

**UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA
GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO**

GUSTAVO BECACICI ESTEVES VIANNA

**UMA APLICAÇÃO WEB DE AUXÍLIO À
ELABORAÇÃO DE OFERTA DE DISCIPLINAS PARA
O SISTEMA DE INFORMAÇÕES PARA ENSINO**

**VITÓRIA – ES
2010**

UMA APLICAÇÃO WEB DE AUXÍLIO À ELABORAÇÃO DE OFERTA DE DISCIPLINAS PARA O SISTEMA DE INFORMAÇÕES PARA ENSINO

Parte manuscrita do Projeto de Graduação
do aluno Gustavo Becacici Esteves
Vianna, apresentado ao Departamento de
Informática do Centro Tecnológico da
Universidade Federal do Espírito Santo,
para obtenção do grau de Engenheiro de
Computação.
Orientador: Prof. M.Sc. Flávio Miguel
Varejão

**VITÓRIA – ES
2010
GUSTAVO BECACICI ESTEVES VIANNA**

UMA APLICAÇÃO WEB DE AUXÍLIO À ELABORAÇÃO DE OFERTA DE DISCIPLINAS PARA O SISTEMA DE INFORMAÇÕES PARA ENSINO

Parte manuscrita do Projeto de Graduação do aluno Gustavo Becacici Esteves Vianna, apresentado ao Departamento de Informática do Centro Tecnológico da Universidade Federal do Espírito Santo, para obtenção do grau de Engenheiro de Computação.

COMISSÃO EXAMINADORA:

**Prof. MsC
Flávio Miguel Varejão
Orientador**

**Prof. Dr.
Hans-Jorg Andreas Schneebeli
Co-orientador**

**Renan Teixeira de Souza
Examinador**

**Profª. MsC
Rosane Santos Caruso de Oliveira
Examinadora**

DEDICATÓRIA

À minha família, pais, irmão e avós, que sempre estiveram presentes, mesmo que em espírito, olhando por mim e sempre preparando tudo para que eu pudesse atingir esse objetivo, a graduação. Dedico a vocês, especialmente ao meu pai, à minha mãe e ao meu irmão, o meu sucesso é de vocês!

AGRADECIMENTOS

Apesar do clichê, não posso deixar de agradecer primeiramente a Deus, pela jornada concluída e por tudo que conquistei nesta longa fase de intenso aprendizado.

À minha família, pelo suporte e compreensão, sempre! Amo vocês!

A você, namorada, pelo apoio e cobrança constantes, que me fizeram vencer a preguiça e outros fatores para obter esse sucesso. Te amo!

Aos amigos! Incluo aqui meus colegas de curso, pois a grande maioria deles se tornou parte da minha vida. Agradeço a vocês pela amizade, que me deu forças durante esses longos anos dentro da universidade, se não fossem vocês não teria conseguido; pelas noites em claro que valeram muito à pena. Não somente pelo objetivo atingido, ou, muitas vezes não atingido, mas pela diversão de passar noites a base de café, guaraná e de muitas risadas. E aos amigos de longa data, que fora do ambiente acadêmico, em bares, festas e mesmo em casa reabasteceram a energia que me era consumida nos estudos. Valeu galera!

Aos meus colegas do NPD, pelo apoio incondicional, pela paciência e ambiente descontraído e acolhedor, que muito facilitou o sucesso desse trabalho, aqui apresentado.

Ao Renan Teixeira, pela paciência, confiança em mim depositada e pela amizade que foi construída nesses anos de convivência, intra e extra-NPD.

Ao Prof. Hans, por ceder o espaço do NPD para que esta monografia pudesse ser concluída, e pela forma que explicava e re-explicava tudo com a maior paciência do mundo. Muito obrigado!

Ao Rafael Valverde, pela ajuda imprescindível na implementação do projeto e pelo aprendizado adquirido ao trabalharmos juntos.

Ao Prof. Flávio Varejão, meu orientador, pelo tema do projeto e pelo suporte.

À Prof^a. Rosane que, apesar dos problemas pôde compor a banca.

Aos colaboradores Márcio Sfalsin e Nivaldo Arruda que muito me ajudaram tirando dúvidas e, sempre muito solícitos, estiveram em contato durante períodos sombrios.

Ao Flávio Lisboa, autor do livro que mais utilizei como base para o desenvolvimento da aplicação.

“Não há vitória sem luta... Não há sonho que não possa ser realizado!”
Autor desconhecido

“As pessoas boas devem amar seus inimigos.”
Seu Madruga

RESUMO

Neste trabalho foi implementado um novo módulo para oferta de disciplinas do SIE (Sistema de Informações para Ensino) – sistema responsável pelo gerenciamento acadêmico de instituições de ensino. É uma aplicação que usa a plataforma web e, de forma amigável, dá aos operadores a facilidade de efetuar a oferta de turmas favorecendo-o na alocação de recursos como sala de aula, horário e professores, evitando a concorrência no uso desses recursos e simplificando o trabalho a ser feito. Permite ao usuário, o coordenador de curso, por exemplo, entrar com os dados de oferta de disciplina, onde as restrições de permissão, de unidade de lotação, de encargos didáticos e diversas outras, são tratadas automaticamente, de acordo com os requisitos do sistema, impedindo que diferentes professores ministrem a mesma aula (salvo casos excepcionais), ou mesmo alertando quando houver a ocupação de diversas turmas em um mesmo horário e/ou local na agenda do docente.

Palavras-chave: Sistema de gerenciamento acadêmico; Oferta de disciplinas; Oferta de turmas; Sistema Web; Zend Framework; Dojo Toolkit.

ABSTRACT

In this work a new module of the SIE (Education Information System) – the system responsible for managing academic teaching institutions – was implemented: the module responsible for building the offer of subjects. It is an application that uses the Web platform, and in a friendly way, gives the operators the facility to make the offering of classes helping them in the allocation of resources as the classroom, the time of each classes and the teachers, to avoid competition in the use of these resources and simplifying the work that has to be done. Allows the user, the course coordinator, for example, to enter the data for the provision of discipline, which permission restrictions, unit manning, amount of work-hours of each teacher and some others are processed automatically, according to system requirements, preventing that different teachers minister the same class (except in exceptional cases), or even if the occupation of several classes in the same time and / or to be placed on the agenda of the teacher a warning is showed.

Keywords: Academic Management System; Offer Subjects; Offer Classes; Web System, Zend Framework; Dojo Toolkit. Oferta de disciplinas; Oferta de turmas

LISTA DE FIGURAS

Figura 1 - Exemplo de fluxo do trâmite de uma turma.....	22
Figura 2 - Máquina de estados de tramitação de uma turma.	23
Figura 3 - Diagrama de entradas e saídas do sistema.	24
Figura 4 - Tela inicial do SIE para a função de Oferta de Disciplinas	25
Figura 5 - Oferta de Disciplinas por Curso.....	26
Figura 6 - Listagem de disciplinas do Curso de Engenharia de Computação, usado neste exemplo.....	27
Figura 7 - Criar nova oferta de disciplina.....	27
Figura 8 - Formulário para uma nova oferta de disciplina.	28
Figura 9 - Tela de escolha da disciplina para a qual a turma será ofertada.	29
Figura 10 - Tela de busca da disciplina, com uma lista de disciplinas exibida.	29
Figura 11 - Após seleção da disciplina o formulário é preenchido.	30
Figura 12 – Oferta de turma feita, para o Curso de Engenharia de Computação, na disciplina Algoritmos e Fundamentos da Teoria de Computação, com código da turma 22 e 02 vagas oferecidas.....	31
Figura 13 – Tela inicial do módulo de Oferta de Disciplinas.....	32
Figura 14 – Funcionalidade de Oferta de Disciplinas por Departamento.	32
Figura 15 – Formulário de seleção de Departamento.....	33
Figura 16 – Formulário de busca, preenchido para filtrar o Departamento cuja sigla é “INF”.	34
Figura 17 – Disciplina de Algoritmos Numéricos 1, do Departamento de Informática.....	35
Figura 18 – Mostra que os departamentos não têm poder de criar novas turmas.....	35
Figura 19 – Após clicar em “alterar” quase todos os campos são habilitados para edição. ...	36
Figura 20 – Tela de alteração do escopo de uma determinada turma.....	37
Figura 21 - Arquitetura do SIE.	39
Figura 22 - Fluxo de dados na arquitetura cliente-servidor do SIE.....	40
Figura 23 - Fluxo de dados na arquitetura cliente-servidor usual.	41
Figura 24 - Arquitetura cliente-servidor usada pelo novo sistema.....	45
Figura 25 - Diagrama de caso de uso do Departamento.....	50
Figura 26 - Diagrama de casos de uso do Colegiado de Curso.	51
Figura 27 - Arquitetura MVC.....	62
Figura 28 - Funcionamento da arquitetura MVC.	63

Figura 29 - Diagrama de classes simplificado - Zend_View com Zend_Form.....	70
Figura 30 - Diagrama de classes simplificado - Zend_Controller com Zend_View.	71
Figura 31 - Tela de login do novo sistema	73
Figura 32- Página inicial. Seleção de parâmetro para listagem de disciplinas por departamento.....	73
Figura 33 - Tela inicial com detalhamento explicativo de cada elemento (links, campos e botões) da página.	74
Figura 34 - Listagem de disciplinas ofertadas do Departamento de Informática, 2010/2.	74
Figura 35 - Listagem de disciplinas do Departamento de Informática, 2010/2, com exibição das Informações Gerais da turma 01 de Algoritmos Numéricos 1, versão de grade curricular 2008.	76
Figura 36 – Tela de alteração de dados de Algoritmos Numéricos 1, do Curso de Física (Bacharelado).....	77
Figura 37 – Tela de progresso do salvamento das alterações efetuadas.....	78
Figura 38 – Tela de confirmação do salvamento das alterações feitas.....	79
Figura 39 - Tela de escolha de outra disciplina do Departamento de Informática, anteriormente escolhido.....	80
Figura 40 – Na mesma tela de alterações de dados, na aba de horários pode-se editar os horários das aulas da turma de Algoritmos Numéricos 1.	81
Figura 41 – Tela de alteração de um determinado horário.	82
Figura 42 - Tela de inserção de um novo horário na turma 01 de Algoritmos Numéricos 1.	83
Figura 43 - Confirmação de exclusão de um dado horário da turma de Algoritmos Numéricos 1.....	84
Figura 44 - Alerta de conflito de horários, solicitando que um novo horário seja escolhido, e fazendo uma sugestão.	85
Figura 45 - MVC do novo sistema de oferta de disciplinas.	90
Figura 46 - PHPUnit instalado no Netbeans.....	92
Figura 47 - Configuração do Apache Tomcat.	101
Figura 48 - Página inicial do Apache Tomcat.	102
Figura 49 - Apache Tomcat Manager.....	103
Figura 50 - Tela de implantação de um novo projeto.	103
Figura 51 - Listagem dos projetos existentes no servidor.	104
Figura 52 - Tela de instalação do Cliente DB2.....	105

Figura 53 - Segunda tela de instalação do Cliente DB2.	106
Figura 54 - Arquivo “.bat” usado para instalar o “PEAR”.	110
Figura 55 - Adicionando o caminho da instalação do “PEAR” nas variáveis de ambiente manualmente ou executando o arquivo “PEAR_ENV.reg” que foi gerado na instalação do “PEAR” e se encontra dentro da pasta “php/PEAR”.....	111
Figura 56 - Arquivo .reg utilizado para adicionar o caminho de instalação do PEAR nas variáveis de ambiente.	112
Figura 57 – Local onde o caminho deve ser mudado.	112

LISTA DE TABELA

Tabela 1 - Permissões de alteração de dados de acordo com o perfil do usuário.....	48
Tabela 2 - Permissões de inserção de dados de acordo com o perfil do usuário.....	48
Tabela 3 - Permissões de exclusão de dados de acordo com o perfil do usuário	49
Tabela 4 - Descrição do caso de uso "Alterar Permissões"	52
Tabela 5 - Descrição do caso de uso "Alterar Conta"	53
Tabela 6 - Descrição do caso de uso "Consultar Ajuda"	53
Tabela 7 - Descrição do caso de uso "Listar Cursos Por Departamento"	54
Tabela 8 - Descrição do caso de uso "Visualizar Turma"	54
Tabela 9 - Descrição do caso de uso "Alterar Turma"	55
Tabela 10 - Descrição do caso de uso "Cadastrar Turma"	56
Tabela 11 - Descrição do caso de uso "Excluir Turma"	57
Tabela 12 - Descrição do caso de uso "Solicitar Turma"	57
Tabela 13 - Comparativo de frameworks.	64

LISTA DE ABREVIATURAS E SIGLAS

ZF	<i>Zend Framework</i>
SIE	<i>Sistema de Informações para Ensino</i>
UFES	<i>Universidade Federal do Espírito Santo</i>
UFSM	<i>Universidade Federal de Santa Maria</i>
NPD	<i>Núcleo de Processamento de Dados</i>
DLL	<i>Dynamic-link Library</i>
COM	<i>Component Object Model</i>
DCOM	<i>Distributed Component Object Model</i>
UML	<i>Unified Modeling Language</i>
ODBC	<i>Open Data Base Connectivity</i>
OLEDB	<i>Object Linking and Embedding, Database</i>
JDBC	<i>Java Database Connectivity</i>
API	<i>Application Programming Interface</i>
MVC	<i>Model-view-controller</i>
PHP	<i>PHP: Hypertext Preprocessor</i>
SGBD	<i>Sistema de Gerenciamento de Banco de Dados</i>
ENADE	<i>Exame Nacional de Desempenho de Estudantes</i>
HTML	<i>HyperText Markup Language</i>
W3C	<i>Consórcio World Wide Web</i>
PDO	<i>PHP Data Objects</i>
L10n	<i>Localização</i>
I18n	<i>Internacionalização</i>
JSON	<i>JavaScript Object Notation</i>
RPC	<i>Remote Procedure Call</i>
XML	<i>eXtensible Markup Language</i>
AJAX	<i>Asynchronous Javascript And XML</i>
DOM	<i>Document Object Model</i>
RoR	<i>Ruby on Rails</i>
IDE	<i>Integrated Development Environment</i>
OO	<i>Orientada a objetos</i>
DAO	<i>Data Access Object</i>
ADO	<i>ActiveX Data Objects</i>

URL	<i>Uniform Resource Locator</i>
PEAR	<i>PHP Extension and Application Repository</i>
WAMP	<i>Windows, Apache, MySQL, PHP ou Perl ou Python</i>
IDE	<i>Integrated Development Environment</i>

GLOSSÁRIO

Chefe de Departamento – Representantes oficiais de um Departamento Acadêmico.

Colegiado de Curso – Órgão responsável pela coordenação didática dos cursos e elaboração de currículos. Além disso, decide questões referentes à matrícula e coordena e executa os procedimentos de avaliação do curso.

Coordenador de Curso – Representantes oficiais de um Curso Acadêmico.

Curso Acadêmico – Conjunto de disciplinas onde alunos se matriculam para aprender um determinado tema.

Departamento Acadêmico – O Departamento é a base da estrutura da universidade. Ele reúne os professores por área de conhecimento e as atividades-fins, como ensino, pesquisa e extensão. Todo professor faz parte de um departamento.

Disciplina – Área de conhecimento estudada e ministrada em um ambiente acadêmico, sendo proveniente de uma matriz curricular.

Grade curricular – Conjunto de relações de interdependência e temporalidade entre as disciplinas e atividades do currículo de um curso.

Oferta de disciplinas – Listagem das turmas oferecidas em um determinado semestre, com o intuito de serem ocupadas por um grupo de alunos.

Pró-Reitoria de Graduação – Órgão responsável pelo gerenciamento da vida acadêmica dos alunos sejam eles presenciais ou de ensino a distância.

Turma – Conjunto de alunos que aprendem um determinado tema ou assunto de uma disciplina específica, em um determinado horário e espaço-físico.

SUMÁRIO

1	INTRODUÇÃO	17
1.1	Motivação	17
1.2	Solução desenvolvida	18
1.3	Metodologia.....	18
1.4	Estrutura do Trabalho	19
2	MÓDULO DE OFERTA DE DISCIPLINAS DO SISTEMA DE	
	INFORMAÇÕES PARA O ENSINO – SIE.....	20
2.1	Módulo Acadêmico	20
2.2	Oferta de disciplinas	21
2.3	Especificação funcional.....	24
2.4	Algumas deficiências observadas	38
3	UM MÓDULO DE OFERTA DE DISCIPLINAS WEB PARA O SIE	44
3.1	Concepção do projeto	44
3.2	Agentes envolvidos	45
3.3	Levantamento de requisitos	47
3.3.1	Requisitos funcionais	47
3.3.2	Casos de uso simplificado*****	50
4	IMPLEMENTAÇÃO E AVALIAÇÃO.....	59
4.1	Implementação do projeto	59
4.1.1	PHP 5.....	59
4.1.2	A arquitetura <i>Model-View-Controller</i> (MVC)	60
4.1.3	Frameworks	63
4.1.4	Escolha do <i>framework</i> e da biblioteca.....	64
4.1.5	Zend Framework	67
4.1.6	Instalação e configurações.....	68
4.1.7	Desenvolvimento	69
5	CONCLUSÃO E TRABALHOS FUTUROS	94
5.1	Conclusão	94
5.2	Trabalhos futuros.....	94
	ANEXOS.....	96
	APÊNDICES	114
	BIBLIOGRAFIA	117

1 INTRODUÇÃO

1.1 Motivação

Este trabalho foi motivado pela demanda de assegurar que as informações fornecidas ao sistema de oferta de disciplinas estejam em conformidade com as normas e com os requisitos da instituição, contidas no ANEXO 1, como recomendação da PROGRAD (Pró-Reitoria de Graduação).

A UFES (Universidade Federal do Espírito Santo) usa, atualmente, o SIE, que é um sistema de plataforma desktop, desenvolvido em 1998, pela UFSM (Universidade Federal de Santa Maria / RS), com o intuito de permitir a circulação, o compartilhamento e a gestão de informações referentes às diversas atividades de uma Instituição de Ensino Superior, por meio da integração de módulos [1] e a partir da utilização de uma base de dados unificada, independente do Sistema de Gerenciamento de Banco de Dados (SGBD) existente.

Partindo do pressuposto que, na época de seu desenvolvimento, o sistema supria a demanda existente, e que a web não era tão conhecida e amplamente utilizada quanto hoje, é fácil perceber, que o aumento da quantidade de dados e o surgimento de novas linguagens exigiu uma mudança na forma e nas ferramentas de programação, o que evidenciou as principais deficiências do SIE, principalmente no que tange à integridade de dados e à interface apresentada.

Essa interface dificulta a utilização do sistema, uma vez que inúmeras janelas são abertas confundindo o usuário. E, pela sua plataforma, limita a distribuição e torna mais árdua a manutenção e a atualização do sistema, já que se encontra instalada em diversos clientes, tornando sua atualização mais complexa e onerosa à instituição.

Além disso, a arquitetura de 2 camadas adotada pelo SIE, que, na época se tornou popular em função do baixo preço e da simplicidade de implantação, quando comparada com um modelo de 3 ou mais camadas (multi-tiers [2]), não supre mais as necessidades da universidade. Foi constatada, portanto, a urgência de um novo módulo de oferta de disciplinas, visando reduzir o tempo de espera dos usuários, evitar a ocorrência de possíveis erros, automatizar o uso dos recursos, facilitar a acessibilidade, centralizar o gerenciamento, simplificar a manutenção e muitos outros atributos que serão mencionados mais à frente.

1.2 Solução desenvolvida

O núcleo deste trabalho de graduação é o desenvolvimento de um novo módulo de oferta de disciplinas para o SIE, fazendo uso, porém, de outras tecnologias e partindo para uma abordagem mais atual, aplicando as devidas restrições, evitando a ocorrência de conflitos e auxiliando o usuário nas suas tarefas diárias, e, por conseguinte, sanando os gargalos do sistema existente.

A nova aplicação usa a plataforma web e, de forma amigável, dá aos operadores a opção de efetuar a oferta de turmas fazendo a alocação de recursos como sala de aula, horário e professores, e impedindo que haja concorrência no uso desses recursos. Restrições de permissão, de unidade de lotação, de encargos didáticos e diversas outras, são tratadas automaticamente e são transparentes ao usuário.

1.3 Metodologia

É necessário que o sistema desenvolvido esteja em consonância com as demandas da UFES. Para tanto, foi imprescindível uma “reengenharia” do módulo existente e, a partir de então, com auxílio de ferramentas de análise e implementação, um novo modelo de sistema foi definido.

Inicialmente fez-se um dimensionamento das necessidades junto aos atores envolvidos, que são: os Colegiados de Curso, os Departamentos Acadêmicos e a Pró-Reitoria de Graduação, que irão utilizar o sistema. Então se definiu, com a identificação da necessidade de negócio, a abrangência e a profundidade do escopo do projeto. Isso propiciou uma visão geral dos processos e dos detalhes específicos demandados pelos usuários nele envolvidos.

Houve também um levantamento dos requisitos para que as restrições pudessem ser identificadas, as devidas funcionalidades e características fossem delineadas e os casos de uso descritos. É nessa fase que os atributos e os serviços foram mapeados e, usando uma ferramenta de modelagem, por exemplo, através de diagramas UML, o modelo foi produzido, onde as informações sobre o domínio, as funções e o comportamento do software foram expostas.

A escolha do PHP (PHP: Hypertext Preprocessor), como linguagem de programação utilizada, se deu pela rapidez de prototipagem e pela existência de um trabalho

já realizado pelo NPD (Núcleo de Processamento de Dados / UFES, local de experimentação e implementação desse projeto).

Visando o aumento da necessidade de módulos como este, foram utilizados frameworks na implementação do módulo mencionado acima. Com a análise detalhada de diversos frameworks, em cruzamento com a demanda, foram escolhidos os mais adequados.

Utilizou-se o paradigma de Orientação a Objeto (OO) [3], com o intuito de aumentar a reutilização do código e, conseqüentemente, a produtividade do programador, diminuir a complexidade das rotinas através de um melhor grau de abstração e facilitar a comunicação entre clientes e programadores, graças à padronização da metodologia, tendo como resultado um produto mais completo e moderno.

Por fim, uma seqüência de testes, validações e verificações foi feita junto aos operadores finais do sistema, buscando a identificação e correção de erros produzidos inadvertidamente pelos desenvolvedores do software. Essa etapa garante a qualidade do produto gerado.

1.4 Estrutura do Trabalho

Neste trabalho foi documentado todo o processo de desenvolvimento do módulo de oferta de disciplina web, considerando as principais fases a seguir:

- análise da demanda,
- levantamento de requisitos,
- estudo para a escolha da tecnologia,
- implementação, e
- testes.

No Capítulo 2 é descrito detalhadamente o módulo de Oferta de Disciplinas do SIE, atualmente utilizado. A solução proposta é explanada no Capítulo 3, onde são mencionados todos os prós e contras analisados para que o desenvolvimento seguisse o modelo desejável. Detalhes da implementação estão no Capítulo 4. E por fim nos Capítulos 5 e 6 é feita uma avaliação dos resultados e a conclusão.

2 MÓDULO DE OFERTA DE DISCIPLINAS DO SISTEMA DE INFORMAÇÕES PARA O ENSINO – SIE.

2.1 Módulo Acadêmico

Visando facilitar os processos que envolvem docentes e discentes, esse módulo registra a vida escolar/acadêmica dos alunos, desde o seu ingresso até o fim da sua graduação, emitindo horários individuais, históricos parciais, certificados, diplomas e outros documentos. Em instituições privadas é possível até mesmo emitir carnês de pagamento e gerenciar a vida financeira do aluno.

O SIE trabalha de forma descentralizada em relação às suas diversas funcionalidades [4], que englobam as seguintes interfaces:

- **Controle Acadêmico;**
 - **Organização do Ensino;**
 - **Cadastro do Aluno;**
 - **Oferta de Disciplinas;**
 - **Matrícula;**
 - **Histórico Escolar;**
 - **Estatísticas – MEC (e-Mec [5] ¹, Enade [6] ², PingIFES [7] ³).**
- Controle Financeiro (geralmente não utilizado por instituições públicas);
- Produção Institucional;
- Assistência Estudantil;
- Atividades Docentes – Portais;
- Aplicações de Chamadas;
- Atividades Discentes – Portais;
- *Moodle*;
- PingIFES.

Encontram-se, em destaque, as funcionalidades relativas ao “sub-módulo” detentor da aplicação de oferta de disciplinas, que é objeto desse trabalho. Portanto, as funções que não são contidas nele, serão desconsideradas.

¹ O *e-MEC* é um sistema eletrônico de acompanhamento dos processos que regulam a educação superior no Brasil.

² O Enade tem o objetivo de aferir o rendimento dos alunos dos cursos de graduação em relação aos conteúdos programáticos, suas habilidades e competências.

³ O PingIFES é um sistema de coleta de dados de gestão das IFES.

2.2 Oferta de disciplinas

O objetivo desta funcionalidade é efetuar o ajuste da montagem da oferta de disciplinas para a realização da matrícula, além de montar e/ou alterar a grade de alocação de docentes e do espaço físico para cada turma ofertada em cada curso [8].

A fim de auxiliar e automatizar o procedimento da oferta de turmas, esse processo foi concebido com diferentes funções, como:

- Oferta de Disciplinas por Curso;
- Oferta de Disciplinas por Departamento;
- Relação Turmas de Blocos;
- Alteração de Escopo da Turma;
- Oferta de Disciplinas por nível de ensino (médio/técnico/fundamental);
- Geração de agenda para turmas sem agenda;
- Consultas e Geração de Relatórios.

Essas funções são utilizadas pelos coordenadores de curso, que são representantes oficiais de um Curso Acadêmico – cuja incumbência está descrita na resolução nº 11/87 do CEPE/UFES⁴ – e pelos chefes de departamento, representantes oficiais de um Departamento Acadêmico – que seguem as restrições descritas no capítulo 2, seção III, artigo 63 do Estatuto da UFES⁵, e atribuições definidas no capítulo 3, artigo 30 do Regimento Interno da UFES⁶ – para efetivar a oferta de disciplinas e atuar devidamente no trâmite de cada turma existente.

A oferta de uma turma segue um roteiro padrão (ANEXO 2), definido pela instituição, e seguido pelo sistema. Na oferta de uma ou mais turmas um documento é criado, marcando assim o início, pelo Colegiado, do procedimento de oferta através do SIE. A partir de então, é feito o registro e a documentação de todas as ações realizadas, em cada setor pelo qual o documento tramitou (ou foi tramitado) e, finalmente, chegando ao local em que se encontra atualmente.

A oferta de disciplinas pode ser entendida da seguinte forma:

- 1º. O Curso cria a turma na situação “Solicitada”, sugere ao Departamento as vagas e os horários da mesma e envia o pedido para “Análise” pelo Departamento responsável pela disciplina;

⁴ http://www.daocs.ufes.br/corpo.asp?pagina=resolucoes/resolucao_n11_87

⁵ http://www.daocs.ufes.br/corpo.asp?pagina=Estatuto_UFES

⁶ <http://www.daocs.ufes.br/Regimento%20Geral%20da%20UFES.pdf>

2°. O Departamento poderá vincular os docentes e espaços físicos à turma. Após, o Departamento analisa o pedido do Colegiado e pode optar por um dos seguintes fluxos:

- Retornar a turma para o Curso para “Ajuste de horários” ou
- Liberar a turma para “Matrícula”.

3°. Se o Departamento retornou a turma para o Curso, este último altera o que for necessário nos horários e novamente envia a turma para re-análise do Departamento, que liberará a turma para a “Matrícula”.

4°. O Departamento envia para o “Lançamento de Notas”;

5°. O Departamento torna uma turma “Inativa”.

