Sistema solicita a identificação do funcionário a ser consultado
3. Usuário insere a identificação do funcionário (nusp / nome)
4. Sistema apresenta os dados do funcionário
5. Usuário finaliza a operação

Pós-condição: dados do funcionário apresentados

Extensão: não existe

Inclusão: busca funcionário (passo 4)

3.1.4 Cadastro de disciplina

3.1.4.1 Caso de uso: Inserção de disciplina

Descrição: Este caso de uso descreve a inserção de uma nova disciplina

Evento iniciador: solicitação de inserção de disciplina

Atores: administrador, funcionário

Pré-condição: usuário autenticado como administrador ou funcionário

Seqüência de eventos:

1. Usuário solicita operação de cadastro de disciplina no menu de cadastro
2. Sistema solicita os dados preliminares relativos à disciplina
3. Usuário solicita inserção da disciplina
4. Usuário insere os dados da nova disciplina
5. Sistema insere o novo cadastro e exibe mensagem de sucesso
6. Usuário finaliza a operação

Pós-condição: nova disciplina inserida

Extensões:

1. Disciplina já existe: sistema apresenta mensagem ao usuário (passo 4)
2. Dados fornecidos da nova disciplina não consistentes (passo 4)

Inclusão: busca disciplina (passo 4)

3.1.4.2 Caso de uso: Remoção de disciplina

Descrição: Este caso de uso descreve a remoção de uma disciplina

Evento iniciador: solicitação de remoção de disciplina

Atores: administrador, funcionário

Pré-condição: usuário autenticado como administrador ou funcionário

Seqüência de eventos:

1. Usuário solicita operação de cadastro de disciplina
2. Sistema solicita a identificação da disciplina a ser manipulada
3. Usuário insere a identificação da disciplina (sigla / nome)



4. Sistema exibe a disciplina desejada
5. Usuário seleciona a opção de remoção de disciplina
6. Sistema confirma a remoção de disciplina
7. Usuário finaliza a operação

Pós-condição: disciplina removida

Extensão: não existe

Inclusão: busca disciplina (passo 4)

3.1.4.3 Caso de uso: Alteração de disciplina

Descrição: Este caso de uso descreve a alteração de uma disciplina

Evento iniciador: solicitação de alteração de disciplina

Atores: administrador, funcionário

Pré-condição: usuário autenticado como administrador ou funcionário

Seqüência de eventos:

1. Usuário solicita operação de cadastro de disciplina
2. Sistema solicita a identificação da disciplina a ser alterado
3. Usuário insere a identificação da disciplina (sigla / nome)
4. Sistema exibe a disciplina desejada.
5. Usuário seleciona a opção de alteração de disciplina
6. Usuário faz as alterações desejadas.
7. Usuário finaliza a operação

Pós-condição: disciplina alterada

Extensão: dados fornecidos da disciplina não consistentes (passo 6): sistema apresenta mensagem ao usuário

Inclusão: busca disciplina (passos 4)

3.1.4.4 Caso de uso: Consulta de disciplina

Descrição: Este caso de uso descreve a consulta de uma disciplina

Evento iniciador: solicitação de consulta de disciplina

Atores: funcionário, administrador, professor, aluno

Pré-condição: usuário autenticado como funcionário, administrador, professor ou aluno

Seqüência de eventos:

1. Usuário solicita operação de cadastro de disciplina
2. Sistema solicita a identificação da disciplina a ser consultada
3. Usuário insere a identificação da disciplina (sigla / nome)
4. Sistema apresenta os dados da disciplina
5. Usuário finaliza a operação

Pós-condição: dados da disciplina apresentados

Extensão: não existe

Inclusão: busca disciplina (passo 4)



3.1.5 Cadastro de turma

3.1.5.1 Caso de uso: Inserção de turma

Descrição: Este caso de uso descreve a inserção de uma nova turma

Evento iniciador: solicitação de inserção de turma

Atores: administrador, funcionário

Pré-condição: usuário autenticado como administrador ou funcionário

Seqüência de eventos:

1. Usuário solicita operação de cadastro de turma no menu de cadastro
2. Sistema solicita os dados preliminares relativos à turma
3. Usuário solicita inserção da turma
4. Usuário insere os dados da nova turma
5. Sistema insere o novo cadastro e exibe mensagem de sucesso
6. Usuário finaliza a operação

Pós-condição: nova turma inserida, com as aulas correspondentes ao período

Extensões:

1. Turma já existe: sistema apresenta mensagem ao usuário (passo 4)
2. Dados fornecidos da nova turma não consistentes (passo 4)

Inclusão: busca turma (passo 4)

3.1.5.2 Caso de uso: Remoção de turma

Descrição: Este caso de uso descreve a remoção de uma turma

Evento iniciador: solicitação de remoção de turma

Atores: administrador, funcionário

Pré-condição: usuário autenticado como administrador ou funcionário

Seqüência de eventos:

1. Usuário solicita operação de cadastro de turma
2. Sistema solicita a identificação da turma a ser manipulada
3. Usuário insere a identificação da turma (disciplina / número / ano)
4. Sistema exibe a turma desejada
5. Usuário seleciona a opção de remoção de turma
6. Sistema confirma a remoção de turma
7. Usuário finaliza a operação

Pós-condição: turma removida

Extensão: não existe

Inclusão: busca turma (passo 4)

3.1.5.3 Caso de uso: Alteração de turma

Descrição: Este caso de uso descreve a alteração de uma turma

Evento iniciador: solicitação de alteração de turma

Atores: administrador, funcionário

Pré-condição: usuário autenticado como administrador ou funcionário



Seqüência de eventos:

1. Usuário solicita operação de cadastro de turma
2. Sistema solicita a identificação da turma a ser alterado
3. Usuário insere a identificação da turma (disciplina / número / ano)
4. Sistema exibe a turma desejada.
5. Usuário seleciona a opção de alteração de turma
6. Usuário faz as alterações desejadas
7. Usuário finaliza a operação

Pós-condição: turma alterada

Extensão: dados fornecidos da turma não consistentes (passo 6): sistema apresenta mensagem ao usuário

Inclusão: busca turma (passos 4)

3.1.5.4 Caso de uso: Consulta de turma

Descrição: Este caso de uso descreve a consulta de uma turma

Evento iniciador: solicitação de consulta de turma

Atores: funcionário, administrador, professor, aluno

Pré-condição: usuário autenticado como funcionário, administrador, professor ou aluno

Seqüência de eventos:

1. Usuário solicita operação de cadastro de turma
2. Sistema solicita a identificação da turma a ser consultada
3. Usuário insere a identificação da turma (disciplina / número / ano)
4. Sistema apresenta os dados da turma
5. Usuário finaliza a operação

Pós-condição: dados da turma apresentados

Extensão: não existe

Inclusão: busca turma (passo 4)

3.1.6 Cadastro de terminal

3.1.6.1 Caso de uso: Inserção de terminal

Descrição: Este caso de uso descreve a inserção de um novo terminal

Evento iniciador: solicitação de inserção de terminal

Atores: administrador, funcionário

Pré-condição: usuário autenticado como administrador ou funcionário

Seqüência de eventos:

1. Usuário solicita operação de cadastro de terminal no menu de cadastro
2. Sistema solicita os dados preliminares relativos ao novo terminal
3. Usuário solicita inserção do terminal
4. Usuário insere os dados do novo terminal
5. Sistema insere o novo cadastro e exibe mensagem de sucesso
6. Usuário finaliza a operação



Pós-condição: novo terminal inserido

Extensões:

1. Terminal já existe: sistema apresenta mensagem ao usuário (passo 4)
2. Dados fornecidos do novo terminal não consistentes (passo 4)

Inclusão: busca terminal (passo 4)

3.1.6.2 Caso de uso: Remoção de terminal

Descrição: Este caso de uso descreve a remoção de um terminal

Evento iniciador: solicitação de remoção de terminal

Atores: administrador, funcionário

Pré-condição: usuário autenticado como administrador ou funcionário

Seqüência de eventos:

1. Usuário solicita operação de cadastro de terminal
2. Sistema solicita a identificação do terminal a ser manipulado
3. Usuário insere a identificação do terminal (id / prédio)
4. Sistema exibe o terminal desejado
5. Usuário seleciona a opção de remoção de terminal
6. Sistema confirma a remoção de terminal
7. Usuário finaliza a operação

Pós-condição: terminal removido

Extensão: não existe

Inclusão: busca terminal (passo 4)

3.1.6.3 Caso de uso: Alteração de terminal

Descrição: Este caso de uso descreve a alteração de um terminal

Evento iniciador: solicitação de alteração de terminal

Atores: administrador, funcionário

Pré-condição: usuário autenticado como administrador ou funcionário

Seqüência de eventos:

1. Usuário solicita operação de cadastro de terminal
2. Sistema solicita a identificação do terminal a ser alterado
3. Usuário insere a identificação do terminal (id / prédio)
4. Sistema exibe o terminal desejado.
5. Usuário seleciona a opção de alteração de terminal
6. Usuário faz as alterações desejadas
7. Usuário finaliza a operação

Pós-condição: terminal alterado

Extensão: Dados fornecidos do terminal não consistentes (passo 6): sistema apresenta mensagem ao usuário

Inclusão: busca terminal (passos 4)



3.1.6.4 Caso de uso: Consulta de terminal

Descrição: Este caso de uso descreve a consulta de um terminal

Evento iniciador: solicitação de consulta de terminal

Atores: funcionário, administrador

Pré-condição: usuário autenticado como funcionário ou administrador

Seqüência de eventos:

1. Usuário solicita operação de cadastro de terminal
2. Sistema solicita a identificação do terminal a ser consultado
3. Usuário insere a identificação do terminal (id / prédio)
4. Sistema apresenta os dados do terminal
5. Usuário finaliza a operação

Pós-condição: dados do terminal apresentados

Extensão: não existe

Inclusão: busca terminal (passo 4)

3.2 Cadastros Complementares

O módulo *Cadastros Complementares* deverá permitir a manipulação dos cadastros adicionais do sistema: Grande Área, Ênfase, Unidade, Departamento, Prédio e Sala.

3.2.1 Cadastro de grande área

3.2.1.1 Caso de uso: Inserção de grande área

Descrição: Este caso de uso descreve a inserção de uma nova grande área

Evento iniciador: solicitação de inserção de grande área

Atores: administrador, funcionário

Pré-condição: usuário autenticado como administrador ou funcionário

Seqüência de eventos:

1. Usuário solicita operação de cadastro de grande área no menu de cadastro
2. Sistema solicita os dados preliminares relativos à nova grande área
3. Usuário solicita inserção da grande área
4. Usuário insere os dados da nova grande área
5. Sistema insere o novo cadastro e exibe mensagem de sucesso
6. Usuário finaliza a operação

Pós-condição: nova grande área inserida

Extensões:

1. Grande área já existe: sistema apresenta mensagem ao usuário (passo 4)
2. Dados fornecidos da nova grande área não consistentes (passo 4)

Inclusão: busca grande área (passo 4)



3.2.1.2 Caso de uso: Remoção de grande área

Descrição: Este caso de uso descreve a remoção de uma grande área

Evento iniciador: solicitação de remoção de grande área

Atores: administrador, funcionário

Pré-condição: usuário autenticado como administrador ou funcionário

Seqüência de eventos:

1. Usuário solicita operação de cadastro de turma
2. Sistema solicita a identificação da turma a ser manipulada
3. Usuário insere a identificação da turma (disciplina / número / ano)
4. Sistema exibe a turma desejada
5. Usuário seleciona a opção de remoção de turma
6. Sistema confirma a remoção de turma
7. Usuário finaliza a operação

Pós-condição: turma removida

Extensão: não existe

Inclusão: busca turma (passo 4)

3.2.1.3 Caso de uso: Alteração de grande área

Descrição: Este caso de uso descreve a alteração de uma grande área

Evento iniciador: solicitação de alteração de grande área

Atores: administrador, funcionário

Pré-condição: usuário autenticado como administrador ou funcionário

Seqüência de eventos:

1. Usuário solicita operação de cadastro de turma
2. Sistema solicita a identificação da turma a ser alterado
3. Usuário insere a identificação da turma (disciplina / número / ano)
4. Sistema exibe a turma desejada.
5. Usuário seleciona a opção de alteração de turma
6. Usuário faz as alterações desejadas
7. Usuário finaliza a operação