A Figura 1 abaixo exhibe o fluxo de informações desse módulo, na qual é possível visualizar claramente onde o Departamento e o Colegiado atuam de modo a se organizarem segundo os moldes da universidade.

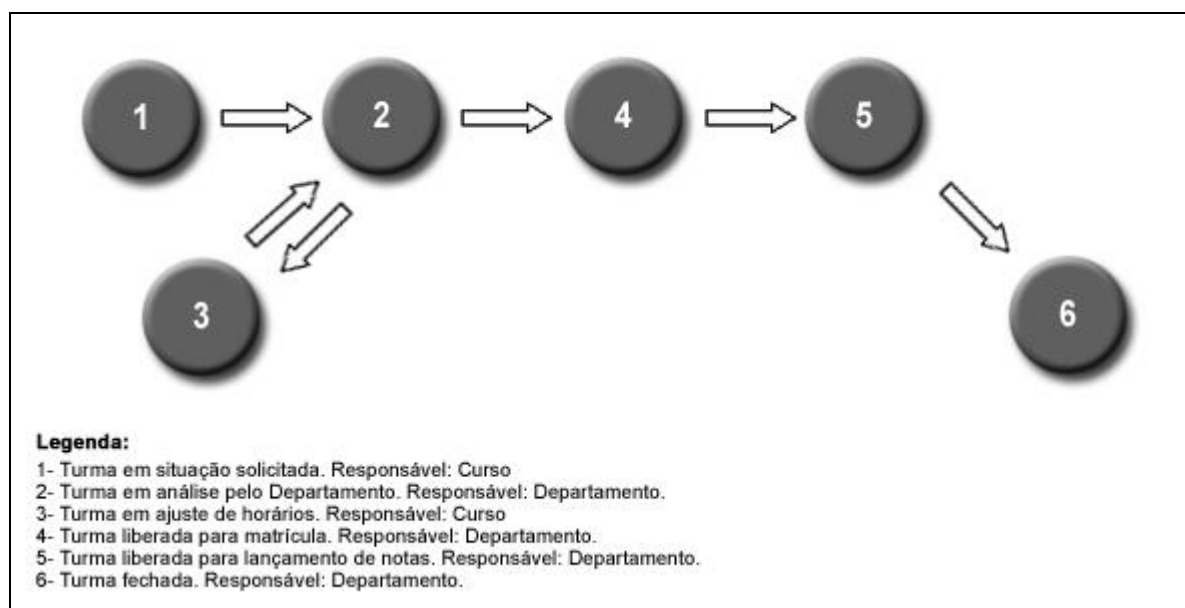


Figura 1 - Exemplo de fluxo do trâmite de uma turma.

Outra forma de visualização desse fluxo entre o Colegiado de Curso e o Departamento Acadêmico é o diagrama de estados, ilustrado na Figura 2 a seguir.

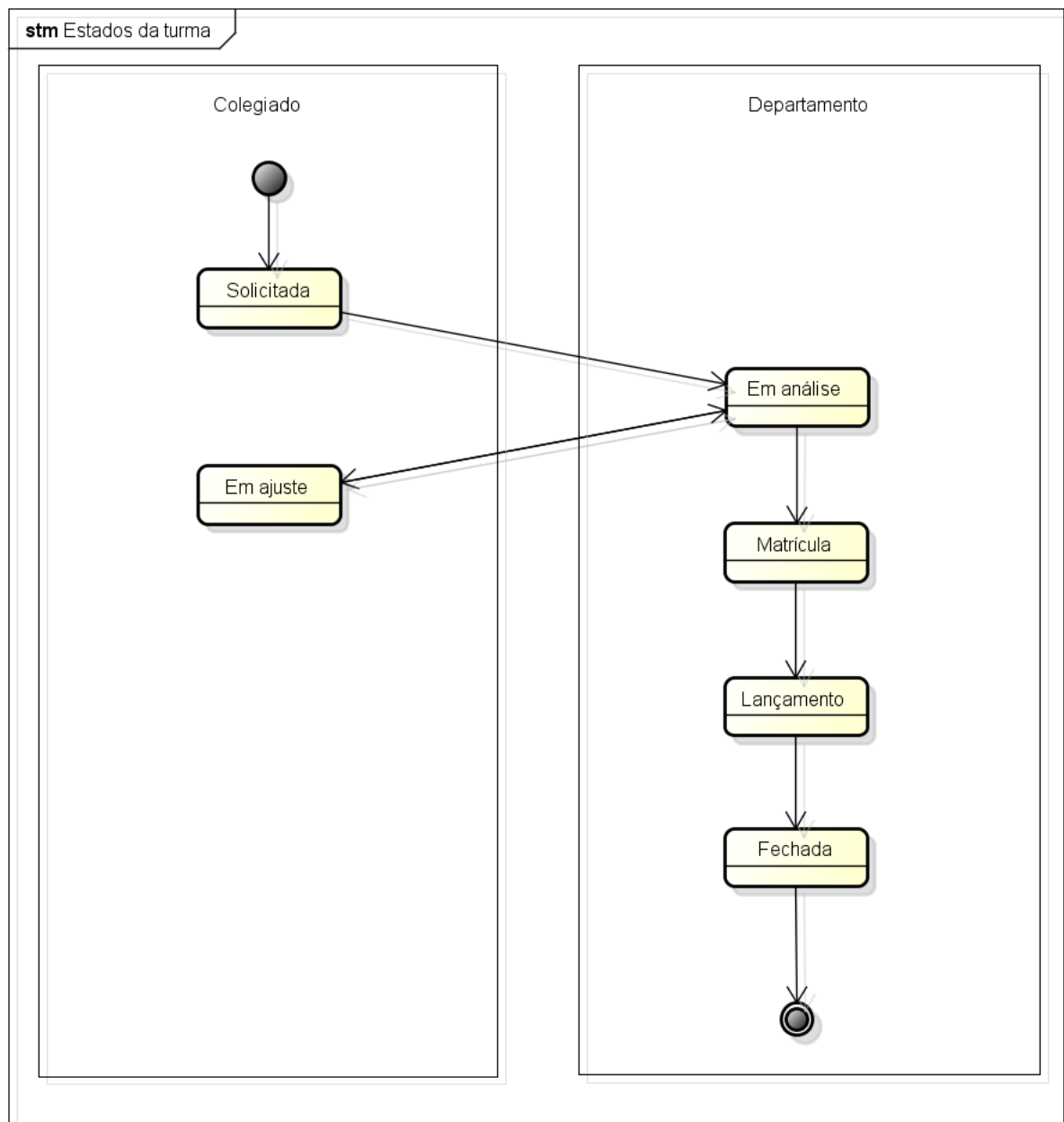


Figura 2 - Máquina de estados de tramitação de uma turma.

A Figura 3 ilustra um diagrama de entradas e saídas do sistema. Considerando os usuários, Colegiado de Curso e Departamento Acadêmico, tem-se que as entradas do sistema são, a saber:

- Currículo do Curso que respeitam os devidos pré-requisitos da grade curricular e sua versão, a unidade para a qual o Curso se reporta, auxiliando a montagem da oferta;

- Estatísticas e histórico das ofertas realizadas em semestres passados, que levam em consideração os turnos das aulas de períodos anteriores e a periodicidade de cada disciplina;
- Docentes que serão alocados em cada disciplina sem descumprir limitações como o encargo didático máximo de cada professor, a sua unidade de lotação, o seu papel na referida turma (professor principal, secundário A, secundário B, secundário C, secundário D e secundário E), entre outras restrições existentes;
- Espaço-físico onde serão ministradas as aulas atendendo ao tipo de aula (teórica, prática ou de exercícios), ao tamanho da sala, à quantidade de alunos, à estrutura da sala, à localização (em qual campus e centro se encontra), entre outras.

A saída do sistema é uma listagem de turmas ofertadas, com horários determinados, professor(es) alocados, espaço-físico definido, obedecendo todas as restrições e especificidades de cada curso, baseada nas entradas supracitadas.

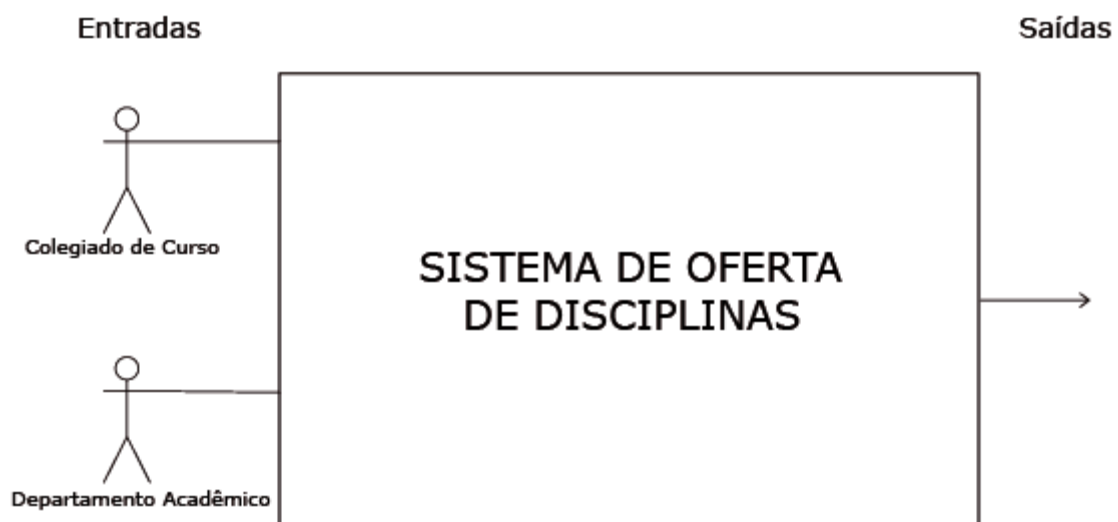


Figura 3 - Diagrama de entradas e saídas do sistema.

2.3 Especificação funcional

A especificação funcional será limitada às tarefas que afetam diretamente a oferta de disciplinas e a solução do problema. Ela está dividida em: oferta de disciplinas por Curso e

oferta de disciplinas por Departamento, uma vez que suas especificidades são determinantes no resultado final.

OFERTA DE DISCIPLINAS POR CURSO

Esta aplicação tem como objetivo efetuar a montagem da sugestão de oferta de disciplinas para a realização da matrícula dos alunos nos cursos [9], informando o número de vagas (ofertadas, ocupadas, disponíveis e aumentadas) e os horários das diferentes turmas. Como se pode ver no exemplo abaixo:

Na Figura 4, após entrar no sistema, optamos por fazer a “Oferta de Disciplinas por Curso”.

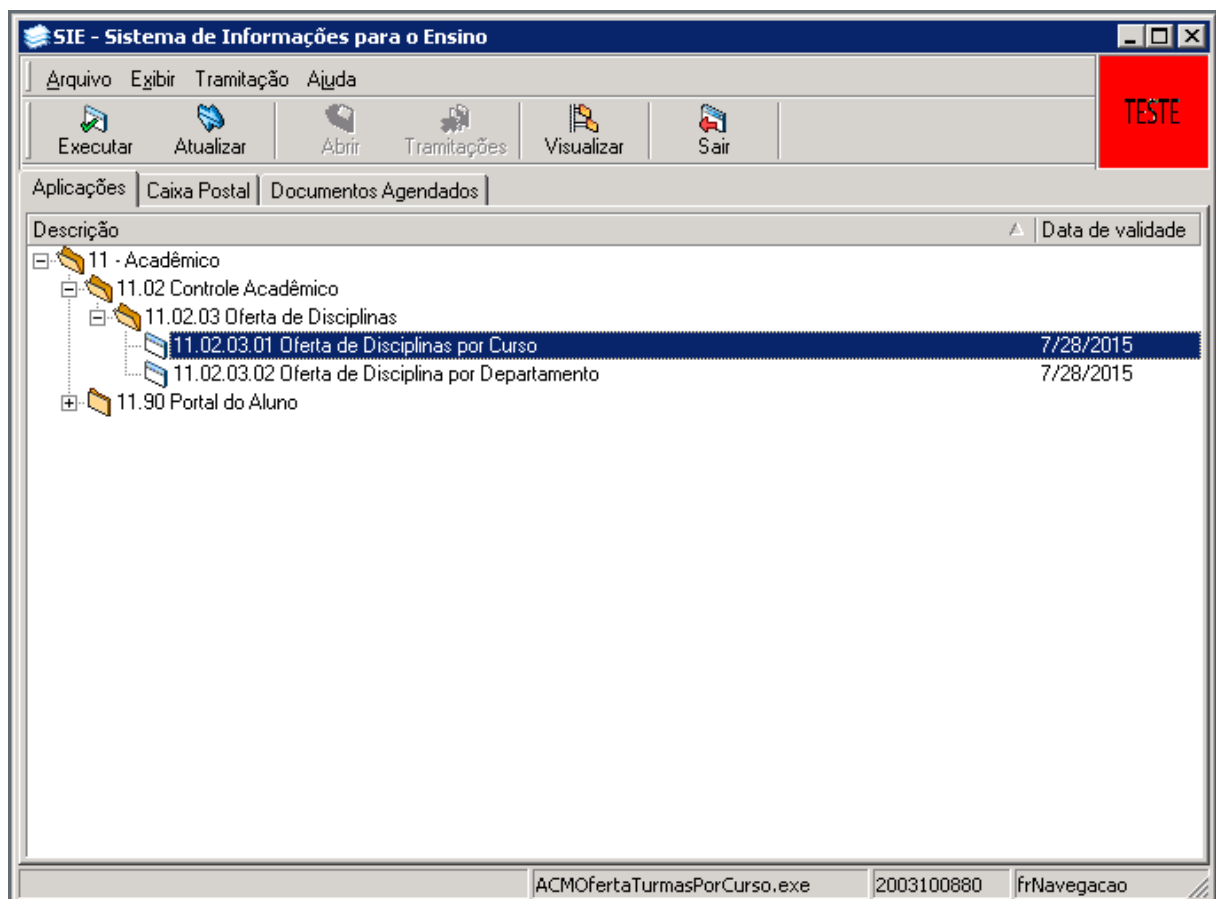


Figura 4 - Tela inicial do SIE para a função de Oferta de Disciplinas

Logo em seguida, ao clicar 02 vezes na opção desejada, uma nova janela surge como na Figura 5, solicitando a seguir, que o código do Curso seja digitado.

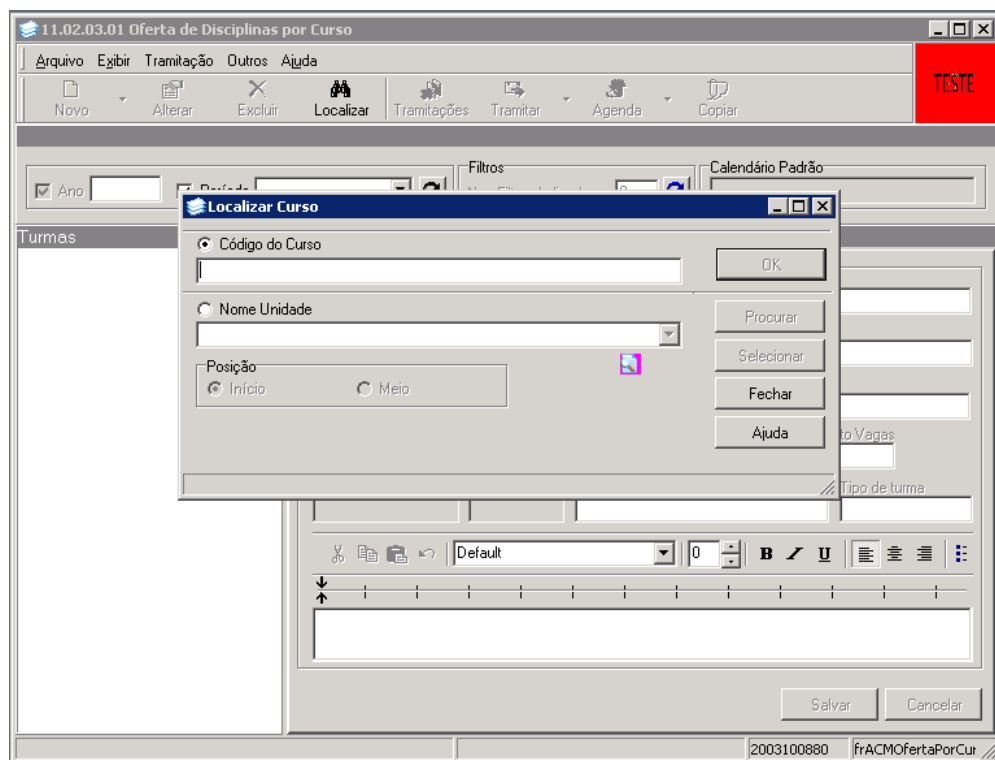


Figura 5 - Oferta de Disciplinas por Curso.

Após digitar o código pedido, uma nova tela é mostrada (Figura 6), onde é possível ver a listagem de disciplinas ofertadas no Curso em questão, em um dado ano e semestre (exibindo como padrão o ano e o semestre corrente).

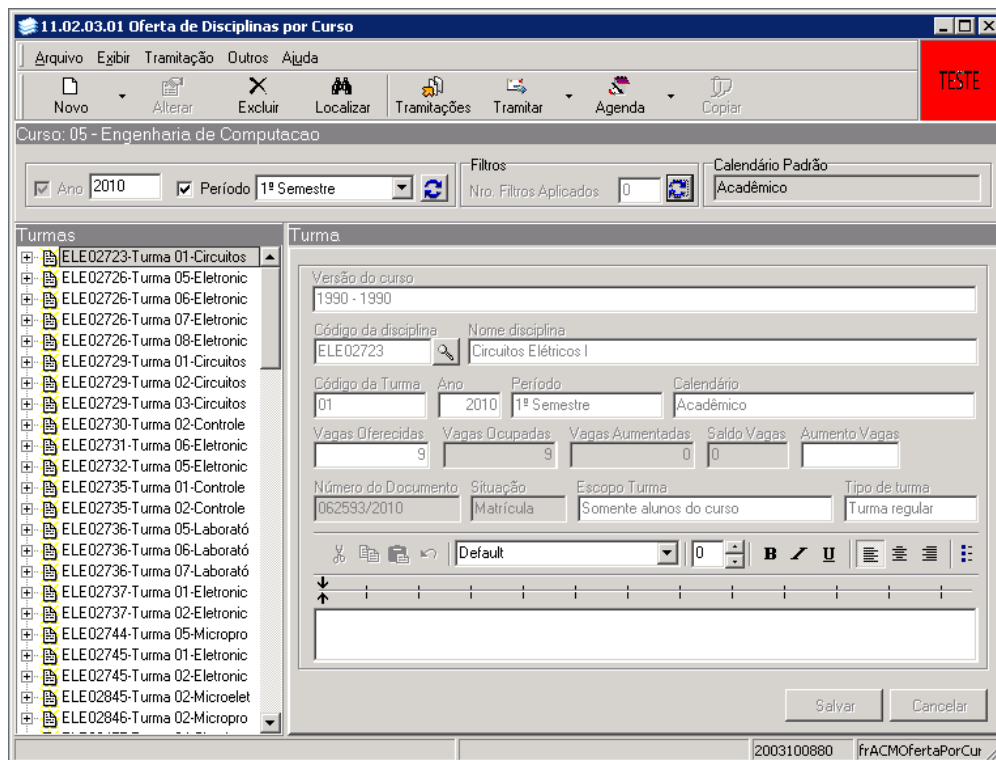


Figura 6 - Listagem de disciplinas do Curso de Engenharia de Computação, usado neste exemplo.

Para efetuar uma nova oferta, é preciso clicar na opção “Novo” na barra superior, como ilustrado na Figura 7 e selecionar alternativa “Turma”.

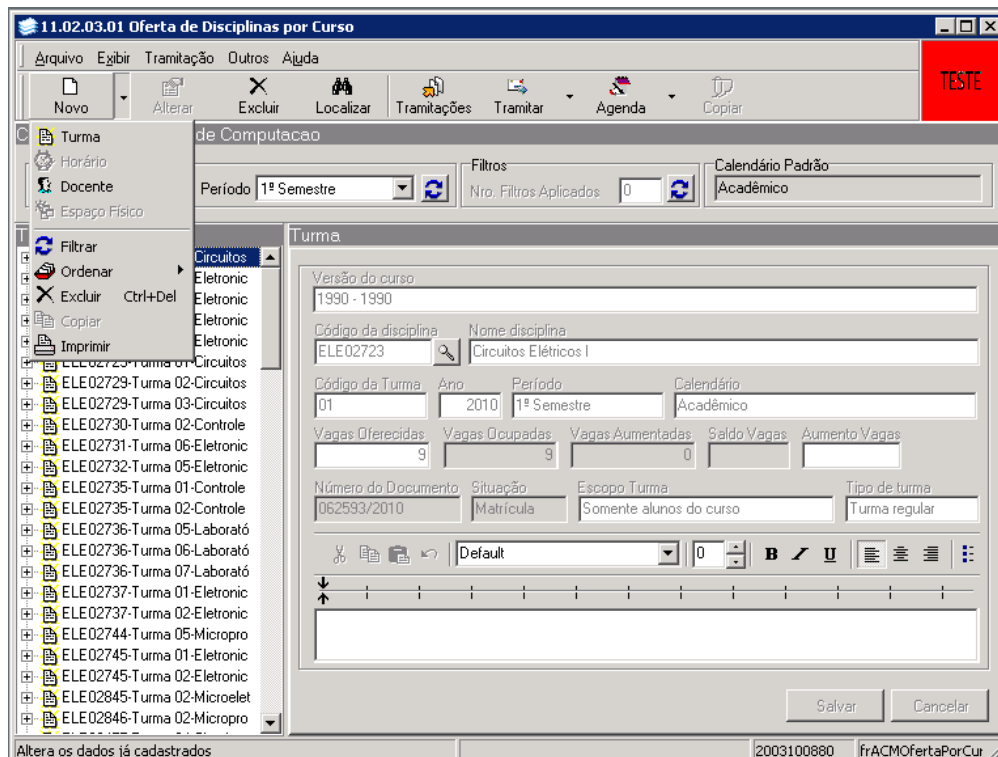


Figura 7 - Criar nova oferta de disciplina.

Aparecerá então o mesmo formulário anterior, porém, com seus campos vazios, aguardando que novos dados sejam inseridos, como mostrado na Figura 8.

Figura 8 - Formulário para uma nova oferta de disciplina.

É preciso então escolher a disciplina para a qual a turma será ofertada, clicando no botão de busca, simbolizado pela “lupa” na figura anterior. Com isso, surgirá mais uma janela, a da Figura 9.

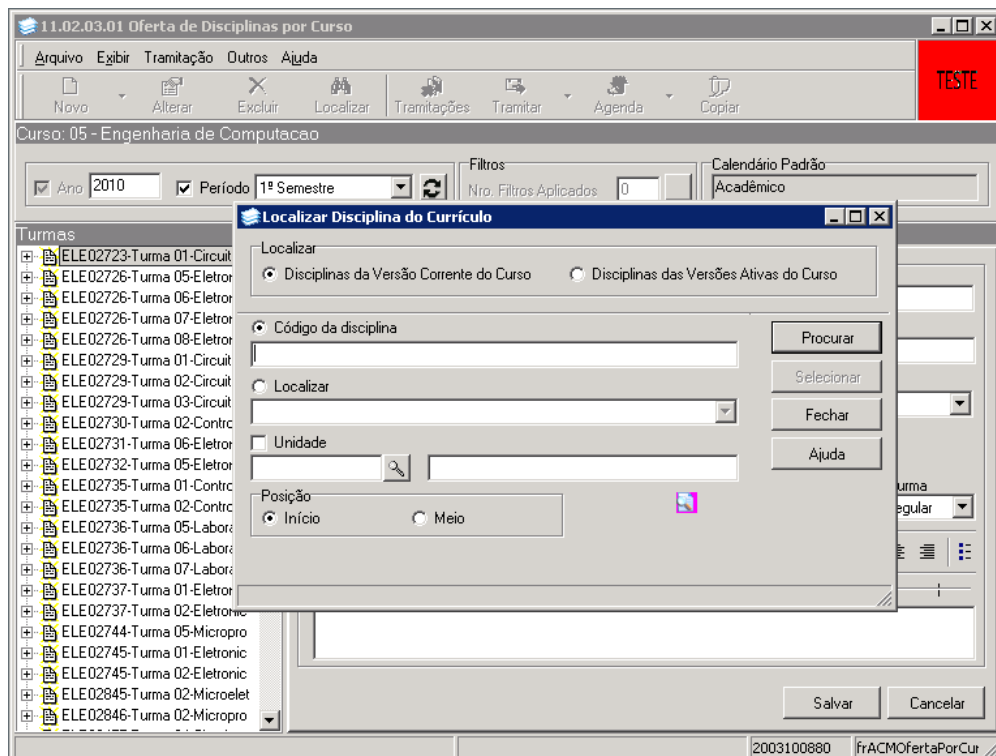


Figura 9 - Tela de escolha da disciplina para a qual a turma será ofertada.

Clicando no “Procurar”, uma lista de disciplinas é exibida, como na Figura 10.

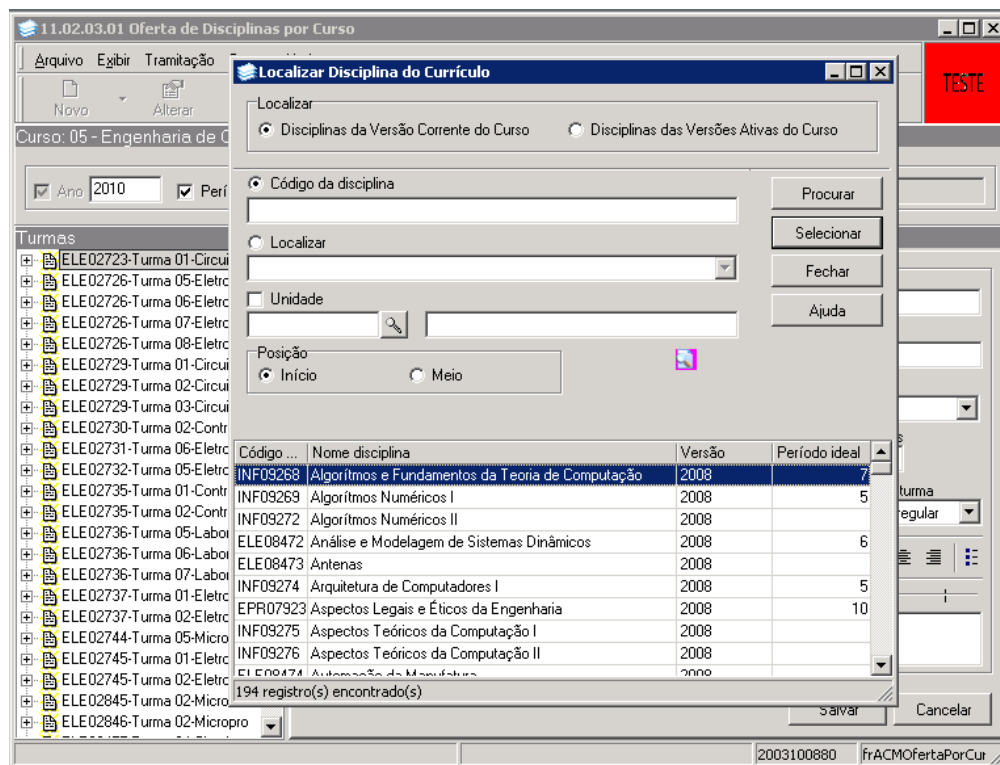


Figura 10 - Tela de busca da disciplina, com uma lista de disciplinas exibida.