Pós-condição: turma alterada

Extensão: dados fornecidos da turma não consistentes (passo 6): sistema apresenta mensagem ao usuário

Inclusão: busca turma (passos 4)

3.2.1.4 Caso de uso: Consulta de grande área

Descrição: Este caso de uso descreve a consulta de uma grande área

Evento iniciador: solicitação de consulta de grande área

Atores: funcionário, administrador

Pré-condição: usuário autenticado como funcionário ou administrador

Seqüência de eventos:

1. Usuário solicita operação de cadastro de grande área



2. Sistema solicita a identificação da grande área a ser consultada.
3. Usuário insere a identificação da grande área (disciplina / número / ano)
4. Sistema apresenta os dados da grande área
5. Usuário finaliza a operação

Pós-condição: dados da grande área apresentados

Extensão: não existe

Inclusão: busca grande área (passo 4)

3.2.2 Cadastro de ênfase

3.2.2.1 Caso de uso: Inserção de ênfase

Descrição: Este caso de uso descreve a inserção de uma nova ênfase

Evento iniciador: solicitação de inserção de ênfase

Atores: administrador, funcionário

Pré-condição: usuário autenticado como administrador ou funcionário

Seqüência de eventos:

1. Usuário solicita operação de cadastro de ênfase no menu de cadastro
2. Sistema solicita os dados preliminares relativos à nova ênfase
3. Usuário solicita inserção da ênfase
4. Usuário insere os dados da nova ênfase
5. Sistema insere o novo cadastro e exibe mensagem de sucesso
6. Usuário finaliza a operação

Pós-condição: nova ênfase inserida

Extensões:

1. Ênfase já existe: sistema apresenta mensagem ao usuário (passo 4)
2. Dados fornecidos da nova ênfase não consistentes (passo 4)

Inclusão: busca ênfase (passo 4)

3.2.2.2 Caso de uso: Remoção de ênfase

Descrição: Este caso de uso descreve a remoção de uma ênfase

Evento iniciador: solicitação de remoção de ênfase

Atores: administrador, funcionário

Pré-condição: usuário autenticado como administrador ou funcionário

Seqüência de eventos:

1. Usuário solicita operação de cadastro de turma
2. Sistema solicita a identificação da turma a ser manipulada
3. Usuário insere a identificação da turma (disciplina / número / ano)
4. Sistema exibe a turma desejada
5. Usuário seleciona a opção de remoção de turma
6. Sistema confirma a remoção de turma
7. Usuário finaliza a operação

Pós-condição: turma removida

Extensão: não existe



Inclusão: busca turma (passo 4)

3.2.2.3 Caso de uso: Alteração de ênfase

Descrição: Este caso de uso descreve a alteração de uma ênfase

Evento iniciador: solicitação de alteração de ênfase

Atores: administrador, funcionário

Pré-condição: usuário autenticado como administrador ou funcionário

Seqüência de eventos:

1. Usuário solicita operação de cadastro de turma
2. Sistema solicita a identificação da turma a ser alterado
3. Usuário insere a identificação da turma (disciplina / número / ano)
4. Sistema exibe a turma desejada.
5. Usuário seleciona a opção de alteração de turma
6. Usuário faz as alterações desejadas
7. Usuário finaliza a operação

Pós-condição: turma alterada

Extensão: dados fornecidos da turma não consistentes (passo 6): sistema apresenta mensagem ao usuário

Inclusão: busca turma (passos 4)

3.2.2.4 Caso de uso: Consulta de ênfase

Descrição: Este caso de uso descreve a consulta de uma ênfase

Evento iniciador: solicitação de consulta de ênfase

Atores: funcionário, administrador

Pré-condição: usuário autenticado como funcionário ou administrador

Seqüência de eventos:

1. Usuário solicita operação de cadastro de ênfase
2. Sistema solicita a identificação da ênfase a ser consultada
3. Usuário insere a identificação da ênfase (disciplina / número / ano)
4. Sistema apresenta os dados da ênfase
5. Usuário finaliza a operação

Pós-condição: dados da ênfase apresentados

Extensão: não existe

Inclusão: busca ênfase (passo 4)

3.2.3 Cadastro de unidade

3.2.3.1 Caso de uso: Inserção de unidade

Descrição: Este caso de uso descreve a inserção de uma nova unidade

Evento iniciador: solicitação de inserção de unidade

Atores: administrador, funcionário



Pré-condição: usuário autenticado como administrador ou funcionário

Seqüência de eventos:

1. Usuário solicita operação de cadastro de unidade no menu de cadastro
2. Sistema solicita os dados preliminares relativos à nova unidade
3. Usuário solicita inserção da unidade
4. Usuário insere os dados da nova unidade
5. Sistema insere o novo cadastro e exibe mensagem de sucesso
6. Usuário finaliza a operação

Pós-condição: nova unidade inserida

Extensões:

1. Unidade já existe: sistema apresenta mensagem ao usuário (passo 4)
2. Dados fornecidos da nova unidade não consistentes (passo 4)

Inclusão: busca unidade (passo 4)

3.2.3.2 Caso de uso: Remoção de unidade

Descrição: Este caso de uso descreve a remoção de uma unidade

Evento iniciador: solicitação de remoção de unidade

Atores: administrador, funcionário

Pré-condição: usuário autenticado como administrador ou funcionário

Seqüência de eventos:

1. Usuário solicita operação de cadastro de turma
2. Sistema solicita a identificação da turma a ser manipulada
3. Usuário insere a identificação da turma (disciplina / número / ano)
4. Sistema exibe a turma desejada
5. Usuário seleciona a opção de remoção de turma
6. Sistema confirma a remoção de turma
7. Usuário finaliza a operação

Pós-condição: turma removida

Extensão: não existe

Inclusão: busca turma (passo 4)

3.2.3.3 Caso de uso: Alteração de unidade

Descrição: Este caso de uso descreve a alteração de uma unidade

Evento iniciador: solicitação de alteração de unidade

Atores: administrador, funcionário

Pré-condição: usuário autenticado como administrador ou funcionário

Seqüência de eventos:

1. Usuário solicita operação de cadastro de turma
2. Sistema solicita a identificação da turma a ser alterado
3. Usuário insere a identificação da turma (disciplina / número / ano)
4. Sistema exibe a turma desejada.
5. Usuário seleciona a opção de alteração de turma
6. Usuário faz as alterações desejadas
7. Usuário finaliza a operação

Pós-condição: turma alterada

Extensão: dados fornecidos da turma não consistentes (passo 6): sistema apresenta mensagem ao usuário



Inclusão: busca turma (passos 4)

3.2.3.4 Caso de uso: Consulta de unidade

Descrição: Este caso de uso descreve a consulta de uma unidade

Evento iniciador: solicitação de consulta de unidade

Atores: funcionário, administrador

Pré-condição: usuário autenticado como funcionário ou administrador

Seqüência de eventos:

1. Usuário solicita operação de cadastro de unidade
2. Sistema solicita a identificação da unidade a ser consultada
3. Usuário insere a identificação da unidade (disciplina / número / ano)
4. Sistema apresenta os dados da unidade
5. Usuário finaliza a operação

Pós-condição: dados da unidade apresentados

Extensão: não existe

Inclusão: busca unidade (passo 4)

3.2.4 Cadastro de departamento

3.2.4.1 Caso de uso: Inserção de departamento

Descrição: Este caso de uso descreve a inserção de um novo departamento

Evento iniciador: solicitação de inserção de departamento

Atores: administrador, funcionário

Pré-condição: usuário autenticado como administrador ou funcionário

Seqüência de eventos:

1. Usuário solicita operação de cadastro de departamento no menu de cadastro
2. Sistema solicita os dados preliminares relativos ao novo departamento
3. Usuário solicita inserção do departamento
4. Usuário insere os dados do novo departamento
5. Sistema insere o novo cadastro e exibe mensagem de sucesso
6. Usuário finaliza a operação

Pós-condição: novo departamento inserido

Extensões:

1. Departamento já existe: sistema apresenta mensagem ao usuário (passo 4)
2. Dados fornecidos do novo departamento não consistentes (passo 4)

Inclusão: busca departamento (passo 4)

3.2.4.2 Caso de uso: Remoção de departamento

Descrição: Este caso de uso descreve a remoção de um departamento

Evento iniciador: solicitação de remoção de departamento



Atores: administrador, funcionário

Pré-condição: usuário autenticado como administrador ou funcionário

Seqüência de eventos:

1. Usuário solicita operação de cadastro de departamento
2. Sistema solicita a identificação do departamento a ser manipulado
3. Usuário insere a identificação do departamento (id / prédio)
4. Sistema exibe o departamento desejado
5. Usuário seleciona a opção de remoção de departamento
6. Sistema confirma a remoção de departamento
7. Usuário finaliza a operação

Pós-condição: departamento removido

Extensão: não existe

Inclusão: busca departamento (passo 4)

3.2.4.3 Caso de uso: Alteração de departamento

Descrição: Este caso de uso descreve a alteração de um departamento

Evento iniciador: solicitação de alteração de departamento

Atores: administrador, funcionário

Pré-condição: usuário autenticado como administrador ou funcionário

Seqüência de eventos:

1. Usuário solicita operação de cadastro de departamento
2. Sistema solicita a identificação do departamento a ser alterado
3. Usuário insere a identificação do departamento (id / prédio)
4. Sistema exibe o departamento desejado.
5. Usuário seleciona a opção de alteração de departamento
6. Usuário faz as alterações desejadas
7. Usuário finaliza a operação

Pós-condição: departamento alterado

Extensão: dados fornecidos do departamento não consistentes (passo 6): sistema apresenta mensagem ao usuário

Inclusão: busca departamento (passos 4)

3.2.4.4 Caso de uso: Consulta de departamento

Descrição: Este caso de uso descreve a consulta de um departamento

Evento iniciador: solicitação de consulta de departamento

Atores: funcionário, administrador

Pré-condição: usuário autenticado como funcionário ou administrador

Seqüência de eventos:

1. Usuário solicita operação de cadastro de departamento
2. Sistema solicita a identificação do departamento a ser consultado
3. Usuário insere a identificação do departamento (id / prédio)
4. Sistema apresenta os dados do departamento
5. Usuário finaliza a operação

Pós-condição: dados do departamento apresentados

Extensão: não existe



Inclusão: busca departamento (passo 4)

3.2.5 Cadastro de prédio

3.2.5.1 Caso de uso: Inserção de prédio

Descrição: Este caso de uso descreve a inserção de um novo prédio

Evento iniciador: solicitação de inserção de prédio

Atores: administrador, funcionário

Pré-condição: usuário autenticado como administrador ou funcionário

Seqüência de eventos:

1. Usuário solicita operação de cadastro de prédio no menu de cadastro
2. Sistema solicita os dados preliminares relativos ao novo prédio
3. Usuário solicita inserção do prédio
4. Usuário insere os dados do novo prédio
5. Sistema insere o novo cadastro e exibe mensagem de sucesso
6. Usuário finaliza a operação

Pós-condição: novo prédio inserido

Extensões:

1. Prédio já existe: sistema apresenta mensagem ao usuário (passo 4)
2. Dados fornecidos do novo prédio não consistentes (passo 4)

Inclusão: busca prédio (passo 4)

3.2.5.2 Caso de uso: Remoção de prédio

Descrição: Este caso de uso descreve a remoção de um prédio

Evento iniciador: solicitação de remoção de prédio

Atores: administrador, funcionário

Pré-condição: usuário autenticado como administrador ou funcionário

Seqüência de eventos:

1. Usuário solicita operação de cadastro de prédio
2. Sistema solicita a identificação do prédio a ser manipulado
3. Usuário insere a identificação do prédio (id / prédio)
4. Sistema exibe o prédio desejado
5. Usuário seleciona a opção de remoção de prédio
6. Sistema confirma a remoção de prédio
7. Usuário finaliza a operação

Pós-condição: prédio removido

Extensão: não existe

Inclusão: busca prédio (passo 4)



3.2.5.3 Caso de uso: Alteração de prédio

Descrição: Este caso de uso descreve a alteração de um prédio

Evento iniciador: solicitação de alteração de prédio

Atores: administrador, funcionário

Pré-condição: usuário autenticado como administrador ou funcionário

Seqüência de eventos:

1. Usuário solicita operação de cadastro de prédio
2. Sistema solicita a identificação do prédio a ser alterado
3. Usuário insere a identificação do prédio (id / prédio)
4. Sistema exibe o prédio desejado.
5. Usuário seleciona a opção de alteração de prédio
6. Usuário faz as alterações desejadas
7. Usuário finaliza a operação

Pós-condição: prédio alterado

Extensão: dados fornecidos do prédio não consistentes (passo 6): sistema apresenta mensagem ao usuário

Inclusão: busca prédio (passos 4)

3.2.5.4 Caso de uso: Consulta de prédio

Descrição: Este caso de uso descreve a consulta de um prédio

Evento iniciador: solicitação de consulta de prédio

Atores: funcionário, administrador

Pré-condição: usuário autenticado como funcionário ou administrador

Seqüência de eventos:

1. Usuário solicita operação de cadastro de prédio
2. Sistema solicita a identificação do prédio a ser consultado
3. Usuário insere a identificação do prédio (id / prédio)
4. Sistema apresenta os dados do prédio
5. Usuário finaliza a operação

Pós-condição: dados do prédio apresentados

Extensão: não existe

Inclusão: busca prédio (passo 4)

3.2.6 Cadastro de sala

3.2.6.1 Caso de uso: Inserção de sala

Descrição: Este caso de uso descreve a inserção de uma nova sala

Evento iniciador: solicitação de inserção de sala

Atores: administrador, funcionário

Pré-condição: usuário autenticado como administrador ou funcionário

Seqüência de eventos:

1. Usuário solicita operação de cadastro de sala no menu de cadastro



2. Sistema solicita os dados preliminares relativos à nova sala
3. Usuário solicita inserção da sala
4. Usuário insere os dados da nova sala
5. Sistema insere o novo cadastro e exibe mensagem de sucesso
6. Usuário finaliza a operação

Pós-condição: nova sala inserida

Extensões:

1. Sala já existe: sistema apresenta mensagem ao usuário (passo 4)
2. Dados fornecidos da nova sala não consistentes (passo 4)

Inclusão: busca sala (passo 4)

3.2.6.2 Caso de uso: Remoção de sala

Descrição: Este caso de uso descreve a remoção de uma sala

Evento iniciador: solicitação de remoção de sala

Atores: administrador, funcionário

Pré-condição: usuário autenticado como administrador ou funcionário

Seqüência de eventos:

1. Usuário solicita operação de cadastro de turma
2. Sistema solicita a identificação da turma a ser manipulada
3. Usuário insere a identificação da turma (disciplina / número / ano)
4. Sistema exibe a turma desejada
5. Usuário seleciona a opção de remoção de turma
6. Sistema confirma a remoção de turma
7. Usuário finaliza a operação

Pós-condição: turma removida

Extensão: não existe

Inclusão: busca turma (passo 4)

3.2.6.3 Caso de uso: Alteração de sala

Descrição: Este caso de uso descreve a alteração de uma sala

Evento iniciador: solicitação de alteração de sala

Atores: administrador, funcionário

Pré-condição: usuário autenticado como administrador ou funcionário

Seqüência de eventos:

1. Usuário solicita operação de cadastro de turma
2. Sistema solicita a identificação da turma a ser alterado
3. Usuário insere a identificação da turma (disciplina / número / ano)
4. Sistema exibe a turma desejada
5. Usuário seleciona a opção de alteração de turma
6. Usuário faz as alterações desejadas
7. Usuário finaliza a operação

Pós-condição: turma alterada

Extensão: dados fornecidos da turma não consistentes (passo 6): sistema apresenta mensagem ao usuário

Inclusão: busca turma (passos 4)



3.2.6.4 Caso de uso: Consulta de sala

Descrição: Este caso de uso descreve a consulta de uma sala.

Evento iniciador: solicitação de consulta de sala

Atores: funcionário, administrador

Pré-condição: usuário autenticado como funcionário ou administrador

Seqüência de eventos:

1. Usuário solicita operação de cadastro de sala
2. Sistema solicita a identificação da sala a ser consultada
3. Usuário insere a identificação da sala (disciplina / número / ano)
4. Sistema apresenta os dados da sala
5. Usuário finaliza a operação

Pós-condição: dados da sala apresentados

Extensão: não existe

Inclusão: busca sala (passo 4)

3.3 Grade Horária

O módulo de *Grade Horária* deverá permitir a consulta de aulas agendadas / lecionadas no período selecionado, de acordo com o professor ou aluno que estará realizando o acesso. O administrador e os funcionários poderão escolher qualquer professor ou aluno cadastrado, para consultar sua grade horária.

3.3.1 Caso de uso: Consulta de grade horária

Descrição: Este caso de uso descreve a consulta da grade horária de aluno / professor no sistema.

Evento iniciador: solicitação de consulta de grade horária no menu principal

Atores: aluno, professor, funcionário e administrador

Pré-condição: usuário autenticado como aluno, professor, funcionário ou administrador

Seqüência de eventos:

1. Usuário solicita operação de consulta de grade horária
2. Sistema solicita o período (semana) da grade horária a ser consultada. Caso o usuário seja um professor ou administrador, é necessário escolher um aluno / professor para consultar a grade horária correspondente. Caso contrário, essa seleção é feita automaticamente
3. Usuário insere o período e a identificação do aluno ou professor (nusp / nome)
4. Sistema apresenta os dados da grade horária
5. Usuário finaliza a operação.

Pós-condição: dados da grade horária apresentadas

Extensão: não existe

Inclusão: busca aluno / professor (passo 3)



3.4 Disciplinas

O módulo de *Disciplinas* deverá permitir a associação de alunos nas turmas cadastradas, o recálculo de presença, a programação aula-a-aula, a consulta de presenças e o requerimento de recálculo de presença.

3.4.1 Caso de uso: Associação de alunos em uma disciplina – por ênfase

Descrição: Este caso de uso descreve o processo pelo qual o administrador (ou funcionário) associam alunos (por ênfase) em uma turma

Evento iniciador: usuário seleciona a opção de associação de alunos por ênfase, no menu *Disciplinas*.

Atores: administrador, funcionário

Pré-condição: usuário autenticado como administrador ou funcionário

Seqüência de eventos:

1. Usuário seleciona a opção de associação por ênfase
2. Sistema exibe lista das disciplinas e turmas cadastradas no sistema
3. Usuário seleciona a disciplina e a turma desejada
4. Sistema exibe a lista das ênfases para realizar a associação
5. Usuário seleciona a ênfase
6. Sistema exibe então os alunos pertencentes à ênfase
7. Usuário filtra os alunos que deseja associar e confirma
8. Sistema exibe mensagem de sucesso
9. Usuário finaliza a operação

Pós-condição: alunos associados à turma

Extensões: não existe

Inclusões: busca disciplinas e turmas (passo 2), ênfases (passo 4) e alunos (passo 6)

3.4.2 Caso de uso: Associação de alunos em uma disciplina – um-a-um

Descrição: Este caso de uso descreve o processo pelo qual o administrador (ou funcionário) associam alunos (um-a-um) em uma turma

Evento iniciador: usuário seleciona a opção de associação de alunos um-a-um, no menu *Disciplinas*.

Atores: administrador, funcionário

Pré-condição: usuário autenticado como administrador ou funcionário

Seqüência de eventos:

1. Usuário seleciona a opção de associação um-a-um
2. Sistema exibe lista das disciplinas e turmas cadastradas no sistema
3. Usuário seleciona a disciplina e a turma desejada
4. Sistema exibe campo para adicionar individualmente os alunos
5. Usuário adiciona alunos e confirma



6. Sistema exibe mensagem de sucesso

7. Usuário finaliza a operação

Pós-condição: alunos associados à turma

Extensões: não existe

Inclusões: busca disciplinas e turmas (passo 2) e alunos (passo 5)

3.4.3 Caso de uso: Associação de alunos em uma disciplina – consultar

Descrição: Este caso de uso descreve o processo pelo qual o administrador (ou funcionário) consultam a associação de alunos em uma turma

Evento iniciador: usuário seleciona a opção de consultar associação de alunos, no menu *Disciplinas*.

Atores: administrador, funcionário, professor, aluno

Pré-condição: usuário autenticado como administrador, funcionário, professor ou aluno

Seqüência de eventos:

1. Usuário seleciona a opção de consultar associação
2. Sistema exibe lista das disciplinas e turmas cadastradas no sistema
3. Usuário seleciona a disciplina e a turma desejada
4. Sistema exibe alunos associados
5. Usuário finaliza a operação

Pós-condição: apresentação dos alunos associados à turma

Extensões: não existe

Inclusões: busca disciplinas e turmas (passo 2)

3.4.4 Caso de uso: Recálculo de presença

Descrição: Este caso de uso descreve o processo pelo qual um professor pode ajustar a presença da turma (ou de apenas um aluno) para uma determinada aula, devido a fatores externos (atrasos, justificativas, etc)

Evento iniciador: usuário seleciona a opção de recálculo de presença no menu *Disciplinas*

Atores: administrador, funcionário, professor

Pré-condição: usuário autenticado como administrador, funcionário ou professor

Seqüência de eventos:

1. Usuário seleciona a opção de recálculo de presença
2. Sistema exibe caixas para seleção com a disciplina, turma e aula onde será efetuado o recálculo
3. Usuário seleciona a aula desejada e confirma
4. Sistema recalcula a presença, de acordo com os registros e os parâmetros da aula escolhida
5. Sistema exibe mensagem de sucesso
6. Usuário finaliza operação

Pós-condição: presenças recalculadas para a aula desejada

Extensões: não existe

Inclusões: busca disciplinas, turmas e aulas (passo 2)



3.4.5 Caso de uso: Programação aula-a-aula

Descrição: Este caso de uso descreve o processo pelo qual o usuário pode ajustar / consultar a programação aula-a-aula de uma determinada turma

Evento iniciador: usuário seleciona a opção de programação aula-a-aula de presença no menu *Disciplinas*

Atores: administrador, funcionário, professor, aluno (apenas consulta)

Pré-condição: usuário autenticado como administrador, funcionário, professor ou aluno

Seqüência de eventos:

1. Usuário seleciona a opção de programação aula-a-aula
2. Sistema exibe caixas para seleção com a disciplina e turma
3. Usuário seleciona a turma desejada e confirma
4. Sistema exibe a programação aula-a-aula
5. Usuário altera / visualiza a programação. Se o usuário for um aluno, pula para o passo 7
6. Sistema exibe resultado
7. Usuário finaliza operação

Pós-condição: apresentação / alteração da programação aula-a-aula

Extensões: não existe

Inclusões: busca disciplinas e turmas (passo 2)

3.4.6 Caso de uso: Consulta / Ajuste de presença (professor).

Descrição: Este caso de uso descreve o processo pelo qual um professor pode consultar / ajustar a presença dos alunos em uma disciplina

Evento iniciador: usuário seleciona a opção de consulta / ajuste de presença no menu *Disciplinas*

Atores: administrador, funcionário e professor

Pré-condição: usuário autenticado como administrador, funcionário ou professor

Seqüência de eventos:

1. Usuário seleciona a opção de consulta / ajuste de presença
2. Sistema exibe caixas para seleção com a disciplina, turma e aula onde será efetuada a consulta / ajuste
3. Professor seleciona a aula desejada
4. Sistema exibe a lista dos alunos pertencentes àquela turma, com suas presenças na aula selecionada
5. Professor solicita a validação de presença para todos os alunos listados, ou apenas para um aluno escolhido
6. Sistema exibe resultado
7. Usuário finaliza operação

Pós-condição: presença recalculada para os alunos desejados

Extensões: não existe

Inclusões: consulta de disciplinas, turmas, aulas (passo 2) e alunos (passo 4)

3.4.7 Caso de uso: Consulta de presença (aluno).

Descrição: Este caso de uso descreve o processo pelo qual um aluno pode consultar sua presença nas turmas em que está matriculado

Evento iniciador: aluno seleciona a opção de consulta de presença no menu *Disciplinas*

Atores: aluno



Pré-condição: usuário autenticado como aluno

Seqüência de eventos:

1. Aluno seleciona a opção de consulta de presença
2. Sistema exibe caixas para seleção da disciplina e turma onde será efetuada a consulta
3. Aluno seleciona a turma sobre a qual quer saber mais detalhes de presença
4. Sistema exibe presença do aluno para a turma selecionada
5. Aluno finaliza operação

Pós-condição: presença do aluno consultada.