Dessa forma, seleciona-se a disciplina (no exemplo foi Algoritmos e Fundamentos da Teoria de Computação) e, automaticamente, o formulário é preenchido com informações já contidas no banco de dados, exemplificado na Figura 11.

Figura 11 - Após seleção da disciplina o formulário é preenchido.

A partir de então, deve-se preencher os campos: código de turma, vagas oferecidas e confirmar a oferta feita clicando no botão “Salvar”, como ilustrado na Figura 12.

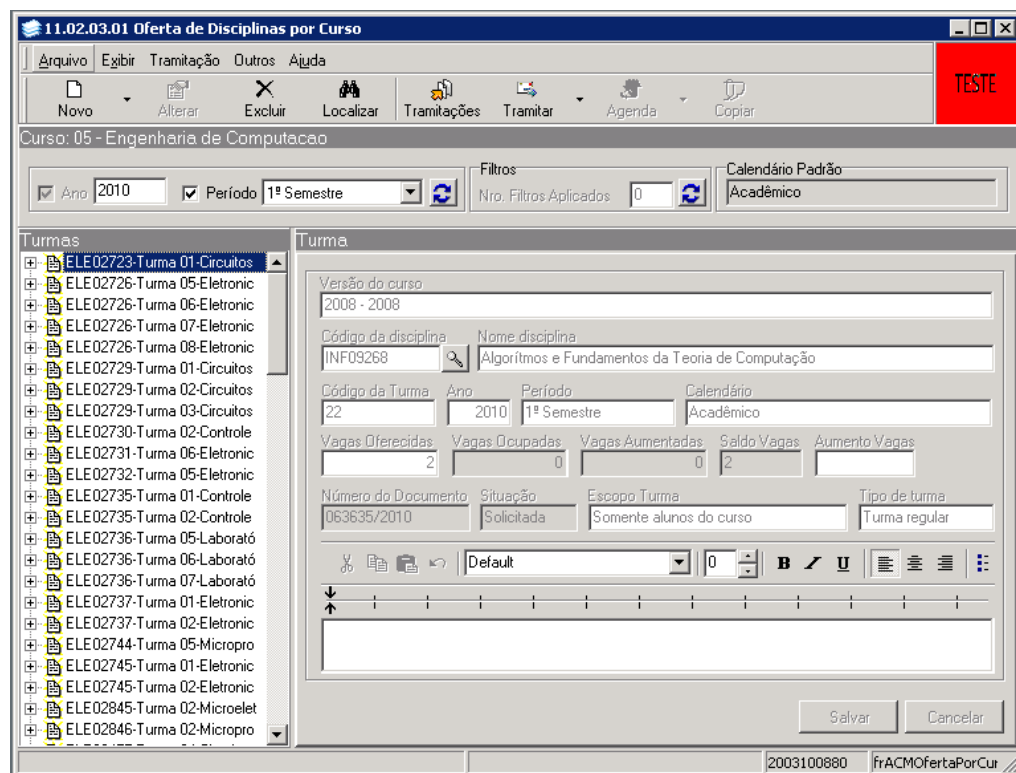


Figura 12 – Oferta de turma feita, para o Curso de Engenharia de Computação, na disciplina Algoritmos e Fundamentos da Teoria de Computação, com código da turma 22 e 02 vagas oferecidas.

Com as telas acima se pode constatar o grande número de decisões a ser tomada pelo operador, um número excessivo de cliques (mais de 10) para chegar ao fim da operação desejada, campos numéricos podendo ser preenchidos com letras e muitas buscas (disciplinas, Departamentos, Cursos, entre outros) para chegar ao ponto que se deseja.

OFERTA DE DISCIPLINAS POR DEPARTAMENTO

Esta aplicação faz a montagem da grade de docentes e dos espaços físicos destinados às turmas solicitadas pelos cursos [9], liberando as turmas para matrícula e lançamento de notas. Não é concedida ao Departamento a permissão de criar novas ofertas, lhes é oferecido a opção de inserir alguns atributos, como por exemplo, docentes e espaço físico. Porém, o Departamento pode editar uma oferta realizada pelo Colegiado de um Curso pertencente a ele. Segue um exemplo:

Após entrar no sistema, escolhe a opção “Oferta de Disciplinas por Departamento”, clicando 02 vezes no link, como na Figura 13.

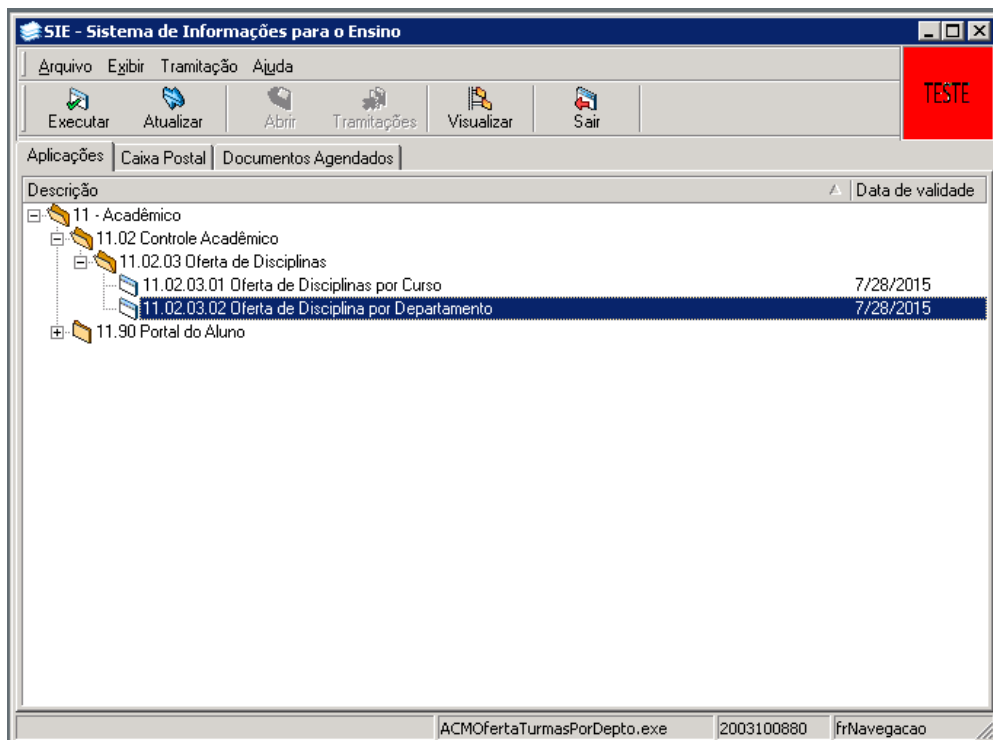


Figura 13 – Tela inicial do módulo de Oferta de Disciplinas.

A partir daí, surge a primeira janela com um formulário em branco, Figura 14, o que sugere que o botão localizar no menu superior deva ser clicado.

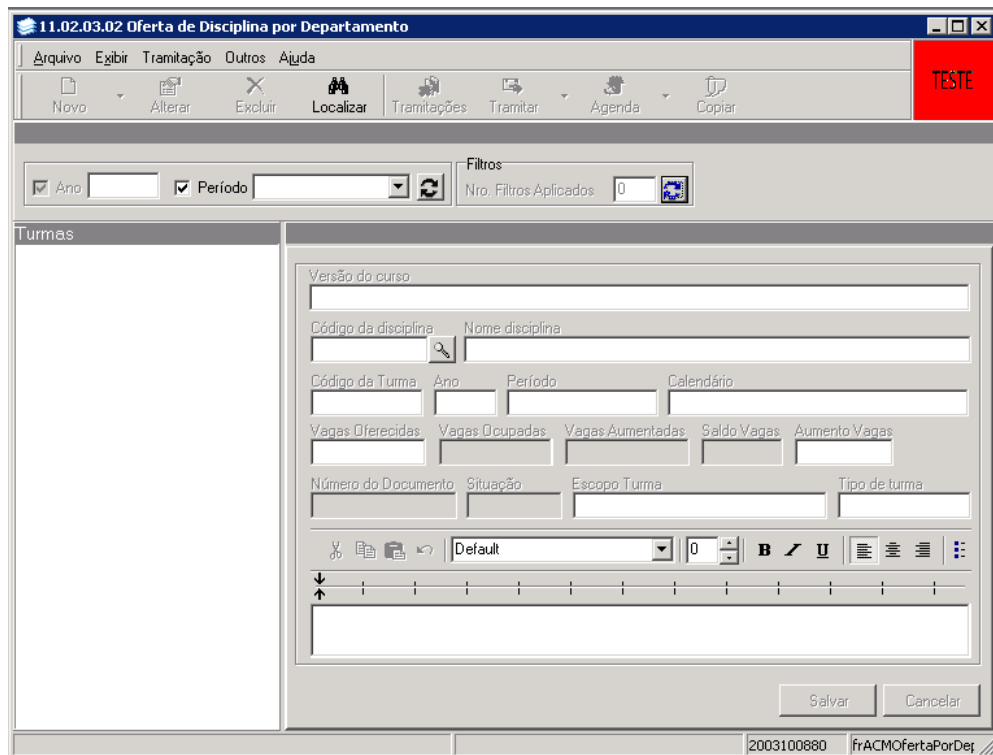


Figura 14 – Funcionalidade de Oferta de Disciplinas por Departamento.

Ao clicar em “Localizar”, mais uma janela é aberta, ilustrada na Figura 15, onde se tem como opções de filtro o “Código Estruturado”, o “Nome da Unidade” e a “Sigla” do Departamento. Feito isso clica-se no botão “OK”.

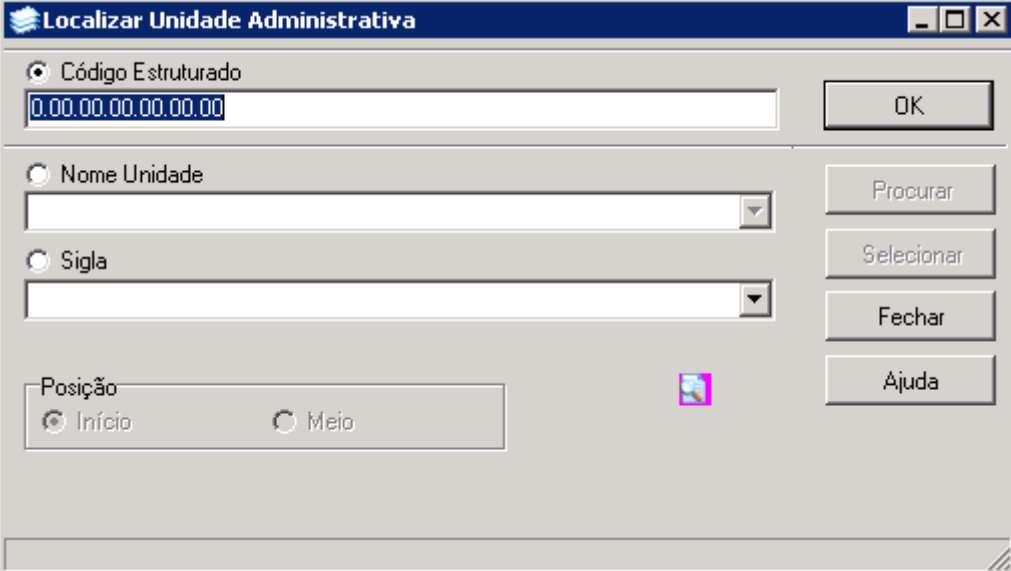
A imagem mostra uma janela de software intitulada "Localizar Unidade Administrativa". No topo, há uma barra de título com o nome da janela e ícones de minimizar, maximizar e fechar. Abaixo, há três opções de filtro: "Código Estruturado" (selecionado com um botão de rádio), "Nome Unidade" e "Sigla". Cada opção tem um campo de entrada correspondente. O campo "Código Estruturado" contém o texto "0.00.00.00.00.00.00". À direita dos campos, há uma coluna de botões: "OK", "Procurar", "Selecionar", "Fechar" e "Ajuda". Na parte inferior esquerda, há uma seção "Posição" com dois botões de rádio: "Início" (selecionado) e "Meio".

Figura 15 – Formulário de seleção de Departamento.

Selecionado o filtro e confirmado essa seleção, uma listagem de unidades será exibida logo abaixo, como na Figura 16, em outra janela.

Localizar Unidade Administrativa

☐ Código Estruturado
 0.00.00.00.00.00.00

☐ Nome Unidade

☒ Sigla
 inf

Posição
☒ Início ☐ Meio

OK
 Procurar
 Selecionar
 Fechar
 Ajuda

Sigla	Código Estruturado	Nome Unidade
INF	1.10.06.00.00.00.00	Departamento de Informatica - CT

1 registro(s) encontrado(s)

Figura 16 – Formulário de busca, preenchido para filtrar o Departamento cuja sigla é “INF”.

Então, no exemplo, foi escolhido o Departamento de Informática. O que, após clicar 02 vezes, abre uma nova janela, mostrada na Figura 17, com um formulário preenchido com os dados provenientes do banco de dados.

Figura 17 – Disciplina de Algoritmos Numéricos 1, do Departamento de Informática.

É possível perceber na Figura 18, que a opção de novas turmas está desabilitada, o que demonstra que o Departamento não tem permissão para criar uma nova turma.

Figura 18 – Mostra que os departamentos não têm poder de criar novas turmas.

Porém, vê-se que a opção “Alterar” está habilitada e, depois de clicada, como na Figura 19, praticamente todos os campos se tornam editáveis, o que torna demonstra que os departamentos têm essa liberdade.

Figura 19 – Após clicar em “alterar” quase todos os campos são habilitados para edição.

Como na funcionalidade de Oferta de Disciplinas por Curso, se vê que diversas janelas são abertas confundindo o usuário, além dos problemas mencionados, como a falta de uma validação de dados, evidenciando uma vulnerabilidade do sistema, comprometendo a integridade dos dados, e da quantidade de decisões necessárias para se obter um resultado.

ALTERAÇÃO DA PROPRIEDADE ESCOPO DA TURMA

Esta aplicação permite a troca de escopo [9] e [10], isto é, alterar a permissão de matrícula em determinada turma, tomando por base que podem existir 05 (cinco) escopos diferentes para uma turma:

Escopo tipo 1 - Apenas os alunos do Curso responsável pela oferta da turma podem nela se matricular.

Escopo tipo 2 - Apenas os alunos cuja disciplina da turma faça parte do currículo do seu Curso podem se matricular.

Escopo tipo 3 - Qualquer aluno da instituição pode se matricular (são as disciplinas de livre escolha do aluno, disciplinas eletivas, que devem seguir as definições da resolução nº 57/2000 do CEPE/UFES⁷).

Escopo tipo 4 - Podem se matricular apenas os alunos do Curso responsável pela solicitação da turma e os alunos das ênfases e habilitações obedecendo a hierarquia de cursos no SIE, pois os cursos devem estar relacionados entre si na estrutura organizacional da instituição de ensino.

Escopo tipo 5 - Turmas em resolução 05 - apenas os alunos do Curso responsável pela solicitação da turma podem se matricular, desde que tenham sido reprovados por nota no semestre imediatamente anterior (essas turmas não consideram choque de horário na hora de matrícula).

Para facilitar o entendimento dessa função, a Figura 20 mostra a interface exibida por ela.

Figura 20 – Tela de alteração do escopo de uma determinada turma.

Para que seja feita essa alteração, é preciso saber, além do escopo atual e o escopo desejado, e informar a unidade à qual a turma pertence, clicando no ícone de busca, representado pela lupa, que exibirá uma tela semelhante à Figura 15, o nome da disciplina ou

⁷ http://www.daocs.ufes.br/corpo.asp?pagina=resolucoes/resolucao_n57_00

o código da turma, também buscando na listagem que será exibida (aplicando-se os devidos filtros) e, para confirmar a alteração, clicar em “alterar”, na barra superior desta nova janela.

A escassez de figuras simbolizando as funcionalidades e facilitando a acessibilidade, é um ponto a se evidenciar, bem como a falta de clareza nas informações exibidas, necessárias para que a mudança da propriedade de escopo seja efetivada.

2.4 Algumas deficiências observadas

Análises feitas no SIE evidenciaram algumas vulnerabilidades do sistema e os gargalos de infra-estrutura, que estão detalhadas a seguir.

PROTOCOLO UTILIZADO

O protocolo usado, Component Object Model (COM [11]), que precedeu o Distributed Component Object Model (DCOM [11]), é proprietário⁸ e também possui vulnerabilidades como: o processamento de mensagens que, caso o roteamento dessas mensagens entre processos e threads não seja feito com regularidade, poderá causar o travamento do sistema; a contagem de referências que, caso dois ou mais objetos sejam referenciados circularmente, pode causar problemas. Além disso, por ter como base as “Dynamic-link Libraries” (DLLs), traz consigo todos os pontos negativos nelas contidos, tais como:

- Falhas de versionamento e falha no acesso de diferentes versões: Quando um programa é instalado, por exemplo, e esse necessita de uma versão anterior àquela contida no *system directory* (repositor padrão dessas bibliotecas no sistema operacional), ele não funcionará corretamente se essa DLL for incompatível àquela necessária. Ou ainda, ao remover um *software*, pode-se remover também as DLLs usadas por ele, mas acessadas por algum outro programa, o que poderá resultar em um mal funcionamento do último;
- Erro de registro de DLL: quando um programa registra uma biblioteca, necessitando na verdade de outra.
- Compartilhamento de módulos de memória por diferentes aplicações: a memória pode instanciar módulos de um mesmo diretório, fazendo com que

⁸ Demanda a compra de uma licença específica para sua utilização.

outras aplicações não possam utilizar essa biblioteca enquanto essa instancia existir.

Essas deficiências que deram origem à expressão DLL-hell [12]. Tais métodos de comunicação foram substituídos na plataforma .NET pela API .NET Remoting [13].

ARQUITETURA DO SISTEMA

Existem inúmeras desvantagens na arquitetura desktop, das quais as mais expressivas são: os problemas de deployment (implantação), a massiva comunicação entre o cliente e o servidor de aplicação, a difícil interação com clientes remotos, a necessidade de implementar a segurança no banco de dados (resultando em uma exposição da base de dados a acessos e a ataques de inúmeras máquinas, até mesmo de redes, além da difícil administração das diversas políticas de segurança atreladas ao mecanismo de armazenamento da aplicação) e a criticidade de sua distribuição (dependendo de periféricos, sistemas operacionais e drivers), no que cabe aos desenvolvedores.

A Figura 21, a seguir, mostra claramente o funcionamento do SIE, através diversos clientes, onde o software está instalado, ou seja, onde se encontra a lógica de negócios e a apresentação da aplicação, e se comunica com o(s) servidor(es) de banco de dados diretamente, por meio de objetos (ODBC, OLEDB, JDBC, entre outros) criados pelas DLLs do programa. Mas todo o processamento se encontra no lado do cliente, onde as máquinas trabalham a informação e exibem-na em tela para o usuário.

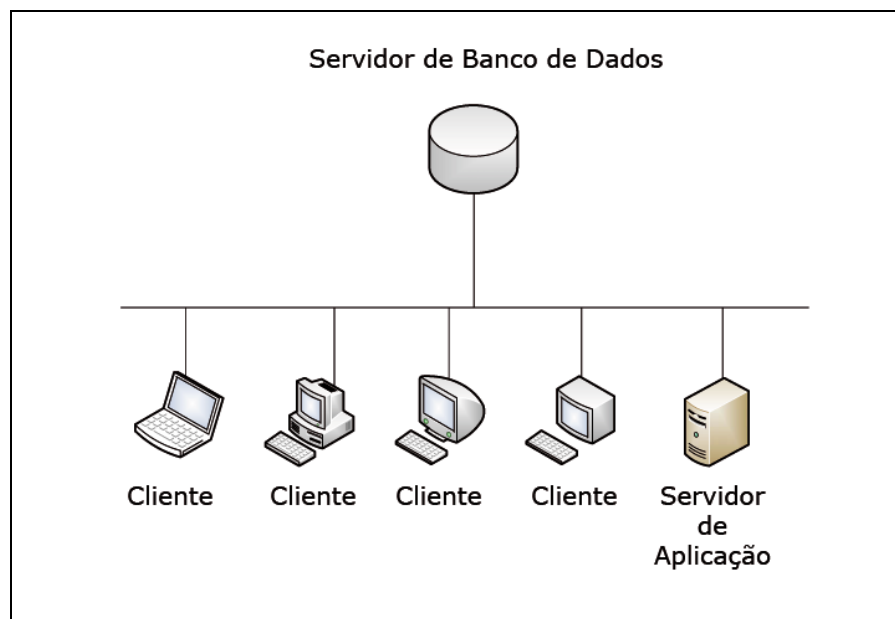


Figura 21 - Arquitetura do SIE.

Na Figura 22 abaixo, vê-se o fluxo de informações da arquitetura cliente-servidor usada pelo SIE.

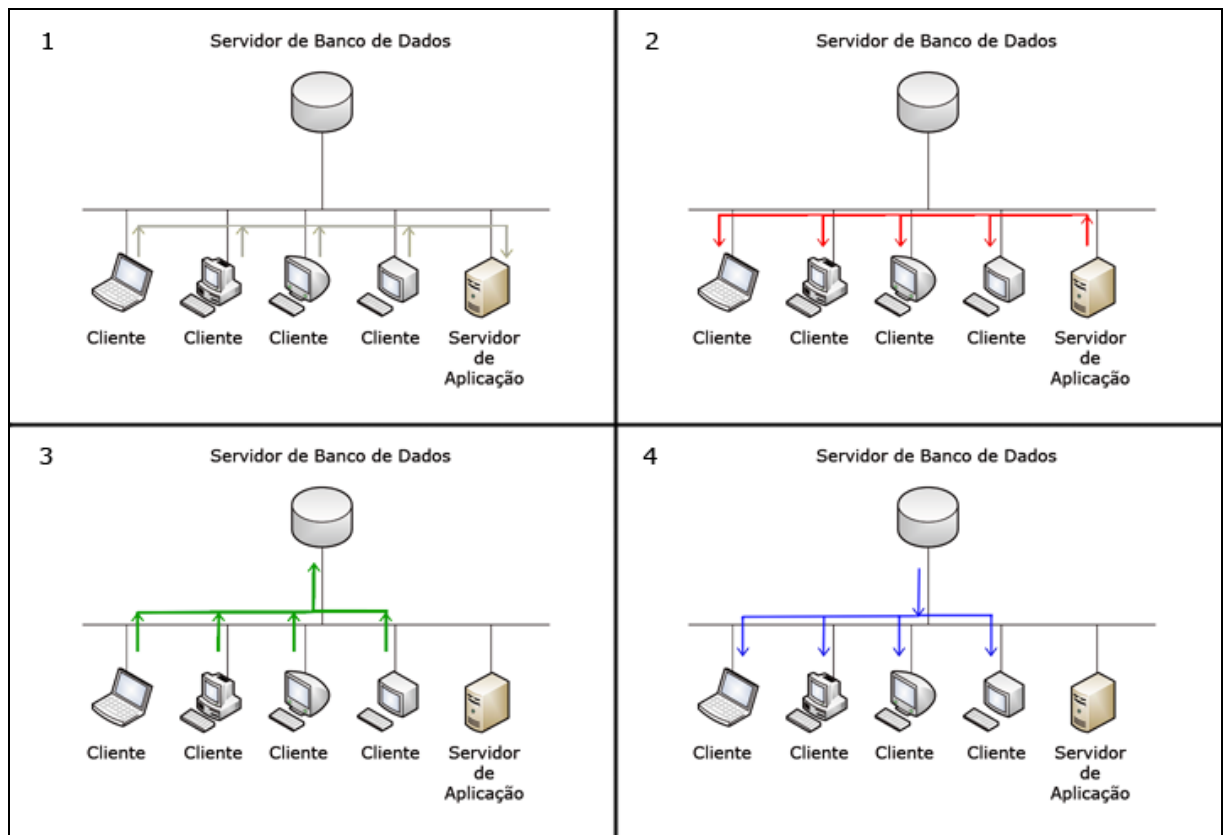


Figura 22 - Fluxo de dados na arquitetura cliente-servidor do SIE.

Todo cliente, ao abrir um módulo do SIE, comunica-se com o servidor de aplicação (parte 1 da Figura 22), que, em sua resposta, retorna o módulo solicitado a cada cliente (parte 2 da Figura 22). Feito isso, o cliente acessam o servidor de banco de dados (parte 3 da Figura 22), diretamente, em busca das informações que serão exibidas. Então, o servidor de banco envia esses dados aos usuários (parte 4 da Figura 22). O que demonstra a constante comunicação entre o cliente e o servidor de aplicação e a necessidade de implementação de políticas de segurança na máquina que hospeda o banco de dados.

Já a Figura 23, exemplifica o fluxo de informações de uma arquitetura cliente-servidor padrão.

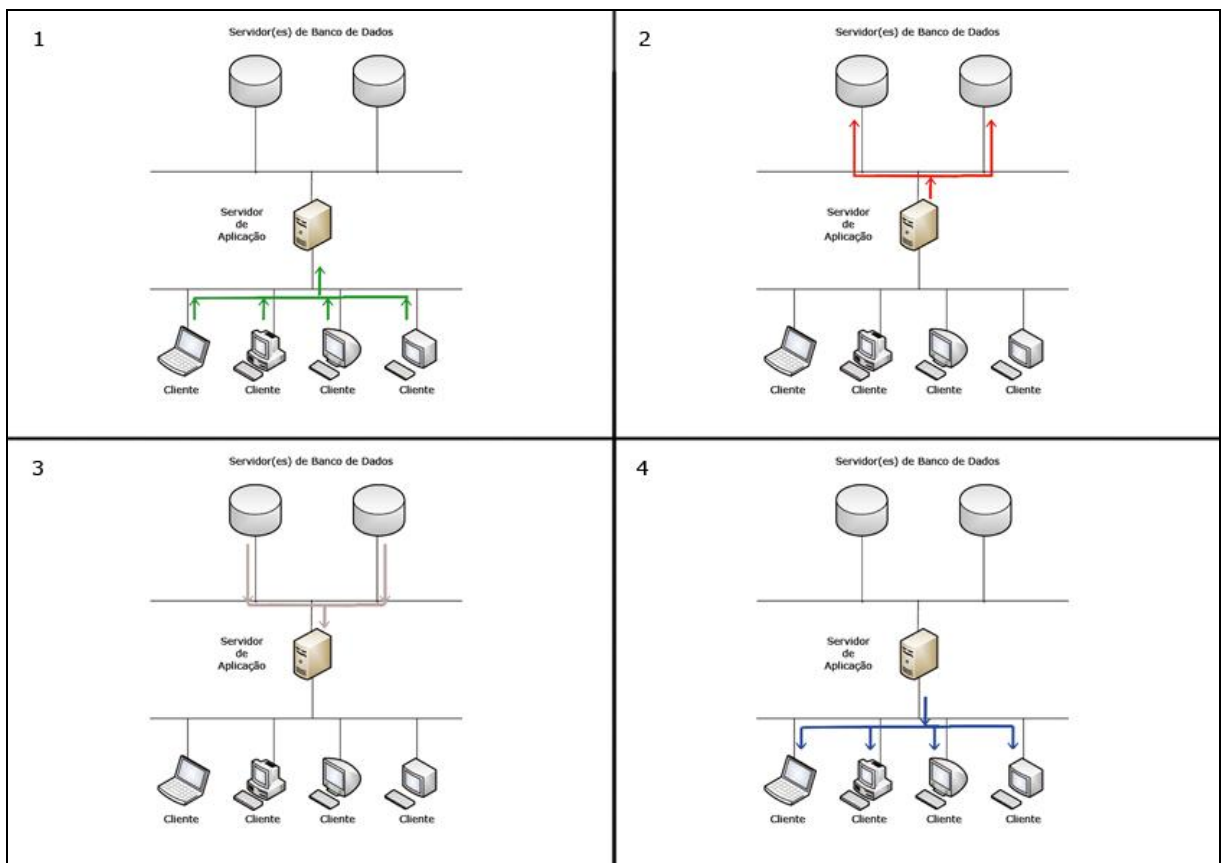


Figura 23 - Fluxo de dados na arquitetura cliente-servidor usual.