Extensões: não existe

Inclusões: consulta de disciplinas, turmas (passo 2) e aulas (passo 4)

3.4.8 Caso de uso: Requerimento de recálculo de presença

Descrição: Este caso de uso descreve o processo onde o aluno pode requerer que sua presença em determinada aula seja recalculada

Evento iniciador: aluno seleciona a opção de requerimento de recálculo no menu *Disciplinas*

Atores: aluno, administrador, funcionário

Pré-condição: usuário autenticado como aluno, administrador ou funcionário

Seqüência de eventos:

1. Usuário seleciona a opção recálculo de presença
2. Sistema exibe caixas para seleção da disciplina, turma e aula onde será feito o pedido
3. Usuário seleciona a disciplina, turma e aula desejada
4. Sistema exibe caixa de texto para que o usuário entre com sua justificativa do requerimento
5. Usuário preenche o formulário e confirma
6. Sistema exibe mensagem de sucesso
7. Usuário finaliza operação

Pós-condição: pedido de recálculo de presença executado

Extensões: usuário deixou o campo de justificativa vazio (passo 5): retorna para o passo 4

Inclusões: consulta de disciplina, turma e aula (passo 2)

3.5 Estatísticas

Esse módulo permite a consulta de diversas estatísticas sobre as presenças nas disciplinas lecionadas.

3.5.1 Caso de uso: Geração de estatística

Descrição: Este caso de uso descreve o processo de geração de estatística de presença

Evento iniciador: usuário seleciona a opção de geração de estatística

Atores: administrador, funcionário, professor, aluno

Pré-condição: usuário autenticado como administrador, funcionário, professor ou aluno

Seqüência de eventos:

1. Usuário seleciona a opção de geração de estatística
2. Sistema exibe tela de estatísticas



3. Usuário seleciona o formato de saída das estatísticas (gráficos, tabelas, etc...)
4. Usuário seleciona quais disciplinas quer comparar e qual período de tempo deve ser coberto, entre outras variáveis
5. Usuário então confirma a entrada de dados
6. Sistema exibe as estatísticas no formato solicitado
7. Usuário finaliza operação

Pós-condição: estatísticas geradas

Extensões: não existe

Inclusões: não existe

3.6 Listas

Esse módulo permite a impressão de listas de chamada, para casos emergenciais – queda do servidor, problemas com a leitora, etc.

3.6.1 Caso de uso: Impressão de lista

Descrição: Este caso de uso descreve o processo de impressão de lista de presença

Evento iniciador: usuário seleciona a opção *Listas*

Atores: administrador, funcionário, professor

Pré-condição: usuário autenticado como administrador, funcionário ou professor

Seqüência de eventos:

1. Usuário seleciona a opção de impressão de listas
2. Sistema exibe caixas para seleção da disciplina e turma
3. Usuário seleciona a disciplina turma desejada
4. Sistema exibe as lista
5. Usuário confirma a impressão
6. Sistema imprime lista
7. Usuário finaliza operação

Pós-condição: lista impressa

Extensões: não existe

Inclusões: não existe

3.7 Registro de presença

Esse módulo é o responsável por efetuar o registro de presença do aluno, através da passagem da carteira USP.

3.7.1 Caso de uso: Registro de presença

Descrição: Este caso de uso descreve o processo de registro de presença



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

Avenida Professor Luciano Gualberto, travessa 3 nº 158 CEP 05508-900 São Paulo SP
Telefone: (011) 3091-5583 Fax (011) 3091-5294

Departamento de Engenharia de Computação e Sistemas Digitais

Evento iniciador: usuário passa a carteira USP na leitora magnética

Atores: aluno

Pré-condição: nenhuma (mesmo o aluno não estando cadastrado no sistema, o registro será efetuado, possibilitando assim recálculos no futuro e fornecendo flexibilidade ao sistema)

Sequência de eventos:

1. Usuário passa a carteira USP na leitora magnética
2. Leitora sinaliza o sucesso do recebimento de informações da trilha magnética
3. Sistema registra os dados lidos

Pós-condição: registro efetuado

Extensões: erro na leitura das trilhas magnéticas (passo 2)

Inclusões: não existe



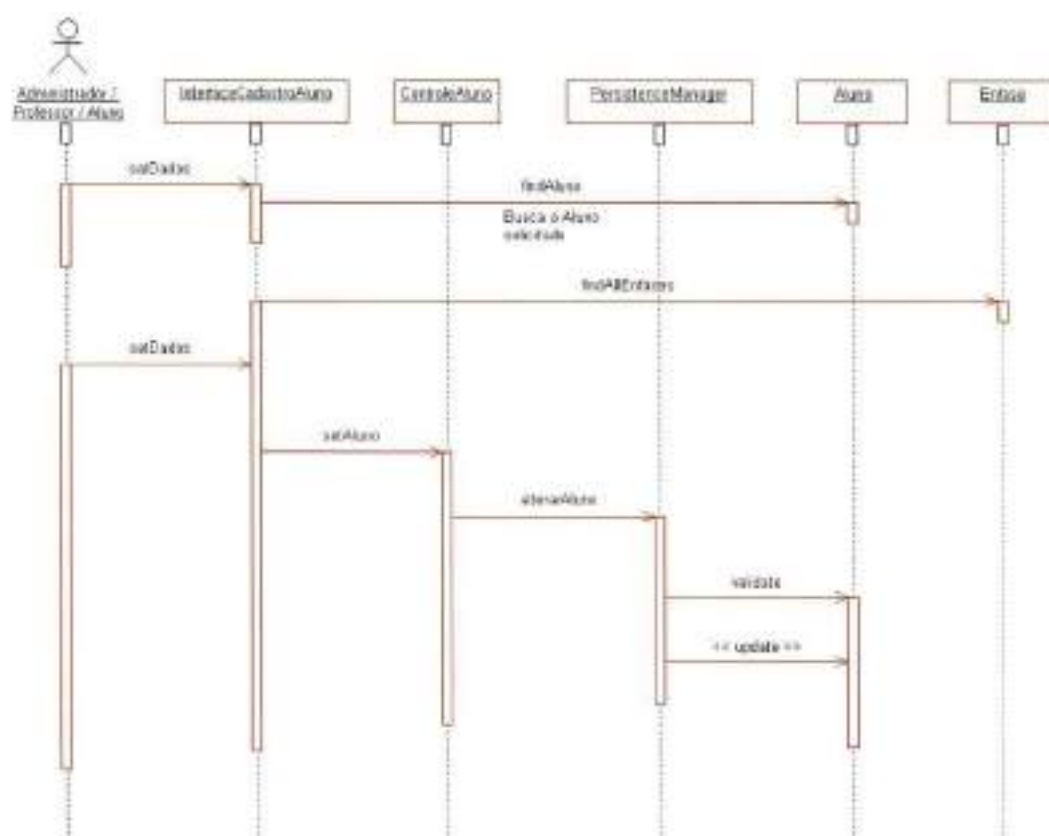
4 Descrição dos Atores

Nome	Papel
Administrador	Gerencia todos os cadastros do sistema; Consulta todas as grades horárias; Gerencia as associações de alunos, os recálculos de presença, as programações aula-a-aula, os ajustes de presença e os requerimentos de recálculo; Imprime listas de presença (emergenciais); Consulta estatísticas de presença.
Funcionário	Gerencia todos os cadastros do sistema, com exceção do cadastro de professores e o cadastro de outros funcionários; Consulta todas as grades horárias; Gerencia as associações de alunos, os recálculos de presença, as programações aula-a-aula, os ajustes de presença e os requerimentos de recálculo; Imprime listas de presença (emergenciais); Consulta estatísticas de presença.
Professor	Consulta todos os cadastros do sistema, com exceção do cadastro de funcionário e o cadastro de outros professores; Altera dados da Turma e da Disciplina; Consulta e altera seus dados pessoais; Consulta sua grade horária; Consulta as associações de alunos, gerencia os recálculos de presença, as programações aula-a-aula, os ajustes de presença e os requerimentos de recálculo, referentes às suas aulas; Imprime listas de presença (emergenciais); Consulta estatísticas de presença referentes às suas aulas.
Aluno	Consulta dados de Turma e Disciplina; Consulta e altera seus dados pessoais; Consulta sua grade horária; Consulta as associações de alunos, as programações aula-a-aula, as presenças em aula e solicita / consulta requerimentos de recálculo; Consulta estatísticas de presença referentes às suas aulas. Efetua registro de presença.

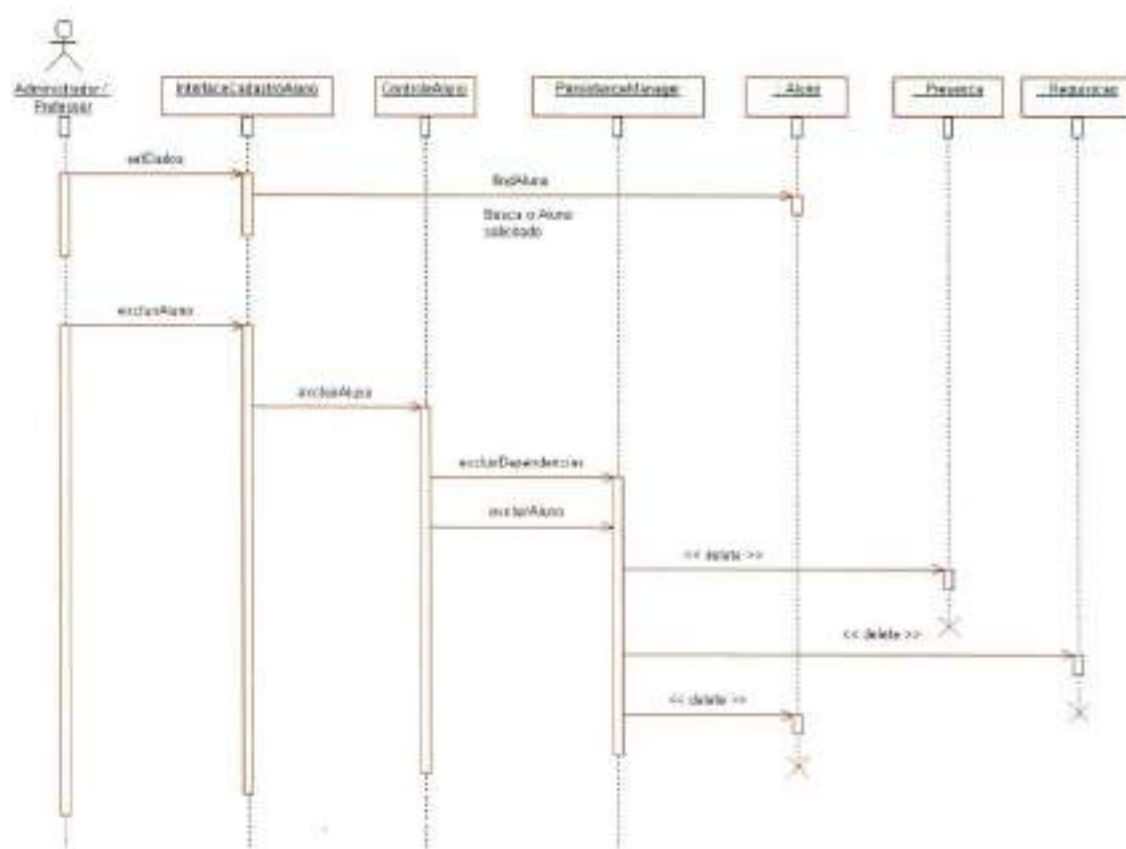
APÊNDICE E – DIAGRAMAS DE SEQUÊNCIA

Nesse apêndice são apresentados os principais diagramas de sequência do sistema SARPA. O restante dos diagramas são análogos aos apresentados.