Solicitações são feitas pelos clientes ao servidor de aplicação (parte 1 da Figura 23). Estes processam a solicitação e repassam ao(s) servidor(es) de banco de dados (parte 2 da Figura 23), que retornam as informações solicitadas ao servidor de aplicação (parte 3 da Figura 23). Assim, as informações, tratadas, validadas e processadas são enviadas para os navegadores dos clientes (parte 4 da Figura 23), onde serão visualizadas pelos operadores. Nesse caso a segurança é implementada no servidor de aplicação, evitando a exposição do banco.

No tocante à dificuldade de implantação, que envolvem atividades como a instalação e a ativação, o lançamento de versões, a desativação, a atualização ou adaptação, o rastreamento de versões, a desinstalação até a retirada do produto. Sabe-se que, por precisar ser instalada em diversos clientes, no caso desta universidade em milhares, fazer essa instalação, ou mesmo de um patch de segurança, lançar uma nova versão, fazer uma adaptação ou atualização (embutida ou não) ao software e desinstalar, torna-se uma tarefa extremamente árdua e de difícil gerência, pois é preciso acessar cada máquina para efetivamente realizar todas essas tarefas e ter uma política de controle de alta qualidade.

Uma solução dos problemas citados acima, e usada no sistema atual é a presença do servidor de aplicação, não como uma camada, mas como uma máquina compartilhada pelas demais, para que todas as aplicações, quando “chamadas” são requisitadas a esse servidor, fazendo com que as instalações estejam com a mesma versão, mas, em contrapartida causa um overhead nele, pois cada requisição feita pelos clientes demanda um tempo excessivo, sobrecarregando a máquina.

A partir dos argumentos acima é possível entender os gargalos da arquitetura usada no sistema atual.

INTERFACE COM USUÁRIOS

Quando temos como alvo os usuários, outros problemas podem ser ressaltados, como, por exemplo, a interface usada. O elo entre a aplicação e o usuário é a tela de exibição da ferramenta, que, portanto, deve ser muito bem planejada e arquitetada para simplificar a usabilidade e aumentar a aceitação do público alvo. Nesse aspecto as seguintes lacunas foram vistas: grande número de decisões para chegar onde deseja, mencionado na seção anterior; grande número de janelas exibidas, confundindo o usuário, também citado anteriormente, exibindo assim, informações desnecessárias e irrelevantes; longo tempo de espera para acessar algumas funcionalidades; falta de metáforas, que faria com que o sistema usasse a mesma língua do usuário; falta de comunicação visual (imagens e gráficos); o sistema não possui nenhum manual ou menu de ajuda; os rótulos dos campos não são intuitivos; dentre outras.

Um engenheiro de software deve sempre objetivar a redução do esforço cognitivo, isso facilita a compreensão do usuário.

VALIDAÇÃO DE DADOS

O problema mais crítico e determinante observado na aplicação existente é a falta de verificações de segurança, validação de requisitos e de informações fornecidas pelos operadores, não prevenindo a ocorrência de possíveis conflitos, com relação à alocação de recursos humanos e materiais, não garantindo a confiabilidade da informação, permitindo que o usuário possa burlar restrições e fazendo com que um mesmo recurso possa ser alocado concorrentemente a outro.

Isso causaria transtornos durante a etapa de matrícula e inúmeros problemas para os colegiados, departamentos acadêmicos e, também, para os alunos, como por exemplo,

diferentes professores serem alocados para ministrar a mesma aula (salvo em casos especiais), ou várias turmas ocuparem o mesmo horário de um único docente sem que o operador do sistema saiba, ou ainda duas disciplinas de serem lecionadas em um mesmo espaço físico, dentre outros casos mais críticos. Vale salientar ainda que a ausência de ferramentas gráficas para facilitar a identificação de conflitos é mais um ponto negativo a ser trabalhado.

O trabalho desenvolvido, objeto desta monografia, aborda o processo de “Oferta de Disciplinas” do módulo Acadêmico, na seção Controle Acadêmico, que será detalhado a seguir.

3 UM MÓDULO DE OFERTA DE DISCIPLINAS WEB PARA O SIE

O sistema implementado visa a elaborar a oferta de disciplinas da UFES e dar agilidade ao processo, proporcionando ao usuário uma melhor usabilidade da ferramenta, em uma plataforma simples e rápida, ou seja, na plataforma WEB.

3.1 Concepção do projeto

Sabe-se que atualmente é uma tarefa trabalhosa efetuar a oferta, já que é necessário avaliar o histórico (usualmente de períodos anteriores), as grades curriculares, alternatividade de turnos (caso seja uma disciplina ofertada anualmente ou semestralmente), todos os pré-requisitos (matérias, créditos, período vencido e outros), horários, docentes aptos e disponíveis, salas de aula com estrutura para receber a referida disciplina e diversas especificidades inerentes de cada Curso ou Departamento.

As turmas a serem ofertadas são inicialmente inseridas na aplicação pela Coordenação de Curso, onde este irá sugerir ao Departamento a alocação de horário. Após a inserção dessas informações a aplicação envia esses dados aos respectivos departamentos responsáveis por tais disciplinas para definição de docente e espaço-físico e posterior aprovação e liberação à Matrícula.

Seguindo as mesmas entradas explicitadas no Capítulo 2 (Grade Curricular, Histórico, Estatísticas, Docentes e Espaço-Físico) e as restrições existentes expostas, após o envio dessas informações, o sistema valida cada entrada e verifica quaisquer possíveis conflitos, acusando-o visualmente, caso exista, já que a consistência da informação é feita dessa forma, mostrando ao operador os erros ou incompatibilidade de dados, até mesmo caso alguma das restrições existentes sejam burladas ou sobrescritas, inadvertidamente, assegurando que o que for fornecido esteja em coerência e que respeite os regulamentos da instituição. Além disso, sugere dados adicionais e possíveis mudanças, como: as disciplinas que cada professor está apto a ministrar, a periodização da turma, a listagem de professores, a listagem de horários comumente utilizados em outros períodos para a matéria em questão, e outros detalhes. Caso tudo prossiga corretamente o resultado é uma listagem de novas ofertas realizadas.

A arquitetura utilizada no novo sistema está ilustrada na Figura 24, bem como seu fluxo de informações. As deficiências encontradas na arquitetura anterior foram sanadas ou, ao menos, reduzidas, já que os problemas de implantação são eliminados, uma vez que a

aplicação estará instalada em uma única máquina (o servidor WEB); a interação entre cliente e servidor é simples, bastando que o cliente possua um navegador para que esteja apto a acessar o sistema; a comunicação é limitada a dados “leves”, diminuindo o overhead; a segurança está implementada no servidor de aplicação, tirando o foco de ataques do servidor de banco de dados; e a distribuição é simples, independente de periféricos.

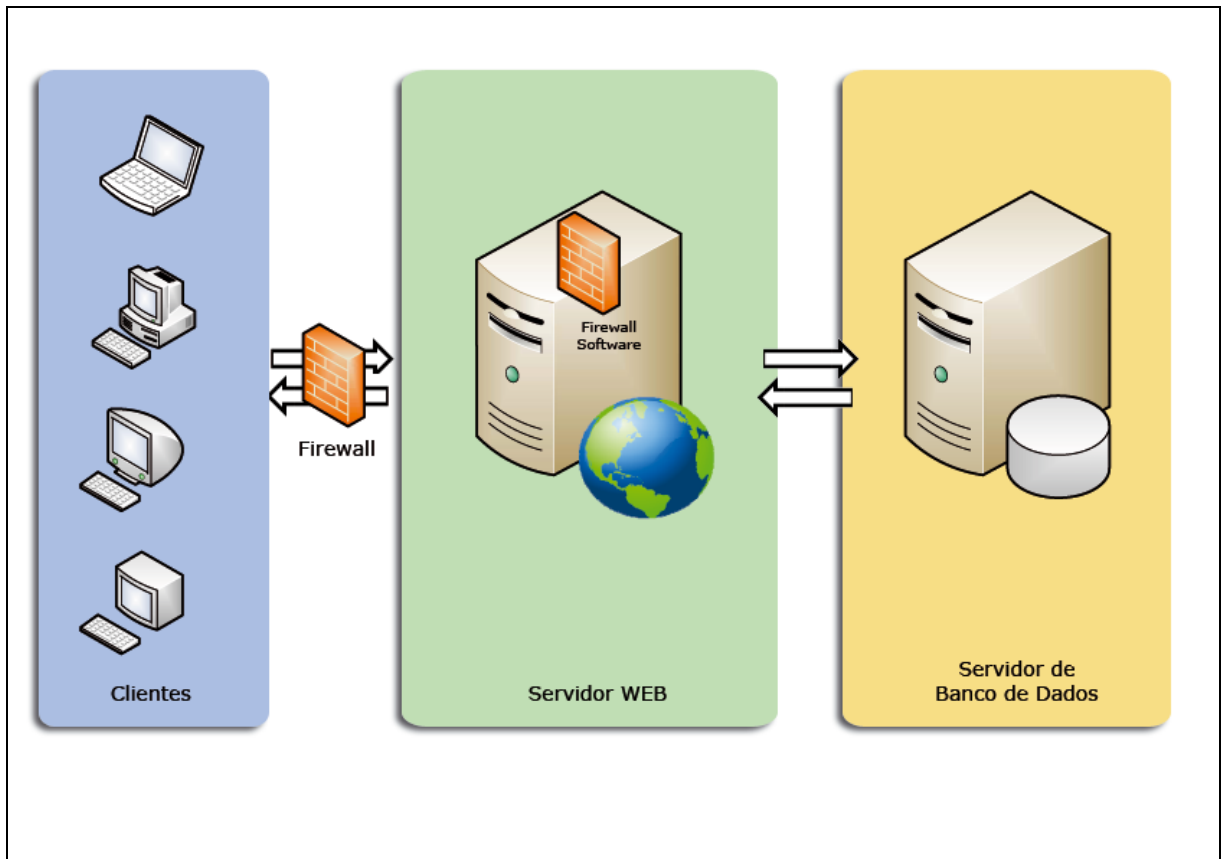


Figura 24 - Arquitetura cliente-servidor usada pelo novo sistema.

3.2 Agentes envolvidos

Os principais agentes envolvidos no procedimento de oferta de disciplina são: o Chefe de Departamento, o Coordenador de Curso, a Pró-Reitoria de Graduação e o aluno, quando as turmas ficarão disponíveis para Solicitação de Matrícula via Portal do Aluno. O Calendário Acadêmico da UFES regulamenta as datas de execução dessas etapas.

O novo módulo implementa diversas novas restrições, tais como:

- Mais de um professor não pode ministrar uma mesma disciplina, exceto casos especiais, como quando a mesma é dividida em aulas teóricas e práticas, entre outros.

- Mais de uma turma alocada em um mesmo espaço-físico, salvaguardando casos de serem as mesmas disciplinas ofertadas para cursos ou versões de currículos diferentes, e algumas outras exceções.
- O mesmo docente não pode ser alocado em um mesmo horário para ministrar diferentes disciplinas, sem que o operador do sistema esteja ciente.
- Disciplinas de um mesmo período não podem estar em horários concorrentes.
- Professores não podem ser alocados em local já disponibilizado para outro docente ou outra disciplina, em um mesmo horário, exceto em casos especiais.

Não obstante a presença dessas restrições já implementadas no sistema, vale mencionar que, pela estrutura usada no desenvolvimento, caso haja necessidade de incluir novas restrições ou de alterar as existentes, basta fazer algumas adequações extremamente simples e de fácil configuração.

A seguir foram listadas as possíveis ações de cada agente do sistema, a saber:

- **AÇÃO DO COLEGIADO DO CURSO:**
 - Insere horário (em situação de “Solicitada” e “Ajuste”);
 - Insere disciplina das versões curriculares vigentes (em situação de “Solicitada”);
 - Insere espaço-físico (em situação de “Ajuste”);
 - Define código da turma (em situação de “Solicitada”);
 - Submete ao Departamento as informações citadas (em situação de “Solicitada”).
- **AÇÃO DO DEPARTAMENTO:**
 - Faz a análise do pedido das disciplinas/turmas pela coordenação de curso (em situação de “Análise”);
 - Ajusta código da turma (em situação de “Análise”);
 - Define docentes responsáveis (em situação de “Análise”);
 - Em caso de não concordância da solicitação encaminha a disciplina/turma para que o Colegiado do Curso faça as devidas adequações, como o ajuste de horário e de número de vagas (exclusividade do Colegiado de Curso).

Vale salientar que o Colegiado do Curso só pode alterar os dados de horário da turma e número de vagas quando a turma está em situação de “Solicitada” ou “Ajuste”. Já o Departamento só altera a quantidade de vagas quando a turma está em “Análise”.

3.3 Levantamento de requisitos

Segundo Sommerville e Sawyer, a Engenharia de Requisitos de Software é uma atividade que engloba as tarefas de descoberta, documentação e manutenção do conjunto de requisitos de um sistema de software [14]. Essa foi, portanto, a fase de maior relevância para o desenvolvimento e a implementação do novo módulo. Nela foram mapeados os gargalos a serem superados, definidas as funcionalidades e os requisitos exigidos para sua criação.

3.3.1 Requisitos funcionais

A partir de um estudo junto aos atores envolvidos e após análise das demandas, constatou-se que a nova aplicação deverá possuir todas as funcionalidades pré-estabelecidas, a saber:

- Autenticação de usuário;
- Controle de acesso (através de perfis de usuários);
- Exibição, alteração, exclusão e inclusão de dados;
- Tramitação de documentos;
- Cópia da oferta anterior de disciplinas.

Estas serão detalhadas a seguir.

AUTENTICAÇÃO DE USUÁRIO:

Tomando como ponto de partida que o requisito primordial é a segurança das informações, garantindo sua inviolabilidade e integridade, neste projeto foi idealizada, como primeira funcionalidade, a autenticação de usuários. Para isso a tela inicial exige 3 informações do usuário: sua credencial, a senha e o perfil desejado. Ao submeter esses dados, as informações são criptografadas e comparadas com as contidas no banco de dados. Somente após a validação, o usuário estará apto a entrar no sistema.

CONTROLE DE ACESSO:

Essa funcionalidade se presta a definir o perfil do usuário: se “Coordenador de Curso” ou “Chefe de Departamento”. Tal diferenciação é necessária em virtude de que há especificidades inerentes a cada tipo de usuário, em decorrência da função que exerce na instituição. Foi preciso criar, então, o caminho mais direto possível para atender cada

categoria de usuário, evitando acessos desnecessários a informações que não sejam de interesse, reduzindo o tempo de utilização do sistema.

Por outro lado, quando de interesse das partes, o sistema foi desenvolvido para permitir, também, a interação simultânea na oferta de disciplinas entre esses dois perfis de usuários, a despeito das especificidades de cada um.

EXIBIÇÃO, ALTERAÇÃO, EXCLUSÃO E INCLUSÃO DE DADOS:

Essas são as funcionalidades essenciais do sistema. Os usuários podem visualizar, alterar, excluir e incluir dados específicos, de acordo com seu perfil. Abaixo, estão as tabelas de permissão de manipulação de dados (Tabelas 1, 2 e 3) apresentam as referências atualmente utilizadas pelo SIE/UFES e por isso foram adotadas no projeto. Foi exatamente para cumprir o que está previsto nessas tabelas que se fez necessária a identificação do perfil do usuário, estabelecendo o que é permitido visualizar, alterar, inserir e excluir, segundo o perfil de cada um.

Enquanto as linhas das tabelas identificam o perfil do usuário, suas colunas indicam o que lhes é permitido alterar, dependendo da situação da turma, que é simbolizada pelo número contido em cada célula. A legenda após as tabelas explica que situação cada número representa.

Tabela 1- Permissões de alteração de dados de acordo com o perfil do usuário

Perfis \ O que alterar?	Horários	Vagas	Turma	Escopo	Docente
Colegiado	1, 2	1, 2	1, 2	1, 2, 3, 4, 5	0
Departamento	0	3, 4	3, 4	1, 2, 3, 4, 5	1, 2, 3, 4, 5

Tabela 2 - Permissões de inserção de dados de acordo com o perfil do usuário

Perfis \ O que inserir?	Horários	Vagas	Turma	Escopo	Docente
Colegiado	1, 2	1, 2	1, 2	1, 2, 3, 4, 5	0
Departamento	0	3, 4	3, 4	1, 2, 3, 4, 5	1, 2, 3, 4, 5

Tabela 3 - Permissões de exclusão de dados de acordo com o perfil do usuário

Perfis \ O que excluir?	Horários	Vagas	Turma	Escopo	Docente
Colegiado	1, 2	1, 2	1, 2	0	0
Departamento	0	0	0	0	1, 2, 3, 4, 5

Legenda de situação da turma:

0. Não há situação de turma em que este usuário pode alterar/inserir/excluir
1. Solicitada
2. Ajuste de horário
3. Análise
4. Matrícula
5. Lançamento

TRAMITAÇÃO DE DOCUMENTOS:

O novo módulo desenvolvido neste projeto precisou ainda ser capaz de gerar um “protocolo” a cada novo registro de informações de turmas/disciplinas realizado, mediante o qual se processa, monitora e controla a tramitação desse registro pelos setores competentes da instituição. Essa funcionalidade proporciona organização de processo, agilidade na difusão de informações vitais para a entidade, além de promover a impessoalidade e a imparcialidade na tomada de decisões. Vale ressaltar que ela é responsável pela integração de todos os módulos do SIE, uma vez que descentraliza todas as ações tomadas, fazendo com que cada atividade seja registrada e receba o aval dos setores interessados.

CÓPIA DA OFERTA ANTERIOR DE DISCIPLINAS:

Essa funcionalidade foi inserida no projeto para permitir que o usuário possa utilizar, em períodos futuros, uma oferta de turma/disciplina registrada anteriormente, sem a necessidade de refazer todo o procedimento, evitando o retrabalho. Dessa forma todas as telas geradas a partir da inserção e/ou alteração de turma/disciplina poderão ser salvas e devidamente armazenadas para resgate posterior, se necessário for.

Assim, foram definidas as funcionalidades desejáveis e idealizados os algoritmos capazes de operacionalizar tais funcionalidades.

3.3.2 Casos de uso simplificado

O diagrama de casos de uso tem o objetivo de ilustrar em um alto nível de abstração quais elementos externos interagem com que funcionalidades do sistema. Então, tem como finalidade, apresentar um “diagrama de contexto” que descreve a interação entre os atores externos e a maneira como eles utilizam a aplicação.

A Figura 25 ilustra o diagrama de casos de uso do Departamento, com as ações do Chefe do Departamento.

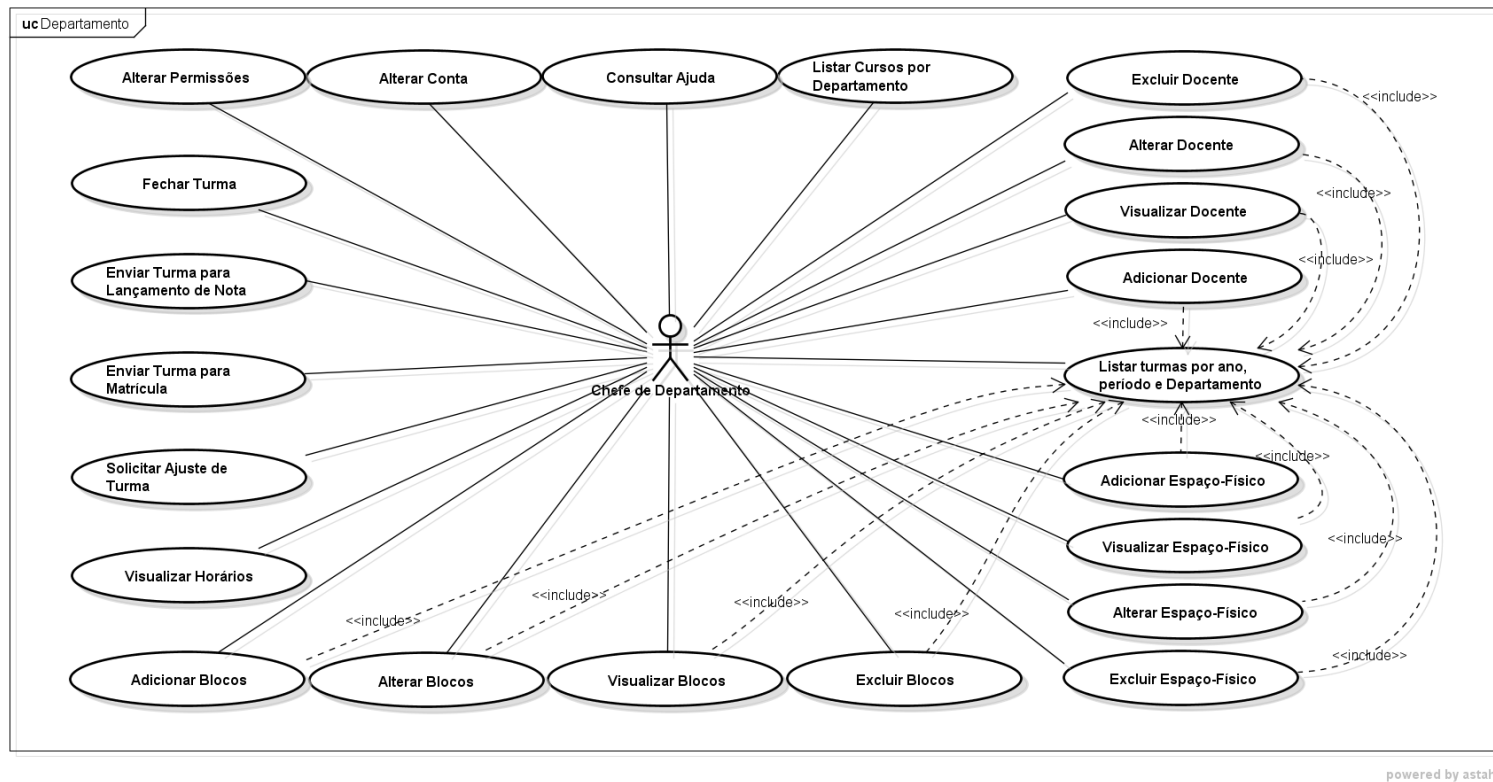
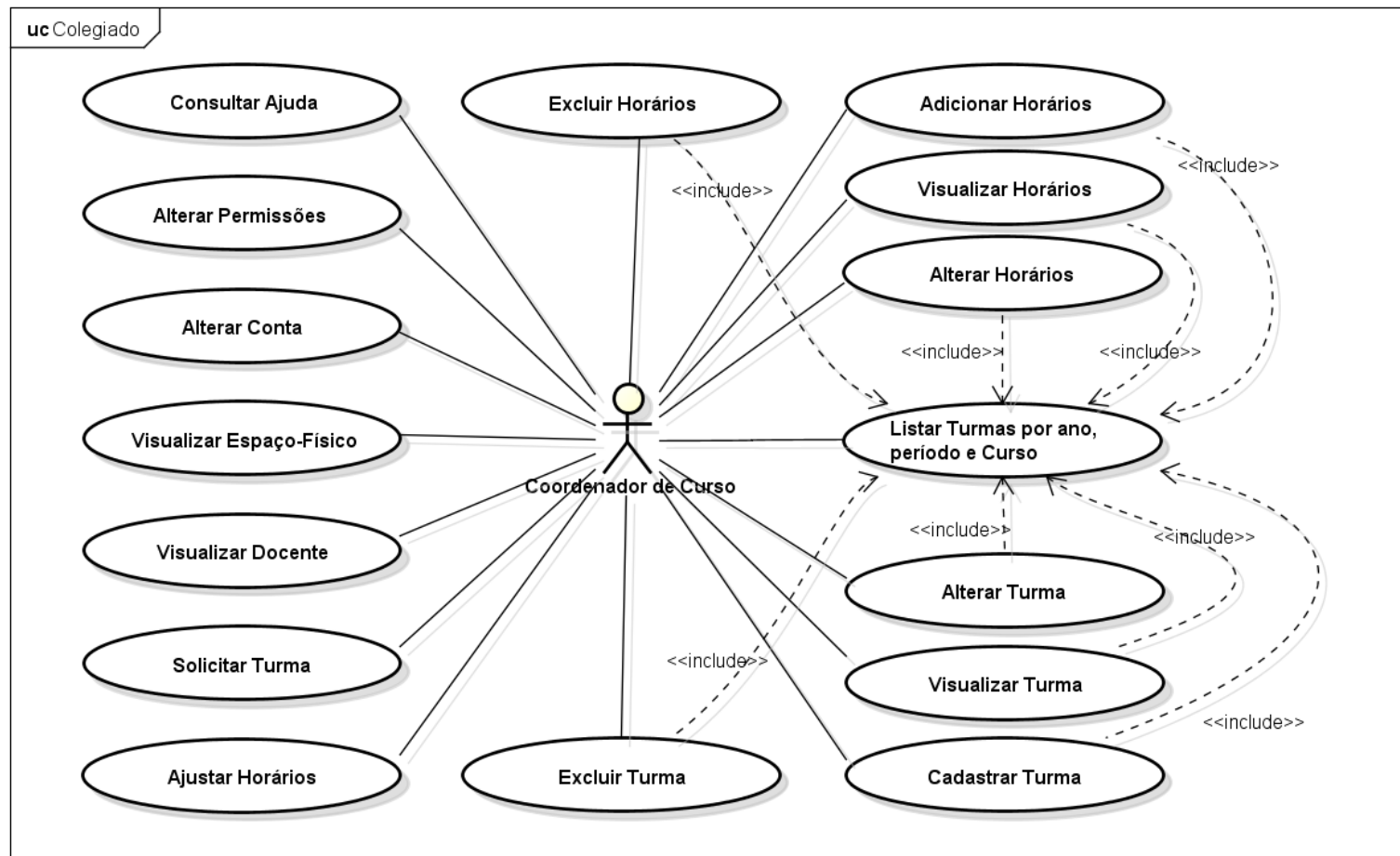


Figura 25 - Diagrama de caso de uso do Departamento.

Já na Figura 26 aparece o diagrama de casos de uso do Colegiado de Curso, onde o Coordenador do Curso é o ator envolvido.



powered by astah®

Figura 26 - Diagrama de casos de uso do Colegiado de Curso.

De posse das imagens acima, que exibem o diagrama simplificado de casos de uso do sistema, agora é preciso detalhar as ações dos operadores e as respostas do sistema às suas ações.

A primeira a ser formalmente explicada é a de ALTERAR PERMISSÕES.

SEÇÃO: ALTERAR PERMISSÕES

Seqüência Típica de Eventos

Tabela 4 - Descrição do caso de uso "Alterar Permissões"

Ação do Ator	Resposta do Sistema
1. Este caso de uso começa quando um usuário clica no <i>link</i> "Alterar Permissões".	2. Uma tela com a grade de permissões é exibida e as permissões editáveis são habilitadas.
3. O usuário (com poder de administrador) seleciona quais permissões que gostaria de fornecer a outro usuário.	
4. Feito isso, ele clica em salvar.	5. As alterações são gravadas e uma tela de confirmação é exibida.

O detalhamento se segue com a ALTERAÇÃO DE CONTA.