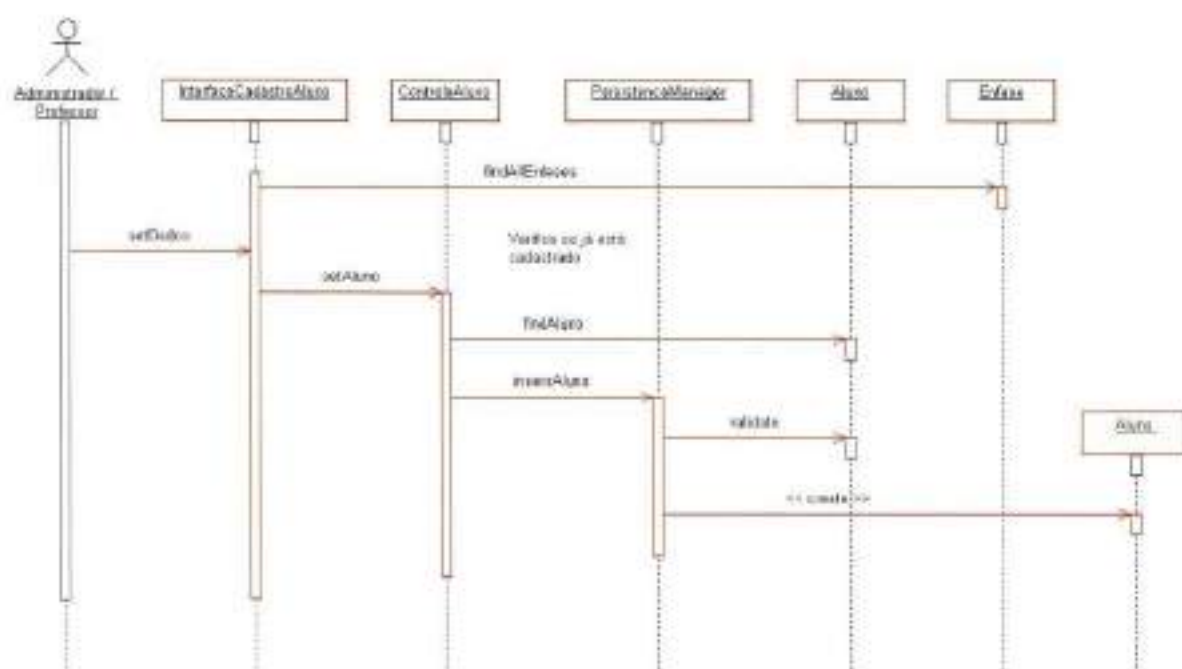
Alteração de Aluno



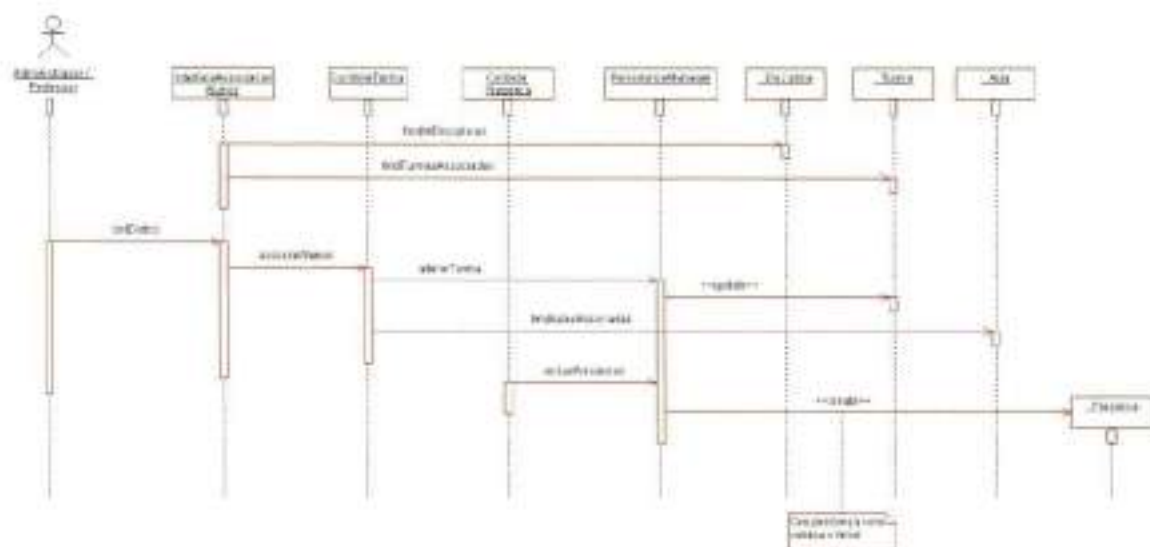
Exclusão de Aluno



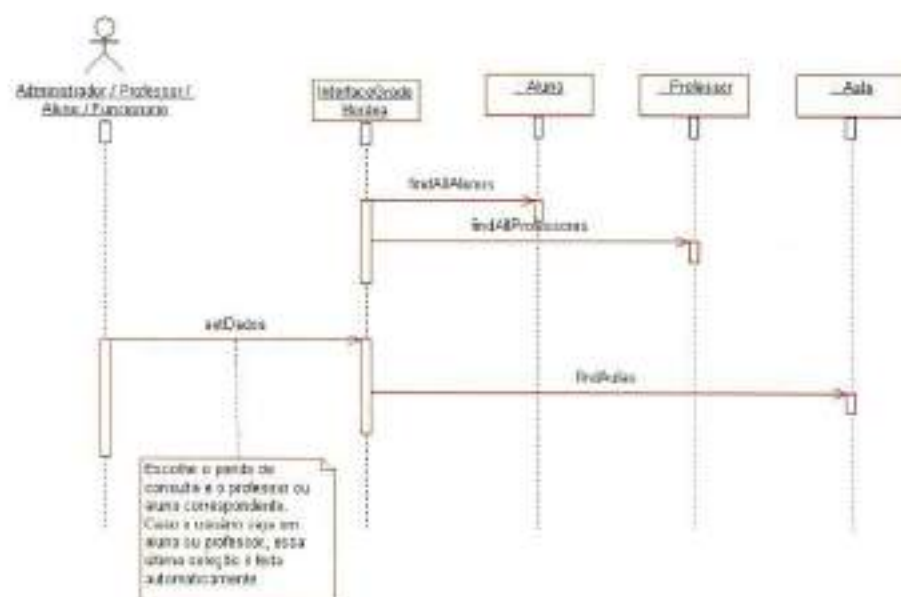
Inserção de Aluno



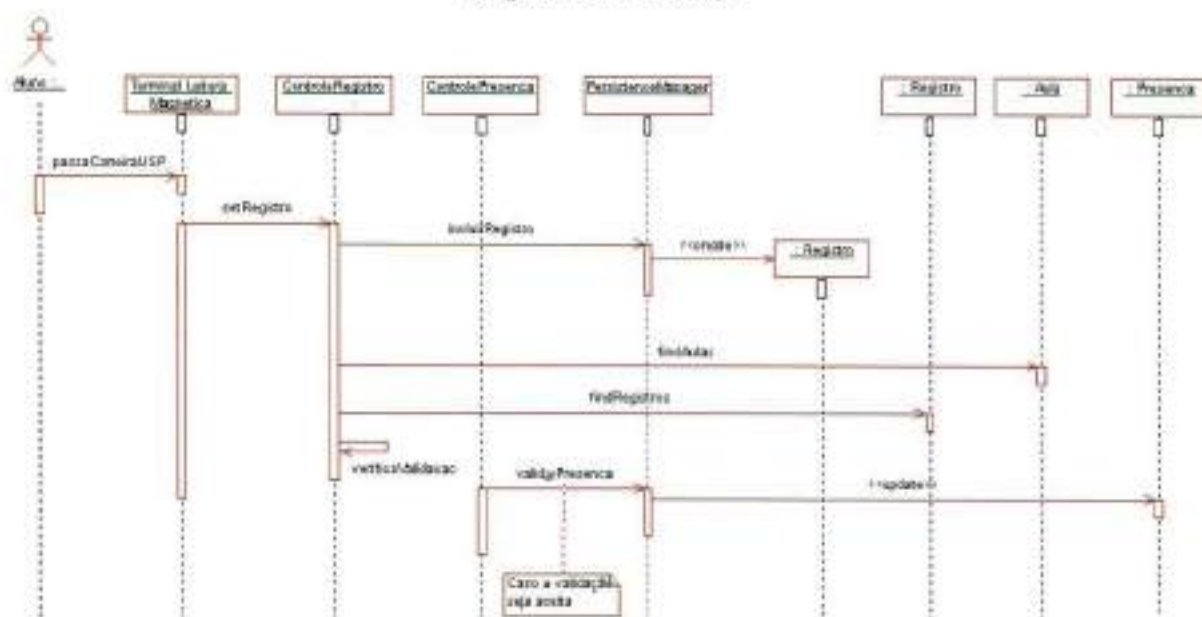
Associação de Aluno um a um



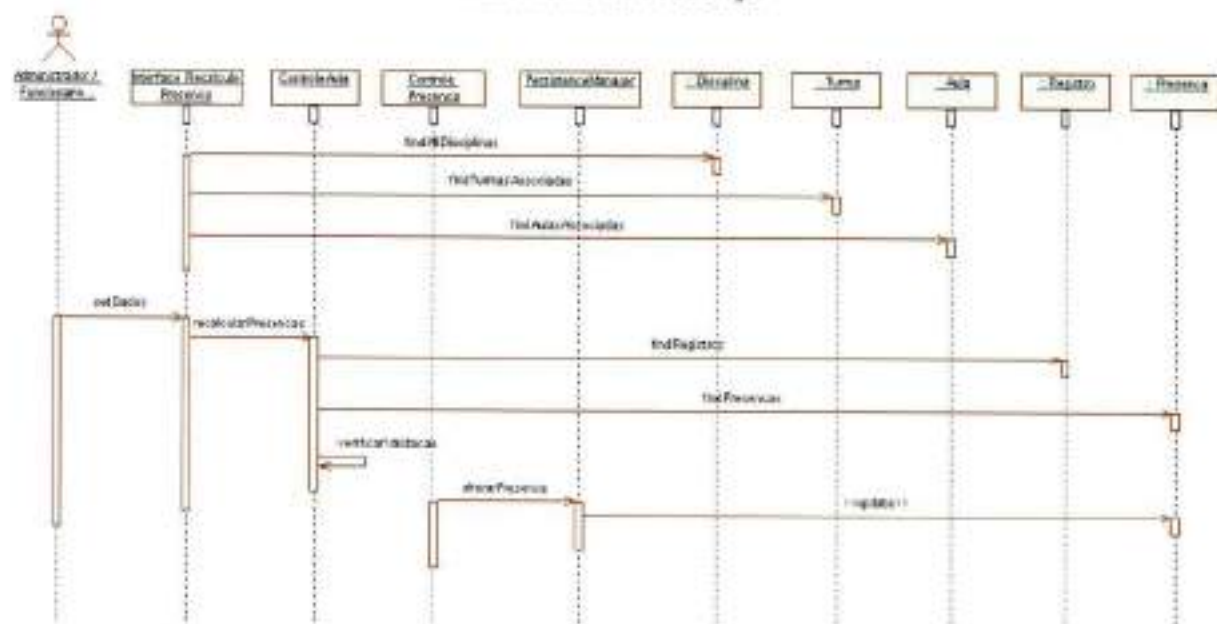
Consulta de Grade



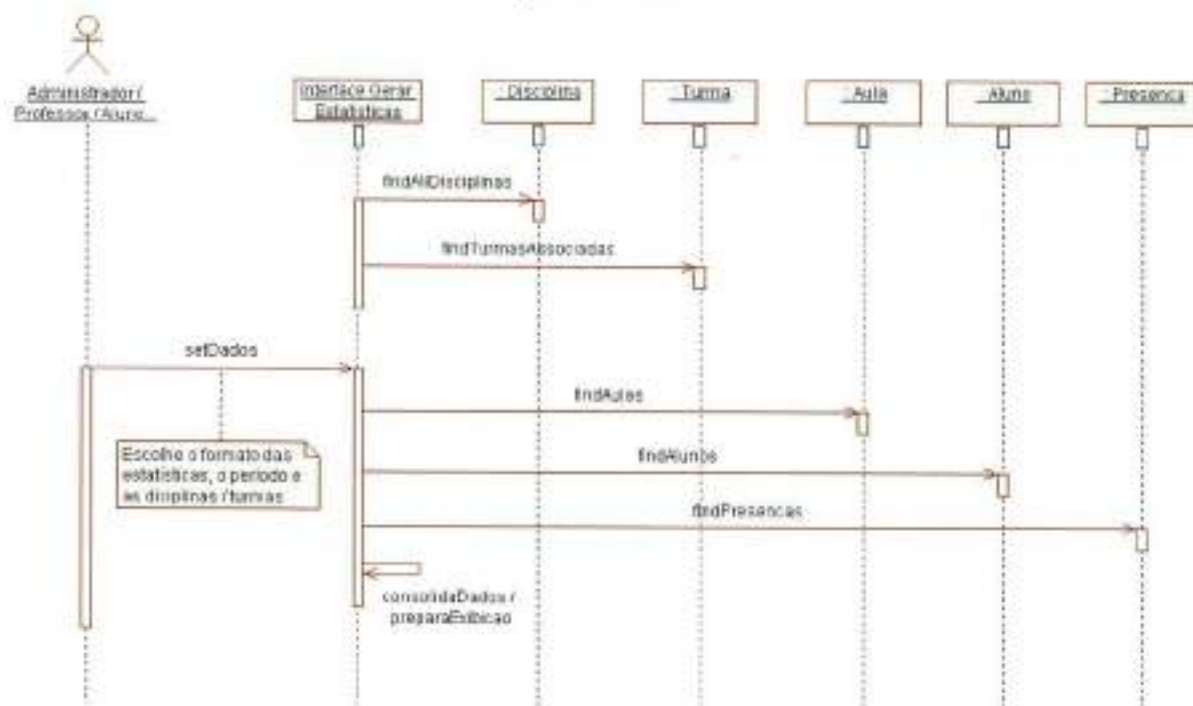
Registro de Presença



Recálculo de Presença



Geração de Estatística



APÊNDICE F – PLANO DE TESTES

Sistema SARPA

Plano de Aceitação

Relatório Técnico

Autores: <i>Carlos Eduardo Pinheiro Vitorino</i> <i>Reginaldo Longano Junior</i> <i>Rodrigo Herculano de Oliveira</i> <i>Vitor Caçador Alexandre</i>	Data de emissão: 05/08/2006
Revisor: <i>Carlos Eduardo Pinheiro Vitorino</i>	Data de revisão 25/11/2006



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

Avenida Professor Luciano Gualberto, travessa 3 nº 158 CEP 05508-900 São Paulo SP
Telefone: (011) 818-5583 Fax (011) 818-5294

Departamento de Engenharia de Computação e Sistemas Digitais

FOLHA DE CONTROLE DE REVISÕES

Número da versão	Data de emissão	Registro de modificações
00	05/08/2006	Criação do documento
01	25/11/2006	Inserção de mais testes e revisão
02		
03		
04		
05		
06		
07		
08		
09		
10		



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

Avenida Professor Luciano Gualberto, travessa 3 nº 158 CEP 05508-900 São Paulo SP
Telefone: (011) 818-5583 Fax (011) 818-5294

Departamento de Engenharia de Computação e Sistemas Digitais

Índice

1	<i>Objetivo do Documento</i>	3
2	<i>Testes</i>	4



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

Avenida Professor Luciano Gualberto, travessa 3 nº 156 CEP 05508-900 São Paulo SP
Telefone: (011) 818-5583 Fax (011) 818-5294

Departamento de Engenharia de Computação e Sistemas Digitais

SARPA - Modelo de Plano de Aceitação

1 Objetivo do Documento

O objetivo deste documento é estabelecer os casos de teste a serem executados para aceitação do sistema. São contemplados os testes dos requisitos funcionais e não funcionais, abrangendo o sistema como um todo.



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

Avenida Professor Luciano Gualberto, travessa 3 nº 158 CEP 05508-900 São Paulo SP
Telefone: (011) 818-5583 Fax (011) 818-5294

Departamento de Engenharia de Computação e Sistemas Digitais

2 Testes

Teste no. 01		Requisito testado Cadastro de Aluno	
Funcional: <input checked="" type="checkbox"/>		Observações:	
Não funcional: <input type="checkbox"/>			
Seqüência para verificação do caso comum		Comportamento esperado do software	Testado
1. Clicar em CADASTROS		O sistema deverá apresentar as opções de cadastros disponíveis.	
2. Clicar em ALUNOS			
3. Clicar em INSERIR		O sistema deverá mostrar uma nova tela com mais dados a serem preenchidos	
4. Preencher os campos presentes na tela			
5. Clicar em OK			
6.			
7.			
Seqüência para verificação de exceções		Comportamento esperado do software	
1. ITEM 5 – Se os campos obrigatórios não forem preenchidos ou algum campo tiver sido preenchido em algum formato inválido		Sistema pede para que os dados sejam inseridos novamente.	
2. ITEM 5 – Se o aluno já estiver cadastrado.		Sistema apresenta mensagem dizendo o cadastro já existe.	
3.			
4.			
5.			
6.			
7.			
<input type="checkbox"/> Aprovado	Comentários		
<input type="checkbox"/> Reprovado			
Cientes: Avaliadores		Apresentador	



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

Avenida Professor Luciano Gualberto, travessa 3 nº 158 CEP 05508-900 São Paulo SP
Telefone: (011) 818-5583 Fax (011) 818-5294

Departamento de Engenharia de Computação e Sistemas Digitais

Teste no. 02		Requisito testado Consulta de Aluno
Funcional: <input checked="" type="checkbox"/>	Observações:	
Não funcional: <input type="checkbox"/>		
Seqüência para verificação do caso comum	Comportamento esperado do software	Testado
1. Clicar em CADASTROS	O sistema deverá apresentar as opções de cadastros disponíveis.	
2. Clicar em ALUNOS		
3. Clicar em BUSCAR	O sistema deverá mostrar uma nova tela com os alunos cadastrados	
4. Clicar em DETALHAR para algum aluno cadastrado.	O sistema deverá mostrar detalhes do aluno cadastrado, sem possibilitar edição.	
5. Clicar em OK		
6. Clicar em ALTERAR para algum aluno cadastrado.	O sistema deverá mostrar detalhes do aluno cadastrado, possibilitando edição.	
7		
Seqüência para verificação de exceções	Comportamento esperado do software	
1. ITEM 6 – Valem as exceções do item CADASTRO DE ALUNO		
2. ITEM 3 – Busca não encontrar nenhum aluno	Tela não apresenta dados de nenhum aluno.	
3.		
4.		
5.		
6.		
7.		
<input type="checkbox"/> Aprovado	Comentários	
<input type="checkbox"/> Reprovado		
Cientes: Avaliadores		Apresentador



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

Avenida Professor Luciano Gualberto, travessa 3 nº 158 CEP 05508-900 São Paulo SP
Telefone: (011) 818-5583 Fax (011) 818-5294

Departamento de Engenharia de Computação e Sistemas Digitais

Teste no. 03		Requisito testado Cadastro de turma	
Funcional: <input checked="" type="checkbox"/>		Observações:	
Não funcional: <input type="checkbox"/>			
Seqüência para verificação do caso comum	Comportamento esperado do software	Testado	
1. Clicar em CADASTROS	O sistema deverá apresentar as opções de cadastros disponíveis.		
2. Clicar em TURMA			
3. Clicar em INSERIR	O sistema deverá mostrar uma nova tela com mais dados a serem preenchidos		
4. Preencher os campos presentes na tela			
5. Preencher os campos sobre as aulas que existem para essa turma			
6. Clicar em ADICIONAR	O sistema deverá listar as aulas que estão sendo adicionadas.		
7. Repetir os passos 5 e 6 enquanto houverem aulas para serem inseridas			
8. Clicar em FINALIZAR	O sistema apresenta uma mensagem de sucesso		
Seqüência para verificação de exceções	Comportamento esperado do software		
1. ITEM 6 e 8 – Se os campos obrigatórios não forem preenchidos ou algum campo tiver sido preenchido em algum formato inválido	Sistema pede para que os dados sejam inseridos novamente.		
2. ITEM 6 e 8 – Se já estiver cadastrado.	Sistema apresenta mensagem dizendo que o cadastro já existe.		
3.			
4.			
5.			
6.			
7.			
<input type="checkbox"/> Aprovado	Comentários		
<input type="checkbox"/> Reprovado			
Cientes: Avaliadores		Apresentador	



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

Avenida Professor Luciano Gualberto, travessa 3 nº 158 CEP 05508-900 São Paulo SP
Telefone: (011) 818-5583 Fax (011) 818-5294

Departamento de Engenharia de Computação e Sistemas Digitais

Teste no. 04		Requisito testado Consulta de Grade-Horária	
Funcional: <input checked="" type="checkbox"/>		Observações:	
Não funcional: <input type="checkbox"/>			
Seqüência para verificação do caso comum		Comportamento esperado do software	Testado
1. Clicar em GRADE HORARIA		O sistema dará a opção de escolha do período em que se deseja consultar a Grade Horária	
2. Seleciona-se o PERÍODO			
3. Clicar em OK		O sistema dará a opção de escolha entre Grade Horária de Professor ou Aluno	
4. Seleciona-se PROFESSOR ou ALUNO para ver a grade horária correspondente.			
5. Clicar em OK			
6.			
7.			
Seqüência para verificação de exceções		Comportamento esperado do software	
1. ITEM 4 – Caso o usuário seja um PROFESSOR ou um ALUNO, a grade horária correspondente é mostrada automaticamente.			
2.			
3.			
4.			
5.			
6.			
7.			
<input type="checkbox"/> Aprovado	Comentários		
<input type="checkbox"/> Reprovado			
Cientes: Avaliadores		Apresentador	



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

Avenida Professor Luciano Gualberto, travessa 3 nº 158 CEP 05508-900 São Paulo SP
Telefone: (011) 818-5583 Fax (011) 818-5294

Departamento de Engenharia de Computação e Sistemas Digitais

Teste no. 05		Requisito testado Associação de Alunos – uma a um	
Funcional: <input checked="" type="checkbox"/>		Observações:	
Não funcional: <input type="checkbox"/>			
Seqüência para verificação do caso comum	Comportamento esperado do software	Testado	
1.Clicar em DISCIPLINAS	O sistema deverá apresentar as opções de disciplinas		
2.Clicar em ASSOCIACAO DE ALUNOS	O sistema deverá apresentar as opções de associações disponíveis.		
3.Clicar em UM-A-UM			
4. Selecione dados para os campos presentes na tela			
5.Clicar em OK			
6.Digite o NUSP do aluno que deve ser associado			
7.Clicar em ADICIONAR	O sistema deverá listar os alunos que estão sendo adicionados.		
8 Repetir os passos 6 e 7 enquanto houverem alunos para serem inseridos			
8. Clicar em FINALIZAR.	O sistema apresenta uma mensagem de sucesso		
Seqüência para verificação de exceções	Comportamento esperado do software		
1. ITEM 6 – Se aluno não estiver cadastrado	É apresentado que nenhum aluno foi associado a turma		
2. ITEM 6 – Se aluno já estiver vinculado	É apresentado que nenhum aluno foi associado a turma		
5.			
6.			
7.			
<input type="checkbox"/> Aprovado	Comentários		
<input type="checkbox"/> Reprovado			
Cientes: Avaliadores		Apresentador	



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

Avenida Professor Luciano Gualberto, travessa 3 nº 158 CEP 05508-900 São Paulo SP
Telefone: (011) 818-5583 Fax (011) 818-5294