SEÇÃO: ALTERAR CONTA

Seqüência Típica de Eventos

Tabela 5 - Descrição do caso de uso "Alterar Conta"

Ação do Ator	Resposta do Sistema
1. Este caso de uso começa com o clique no <i>link</i> de alteração de dados de conta.	2. Uma tela é exibida, onde os dados cadastrais do usuário é exibido. Os campos passíveis de alteração estão habilitados.
3. O usuário seleciona os campos que deseja mudar e insere os dados referentes a cada campo.	
4. Após isso clica no botão de confirmação.	5. Uma tela de confirmação de armazenamento é mostrada.

A opção de CONSULTAR AJUDA será descrita.

SEÇÃO: CONSULTAR AJUDA

Seqüência Típica de Eventos

Tabela 6 - Descrição do caso de uso "Consultar Ajuda"

Ação do Ator	Resposta do Sistema
1. Este caso de uso começa com o clique no ícone de ajuda, presente no <i>menu</i> superior.	2. Uma tela com dicas e instruções é exibida.
3. O usuário busca a informação que necessita: <ol style="list-style-type: none"> Na listagem exibida; Digita no campo de busca a informação que deseja e clica em "Buscar"; 	4. Novas instruções, dicas e sugestões são listadas em tela.

A opção a seguir é LISTAR CURSOS POR DEPARTAMENTO, que é análoga LISTAR TURMAS POR ANO, PERÍODO E DEPARTAMENTO e LISTAR TURMAS POR ANO, PERÍODO E CURSO, portanto a formalização se aterá em uma única tabela, onde os parâmetros serão: **ano, período, departamento e cursos**.

SEÇÃO: LISTAR CURSOS POR DEPARTAMENTO

Seqüência Típica de Eventos

Tabela 7 - Descrição do caso de uso "Listar Cursos Por Departamento"

Ação do Ator	Resposta do Sistema
1. Este caso de uso começa na página inicial. O usuário insere os parâmetros desejados para a filtragem e clica em “Buscar”.	2. O sistema faz uma busca no banco de dados e retorna uma listagem de cursos ou disciplinas, com as seguintes informações relevantes para o usuário, que será exibida em tela.

A próxima explanação será sobre a funcionalidade VISUALIZAR TURMA, que, da mesma forma que a Seção anterior, é análoga aos seguintes casos de uso: VISUALIZAR HORÁRIOS, VISUALIZAR DOCENTES, VISUALIZAR ESPAÇO-FÍSICO, VISUALIZAR BLOCOS. Então, uma única tabela representará este caso.

SEÇÃO: VISUALIZAR TURMA

Seqüência Típica de Eventos

Tabela 8 - Descrição do caso de uso "Visualizar Turma"

Ação do Ator	Resposta do Sistema
1. Este caso de uso começa após a listagem de ofertas existentes.	
2. O usuário clica em “Exibir” para que a turma desejada seja expandida e as informações exibidas.	3. Um quadro de informações gerais é exibido.

O caso de uso ALTERAR TURMA é semelhante à: ALTERAR HORÁRIOS, AJUSTAR HORÁRIO, ALTERAR DOCENTE, ALTERAR ESPAÇO-FÍSICO, ALTERAR BLOCOS. Da mesma forma exibiremos na próxima tabela o caso de uso que sumariza os casos citados.

SEÇÃO: ALTERAR TURMA

Seqüência Típica de Eventos

Tabela 9 - Descrição do caso de uso "Alterar Turma"

Ação do Ator	Resposta do Sistema
1. Este caso de uso começa após a listagem de ofertas existentes.	
2. O usuário clica em “Alterar” para selecionar a turma que deseja alterar.	3. O usuário é redirecionado para uma nova página, representada por um formulário, com os campos editáveis habilitados.
4. O usuário altera os dados que deseja e clica em salvar.	5. Uma barra que ilustra o processamento surgirá, mostrando o progresso da tarefa.
	6. Com o fim do processamento um alerta de confirmação de alteração surge.
7. O usuário, porém, pode continuar navegando nesta mesma página, clicando na figura da “Lupa”, que denota uma busca por disciplinas.	8. Uma janela com a lista de disciplinas pertencentes ao departamento anteriormente selecionado, surge.

- | | |
|--|--|
| 9. O usuário seleciona a disciplina e clica em “OK”. | 10. O sistema redireciona-o para uma nova página, com novas informações e este caso de uso recomeça. |
|--|--|

Já a função CADASTRAR TURMA é similar aos casos de uso ADICIONAR HORÁRIOS, ADICIONAR DOCENTES, ADICIONAR ESPAÇO-FÍSICO e ADICIONAR BLOCOS, que também serão reduzidos a um único detalhamento, que se segue.

SEÇÃO: CADASTRAR TURMA

Seqüência Típica de Eventos

Tabela 10 - Descrição do caso de uso "Cadastrar Turma"

Ação do Ator	Resposta do Sistema
1. Este caso de uso começa na página inicial, onde o usuário poderá clicar em “Criar Nova Turma”, após inserir os parâmetros que deseja, a saber: ano, período e unidade.	2. Um quadro surge, com um formulário semi-preenchido é mostrado (somente alguns campos são preenchidos automaticamente, que estarão desativados para alterações).
3. O usuário preenche/seleciona cada uma das informações que lhe convir e clica em salvar.	4. Uma tela de confirmação de salvamento é exibida e as abas de horários, docentes, espaço-físico e blocos são habilitadas, para que, da mesma forma, sejam preenchidas e novos dados sejam cadastrados.

No tocante a EXCLUIR TURMA, temos como casos de uso semelhantes EXCLUIR HORÁRIO, EXCLUIR DOCENTE, EXCLUIR ESPAÇO-FÍSICO E EXCLUIR BLOCOS. Que pode ser visto como:

SEÇÃO: EXCLUIR TURMA

Seqüência Típica de Eventos

Tabela 11 - Descrição do caso de uso "Excluir Turma"

Ação do Ator	Resposta do Sistema
1. Este caso de uso começa após a listagem de ofertas existentes.	
2. O usuário escolhe a turma que deseja e clica em “Excluir”.	3. Um aviso de irreversibilidade da ação surge na tela.
4. O usuário pode: <ul style="list-style-type: none"> a. Confirmar; b. Cancelar; 	5. Ao confirmar a exclusão é feita e uma nova lista (sem a turma excluída) é exibida. Ao cancelar, nada acontece.

Será visto abaixo a descrição de SOLICITAR TURMA, que é também conhecido como TRAMITAR TURMA. Esse procedimento envolve os seguintes casos de uso: SOLICITAR AJUSTE DE TURMA, ENVIAR TURMA PARA ANÁLISE, ENVIAR TURMA PARA MATRÍCULA, ENVIAR TURMA PARA LANÇAMENTO DE NOTA e FECHAR TURMA, assim será resumido em uma tabela.

SEÇÃO: SOLICITAR TURMA

Seqüência Típica de Eventos

Tabela 12 - Descrição do caso de uso "Solicitar Turma"

Ação do Ator	Resposta do Sistema
1. Este caso de uso começa após a listagem ou criação de uma oferta.	

2. O usuário escolhe a turma que deseja solicitar e clica em “Tramitar”.
3. Uma tela é exibida com as seguintes informações:
 - Setor proveniente;
 - Setor de destino;
 - Fluxo que será dado, o que engloba a situação da turma. Podendo ser: “Solicitada”, “Análise”, “Ajuste de Horários”, “Matrícula”, “Lançamento de Notas” e “Inativa”.
 - Observações (também conhecido como despacho);
 - O documento referente à solicitação.
4. O usuário pode:
 - a. Clicar em “OK”;
 - b. Clicar em “Cancelar”.
5. Ao clicar em “OK” é iniciado o procedimento de solicitação de turma, alterando a situação da turma. É exibida uma barra de progresso do trâmite e o documento segue para outro setor. (Podendo ou não aparecer na listagem, dependendo do perfil do usuário logado)
Ao cancelar, nada acontece, a tela é fechada e retorna à listagem de disciplinas ofertadas.

Todos os casos de uso foram descritos e formalizados. O funcionamento do sistema e seu desenvolvimento acompanhou todos os procedimentos citados acima e na próxima Seção irá delinear tecnicamente como cada funcionalidade foi implementada e quais as tecnologias utilizadas, bem como as configurações feitas e as interfaces criadas.

4 IMPLEMENTAÇÃO E AVALIAÇÃO

No Capítulo anterior foi definido o escopo do projeto e toda a coleta de requisitos necessários junto aos órgãos/departamentos/colegiados interessados efetuada durante o desenvolvimento do trabalho em questão. Assim, teve início uma nova fase, que será exposta a seguir: o desenvolvimento da aplicação.

4.1 Implementação do projeto

Foi feito um estudo aprofundado de diversas tecnologias passíveis de serem usadas, cujos parâmetros analisados foram: confiabilidade, segurança e desempenho, no contexto tecnológico da universidade.

Constatou-se que, na camada de apresentação, deveria ser usada a linguagem de marcação padrão, o HTML (HyperText Markup Language), seguindo as regras da W3C (Consórcio World Wide Web, órgão responsável pelo desenvolvimento de padrões e diretrizes para a Web, <http://www.w3c.br>) e o javascript, como linguagem usada do lado do cliente (client-side). Já na camada de lógica de negócios, a tecnologia que mais se adequou às exigências foi o PHP, correspondendo às demandas de segurança, estabilidade e alta produtividade. E na camada de dados foi usado o IBM DB2, SGBD já utilizado pela instituição.

Nas sessões que se seguem serão descritas detalhadamente as ferramentas, as tecnologias e as metodologias de desenvolvimento utilizadas.

4.1.1 PHP 5

O PHP é uma linguagem de extensão (scripting, i.e., que permite o controle de uma ou mais aplicações) interpretada, de propósito geral, perfeitamente adequada para o desenvolvimento web e que pode ser embutida no código HTML. Foi criada por Rasmus Lerdorf, 1995 e logo foi disponibilizada para que a comunidade pudesse contribuir com sua evolução e na correção de problemas.

O suporte à programação orientada a objetos (OO) se deu a partir da versão 3.0, após participação de Zeev Suraski e Andi Gutmans, que são atuais diretores da Zend Technologies (empresa que é cooperadora e mantenedora do PHP e que lançou o ZF).

Porém, cabe salientar, que o PHP não é uma linguagem de programação orientada a objetos, e sim, que tem suporte à orientação a objetos. O que significa dizer que é possível desenvolver tanto com programação estruturada (baseada em funções) como com OO (baseada em classes) ou até mesmo combinando-as, dando maior versatilidade e liberdade ao programador.

Tecnologicamente, o PHP é uma linguagem idealizada para aplicações web, uma vez que tem suporte a inúmeras bibliotecas de código e tem a capacidade de ser integrada a outras linguagens e/ou plataformas com facilidade. Já economicamente, é uma linguagem aberta, livre, gratuita e sem restrições para o uso comercial. O fácil aprendizado é outro fator que deve ser levado em consideração, haja vista a existência da vasta documentação, dos fóruns e das listas de discussão. Além disso, o PHP é multiplataforma e portátil, uma vez que não há limitação quanto ao sistema operacional.

4.1.2 A arquitetura Model-View-Controller (MVC)

Anteriormente, a grande maioria das aplicações era monolítica, isto é, de uma só camada e eram desenvolvidos para serem usados em uma única máquina. Geralmente todas as funcionalidades se encontravam em um módulo, gerado por uma imensa quantidade de linhas de código e de manutenção extremamente complexa. A entrada do usuário, a verificação, a lógica de negócio e o acesso ao banco de dados estavam centralizados em um mesmo local.

Entretanto, com a demanda de compartilhamento da lógica de acesso aos dados entre diversos usuários simultaneamente, surgiram as aplicações de duas camadas (2-tier), onde a base de dados foi separada da camada de apresentação e da camada da lógica de aplicação. Porém, ainda haviam problemas, como o controle de atualizações, cuja gerência era descentralizada e, a dificuldade de manutenção do sistema, haja vista que os programas eram instalados nos clientes.

Com o intuito de sanar tais problemas, vieram as aplicações de três camadas (3-tier). E, para acompanhar esse novo modelo, passa a ser necessária a adoção de diversos padrões de projetos (design patterns), dentre eles o MVC, que visa aumentar a escalabilidade, a eficiência, a reutilização e, conseqüentemente, a produtividade.

O MVC é um padrão de arquitetura de software que separa a lógica de negócio da lógica de apresentação, permitindo que o desenvolvimento cada uma seja feito separadamente.

Na arquitetura em questão, o modelo representa os dados da aplicação e as regras de negócio que governam o acesso e a manipulação deles. Além disso, ele mantém a persistência dos dados e fornece, ao controlador, o acesso às funcionalidades presentes na aplicação, encapsuladas no próprio modelo.

Já na visualização, um componente processa o conteúdo desejado (proveniente do modelo) e encaminha ao controlador as ações do usuário. Alimenta templates com esses dados, mas de forma alguma os modifica, só os apresenta.

O núcleo da aplicação é a camada de controle, que é onde o comportamento do sistema é definido. É nessa camada que as ações do usuário são interpretadas e mapeadas para as chamadas do modelo. Aqui que são selecionadas as visualizações a serem exibidas, fazendo com que o modelo manipule os dados, a partir da respectiva resposta à solicitação do usuário e os envie para que sejam apresentados.

Na Figura 27 é possível ter uma idéia superficial de como é a estrutura da arquitetura em questão.

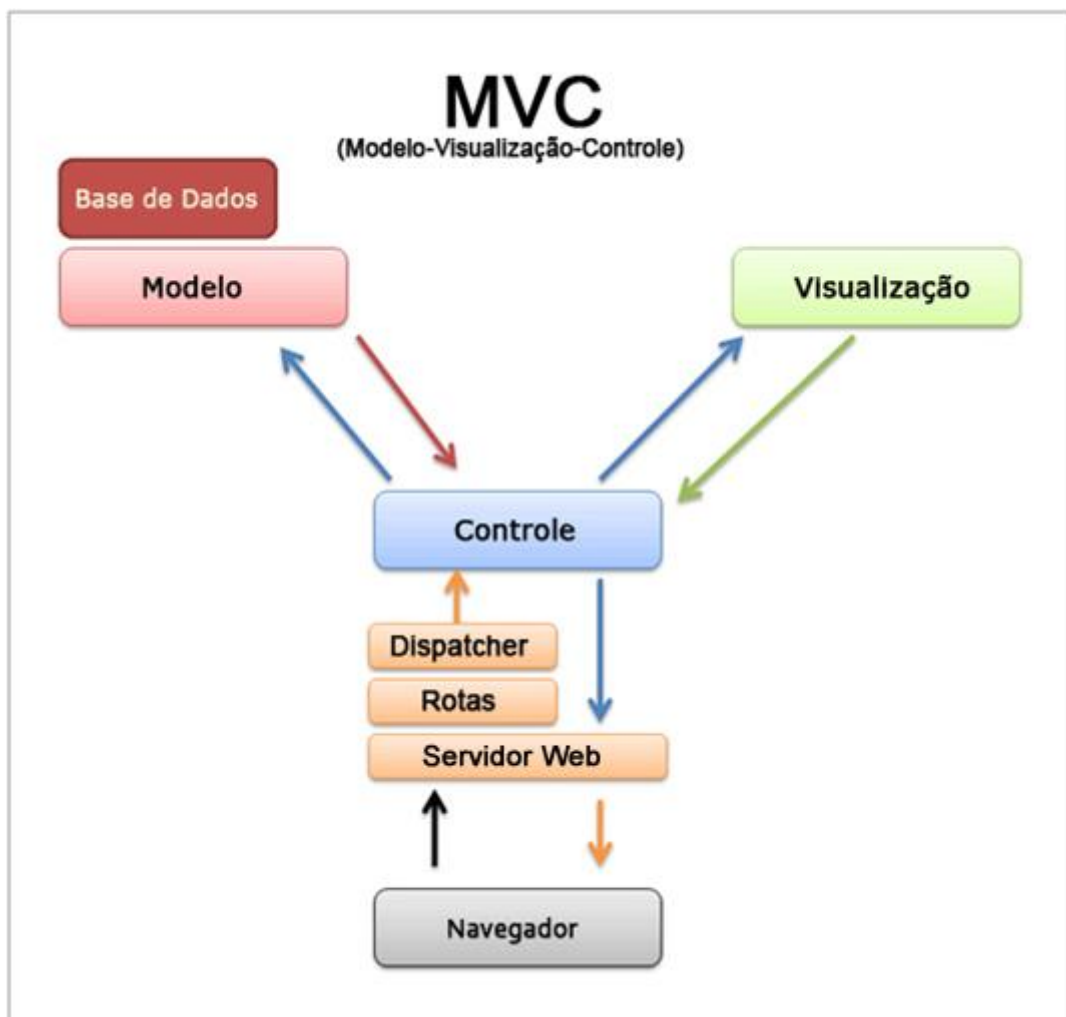


Figura 27 - Arquitetura MVC.

Na Figura 28, é mostrado o funcionamento, propriamente dito, das 3 camadas. O que cada uma delas faz o fluxo de dados e onde acontecem os devidos processamentos.

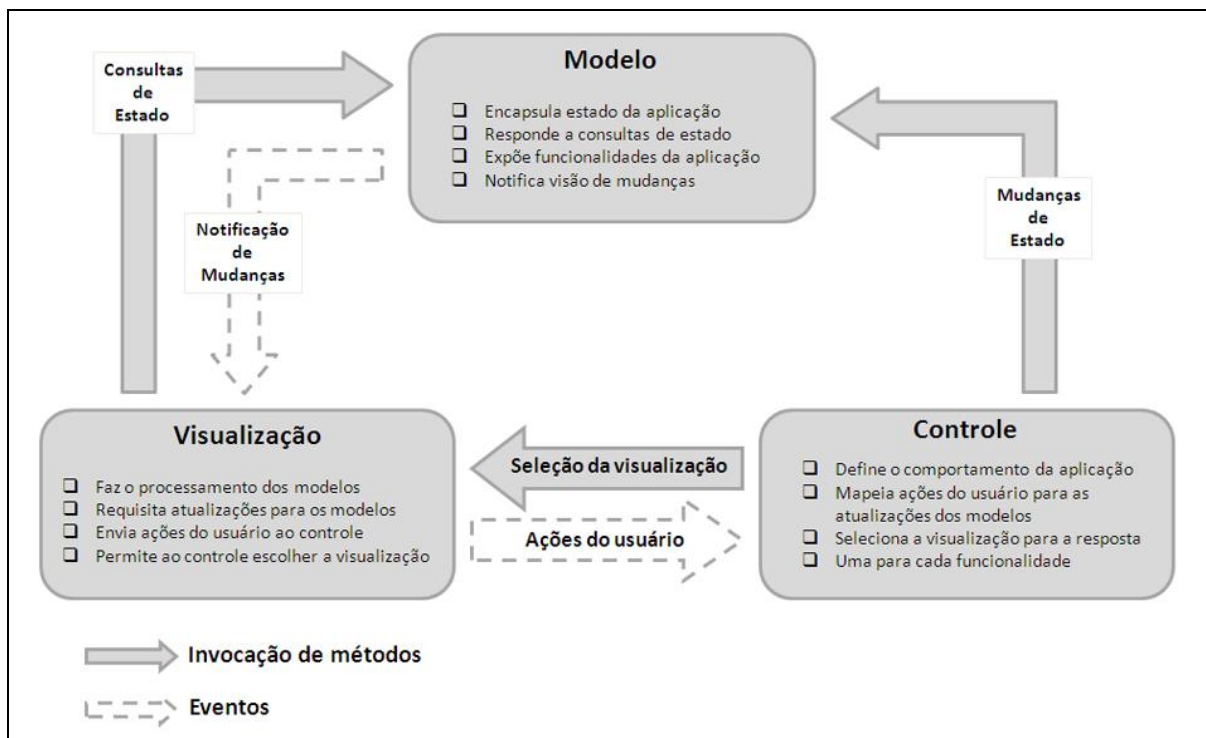


Figura 28 - Funcionamento da arquitetura MVC.

Foi assumido como um padrão de projeto e tem sido considerada uma boa prática, quando se trata de aplicações web, pois o MVC é capaz de gerenciar múltiplos visualizadores utilizando um mesmo modelo, tornando fácil efetuar testes, manutenções e atualizações; torna a aplicação escalável; aumenta, significativamente, a produtividade, por ser possível o desenvolvimento em paralelo do modelo, da visualização (design) e do controles, independentemente.

4.1.3 Frameworks

De acordo com o Prof. Elton Luis Minetto [15]: “O *framework* é uma base de onde se pode desenvolver algo maior ou mais específico. É uma coleção de códigos, classes, funções, técnicas e metodologias que facilitam o desenvolvimento de novos *softwares*.”

A motivação de utilizar um *framework* passa por diversos campos, indo desde o reuso de código já trabalhado (naturalmente isso depende da credibilidade e da experiência dos desenvolvedores), passando pela padronização de uma estrutura de projeto já idealizada e efetivamente elaborada, chegando à complexidade que alguns alcançaram, devido ao número

de colaboradores, de tempo de mercado e também da quantidade de erros que já foram tratados.

Outro quesito de extrema importância é a segurança do projeto, uma vez que a qualidade de uma implementação se dá, em grande parte, na segurança que está embutida no sistema produzido.

Afora o já exposto, os frameworks atuais, adotam a programação orientada a objetos e diversos padrões de projetos, como o MVC, o *singleton*, *observer*, *factory*, DAO (*Data Access Object*), entre outros, o que uniformiza o desenvolvimento, a estrutura dos sistemas e as metodologias utilizadas, diminui o custo de aprendizagem, e com isso, aumenta a manutenabilidade das aplicações.

O conhecimento deve estar sempre documentado, devem haver padrões e métodos que uniformizem os processos, seja em empresas pequenas ou nas grandes, de modo que um projeto não dependa das pessoas, mas das funções que elas exercem. Isso é uma maneira de assegurar qualidade e disciplina, para atingir as metas propostas. A utilização de um *framework* de desenvolvimento estimula esse processo.

4.1.4 Escolha do framework e da biblioteca

Tendo terminada a fase de análise e estudo, surge uma nova fase de pesquisa: a busca pelo melhor conjunto de ferramentas de desenvolvimento, pensando na produtividade, na segurança e na confiabilidade do produto.

Diversos testes foram feitos, com inúmeros frameworks consagrados e alguns recentemente lançados. Os dados que auxiliaram na decisão estão na tabela* abaixo (lembrando que todos os frameworks avaliados são de código aberto e gratuito):

Tabela 13 - Comparativo de frameworks.

Frameworks \ Aspectos avaliados	CakePHP	Spaghetti	Yii	Zend	Prado
PHP 4.0	✓	✓	✗	✓	✗
PHP 5.0	✓	✓	✓	✓	✓
MVC	✓	✓	✓	✓	✓
Suporte a diversos SGBDs	✓ ¹	✓ ¹	✓ ¹	✓	✓ ¹

Segurança	✓	?	✓	✓	✓
Alta velocidade de processamento	✗	✗	?	✓	?
Facilidade de configuração	✓	✓	✗	✓ ²	✗
Facilidade de aprendizado	✓ ²	✓ ²	?	✗	?
Documentação	✓ ³	✗	✗	✓ ³	?
Confiabilidade	?	✓	?	✓	?
Manutenção	?	?	?	✓	✓
Interface	✓	✓	✓	✗	✓
Escalabilidade / Modularização	✓	✓	✓	✓	✓
Adaptabilidade	✓	?	?	✓	?
Ajax	✓	✓	✓	✓	✓

* Tabela adaptada de <http://www.phpframeworks.com/>, junho de 2010

Legenda:

6. Avaliação positiva do aspecto
7. ¹ Avaliação positiva, porém com suporte limitado a alguns SGBD
8. ² Avaliação positiva, porém com análise um pouco subjetiva desta facilidade
- ✓ ³ Avaliação positiva, porém a documentação da comunidade deixa a desejar
- ✗ Avaliação negativa do ponto em questão
- ? Avaliação inconclusiva do quesito analisado

Os *frameworks* mais profundamente analisados foram: CakePHP, Spaghetti e o Zend. O Yii e o Prado foram estudados concomitantemente ao Zend, porém foi dada uma ênfase maior ao ZF devido à quantidade de documentação disponível e ao suporte ao SGBD usado pela instituição.

O Zend Framework (ZF) foi o que mais se adequou ao ambiente existente. Além da credibilidade dos desenvolvedores do ZF, por serem criadores da tecnologia PHP, outros fatores foram determinantes para a escolha desta ferramenta, bem como: a existência de diversas classes já implementadas e exaustivamente testadas, trazendo mais confiabilidade ao

sistema desenvolvido e o excelente desempenho obtido quando submetido a altas cargas de processamento.

Feito isso, partimos para a análise da primeira camada, que é de suma importância, uma vez que é o rosto do sistema para o usuário. Pensando nisso, foi observado a necessidade da utilização de uma biblioteca javascript, visando facilitar a validação de dados e a implementação de requisições assíncronas, o tão conhecido Ajax (*Asynchronous Javascript And XML*), tornar a interface mais amigável e aumentar a interatividade do usuário com o sistema, já que havia a intenção de adequação aos padrões da web 2.0, onde a interatividade é uma constante.

Então, usando como base a escolha do ZF, fizemos testes de adaptabilidade com algumas das mais famosas bibliotecas de javascript, e obtivemos os seguintes resultados:

- jQuery – Integração média. Algumas iniciativas tendem a resolver os problemas existentes, mas ainda não têm tanta força e são muito recentes. Portanto ainda deixa a desejar quando se trata de compatibilidade com o ZF. Porém um amplo leque de efeitos interessantíssimos é oferecido em sua versão mais recente (1.4.2), além de ter sido considerada, em nossa análise, a mais completa, de uso mais simples e, comprovadamente, a biblioteca com maior número de menções na web nos últimos 12 meses (<http://www.google.com.br/trends?q=jquery%2Cdojo%2Cprototype&ctab=0&geo=all&date=ytd&sort=1>), o que nos daria um maior balizamento para usá-la.
- Dojo Toolkit – A grande quantidade de classes implementadas, de funcionalidades conjuntas, com simples usabilidade e a total integração com o ZF, fizeram do Dojo Toolkit a escolha para este sistema. Além disso a extensa documentação, tanto da comunidade quanto dos desenvolvedores e a disponibilidade de uma área de testes e de demonstrativos é oferecida em seu *website* (<http://dojocampus.org/explorer>) são pontos positivos dessa biblioteca. No entanto, se comparado com o jQuery, peca na quantidade de documentação, na usabilidade e alguns outros aspectos, o que fez do quesito integração o ponto decisivo sem seu favor.
- Prototype – É uma biblioteca extremamente forte e a mais antiga das supracitadas. Vale lembrar da alta credibilidade dessa biblioteca, que além de possuir a distribuição *standalone*, faz parte de grandes projetos como Ruby on

Rails (RoR). Entretanto não acompanhou o desenvolvimento dos navegadores, o que a afastou das concorrentes. Há uma grande expectativa para o lançamento da versão 2.0, que deverá corrigir grande parte dos problemas que surgiram em versões de “extensão de métodos” DOM (*Document Object Model* - Modelo de Objetos de Documento, que é uma especificação da W3C, independente de plataforma e tecnologia, onde pode-se dinamicamente alterar e editar a estrutura, conteúdo e estilo de um documento eletrônico, permitindo que o documento seja mais tarde processado e os resultados desse processamento, incorporados de volta no próprio documento, extremamente utilizado em javascript), e que acarretará uma mudança radical na sua sintaxe e conseqüentemente uma eliminação de grande parte do código entrará em desuso. Não há nenhuma compatibilidade com o ZF, mas apesar de não haver nenhum sinal de uma parceria futura ou mesmo indícios de integração com algum *framework*, seu uso é possível por se tratar da camada de apresentação, estando presente somente do lado do cliente.