Departamento de Engenharia de Computação e Sistemas Digitais

Teste no. 06		Requisito testado Associação de Alunos – por ênfase	
Funcional: <input checked="" type="checkbox"/>		Observações:	
Não funcional: <input type="checkbox"/>			
Seqüência para verificação do caso comum	Comportamento esperado do software	Testado	
1.Clicar em DISCIPLINAS	O sistema deverá apresentar as opções de disciplinas		
2.Clicar em ASSOCIACAO DE ALUNOS	O sistema deverá apresentar as opções de associações disponíveis.		
3.Clicar em POR ENFASE			
4. Selecione dados para os campos presentes na tela			
5.Clicar em OK			
6.Escolha a Ênfase desejada			
7.Clicar em ADICIONAR	O sistema deverá listar os alunos que estão sendo adicionados.		
8.Repetir os passos 6 e 7 enquanto houverem alunos para serem inseridos			
8. Clicar em FINALIZAR	O sistema apresenta uma mensagem de sucesso		
Seqüência para verificação de exceções	Comportamento esperado do software		
1. ITEM 6 – Se aluno já estiver vinculado	É apresentado que nenhum aluno foi associado à turma		
2.			
5.			
6.			
7.			
<input type="checkbox"/> Aprovado	Comentários:		
<input type="checkbox"/> Reprovado			
Cientes: Avaliadores		Apresentador	



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

Avenida Professor Luciano Gualberto, travessa 3 nº 158 CEP 05508-900 São Paulo SP
Telefone: (011) 818-5583 Fax (011) 818-5294

Departamento de Engenharia de Computação e Sistemas Digitais

Teste no. 07		Requisito testado Associação de Alunos – consulta	
Funcional: <input checked="" type="checkbox"/>		Observações:	
Não funcional: <input type="checkbox"/>			
Seqüência para verificação do caso comum	Comportamento esperado do software	Testado	
1. Clicar em DISCIPLINAS	O sistema deverá apresentar as opções de disciplinas		
2. Clicar em ASSOCIACAO DE ALUNOS	O sistema deverá apresentar as opções de associações disponíveis.		
3. Clicar em CONSULTA			
4. Selecione dados para os campos presentes na tela			
5. Clicar em OK	O sistema deverá listar os alunos que estão adicionados.		
6.			
7.			
Seqüência para verificação de exceções	Comportamento esperado do software		
1. ITEM 5 – Se nenhum aluno associado	Não é apresentado aluno associado à turma		
2.			
3.			
4.			
5.			
6.			
7.			
<input type="checkbox"/> Aprovado	Comentários		
<input type="checkbox"/> Reprovado			
Cientes: Avaliadores		Apresentador	



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

Avenida Professor Luciano Gualberto, travessa 3 nº 158 CEP 05508-900 São Paulo SP
Telefone: (011) 818-5583 Fax (011) 818-5294

Departamento de Engenharia de Computação e Sistemas Digitais

Teste no. 08		Requisito testado Programação aula-a-aula - consulta	
Funcional: <input checked="" type="checkbox"/>		Observações:	
Não funcional: <input type="checkbox"/>			
Seqüência para verificação do caso comum	Comportamento esperado do software	Testado	
1. Clicar em DISCIPLINAS	O sistema deverá apresentar as opções de disciplinas		
2. Clicar em PROGRAMACAO AULA-A-AULA	O sistema deverá apresentar as opções disponíveis.		
3. Selecione dados para os campos presentes na tela			
4. Clicar em BUSCAR	O sistema deverá listar as aulas adicionadas para a turma		
5.			
6.			
7.			
Seqüência para verificação de exceções	Comportamento esperado do software		
1. ITEM 4 – Se nenhuma aula associada	Não é apresentada aula associada à turma		
2.			
3.			
4.			
5.			
6.			
7.			
<input type="checkbox"/> Aprovado	Comentários		
<input type="checkbox"/> Reprovado			
Cientes: Avaliadores		Apresentador	



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

Avenida Professor Luciano Gualberto, travessa 3 nº 158 CEP 05508-900 São Paulo SP
Telefone: (011) 818-5583 Fax (011) 818-5294

Departamento de Engenharia de Computação e Sistemas Digitais

Teste no. 09		Requisito testado Programação aula-a-aula - alteração	
Funcional: <input checked="" type="checkbox"/>		Observações:	
Não funcional: <input type="checkbox"/>			
Seqüência para verificação do caso comum	Comportamento esperado do software	Testado	
1. Clicar em DISCIPLINAS	O sistema deverá apresentar as opções de disciplinas		
2. Clicar em PROGRAMACAO AULA-A-AULA	O sistema deverá apresentar as opções disponíveis.		
3. Selecione dados para os campos presentes na tela			
4. Clicar em BUSCAR	O sistema deverá listar as aulas adicionadas para a turma.		
5. Clicar em ALTERAR	O sistema deverá disponibilizar a alteração de dados		
6. Clicar em OK			
7.			
Seqüência para verificação de exceções	Comportamento esperado do software		
1. ITEM 4 – Se nenhuma aula associada	Não é apresentada aula associada à turma		
2. ITEM 6 – Se os campos obrigatórios não forem preenchidos ou algum campo tiver sido preenchido em algum formato inválido	Sistema pede para que os dados sejam inseridos novamente.		
3.			
4.			
5.			
6.			
7.			
<input type="checkbox"/> Aprovado	Comentários		
<input type="checkbox"/> Reprovado			
Cientes: Avaliadores		Apresentador	



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

Avenida Professor Luciano Gualberto, travessa 3 nº 158 CEP 05508-900 São Paulo SP
Telefone: (011) 818-5283 Fax (011) 818-5294

Departamento de Engenharia de Computação e Sistemas Digitais

Teste no. 10		Requisito testado Recálculo de Presença	
Funcional: <input checked="" type="checkbox"/>		Observações:	
Não funcional: <input type="checkbox"/>			
Seqüência para verificação do caso comum	Comportamento esperado do software	Testado	
1. Clicar em DISCIPLINAS	O sistema deverá apresentar as opções de disciplinas		
2. Clicar em RECALCULO DE PRESENÇA	O sistema deverá apresentar as opções disponíveis.		
3. Selecione dados para os campos presentes na tela	Sistema exibe a lista dos alunos pertencentes àquela turma, com suas presenças na aula selecionada		
4. Clicar em OK	Sistema exibe resultado		
5.			
6.			
7.			
Seqüência para verificação de exceções	Comportamento esperado do software		
1.			
2.			
3.			
4.			
5.			
6.			
7.			
<input type="checkbox"/> Aprovado	Comentários		
<input type="checkbox"/> Reprovado			
Cientes: Avaliadores:		Apresentador	



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

Avenida Professor Luciano Gualberto, travessa 3 nº 158 CEP 05508-900 São Paulo SP
Telefone: (011) 818-5583 Fax (011) 818-5294

Departamento de Engenharia de Computação e Sistemas Digitais

Teste no. 11		Requisito testado Consulta / Ajuste de presença	
Funcional: <input checked="" type="checkbox"/>		Observações:	
Não funcional: <input type="checkbox"/>			
Seqüência para verificação do caso comum	Comportamento esperado do software	Testado	
1. Clicar em DISCIPLINAS	O sistema deverá apresentar as opções de disciplinas		
2. Clicar em CONSULTA DE PRESENÇA	O sistema deverá apresentar as opções disponíveis.		
3. Selecionar dados para os campos presentes na tela			
4. Clicar em OK	Sistema informações sobre a presença do aluno para a aula solicitada.		
5. Selecionar o tipo de validação de presença (individual ou para o grupo)	Sistema exibe resultado		
6.			
7.			
Seqüência para verificação de exceções	Comportamento esperado do software		
1. ITEM 5 - Se o usuário for aluno	Não realiza o passo		
2.			
3.			
4.			
5.			
6.			
7.			
<input type="checkbox"/> Aprovado	Comentários		
<input type="checkbox"/> Reprovado			
Cientes: Avaliadores		Apresentador	



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

Avenida Professor Luciano Gualberto, travessa 3 nº 158 CEP 05508-900 São Paulo SP
Telefone: (011) 818-5583 Fax (011) 818-5294

Departamento de Engenharia de Computação e Sistemas Digitais

Teste no. 12		Requisito testado Requerimento de Recálculo
Funcional: <input checked="" type="checkbox"/>	Observações:	
Não funcional: <input type="checkbox"/>		
Seqüência para verificação do caso comum	Comportamento esperado do software	Testado
1. Clicar em DISCIPLINAS	O sistema deverá apresentar as opções de disciplinas	
2. Clicar em REQUERIMENTO DE RECALCULO	O sistema apresenta os requerimentos pendentes	
3. Clicar em ADICIONAR REQUERIMENTO	O sistema deverá apresentar as opções disponíveis.	
4. Selecione dados para os campos presentes na tela		
5. Clicar em OK	Sistema deverá apresentar campo para preenchimento da justificativa	
6. Preencher formulário com a justificativa para o requerimento		
7. Clicar em OK		
8.		
Seqüência para verificação de exceções	Comportamento esperado do software	
1. ITEM 3 – Se usuário é professor	Pula para o passo 7	
2.		
3.		
4.		
5.		
6.		
7.		
<input type="checkbox"/> Aprovado	Comentários	
<input type="checkbox"/> Reprovado		
Cientes: Avaliadores		Apresentador



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

Avenida Professor Luciano Gualberto, travessa 3 nº 156 CEP 05508-900 São Paulo SP
Telefone: (011) 818-5583 Fax (011) 818-5294

Departamento de Engenharia de Computação e Sistemas Digitais

Teste no. 13		Requisito testado Estatísticas	
Funcional: <input checked="" type="checkbox"/>		Observações:	
Não funcional: <input type="checkbox"/>			
Sequência para verificação do caso comum		Comportamento esperado do software	Testado
1. Clicar em ESTATÍSTICAS		O sistema deverá apresentar as opções de estatísticas	
2. Selecione dados para os campos presentes na tela (tipo de gráfico, disciplinas envolvidas, períodos, etc.)			
3. Clicar em OK		O sistema apresentará as estatísticas no formato solicitado.	
4.			
5.			
6.			
7.			
Sequência para verificação de exceções		Comportamento esperado do software	
1. ITEM 3 – Se os campos obrigatórios não forem preenchidos ou algum campo tiver sido preenchido em algum formato inválido		Sistema pede para que os dados sejam inseridos novamente.	
2.			
3.			
4.			
5.			
6.			
7.			
<input type="checkbox"/> Aprovado	Comentários		
<input type="checkbox"/> Reprovado			
Cientes: Avaliadores		Apresentador	



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

Avenida Professor Luciano Gualberto, travessa 3 nº 158 CEP 05508-900 São Paulo SP
Telefone: (011) 818-5583 Fax (011) 818-5294

Departamento de Engenharia de Computação e Sistemas Digitais

Teste no. 14		Requisito testado Impressão de Listas	
Funcional: <input checked="" type="checkbox"/>		Observações:	
Não funcional: <input type="checkbox"/>			
Seqüência para verificação do caso comum		Comportamento esperado do software	Testado
1. Clicar em LISTAS		O sistema deverá apresentar as opções disponíveis.	
2. Selecione dados para os campos presentes na tela			
3. Clicar em OK		Lista de Presença será exibida na tela	
4.			
5.			
6.			
7.			
Seqüência para verificação de exceções		Comportamento esperado do software	
1. ITEM 3 – Se turma para aquela disciplina não tiver alunos cadastrados		A lista é impressa em branco.	
2.			
3.			
4.			
5.			
6.			
7.			
<input type="checkbox"/> Aprovado	Comentários		
<input type="checkbox"/> Reprovado			
Cientes: Avaliadores		Apresentador	



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

Avenida Professor Luciano Gualberto, travessa 3 nº 158 CEP 05508-900 São Paulo SP
Telefone: (011) 818-5583 Fax (011) 818-5294

Departamento de Engenharia de Computação e Sistemas Digitais

Teste no. 15		Requisito testado Registro de Presença	
Funcional: <input checked="" type="checkbox"/>		Observações:	
Não funcional: <input type="checkbox"/>			
Seqüência para verificação do caso comum		Comportamento esperado do software	Testado
1. Aluno passa a carteira USP no leitor magnético no início da aula		Registro da presença no banco de dados	
2. Aluno passa a carteira USP no leitor magnético no fim da aula		Registro da presença no banco de dados e cálculo da porcentagem de permanência em aula	
3. Presença é validada ao aluno			
4.			
5.			
6.			
7.			
Seqüência para verificação de exceções		Comportamento esperado do software	
1. ITEM 3 – Se o intervalo de passagem entre a primeira e a última vez que o aluno passou a carteira USP durante a aula for menor que a porcentagem de presença aceita para aquela disciplina		Presença não será validada.	
2.			
3.			
4.			
5.			
6.			
<input type="checkbox"/> Aprovado	Comentários		
<input type="checkbox"/> Reprovado			
Cientes: Avaliadores		Apresentador	

ANEXO A – EXEMPLOS DE RESPOSTAS OBTIDAS AOS QUESTIONÁRIOS

QUESTIONÁRIO DO PROFESSOR – EXEMPLO 1

- 1- Há quantos anos leciona na Escola Politécnica?
- ☐ Até 3 anos
 - ☐ Entre 3 e 5 anos
 - ☐ Entre 5 e 10 anos
 - ☒ Entre 10 e 15 anos
 - ☐ Há mais de 15 anos
- 2- Há quantos anos é professor?
- ☐ Até 3 anos
 - ☐ Entre 3 e 5 anos
 - ☐ Entre 5 e 10 anos
 - ☐ Entre 10 e 15 anos
 - ☒ Há mais de 15 anos
- 3- Concorda com a presença obrigatória? Porquê?
- Sim, concordo porque nosso curso é presencial e, portanto, tem como pressuposto que os alunos assistam a pelo menos 75% das aulas.
- 4- Acha que um controle automatizado da presença poderia trazer melhoria em relação ao processo atual? Porquê?
- Acho que pode trazer melhorias desde que sejam resolvidas questões ligadas com segurança e confiabilidade na obtenção dos dados.
- 5- Que regras para validar a presença você, como professor, acha adequado aplicar?
- a) O aluno deve chegar até um máximo de 15 minutos após o início da aula;
 - b) Deve permanecer na sala de aula até o final da aula.
- 6- De que modo você enxerga que os alunos possam fazer mau uso do sistema, tirando vantagem do mesmo?
- Como se trata de um sistema para validar frequência nas aulas, as tentativas de fraude serão sempre no sentido de um aluno tentar validar a presença de outros alunos (que não estão presentes), seja pela troca das carteirinhas ou passando várias carteirinhas diferentes pela leitora.
- 7- Onde você acha que seria ideal que a leitora de cartões ficasse? Dentro ou fora da sala? Ao lado da porta? Em cima da mesa do professor?
- Acho que deveria ficar dentro da sala de aula, em local de fácil acesso para os alunos, mas não na mesa do professor.
- 8- Espaço livre para comentários:
- Pensar a solução do sistema em duas situações:
- 1) Apenas substituir a atual lista de presença com vantagens (por exemplo: como vocês já mencionaram "melhorar o controle e disponibilidade das informações" etc.);
 - 2) Caso vocês desejem entrar no mérito de fazer com que ele possa detectar e coibir fraudes, as implicações serão bem mais abrangentes, exigindo a criação de procedimentos para alunos e professores, para que posteriormente possa ser implantado com sucesso.

QUESTIONÁRIO DO PROFESSOR – EXEMPLO 2

1- Há quantos anos leciona na Escola Politécnica?

- ☐ Até 3 anos
☐ Entre 3 e 5 anos
☐ Entre 5 e 10 anos
☒ Entre 10 e 15 anos
☐ Há mais de 15 anos

2- Há quantos anos é professor?

- ☐ Até 3 anos
☐ Entre 3 e 5 anos
☐ Entre 5 e 10 anos
☐ Entre 10 e 15 anos
☒ Há mais de 15 anos

3- Concorda com a presença obrigatória? Porquê?

A presença poderia ser flexibilizada mas não totalmente facultativa, isso é, deveria ser obrigatória nas atividades programadas pelo professor, a seu critério, quando as atividades o exigirem, tais como as que envolvem avaliação.

4- Acha que um controle automatizado da presença poderia trazer melhoria em relação ao processo atual? Porquê?

Sim pois o processo atual é absurdo. Uns assinam as listas pelos outros ou então o professor é obrigado a fazer chamada.

Acho o sistema mais útil como sistema de segurança para controle de acesso às salas e demais dependências e evitar a entrada de estranhos.

O objetivo é que aluno aprenda e o fato de estar presente não significa que esteja aprendendo. Deve-se criar situações e atividades que motivem o aluno a estar presente na sala de aula e não ficar controlando se o aluno está presente ou não.

5- Que regras para validar a presença você, como professor, acha adequado aplicar?

O sistema deveria ter um sensor de presença (soar um alarme se o aluno não passar a carteirinha) ou bloqueio para controlar a entrada e saída obrigando o aluno a passar a carteirinha. Leitor de digital seria mais efetivo em lugar de carteirinha que pode ser emprestada para um colega.

6- De que modo você enxerga que os alunos possam fazer mau uso do sistema, tirando vantagem do mesmo?

O esquema proposto pode ser burlado pois basta o aluno deixar a carteirinha com um colega. Precisa ter sensor de presença ou bloqueios para impedir ao aluno de sair sem passar a carteirinha. Outros sistemas alternativos usando biometrias disponíveis no mercado talvez funcionassem melhor (leitor de digital).

7- Onde você acha que seria ideal que a leitora de cartões ficasse? Dentro ou fora da sala? Ao lado da porta? Em cima da mesa do professor?

Na porta de entrada.

8- Espaço livre para comentários:

Com os meios eletrônicos atuais, o professor não deve ser visto como a fonte principal de informações e sim como um agente que promove atividades que favorecem o aprendizado. As atividades podem ser colaborativas e pode haver interação entre alunos e professores mesmo à distância.

No entanto, evoluímos para uma sociedade em que a presença é cada vez mais virtualizada (tele-presença) através dos meios eletrônicos (Internet, celular,...). Não seria o caso de se estimular a presença? Promover atividades que envolvam maior interação presencial entre as pessoas (alunos e professores)?

Vejam o que acontece quando o professor não controla presença:

a) O aluno em geral se desatualiza em relação à matéria dada e deixa acumular matéria.

O aluno ausente das aulas, deve se organizar para acompanhar a matéria dada, reservar um horário semanal para olhar a matéria e realizar as atividades. O professor deve programar atividades que resultem em desafios que forcem os alunos a pesquisar, interagir com o professor e colegas e se aprofundar sobre o assunto.

b) O professor em geral passa durante a aula comentários e observações de sua experiência profissional que muitas vezes não se encontra em livros ou na Internet. Além disso alguns professores ocupam cargos importantes em empresas ou costumam ser solicitados a indicar alunos para oportunidades de trabalho. O aluno ausente está jogando fora esta oportunidade pois dificilmente um professor irá indicar para o cargo um aluno ausente pois o sucesso depende, além da competência técnica, de fatores envolvendo fatores psicológicos da personalidade e caráter do aluno.

c) Existem atividades que deveriam ser estimuladas pelo professor e que envolvem apresentações dos alunos. Na vida profissional, o aluno deve se acostumar a expor suas idéias em público o que é difícil para a maioria das pessoas e é algo que exige treino. Na interação remota verifica-se que muitos alunos que são normalmente quietos em classe se manifestam com muito mais facilidade. Se o aluno não se acostumar e não desenvolver essas características vai limitar sua vida profissional e muitas vezes não será reconhecido.

d) Ainda não existem formas efetivas de avaliação a distância de forma que se garanta que o aluno é o real autor.

e) O aluno que não está presente às aulas, muitas vezes também não está presente na Escola. O aluno deixa de se integrar com os colegas e não usufrui o que a USP apresenta do ponto de vista de recursos, laboratórios, pesquisa, cultura e esportes.

QUESTIONÁRIO DO ALUNO – EXEMPLO 1

1- Com que frequência você diria que costuma assistir às aulas?

- () Sempre
(X) Frequentemente
() Em torno de metade
() Raramente
() Nunca

2- Concorda com a presença obrigatória? Porquê?

Parcialmente. Se o conteúdo da aula não for somente uma repetição do que se pode encontrar nos livros, mas uma complementação a essa e também algo mais prático e dinâmico onde os alunos possam questionar e tirar suas dúvidas, sim. Caso contrário, como ocorre em muitas disciplinas, não.

3- Quais vantagens e desvantagens você enxerga no sistema proposto em relação ao atual?

A vantagem é exatamente haver dados estatísticos sobre as aulas. Através delas será possível efetuar diversas avaliações do curso.

4- Você acha que o sistema dificulta a falsificação da presença?

Não. Da mesma forma que o aluno pode pedir para o seu colega assinar a lista de presença, ele pode emprestar a sua carteirinha. Da mesma forma em que o aluno entra na sala para assinar a lista e depois sai dela, isto pode ser feito com o controle da carteirinha.

5- Que métodos para validar a presença você, como aluno, acha adequado aplicar (exemplo: porcentagem de tempo da aula, passagem da carteirinha uma vez durante a aula, passagem da carteirinha nos primeiros 20 minutos de aula)?

Para entrar na sala de aula, o aluno deverá passar através de uma catraca eletrônica, liberando-a com a carteira USP. O mesmo deve ser feito para sair. Duas aulas sequenciais não podem ocorrer na mesma sala de aula. Assim obrigaria todos a saírem. A presença é avaliada através do tempo em aula do aluno.

6- De que modo você enxerga que o professor pode prejudicar o aluno com o sistema (exemplo: métodos abusivos de validação de presença ou vários métodos distintos que possam confundir o aluno)?

De nenhuma forma. Imagino que com o bom funcionamento do sistema, ambos, alunos e professores, não poderão argumentar fora dos dados estatísticos obtidos.

7- Onde você acha que seria ideal que a leitora ficasse? Dentro ou fora da sala? Ao lado da porta? Em cima da mesa do professor?

Como sugerido, em uma (ou várias) catraca(s) que bloqueia a entrada da sala.

8- Espaço livre para comentários:

A catraca não impede do aluno sair para ir ao banheiro por exemplo. Ela somente controla o horário de entrada e saída durante a aula.

QUESTIONÁRIO DO ALUNO – EXEMPLO 2

- 1- Com que frequência você diria que costuma assistir às aulas?
- ☐ Sempre
 - ☒ Frequentemente
 - ☐ Em torno de metade
 - ☐ Raramente
 - ☐ Nunca
- 2- Concorda com a presença obrigatória? Porquê?
- Sim. Embora em muitas matérias a presença não tenha a menor relevância, acho que não é liberando a presença do aluno que isso será resolvido. É preciso partir do pressuposto que o professor tem mais a acrescentar ao aluno do que os livros. Se é verdade ou não aí já outro problema.
- 3- Quais vantagens e desvantagens você enxerga no sistema proposto em relação ao atual?
- Agilidade nas chamadas, praticidade aos usuários, dificuldade do aluno obter presença mesmo sem estar presente à aula (não que isso seja, mas...).
- 4- Você acha que o sistema dificulta a falsificação da presença?
- Sim, claramente.
- 5- Que métodos para validar a presença você, como aluno, acha adequado aplicar (exemplo: porcentagem de tempo da aula, passagem da carteirinha uma vez durante a aula, passagem da carteirinha nos primeiros 20 minutos de aula)?
- O mais correto e que mais dificulta fraudes é o percentual de tempo durante a aula. O aluno passa a carteirinha sempre que sair ou entrar da sala.
- 6- De que modo você enxerga que o professor pode prejudicar o aluno com o sistema (exemplo: métodos abusivos de validação de presença ou vários métodos distintos que possam confundir o aluno)?
- Se cada professor tiver liberdade para criar suas regras de presença, ele pode fazê-las de modo abusivo: exigindo mais presença às aulas do que o ideal por ex.
- O ideal é q essas regras fossem as mesmas para todos os professores.
- 7- Onde você acha que seria ideal que a leitora ficasse? Dentro ou fora da sala? Ao lado da porta?
- Em cima da mesa do professor?
- Bom, depende. Se quisermos controlar o percentual de tempo q o aluno fica na sala, então a catraca deve ficar na porta (para controlar a saída do aluno no meio da aula).
- Fora da porta facilita fraudes: o aluno pode passar a carteirinha sem entrar na sala e ir dormir no CEE.
- Em cima da mesa do professor pressupõe que cada professor terá a sua leitora, mas acho q uma leitora por sala é mais indicado, já q deve ser mais barato.
- Bom, acho q a localização vai depender dos objetivos do sistema.
- 8- Espaço livre para comentários:
- Boa sorte no projeto.