4.1.5 Zend Framework

Tomando como premissa a simplicidade de desenvolvimento, o ZF conseguiu sintetizar esse espírito. Tem como foco a construção de aplicações web, seguindo as regras da web 2.0 e possui *web services* seguros, confiáveis e atuais, usando API's de grandes empresas como o Google, a Amazon e o Yahoo.

Vale ressaltar que o aumento da produtividade foi idealizado desde a implementação do *framework*. Através de uma biblioteca leve e fracamente acoplada, permitindo a fácil personalização do produto, o Zend usa também uma base de códigos exaustivamente testada e extensível, uma arquitetura flexível, além da integração com diversas ferramentas modernas aumentando a acessibilidade e a usabilidade do sistema.

Com licença segura e confiável, a *Zend Framework's License*, baseada na *BSD License* [16], assegura que seu código seja livre, ou seja, irrepreensível, e protegido, seguindo os padrões, em conformidade com as regras que são consideradas as melhores práticas de programação.

O núcleo do ZF é um diretório de classes, seguindo um padrão semelhante ao usado no Java, simulando uma estrutura de pacotes. O objetivo dessa padronização de

arquitetura é a reutilização do código sem que o desenvolvedor tenha que alterar a biblioteca nativa do *framework*, podendo simplesmente criar pastas e inserir conteúdos dependendo do uso do *software*. As classes do ZF utilizadas nesse trabalho se encontram no APÊNDICE A.

O Zend é uma ferramenta versátil e com uma diversidade de funções que surpreende. Suporta conexão a praticamente todos os SGBDs, uma característica que foi essencial para a definição do *framework* utilizado. Implementa a conexão, consultas e alteração de dados, bem como controle de transações via PDO, uma nova biblioteca, semelhante ao ADOdb e PEAR DB - que são bibliotecas de abstração da camada de banco de dados para PHP, isto é, fornece suporte a diversos SGBDs, tornando o seu código reutilizável - porém muito mais rápida, pois foi escrita em uma linguagem compilada (C/C++), e, ao invés de trabalhar com erros, lida com exceções, o que torna menos traumático o tratamento de inconformidades de dados e outras falhas. O PDO representa uma camada de abstração de acesso a banco de dados, não importando, portanto, qual o banco usado, uma vez que as mesmas funções de manipulação de dados ou recuperação de informações são as mesmas.

Em suma, com o intuito de facilitar qualquer tipo de manutenção ou atualização necessária ao sistema, o Zend se adéqua perfeitamente na solução proposta.

4.1.6 Instalação e configurações

Para que o sistema web funcionasse corretamente foi necessária a transformação da máquina usada em um servidor web. Para tanto, foi definido o seguinte ambiente de desenvolvimento:

- Instalação do WAMP 2.0.
- Com o Apache 2.2.5.
- Usando o PHP 5.2.5.
- Tendo instalado o servidor Tomcat 6.0.
- Foi necessário o download do ZF e do Dojo por motivos óbvios.
- A IDE (*Integrated Development Environment*⁹) utilizada foi o Netbeans.
- A inclusão da biblioteca de criptografia usada pelos desenvolvedores do SIE.
- Inclusão do arquivo *cripto.war*, como web-service desenvolvido.

⁹ Acrônimo, da língua inglesa, de *Integrated Development Environment* ou Ambiente Integrado de Desenvolvimento.

Após a instalação dos devidos programas e o ajuste de detalhes, a aplicação funcionará corretamente assim que toda ela for transferida para o servidor em questão.

Foram feitas configurações iniciais em arquivos “.htaccess”, garantindo que o fluxo das informações siga o desejável, de acordo com a premissa do modelo MVC e seja passível de gerenciamento centralizado.

No arquivo que controla esse fluxo, o chamado *bootstrap* ou “index.php”, diversas definições foram implementadas, como por exemplo:

- Forma de exibição de exceções;
- Caminho de arquivos para inclusão;
- Carregador automático de classes do ZF;
- Tratamento de requisições;
- Tipo de documento;
- Codificação de dados;
- Processamento de máquinas de *templates*;
- Caminho base;
- Início de sessão;
- Diferentes rotas, com o intuito de criar URLs¹⁰ amigáveis;
- Controladores;
- Configurações de banco de dados;
- Localização;
- Internacionalização;
- Entre muitas outras.

No ANEXO 3 é possível entender o procedimento sistemático de instalação de aplicativos e as configurações necessárias para o funcionamento correto do sistema aqui descrito, um roteiro de instalação.

4.1.7 Desenvolvimento

Este é o núcleo desse trabalho. Após um exaustivo período de análises e estudos, é chegada a fase de desenvolvimento. Serão descritas a seguir todas as etapas de implementação do sistema em questão, da interface, passando pela lógica de negócios e chegando aos testes.

¹⁰ Da língua inglesa: *Uniform Resource Locator*.

4.1.7.1 Diagrama de classes simplificado

Vale mencionar que somente estão ilustradas uma parte das classes implementadas, com o intuito de facilitar a compreensão e o entendimento do que foi feito no sistema. Horizontalmente destacou-se o que foi desenvolvido pelo ZF e o que pertence ao Sistema de Oferta de Disciplinas WEB.

A Figura 29 exemplifica algumas classes do ZF e mostra de que forma o sistema interage com elas. As classes desenvolvidas herdam ou estendem às do *framework*. Então, no exemplo abaixo temos que as classes “forms_LoginForm”, “forms_InfoTurmaGeralForm”, “forms_NovaTurmaInfoGeralForm” e “forms_ListaDepartamentosDisciplinasForm” herdam da classe “Zend_Form”.

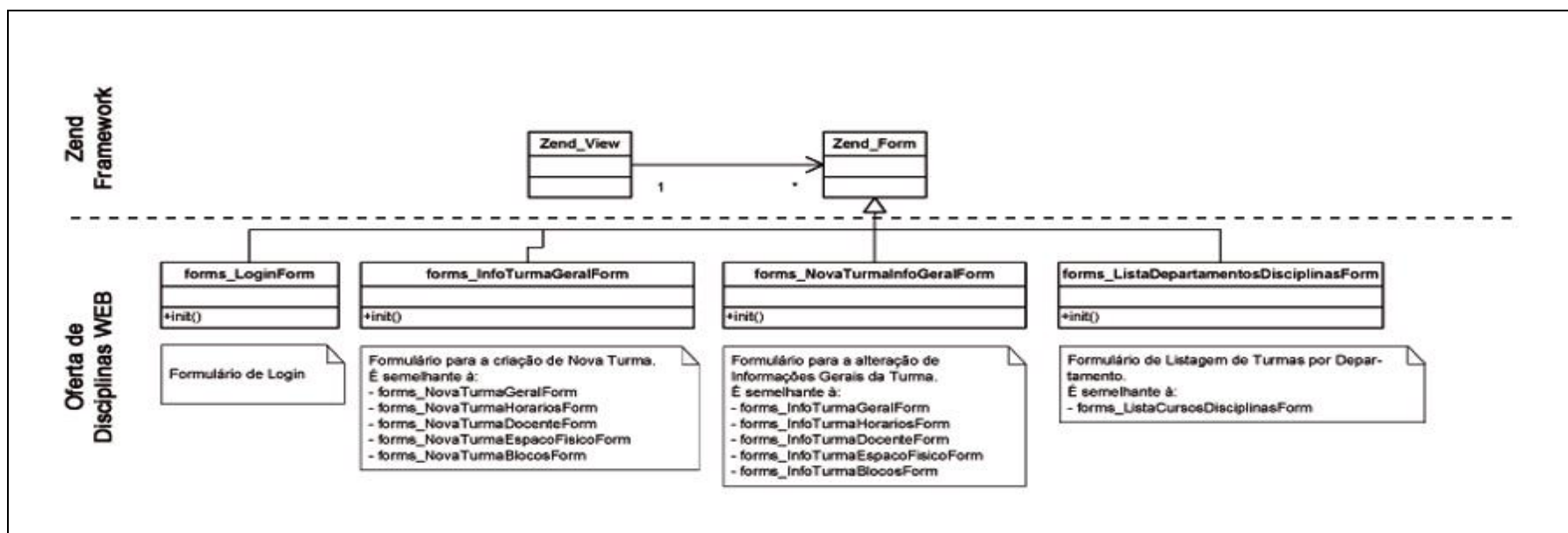


Figura 29 - Diagrama de classes simplificado - Zend_View com Zend_Form.

Da mesma forma, a Figura 30 demonstra que as classes “LoginController”, “ListaDepartamentoController”, “InfoTurmaController”, “CadastraTurmaController” e “ErroController” herdam da classe “Zend_Controller”.

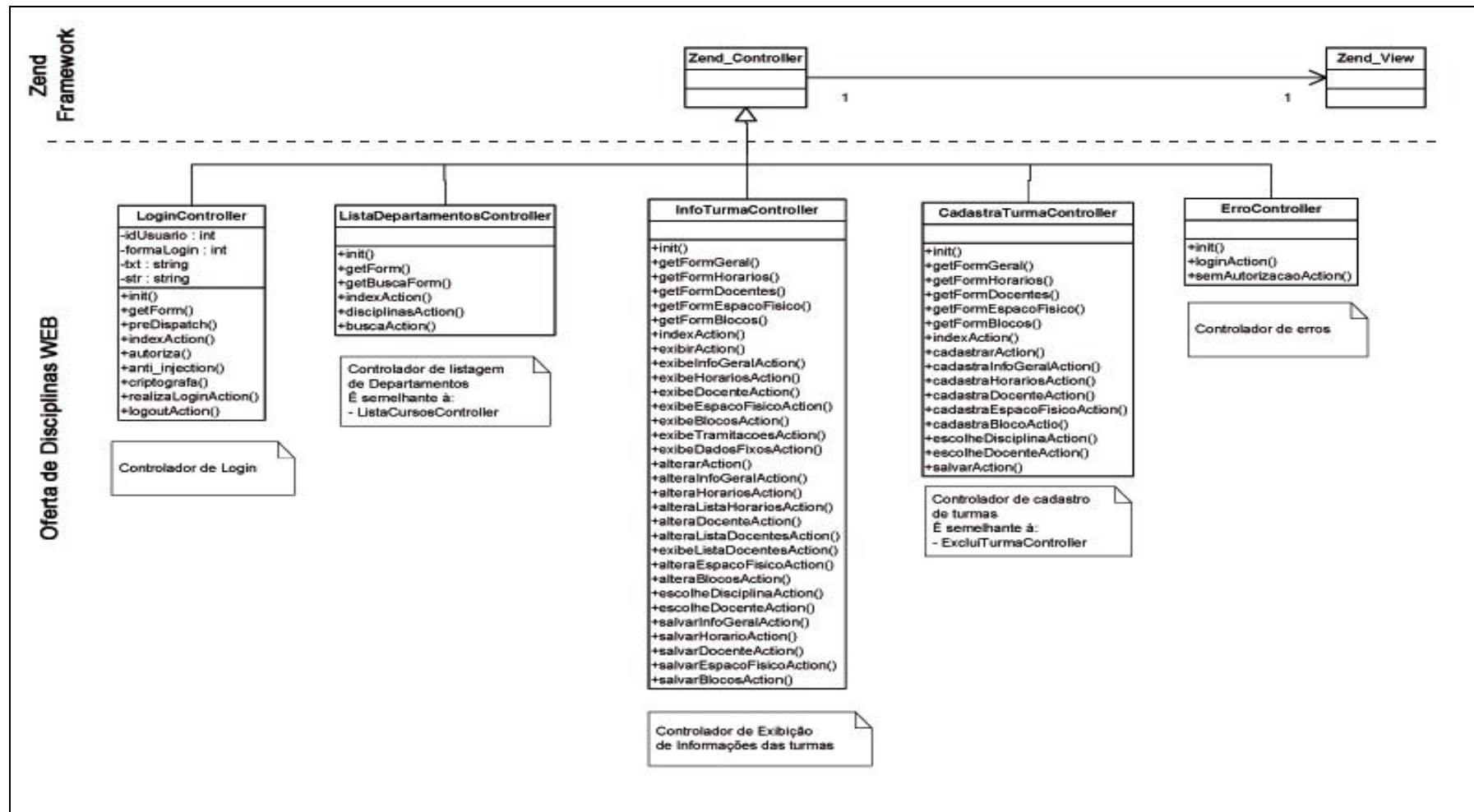


Figura 30 - Diagrama de classes simplificado - Zend_Controller com Zend_View.

As classes da Figura 29, que possuem o prefixo “forms_” implementam formulários, como de *login*, de alteração de dados, de inclusão de novos dados e de listagem de disciplinas por Departamento. Estas herdam atributos da classe do ZF, a “Zend_Form”, que interagem com outra classe do ZF. A “Zend_View”, responsável pela visualização do conteúdo.

Ademais, na Figura 30, as classes com o sufixo “Controller” são os chamados controladores do modelo MVC e são responsáveis pelo processamento dos dados. O “LoginController”, por exemplo, trabalha na validação e na verificação dos dados do usuário que irá entrar no sistema. Já o “CadastraTurmaController” é responsável pela inclusão da turma no banco de dados, e também da validação de cada campo enviado para realizar essa inserção, e assim por diante.

4.1.7.2 A interface

O primeiro contato do usuário com o sistema se dá nessa camada. Tudo o que foi desenvolvido será visível somente nesse nível de abstração.

Por usar uma grande quantidade de formulários, foi necessária uma validação encadeada de dados, através da linguagem do lado do cliente, o javascript alinhado com o Dojo. Todos os cliques e teclas pressionadas aqui serão avaliados e processados pelo navegador de modo a serem “aprovadas” dependendo da classificação do campo e do conteúdo a ser submetido.

Utilizaram-se requisições assíncronas conhecidas como AJAX, evitando que a mesma página fosse recarregada a cada envio de dados, o que traria certo desconforto ao operador do sistema.

Além disso, seguimos algumas práticas desejáveis no desenvolvimento do *layout*, como: uso de imagens para representar funcionalidades; uso de metáforas e de uma linguagem coloquial; liberdade maior ao controlador durante o uso; criou-se um *menu* de ajuda, antecipando possíveis problemas ou dúvidas que surgirão. Portanto um projeto de interface simples e completo foi feito, antecipando possíveis dificuldades e aumentando a usabilidade do sistema.

A seguir uma seqüência de imagens de telas e páginas do sistema desenvolvido, que irão esclarecer e elucidar o que é considerado uma interface limpa, clara e intuitiva.

A Figura 31 representa a tela de ingresso no sistema, onde o usuário informa sua credencial e senha e envia para verificação e validação dos dados.

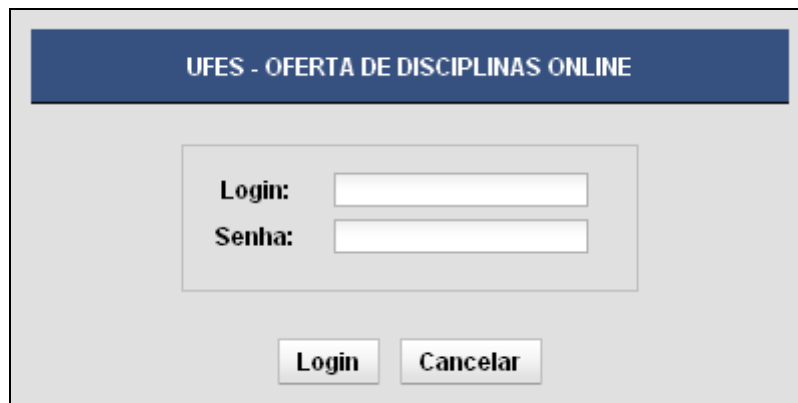
A interface de login do sistema UFES - OFERTA DE DISCIPLINAS ONLINE. No topo, há uma barra azul com o título. Abaixo, um formulário com campos para 'Login:' e 'Senha:'. Na base do formulário, há dois botões: 'Login' e 'Cancelar'.

Figura 31 - Tela de login do novo sistema

Após a conferência dos dados de senha e *login*, o sistema, automaticamente, define o perfil ao qual o operador pertence, redirecionando-o para a página de oferta de disciplinas por Curso ou por Departamento. No exemplo o perfil associado a essa conta representada na Figura 32 é de Chefe de Departamento.

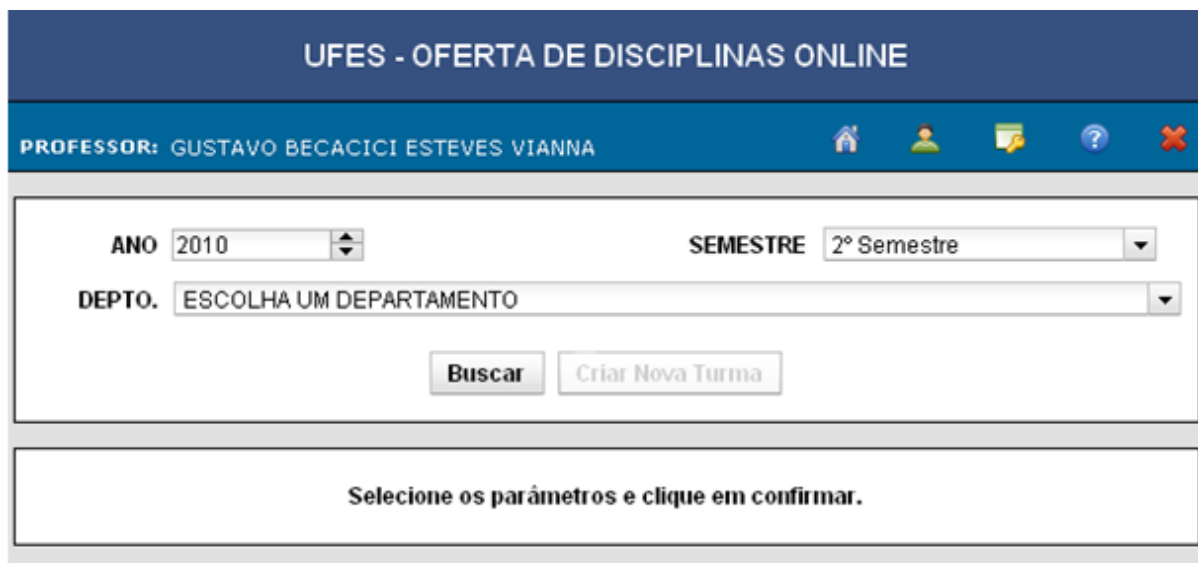
A interface principal do sistema UFES - OFERTA DE DISCIPLINAS ONLINE. No topo, uma barra azul com o título. Abaixo, uma barra de perfil do usuário: 'PROFESSOR: GUSTAVO BECACICI ESTEVES VIANNA' com ícones de navegação. O formulário principal contém campos para 'ANO' (2010), 'SEMESTRE' (2º Semestre) e 'DEPTO.' (ESCOLHA UM DEPARTAMENTO). Há botões 'Buscar' e 'Criar Nova Turma'. Na base, uma caixa de texto indica: 'Selecione os parâmetros e clique em confirmar.'

Figura 32 - Página inicial. Seleção de parâmetro para listagem de disciplinas por departamento.

A Figura 33 é a mesma imagem da Figura 32, porém referenciando o que cada elemento da página de listagem de disciplinas por Departamento significa e localiza o usuário na tela, guiando-o através de ícones intuitivos, uma barra de *menu* fixa no topo e sua identificação, informando o que deve ser feito para prosseguir.

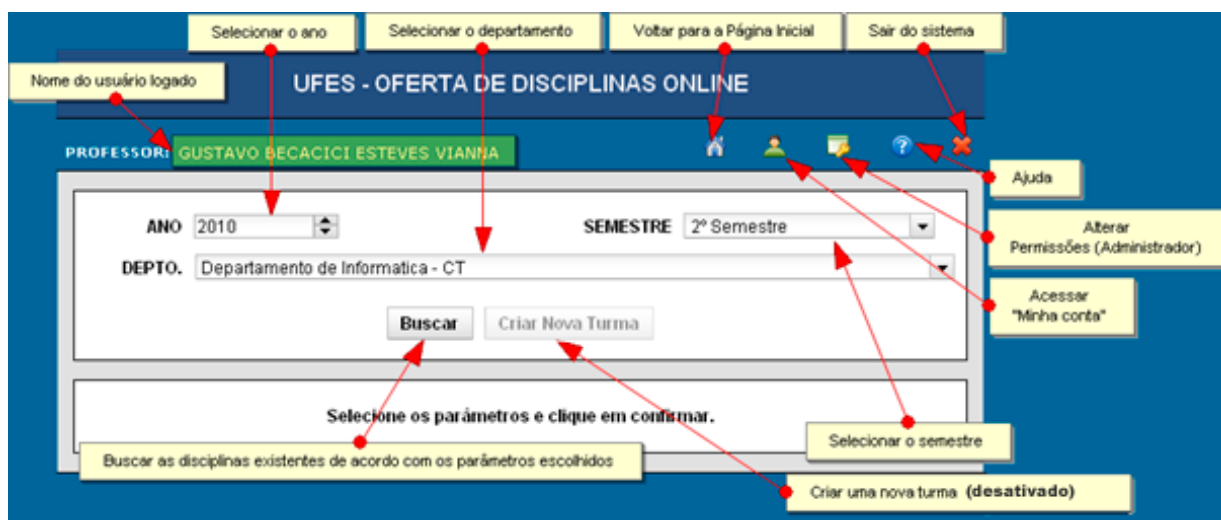


Figura 33 - Tela inicial com detalhamento explicativo de cada elemento (links, campos e botões) da página.

Ele seleciona então o ano (que por padrão é o ano corrente), o semestre (que também é o semestre corrente) e habilita-o a fazer a escolha do Departamento, que, neste caso, é o Departamento de Informática, ilustrado na Figura 34.








Figura 34 - Listagem de disciplinas ofertadas do Departamento de Informática, 2010/2.

De posse da lista de disciplinas já ofertadas no referente ano, semestre e Departamento, basta clicar no ícone “Exibir” (também representado por uma lupa) para que, sem recarregar a página, uma tabela com informações gerais da turma selecionada seja exibida, como na Figura 35, de forma limpa, trabalhando com abas e fornecendo os dados essenciais para identificar a turma, qual o Curso proprietário da disciplina, a que versão de currículo de curso esta disciplina pertence, a quantidade de vagas ofertadas, ocupadas, aumentadas e disponíveis para matrícula, horário, seu respectivo calendário, tipo e escopo da turma.

UFES - OFERTA DE DISCIPLINAS ONLINE

PROFESSOR: GUSTAVO BECACICI ESTEVES VIANNA











ANO 2010

SEMESTRE 1º Semestre

DEPTO. Departamento de Informatica - CT

Refinar Busca

CÓDIGO	TURMA	VERSÃO	DISCIPLINA	OPÇÕES
INF06582	01	2008	ALGORÍTIMOS NUMÉRICOS I	   

INFORMAÇÕES DA TURMA

CURSO:	10 B - FISICA - BACHARELADO		
VERSÃO:	2008		
DISCIPLINA:	INF06582 - Algoritmos Numéricos I		
ANO:	2010	PERÍDO:	1º SEMESTRE
TURMA:	01	CALENDÁRIO:	ACADÊMICO
VAGAS OFERECIDAS:	40	VAGAS AUMENTADAS:	15
VAGAS OCUPADAS:	9	SITUAÇÃO:	MATRÍCULA
TIPO:	TURMA REGULAR	ESCOPO:	SOMENTE ALUNOS DO CURSO

Figura 35 - Listagem de disciplinas do Departamento de Informática, 2010/2, com exibição das Informações Gerais da turma 01 de Algoritmos Numéricos 1, versão de grade curricular 2008.

É possível também, alterar uma turma específica, bastando clicar na opção “Alterar”, que o usuário será redirecionado para a página que exibirá as informações passíveis de alteração, de acordo com sua permissão (associada ao perfil que foi definido ao *logar* no sistema), como visto na Figura 36.

UFES - OFERTA DE DISCIPLINAS ONLINE

PROFESSOR: GUSTAVO BECACICI ESTEVES VIANNA

[← Voltar](#)

CURSO:	10 B - FISICA - BACHARELADO
VERSÃO:	2008
DISCIPLINA:	INF06582 - Algoritmos Numéricos I
TURMA:	01

INFORMAÇÕES GERAIS
HORÁRIOS
DOCENTES
ESPAÇO FÍSICO
BLOCOS

GERAL

CURSO:	10 B - FISICA - BACHARELADO		
VERSÃO:	2008		
DISCIPLINA: *	INF06582 - Algoritmos Numéricos I		
ANO: *	2010	PERÍODO: *	1º Semestre
TURMA: *	01	CALENDÁRIO: *	ACADÊMICO ▼

VAGAS

OFERTADAS:	40	OCUPADAS:	9	
AUMENTO:		AUMENTADAS:	15	SALDO: 16

DETALHES

TIPO:	TURMA REGULAR ▼	ESCOPO:	SOMENTE ALUNOS DO CURSO ▼
DOCUMENTO:	059170/2009	SITUAÇÃO:	MATRÍCULA ▼

Salvar

* Campos Obrigatórios

[← Voltar](#)

Figura 36 – Tela de alteração de dados de Algoritmos Numéricos 1, do Curso de Física (Bacharelado).

Caso alguma alteração seja feita, cabe ao usuário clicar no botão de “Salvar”, armazenando os novos dados no banco de dados. Enquanto os dados são salvos, uma barra de processamento será mostrada, como na Figura 37 abaixo.

The screenshot displays the 'UFES - OFERTA DE DISCIPLINAS ONLINE' web application. At the top, the user is logged in as 'PROFESSOR: GUSTAVO BECACICI ESTEVES VIANNA'. The interface is divided into sections for course information and a progress bar.

Course Information Section:

CURSO:	10 B - FISICA - BACHARELADO
VERSÃO:	2008
DISCIPLINA:	INF06582 - Algoritmos Numéricos I
TURMA:	01

Below this section, there are tabs for 'INFORMAÇÕES GERAIS' (selected), 'HISTÓRICO', and 'BLOCOS'. A progress bar is visible over the 'INFORMAÇÕES GERAIS' tab.

INFORMAÇÕES GERAIS Section:

GERAL			
CURSO:	10 B - FISICA - BACHARELADO		
VERSÃO:	2008		
DISCIPLINA: *	INF06582 - Algoritmos Numéricos I		
ANO: *	2010	PERÍODO: *	1º Semestre
TURMA: *	01	CALENDÁRIO: *	ACADÊMICO

VAGAS Section:

OFERTADAS:	40	OCUPADAS:	9
AUMENTO:		AUMENTADAS:	-15
		SALDO:	16

Figura 37 – Tela de progresso do salvamento das alterações efetuadas.

Ao concluir tal atualização um alerta de confirmação surgirá (Figura 38), informando que não houve erro, com a mensagem “OS DADOS FORAM ALTERADOS COM SUCESSO!”, e que ele está apto a continuar navegando e a dar continuidade às tarefas que estiverem pendentes.

UFES - OFERTA DE DISCIPLINAS ONLINE

PROFESSOR: GUSTAVO BECACICI ESTEVES VIANNA

[Voltar](#)

CURSO:	10 B - FÍSICA - BACHARELADO
VERSÃO:	2008
DISCIPLINA:	INF06582 - Algoritmos Numéricos I
TURMA:	01

OS DADOS FORAM ALTERADOS COM SUCESSO!

INFORMAÇÕES GERAIS

CURSO: 10 B - FÍSICA - BACHARELADO

VERSÃO: 2008

DISCIPLINA: * INF06582 - Algoritmos Numéricos I

ANO: * 2010 **PERÍODO:** * 1º Semestre

TURMA: * 01 **CALENDÁRIO:** * ACADÊMICO

BLOCOS

VAGAS

OFERTADAS: 40	OCUPADAS: 9
AUMENTO:	AUMENTADAS: -15 SALDO: 16

DETALHES

Figura 38 – Tela de confirmação do salvamento das alterações feitas.

Visando evitar o excessivo número de cliques, fazendo o usuário retornar à página de listagem e fazer novamente a busca de uma nova disciplina, ele pode facilmente procurar uma nova disciplina na página de alteração, como mostrado na Figura 39. As disciplinas listadas pertencem ao Departamento escolhido na primeira busca, neste caso o Departamento de Informática. Como uma mesma disciplina pode conter diversas turmas, o nome de cada uma é seguido pela turma existente, facilitando a navegabilidade no sistema.

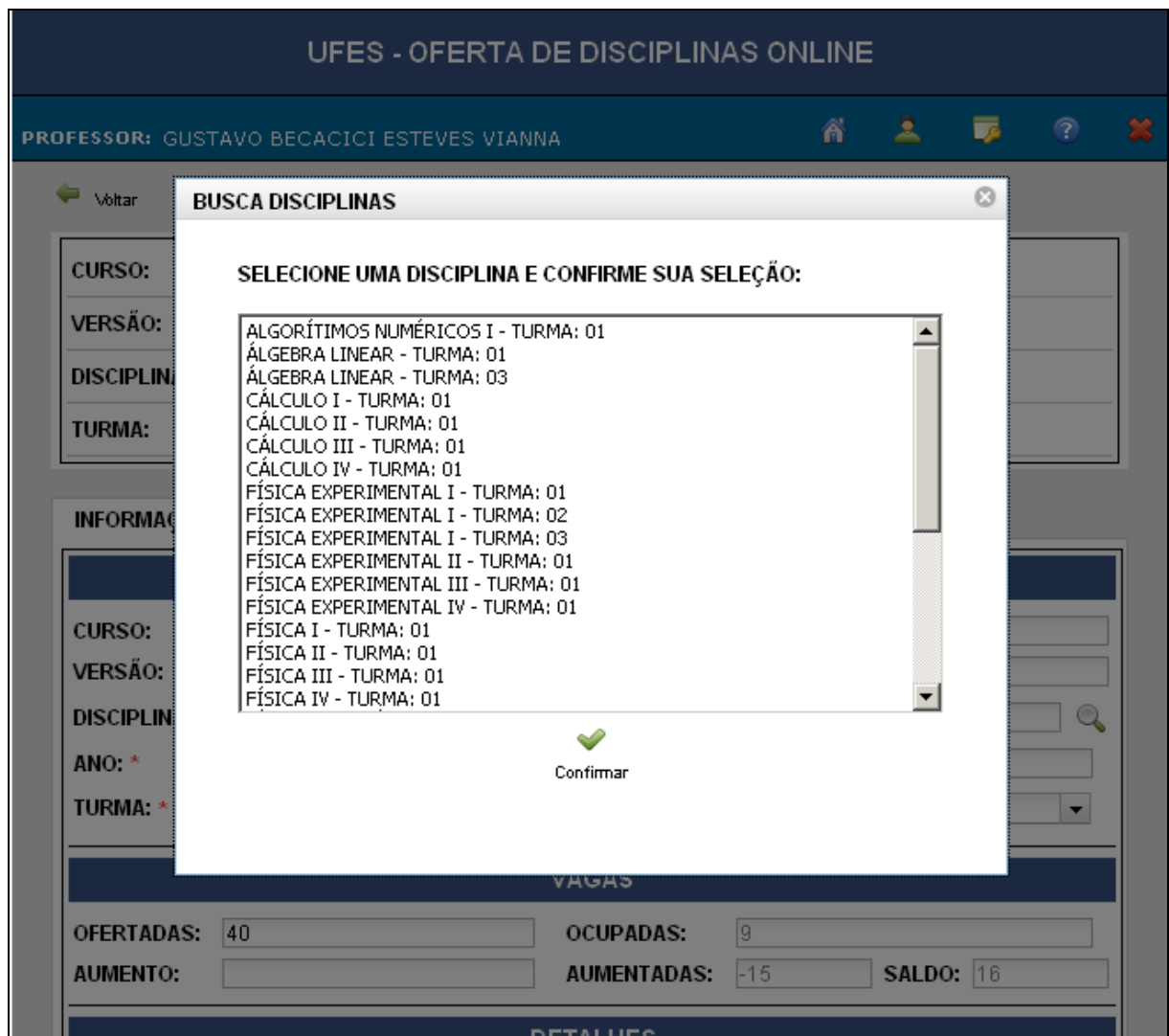


Figura 39 - Tela de escolha de outra disciplina do Departamento de Informática, anteriormente escolhido.

Selecionando a aba de “Horários”, novamente sem redirecionamento, surge a listagem detalhada de horários existentes para a turma 01 de Algoritmos Numéricos 1, cujos dados de Curso, versão de grade curricular, disciplina e código de turma estão fixos no topo da página. Assim são mostrados na Figura 40 o dia da semana, os horários, o tipo de aula e as datas de início e fim dessas turmas, além de opções de alteração, inclusão e exclusão.

UFES - OFERTA DE DISCIPLINAS ONLINE

PROFESSOR: GUSTAVO BECACICI ESTEVES VIANNA

← Voltar

CURSO:	08 - ENGENHARIA MECANICA
VERSÃO:	2008
DISCIPLINA:	INF09270 - Algoritmos Numéricos I
TURMA:	02

Adicionar um novo horário à turma

INFORMAÇÕES GERAIS
HORÁRIOS
DOCENTES
ESPAÇO FÍSICO
BLOCOS

HORÁRIOS							
DIA DA SEMANA	HORÁRIOS	AULA	DATA INÍCIO	DATA FIM	OPÇÕES		
Quarta-feira	14:00:00 às 16:00:00	Aula Teórica	2010-08-09	2010-12-09	Alterar	Excluir	
Sexta-feira	13:00:00 às 15:00:00	Aula Teórica	2010-08-09	2010-12-09	Alterar	Excluir	

← Voltar

Figura 40 – Na mesma tela de alterações de dados, na aba de horários pode-se editar os horários das aulas da turma de Algoritmos Numéricos 1.

Ao clicar em “Alterar” no primeiro horário da lista, uma janela contendo um formulário é exibida (Figura 41). Os campos editáveis são habilitados para os usuários e as diversas opções de mudanças são apresentadas. Além disso, novamente pensando na usabilidade, alguns campos que devem ser preenchidos de acordo com um padrão são automaticamente mascarados pelo sistema, de modo que o usuário não tenha que pensar qual o formato de datas ou de horas que deve ser inserido. Portanto, ao selecionar a data de início, por exemplo, um calendário é visualmente exibido, para que nele sejam selecionadas as datas de início e de fim da turma em questão.

UFES - OFERTA DE DISCIPLINAS ONLINE

PROFESSOR: GUSTAVO BECACICI ESTEVES VIANNA

CURSO: 08 - ENGENHARIA MECANICA

ALTERAR HORÁRIOS

HORÁRIOS			
DIA DA SEMANA: *	QUARTA-FEIRA	TIPO DE AULA: *	AULA TEÓRICA
INTERVALO: *	00:00:00	DURAÇÃO: *	01:00:00
HORA INICIAL: *	14:00:00	HORA TÉMINO: *	16:00
DATA DE INÍCIO: *	09/08/2010	Nº HR/AULA: *	2
DATA DE TÉMINO: *	09/12/2010		

*** Campos Obrigatórios**

agosto

S	T	Q	Q	S	S	D
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

2009 2010 2011

Salvar

Figura 41 – Tela de alteração de um determinado horário.

Há também a opção de inserir um novo horário, ilustrado na listagem na parte superior direita pelo símbolo de adição (+), como evidenciado na Figura 40, que ao ser clicado faz com que apareça em tela uma janela semelhante à figura anterior, porém com todos os campos vazios ou com preenchimento padrão, dando ao operador a liberdade de preenchê-los de acordo com sua conveniência, respeitando as restrições já mencionadas no Capítulo 3, Seção 3.2. Essa janela é mostrada na Figura 42, a seguir.

The screenshot displays the 'UFES - OFERTA DE DISCIPLINAS ONLINE' web application. At the top, the professor's name 'GUSTAVO BECACICI ESTEVES VIANNA' is shown. Below this, a 'CURSO' dropdown menu is set to '10.B - FÍSICA - BACHARELADO'. A modal window titled 'NOVO HORÁRIO' is open, containing a form with the following fields:

HORÁRIOS			
DIA DA SEMANA: *	-	TIPO DE AULA: *	-
INTERVALO: *	-	DURAÇÃO: *	-
HORA INICIAL: *		HORA TÉMINO: *	
DATA DE INÍCIO: *		DATA DE TÉMINO: *	
		Nº HR/AULA: *	1

Below the form fields is a 'Salvar' button. At the bottom of the modal, a red text label reads '* Campos Obrigatórios'.

Figura 42 - Tela de inserção de um novo horário na turma 01 de Algoritmos Numéricos 1.

Quanto à opção de exclusão de um horário listado, ao clicar na tarefa “Excluir”, um alerta é exibido, mostrando a irreversibilidade da ação, Figura 43. Caso tenha certeza, basta clicar em “Confirmar” e o horário será deletado da listagem.

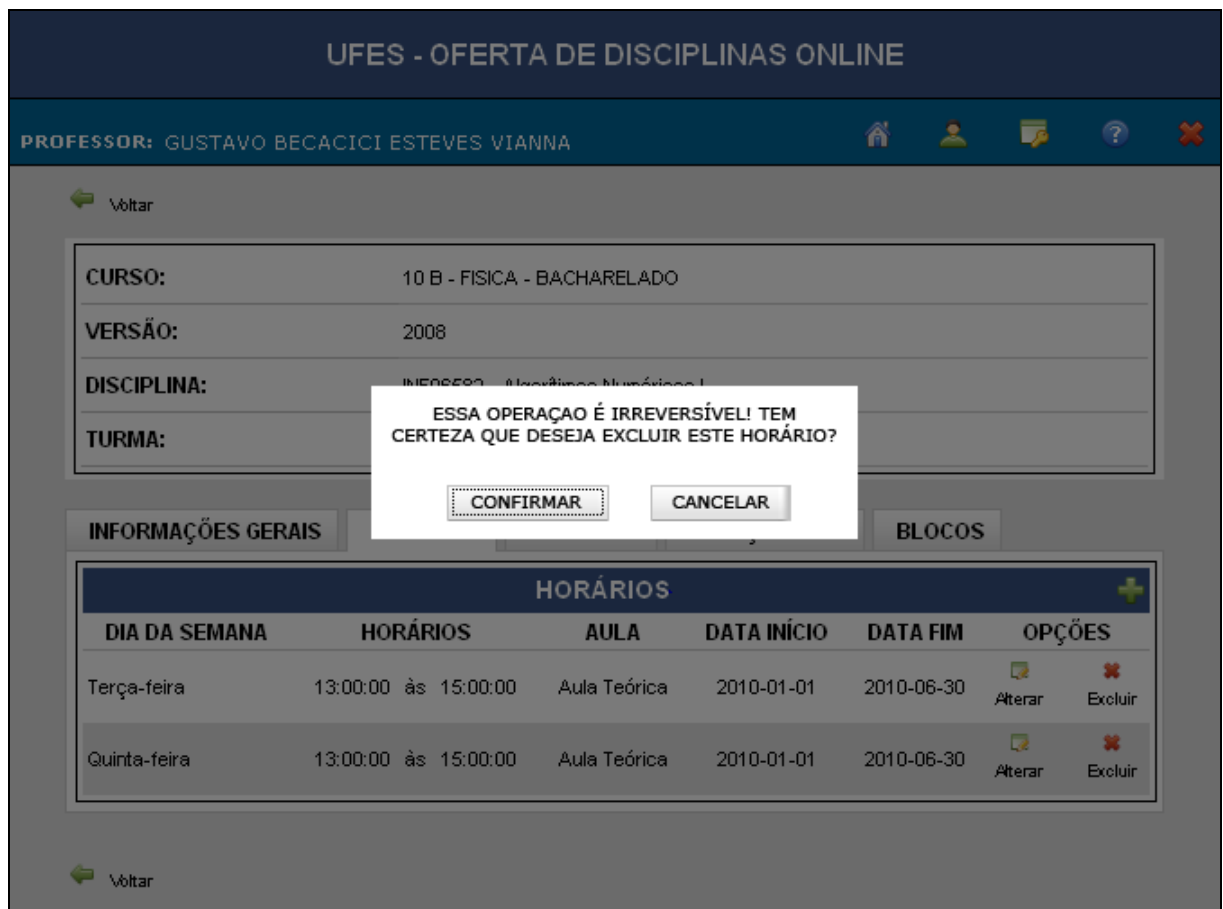


Figura 43 - Confirmação de exclusão de um dado horário da turma de Algoritmos Numéricos 1.

Como foi mencionado anteriormente, o sistema verifica, em tempo de execução, possíveis conflitos e concorrências de recursos como: horários, espaço-físico e docentes. No caso de haver algum choque desse tipo ou de qualquer outro, um alerta será apresentado, explicando o tipo de conflito e o que deve ser feito para saná-lo. Na Figura 44, tem-se um exemplo de choque de horário para um mesmo docente e o alerta que surge para instruir o usuário qual atitude tomar.

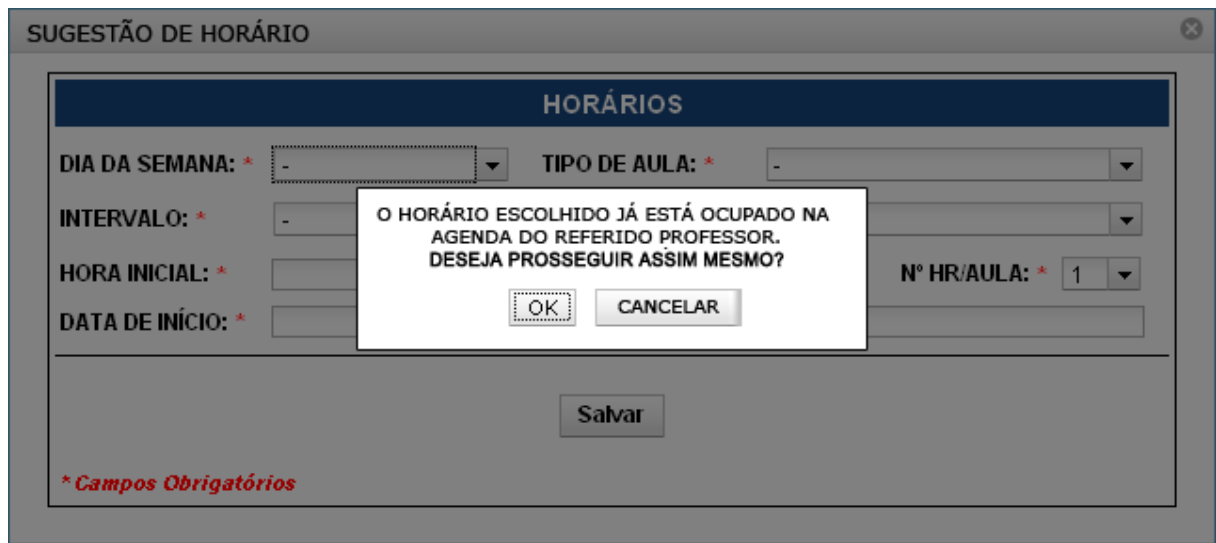


Figura 44 - Alerta de conflito de horários, sugerindo que um novo horário seja escolhido caso lhe seja conveniente, mas dando a opção de prosseguir mesmo assim.

As figuras mostradas acima é um exemplo de telas que o sistema exhibe. Só foi abordado as funcionalidades da aba de horários, mas as abas de docentes, espaço-físico e blocos são análogas às ilustradas. Listagens de professores, salas e dos blocos das turmas são, da mesma forma que a lista de horários, passíveis de visualização, alteração, inclusão e exclusão. Os alertas de confirmação e de conflitos também são mostrados.

A opção de tramitação também segue o mesmo padrão, onde precisa ser definido o setor do qual o documento será tramitado (por padrão o setor de lotação do usuário logado) e o destino da tramitação, bem como um campo para observações, caso seja necessário descrever ou explicar algo.

Sendo assim, é possível dizer que as deficiências de interface existentes no SIE foram sanadas, fazendo com que o uso do sistema seja mais prazeroso e agradável ao usuário, com poucas telas, informações relevantes são dadas, uma distribuição uniforme dos elementos visuais nas páginas e uma quantidade extremamente menor de decisões cabem a quem usa a aplicação, automatizando grande parte das funcionalidades.

4.1.7.3 A lógica de negócios

Levando em consideração a utilização do modelo MVC, de três camadas, a camada de lógica de negócios, em questão, se encontra no servidor de aplicação. É a camada principal do projeto.

São rotinas desenvolvidas para realizarem as entradas de dados, geração de relatórios, preenchimento de formulários e, mais especificamente, todo o processamento dessas informações que ocorrem por trás da aplicação visível ao cliente.

O primeiro ponto a ser destacado é o controle de fluxo, estimulado pelo modelo MVC e aplicado de forma excelente no *framework* utilizado. Todas as requisições, obrigatoriamente passam pelo chamado *bootstrap*, citado anteriormente, que, juntamente às configurações feitas nos arquivos “.htaccess”, fecham o fluxo de informações da aplicação. Neste arquivo, *bootstrap*, definiu-se o tratamento de exceções, exibindo ao usuário erros amigáveis e, previamente, planejados. Foi feito também o carregamento de todas as classes que serão utilizadas pelo sistema, bem como o arquivo base de acesso ao banco de dados que é indicado. E, por fim, tem-se a chamada do método que será responsável pela recepção, interpretação e processamento das requisições, o chamado *dispatch*.

O usuário, só entra no sistema se estiver devidamente validado e com um perfil associado. Então, ele será redirecionado para a página inicial, de acordo com seus privilégios.

A segurança de dados é uma parte crucial da implementação de todo sistema. Portanto, alguns cuidados foram tomados, como a criptografia de dados sensíveis, por exemplo, que foi tida como premissa. Além disso, foi aplicada uma série de políticas de acesso, bloqueando a visualização de certas páginas que não tenham relação com o perfil do referido operador e controlando as alterações que ele possa fazer. Ademais, foram implementadas as rotas, resultando nas famosas URLs amigáveis, que adicionam uma segurança ao sistema, uma vez que mascara a identificação do controlador, da ação e até mesmo dos parâmetros enviados, e ainda aumentam a navegabilidade.

O desenvolvimento das funcionalidades destacadas na fase de coleta de requisitos (visualização, alteração, inserção, exclusão de dados, tramitação de documentos e cópia de ofertas pré-existentes), também foram implementados nessa camada. Todas são acompanhadas de tratamento de exceções e avisos amigáveis de erros ou falhas resultantes do mau uso ou do mau funcionamento do sistema. Abaixo estão listadas uma série de exemplos dessas funcionalidades e como são aplicáveis no sistema:

- LISTAR TURMAS POR ANO, PERÍODO E DEPARTAMENTO

Essa função faz uma simples consulta ao banco, através da chave única do Departamento, usando com parâmetros o ano e o semestre desejado. Cada campo é validado de acordo com seu tipo antes de serem submetidos a real busca dos

dados. O campo ano, por exemplo, por padrão tem 4 dígitos e é numérico, caso o usuário tente digitar alguma letra, um aviso é mostrado para que ele corrija a informação fornecida antes de prosseguir a consulta. Após essa verificação, em caso positivo, a consulta retorna uma lista de turmas já ofertadas.

- **LISTAR TURMAS POR ANO, PERÍODO E CURSO**

Da mesma forma que a função anterior, essa também não passa de uma consulta à base de dados, com parâmetros diferentes, ao invés da chave única de Departamento, é fornecida a chave única de Curso, juntamente com o ano e com o semestre solicitado. Assim, ocorre também a validação e, se estiver correto, há um retorno da listagem das turmas que já foram ofertadas. É um passo obrigatório para as funções que se seguem.

- **VISUALIZAR INFORMAÇÕES GERAIS**

Após a listagem das turmas por ano, período e Departamento/Curso, é que essa opção é fornecida ao usuário. Através dos dados de ano e semestre provenientes da sessão e da disciplina escolhida na listagem, essa tarefa consiste em uma consulta mais complexa, uma vez que solicita uma gama de informações muito maior do que as anteriores, e tem como parâmetro a chave única de cada turma, concatenando dados como versão de grade curricular, curso ao qual pertence, disciplina a qual está vinculada, código da turma, número de vagas ofertadas, disponíveis, ocupadas e aumentadas, calendário ao qual está inserida e outras informações que geram uma visualização mais abrangente da turma selecionada. Nessa implementação estão incluídas a exibição dos horários, dos docentes que irão ministrar as aulas, do espaço-físico alocado e do bloco ao qual a turma pertence. E é aqui que são disponibilizadas as abas para acessar as funcionalidades abaixo.

- **ALTERAR HORÁRIO**

Aqui é onde acontece uma modificação de dados no banco. Na opção de alteração de dados, é fornecida, na aba de horários, a possibilidade de atualização ou edição de algum dado que não esteja correto. Uma classe é responsável pela exibição do

formulário de alteração de horário. E como há entrada de dados, também há uma validação, de modo que garante a integridade da informação. A maioria dos campos habilitados para alteração possui uma lista de alternativas, que condizem com o tipo obrigatório daqueles dados. Assim, ao remeter o formulário para o processamento, é buscado o registro, também a partir da chave única da turma enviada à classe responsável por essa tarefa, e esta linha do banco é alterada de acordo com o que foi pedido. O retorno é a listagem dos horários com as atualizações efetivadas.

- **ADICIONAR HORÁRIO**

Na inserção de uma nova informação, do mesmo modo que na função de alterar, uma modificação nos registros da base de dados. Após preencher os campos do formulário processado, o usuário remete-o para a captura das informações enviadas e, com isso, novas linhas são criadas e há uma vinculação da tabela de horários das turmas à turma em questão, e, caso já haja um professor alocado, a turma é inserida na agenda desse professor, bem como na tabela de turmas. Feito isso, o *software* retorna uma lista com o novo horário adicionado.

- **EXCLUIR HORÁRIO**

Faz uma remoção do horário por um índice único que vincula a turma ao referido horário através de uma solicitação de deleção feita ao banco. Tem como retorno uma nova listagem de horários sem o horário que foi removido.

- **SOLICITAR TURMA**

Este é o passo mais esperado do sistema, uma vez que trabalha com tramitação de documentos entre setores da instituição. O fluxo da solicitação de uma turma, mostrado anteriormente na Figura 1 é alinhado à execução da aplicação na lógica de negócios, que é o cérebro do *software*. Ao iniciar essa tarefa, uma nova linha na tabela de documentos é criada, dando partida no processo. Após isso, uma linha na tabela de tramitações também é inserida. Então os campos que dizem respeito à solicitação de turma, como assunto, setor de origem do documento, setor de destino e outros dados são alterados ou incluídos no procedimento. E, por

fim, a turma tem sua situação alterada para “Solicitada”. É nessa funcionalidade que ocorre uma integração real entre tudo o que acontece no sistema, registrando o que é feito e reportando aos setores responsáveis cada parte do fluxo. O retorno é a confirmação de alteração desta situação da turma, que, dependendo do status, não estará mais visível ao seu usuário, uma vez que foi remetida a outro setor.

Vale lembrar também da existência de restrições do sistema, que são todas efetivamente aplicadas nessa camada, porém não impedem que exceções ocorram, mas com o consentimento dos operadores. A seguir alguns exemplos de restrições de aplicação que são implementadas aqui:

- Restrições de unidades, onde um professor alocado em um Departamento não está apto a lecionar classes em outro.
- Restrição de docentes, onde dois ou mais professores estão lecionando uma mesma disciplina.
- Restrição de horários, onde um professor não leciona duas ou mais disciplinas em um mesmo horário.
- Restrição de espaço físico, onde duas ou mais matérias não serão ministradas em uma mesma sala de aula.
- Restrição de encargo didático, onde um professor não pode extrapolar a quantidade máxima permitida para dar aulas.

Logo, o ponto chave do sistema é a lógica de negócios, pois é onde todo o funcionamento do projeto é definido, as restrições, quais serão os processamentos necessários, onde os dados deverão ser validados, quais serão os redirecionamentos e muitas outras respostas esperadas, a fim de tratar as possíveis falhas causadas pelo usuário, trazendo a conformidade entre o que é permitido e o que a aplicação faz.

4.1.7.4 Os modelos

Conforme visto, a aplicação desenvolvida irá acessar dados de um banco de dados relacional, porém é orientada a objetos. Pensando em manter a metodologia de programação (OO), é necessário transformar as tabelas em objetos. Isto visa desacoplar o sistema do SGBD, para torná-la independente, facilitando possíveis trocas da base de dados.

Essa é a finalidade do mapeamento objeto-relacional, que consiste em criar uma interface OO para um banco de dados relacional.

Existem inúmeras implementações da técnica supracitada, que geralmente utilizam padrões de projetos como a *ActiveRecord*, por exemplo. A *ActiveRecord* preconiza que uma tabela do banco é vista como uma classe, enquanto cada registro será um objeto.

O ZF faz uso deste mesmo padrão de projeto. Foi assim que os modelos foram criados e todas as operações feitas sobre os objetos serão replicadas na base de dados como operação sobre as linhas da tabela.

4.1.7.5 MVC do novo sistema

A Figura 45 exemplifica o trabalho do modelo MVC usado nesse projeto.

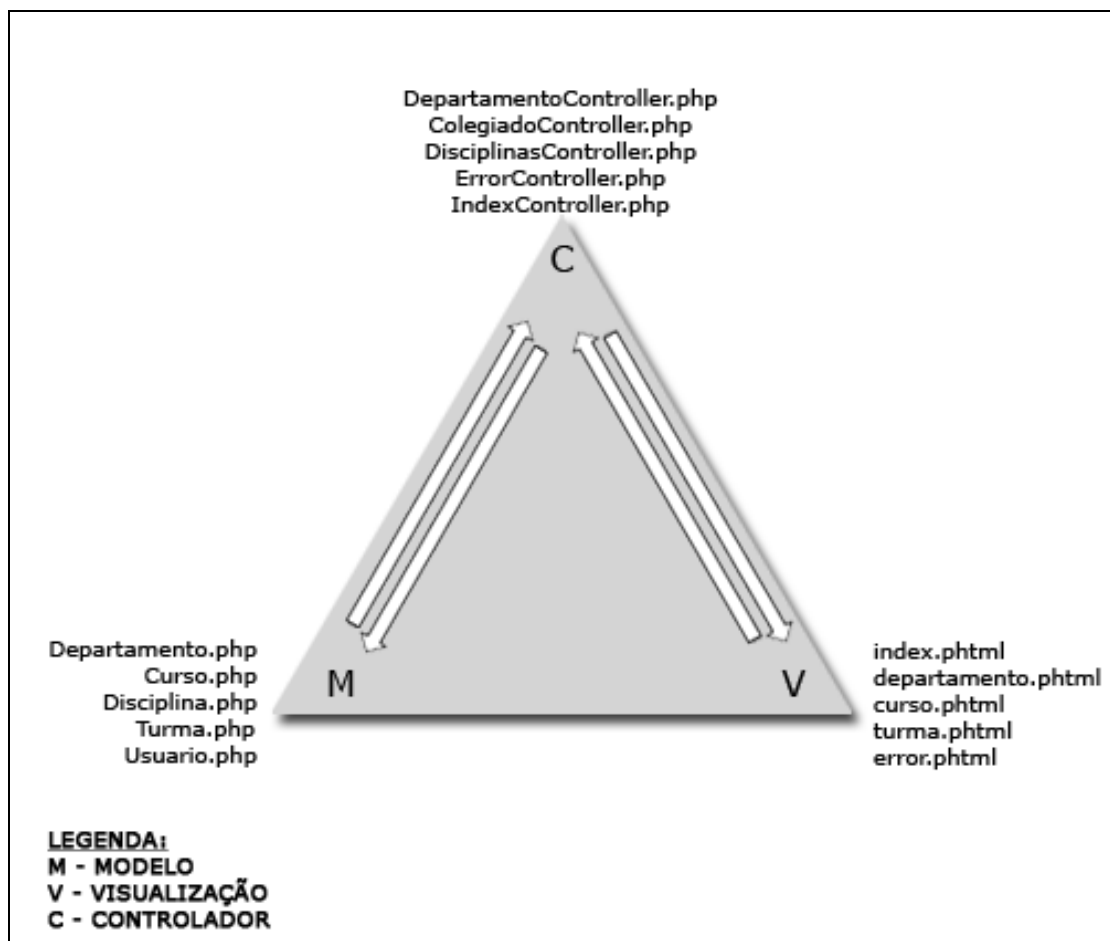


Figura 45 - MVC do novo sistema de oferta de disciplinas.

Ocorre uma interação direta entre o controle e o modelo e a visualização, porém, o modelo nunca é acessado pela interface, o que é uma questão importante, pois se trata da segurança do programa.

Para exemplificar, temos o “DepartamentoController”, que tem um modelo, já com seu objeto mapeado em “Departamento.php”, onde todas as consultas ao banco são estruturadas e feitas, e, ao receber o retorno disso, trabalha as informações, seja em formulários seja em uma simples tabela, repassa para a sua visualização, em “departamento.phtml”.

Outra forma de interação é no “ColegiadoController”, que remete ao modelo a incumbência de acessar o banco e fazer suas consultas, para então tratar os dados e enviá-los para a interface da forma que devem ser exibidos.

É dessa forma que é garantido um único fluxo de dados e, conseqüentemente uma segurança e uma maior organização nos procedimentos.

4.1.7.6 Os testes

Miller relaciona diretamente a fase de testes com a qualidade do serviço produzido. Afirmar que “a motivação subjacente ao teste de programa é afirmar a qualidade do *software* com métodos que podem ser aplicados econômica e efetivamente tanto a sistemas de grande porte quanto de pequeno porte” [17].

A principal ferramenta utilizada foi o PHPUnit [18], que é uma comumente usada para testar *frameworks* e é também um *framework* por si só. Um manual de instalação do PHPUnit se encontra no ANEXO 4.

A FIGURA 46, abaixo, exemplifica a interface usada juntamente com a IDE, como uma extensão do programa.

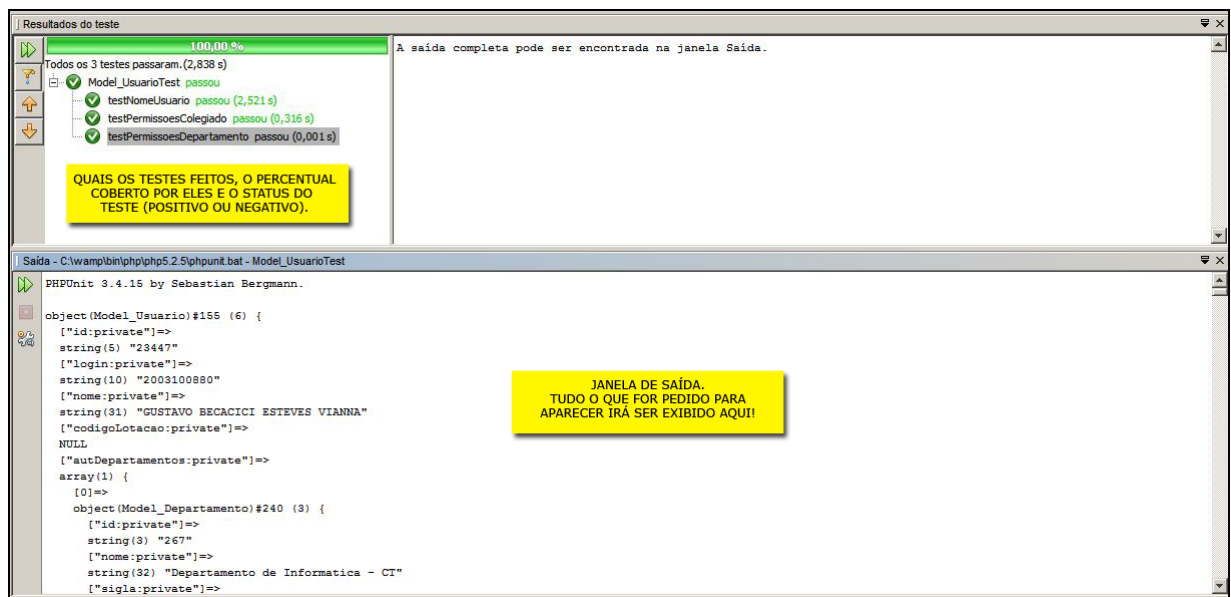


Figura 46 - PHPUnit instalado no Netbeans.

Além de mostrar uma barra que determina o percentual de abrangência dos testes feito na referida classe e uma listagem de classes e métodos testados, bem como o tempo de processamento ao realizar o teste, o software tem uma área de saída de dados, onde o desenvolvedor pode imprimir em tela o valor de diversas variáveis para conferir se estão em conformidade com o planejado.

Cada classe é testada separadamente do código fonte da aplicação. Os testes podem ser feitos em qualquer momento, sem que interfira na produção do programa e são automatizados, sendo executados continuamente durante o ciclo de desenvolvimento. Eles se baseiam em afirmações, onde definimos os valores esperados e verificamos se o resultado procede. Funções que alimentam os testes com valores randômicos podem ser criadas, aumentando assim a assertividade do resultado. São testadas inclusive as exceções lançadas pela aplicação, validando cada caso.

As funcionalidades abaixo foram testadas e devidamente verificadas, quanto ao resultado da resposta e ao tempo de processamento, e todas foram aprovadas nos testes feitos:

- AUTENTICAÇÃO DE USUÁRIO;
- CONTROLE DE ACESSO;
- EXIBIÇÃO, ALTERAÇÃO, EXCLUSÃO E INCLUSÃO DE DADOS;
- TRAMITAÇÃO DE DOCUMENTOS;
- CÓPIA DA OFERTA ANTERIOR DE DISCIPLINAS.

Sem dúvida a realização de testes é uma prática muito bem vista, que produz resultados em curto, médio e longo prazo. Evita surpresas na execução dos sistemas; reduz o tempo gasto com *debug*; auxilia em um possível processo de reengenharia e adaptação do *software*; aumenta a qualidade do produto final; e muitos outros benefícios que são computados no fim do processo de desenvolvimento. Em suma, e concordando com a afirmação inicial dessa Seção, é extremamente válido e necessário a realização de testes antes da entrega do projeto.

5 CONCLUSÃO E TRABALHOS FUTUROS

5.1 Conclusão

Como foi dito ao longo desse trabalho, fazer do SIE uma grande aplicação web, com inúmeros módulos é desejo da instituição. O passo inicial foi dado e com o resultado final em produção vemos que o objetivo foi alcançado. A metodologia utilizada surtiu o efeito desejado, produzindo uma aplicação de alta qualidade, segurança e escalabilidade. Manteve-se um padrão de projeto simples, que pode ser seguido sem muito esforço.

A implementação, apesar da complexidade do aprendizado, obteve êxito. O módulo desenvolvido supera as expectativas e atende a demanda com segurança (controle de acesso, autenticação unificada com outros sistemas, restrições de usuários, tratamento de erros) e com alta velocidade de processamento. A interface “limpa” e intuitiva dá ao usuário o poder de efetuar as requisições de dados de forma simplificada e rápida.

Vale ressaltar que, por se tratar de um problema da classe NP-completo¹¹, a solução proposta é uma tentativa que se mostrou satisfatória e conseguiu suprir a demanda existente. Portanto a avaliação do resultado é positiva e, após os testes feitos, se mostrou estável e confiável.

Sumarizando, espera-se que o trabalho realizado tenha sido de grande valia para a universidade e para o meio docente, que é o público alvo desta monografia.

5.2 Trabalhos futuros

A partir do estudo feito e do resultado da implementação do referido módulo, um leque de opções de trabalhos futuros se abre. A forte tendência, mencionada na introdução, de migração de sistemas do tipo *desktop* para a web, incentiva uma completa transformação do SIE para o novo modelo utilizado, a longo prazo.

Uma demanda da UFES é o cadastro e alocação de espaço físico. O banco de dados atual não contém as salas de aula, a localização física de cada setor, órgão, Departamento e Colegiado. O cadastramento desses, possibilitaria a alocação de cada sala a uma disciplina em algum horário predeterminado e, assim, resultaria em uma maior organização espacial tanto para a instituição como para os estudantes.

¹¹ Da língua inglesa *nondeterministic polynomial time* [20].

No entanto, em curto prazo, existem necessidades mais emergenciais a serem sanadas. Um exemplo disso é o planejamento da oferta de disciplinas, que atualmente não é eficaz e é feito através de planilhas ou até mesmo de documentos manuscritos, dependendo do gosto de cada órgão/departamento/colegiado. A automatização desse processo traria um retorno imenso para a instituição.

Portanto, fica como sugestão de trabalhos futuros a extensão do módulo para que seja possível um planejamento da oferta de turmas, fazendo com que a digitação da oferta esteja o mais próximo possível do esperado, evitando problemas para a comunidade acadêmica e especialmente para os responsáveis por essa árdua tarefa a ser feita todos os períodos.

ANEXOS

ANEXO 1 – INSTRUÇÃO NORMATIVA PARA REGULAMENTAÇÃO DOS DADOS DE OFERTA DE TURMAS

Instrução Normativa xxx/xxx

Estabelece normas de preenchimento de informações de docentes em turmas ofertadas para os cursos de graduação e pós-graduação

Devido a exigência dos Órgãos Controladores do MEC, entre eles, Pingfes, Censo Educacional, Enade e E-Mec), e dos processos decorrentes da avaliação institucional é necessário que o Sistema Acadêmico da UFES contenha a informação correta da carga horária de trabalho dos docentes. Esta mesma informação será usada para o Sistema Informatizado de Avaliação Docente, no qual os alunos matriculados em uma determinada turma serão habilitados a avaliar de acordo com as normas vigentes o(s) docente(s) responsável(is).

No caso de informações que não são atendidas pelo Sistema Acadêmico, a Administração Central disponibilizará oportunamente à Comunidade Universitária, um sistema de coleta de informações sobre as cargas horárias de atividades administrativas, de pesquisa e de extensão.

Para a inserção dos dados Acadêmicos, os departamentos deverão informar no Sistema Acadêmico - SIE, para cada turma as disciplinas sob a sua responsabilidade, a carga horária dos docentes envolvidos.

Para uniformizar o fornecimento destas informações, no SIE, deverão obedecer as seguintes normas:

- 1) Todas as turmas deverão possuir horário definido
- 2) Todas as turmas deverão ter registrado(s) o(s) docente(s) responsável
- 3) Preencher o encargo didático com a carga horária semestral, devendo ser entendido por encargo didático a carga horária efetiva do docente em sala de aula.
- 4) Nos casos de turmas de disciplinas com códigos diferentes ministradas simultaneamente:
 - a. Exemplo: casos de turmas de disciplinas de currículo novo e anterior

- b. Preencher o encargo didático, SOMENTE na versão corrente (mais atual) da disciplina e preencher com ZERO o campo de encargo didático das versões anteriores
- 5) Para os casos de professores ministrando concomitantemente a disciplina, ou seja, docentes efetivamente presentes na mesma sala de aula.
- a. Preencher o encargo didático efetivo dos respectivos docentes
 - b. No campo observações da turma, deverá constar a justificativa da necessidade de mais de um docente
- 6) Para os casos de subdivisão de turmas
- a. Exemplo: disciplinas com laboratório com restrição de vagas e aulas teóricas em comum
 - b. Preencher na primeira turma o encargo didático do docente responsável pela parte comum, e com encargo didático zero nas demais turmas.
 - c. Preencher todas as turmas com a carga horária efetiva dos respectivos docentes responsáveis pela parte subdividida
- 7) Alertamos que serão auditadas as informações fornecidas para identificação de inconsistências e erros de preenchimento.
- 8) Nos casos de estágio supervisionado deverá ser lançada uma carga de 15 horas para cada grupo de 04 alunos quando a supervisão for presencial ou cada grupo de 08 alunos nas demais formas de supervisão.
- 9) Nos casos de disciplinas de graduação relativas a orientações de trabalho, projetos ou monografias de final de Curso deverá ser lançada uma carga horária de 15 horas para cada aluno.

EXEMPLOS

Exemplo 1

A disciplina Cálculo I com carga horária de 90 horas é oferecida com dois códigos diferentes (MAT1000 para as versões antigas e MAT3000 para as novas versões das grades curriculares), mas as turmas 01 do código MAT1000 e a turma 03 do código MAT3000 são ministradas na mesma sala de aula e no mesmo horário pelo professor X. Neste caso, somente deve ser preenchida a carga horária da turma 03 da disciplina com código MAT3000.

Exemplo 2

A turma 01 da disciplina Ambulatório com carga horária de 180 horas é ministrada por dois professores (W e Z), ambos permanecendo no recinto durante todo o horário previsto. Neste caso, deve ser preenchida a carga horária de 180 para o professor W e depois, novamente a carga horária de 180 horas para o professor Z.

Exemplo 3

Disciplinas com subdivisão de turmas.

A disciplina Sistemas Digitais tem uma carga horária de 60 horas, com 45 horas de aulas teóricas e 15 horas de laboratório. As aulas teóricas são ministradas para turmas de 40 alunos e as aulas de laboratório para turmas de 10 alunos. Estão registradas então 04 turmas (01, 02, 03, 04) com 10 alunos cada. O professor A é responsável pelas aulas teóricas e uma turma de laboratório (Turma 04) e o professor B pelas demais turmas de laboratório. Neste caso na turma 01 deve ser preenchida a carga horária de 45 horas para o professor A e uma carga horária de 15 horas para o professor B. Nas demais turmas somente devem ser preenchidas as cargas horárias de laboratório (15 horas); para o professor B nas turmas 02 e 03; para o professor A na turma 04.

ANEXO 2 – MANUAL DE OFERTA DE DISCIPLINAS.

Este manual será incluído no CD que irá em anexo à este trabalho.

ANEXO 3 – MANUAL DO EMC

MANUAL DE INSTALAÇÃO DO EMC

Introdução

O Editor de Matriz Curricular (EMC) oferece uma interface amigável para que os usuários dos colegiados e departamentos da UFES possam controlar a oferta de disciplinas aos alunos de maneira mais ágil e prática.

É um sistema web desenvolvido em PHP, utiliza o *framework* Zend, faz uso do banco de dados DB2 do SIE para operar de maneira integrada com os sistemas existentes, a autenticação é feita através de um *web-service* desenvolvido em JAVA.

Requisitos

Os requisitos para instalação da aplicação no servidor são:

- *Client* do banco de dados DB2
- Apache 2.2.15
- PHP 5.2.5
- Apache Tomcat 6.0

Instalação do Tomcat

Para instalar o servidor Tomcat siga os passos a seguir:

1. Obtenha o instalador do Tomcat 6.0
2. Execute o instalador.
3. Avance até a seguinte tela:

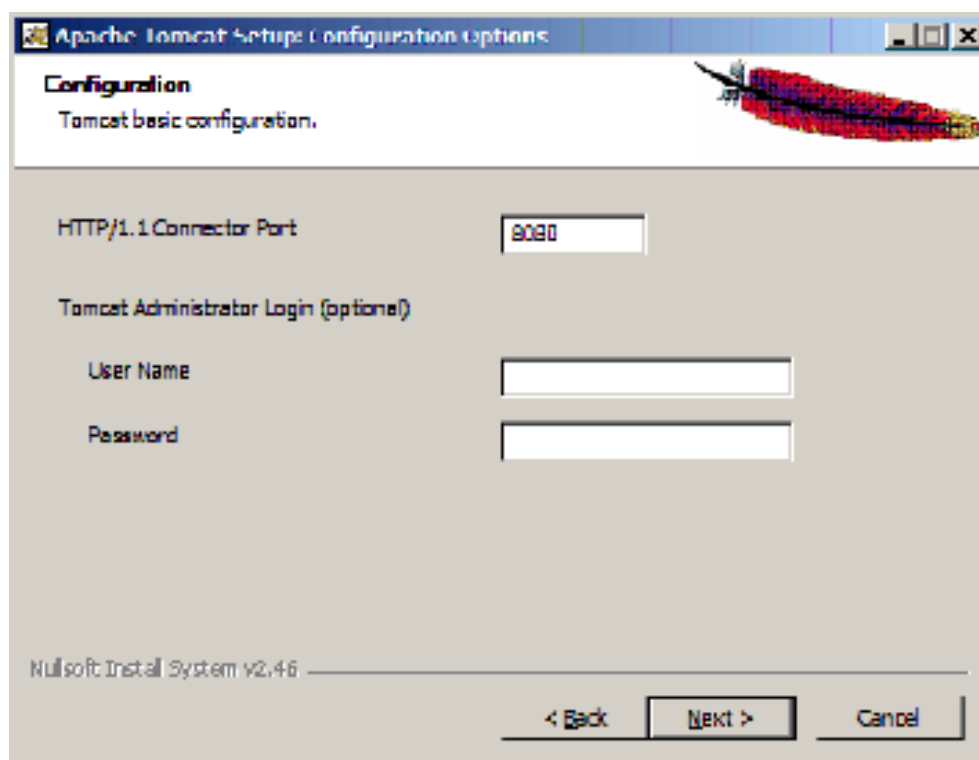


Figura 47 - Configuração do Apache Tomcat.

4. Informe a porta que o servidor usará e um *login* e senha para o administrador do servidor. Lembre-se dessas configurações, pois serão necessárias posteriormente.
5. Inicie o servidor.

Deploy do web-service de autenticação

Após a instalação do servidor Tomcat, é necessário implantar o *web-service* de autenticação. Para isso será necessário o arquivo “cripto.war” que é o *web-service*. Siga os passos a seguir para implantá-lo:

1. Caso não esteja sendo executado, iniciar o monitor do Tomcat. Normalmente, pode ser encontrado em Iniciar > Todos os programas > Apache Tomcat 6.0 > Monitor Tomcat.
2. Se o serviço não estiver iniciado, execute-o clicando com o botão direito no seu ícone perto do relógio do sistema e em seguida clique em “Start Service”.
3. Abra um navegador e digite o seguinte endereço: “http://localhost:8080”. Sendo “localhost” o endereço do computador onde o servidor foi instalado e 8080 a porta escolhida na instalação do Tomcat.
4. Será exibida a página inicial do Tomcat.

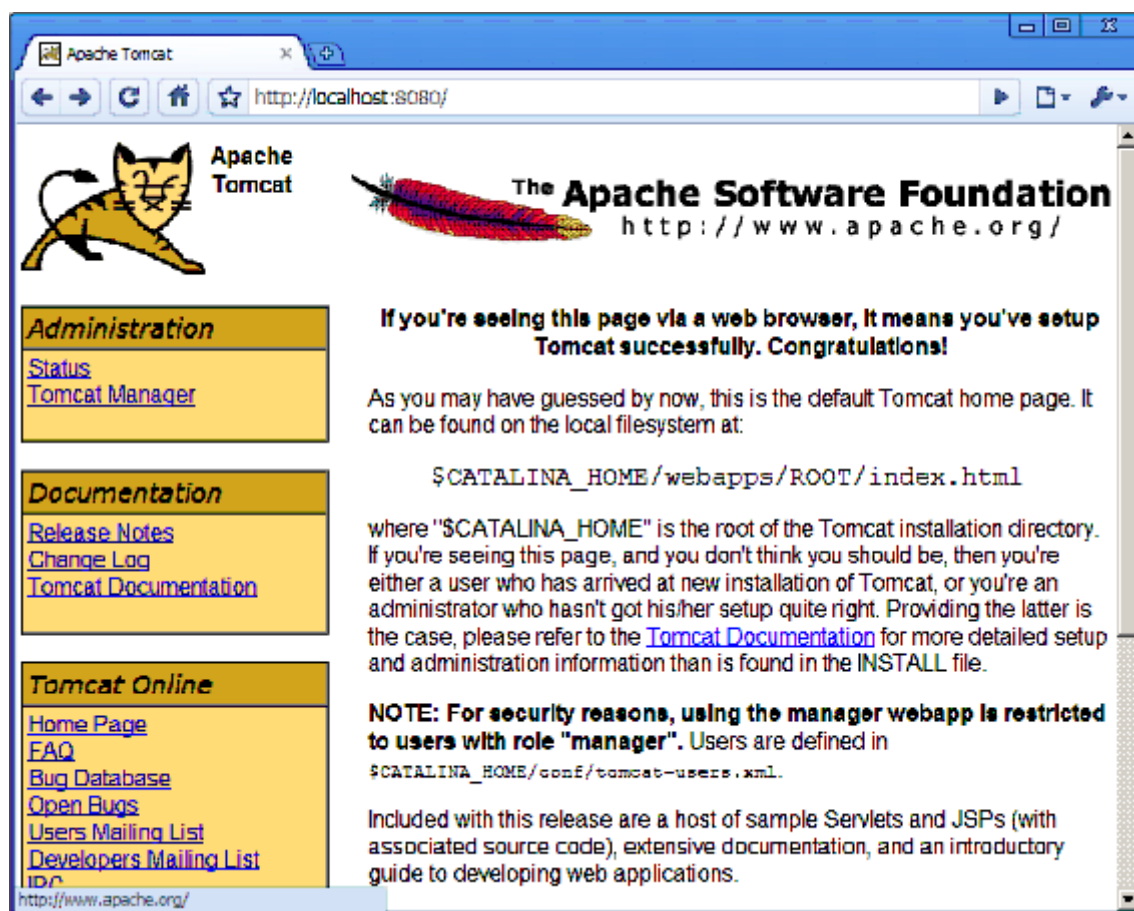


Figura 48 - Página inicial do Apache Tomcat.

5. Clique no link “Tomcat Manager”. Será exibida a página de gerenciamento do servidor.

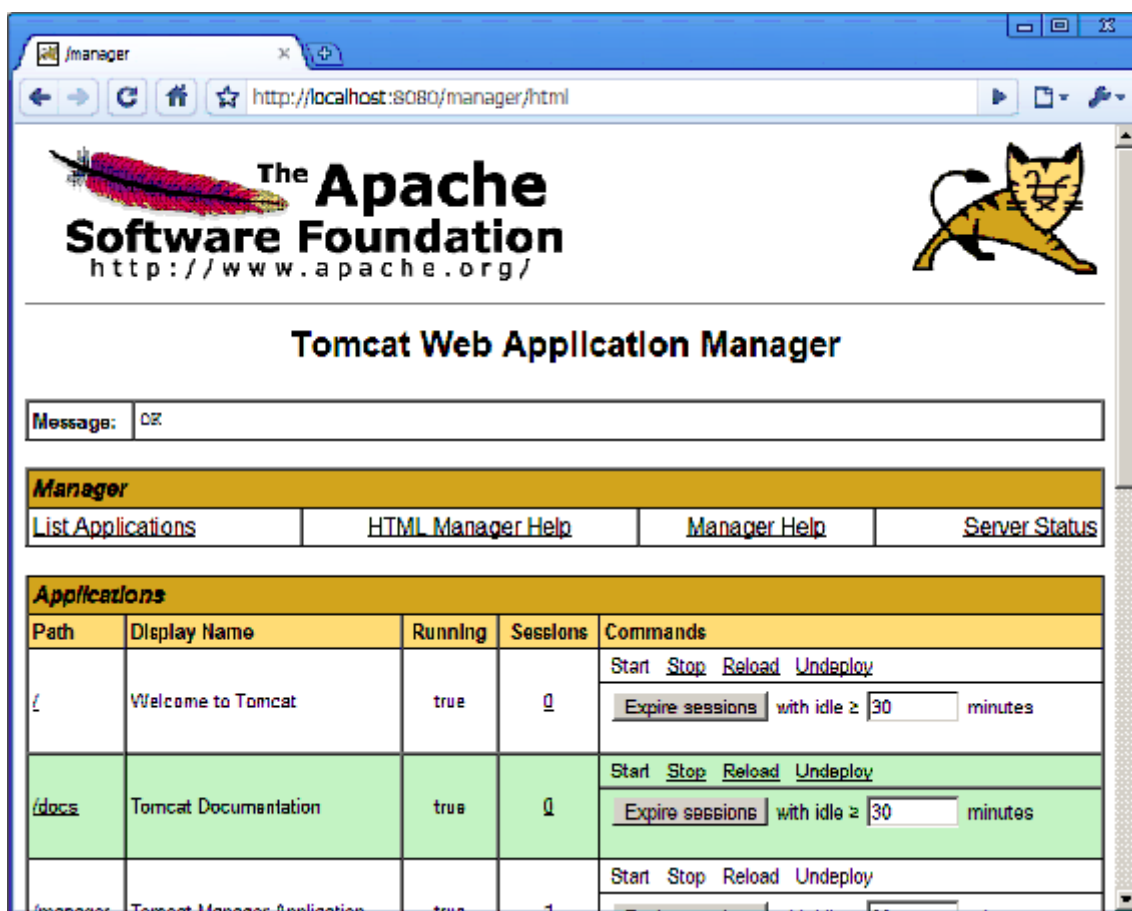


Figura 49 - Apache Tomcat Manager.

6. Encontre a sessão “Deploy” e dentro dela a “WAR file to deploy”.

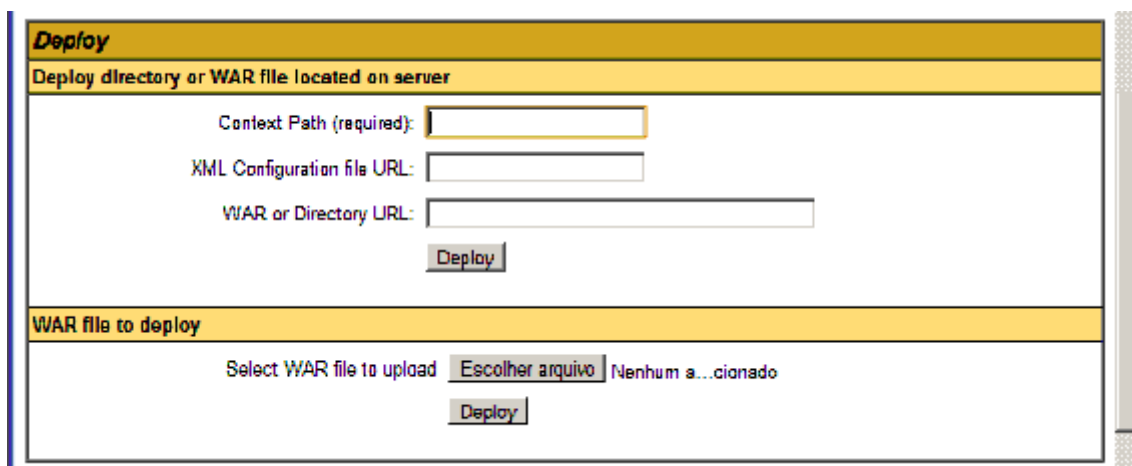


Figura 50 - Tela de implantação de um novo projeto.

7. Clique no botão “Escolher arquivo” e selecione o arquivo WAR com o web-service.

8. Clique no botão “Deploy”.

Applications				
Path	Display Name	Running	Sessions	Commands
/	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ <input type="text" value="30"/> minutes
/Crypto		true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ <input type="text" value="30"/> minutes
/docs	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ <input type="text" value="30"/> minutes
/manager	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ <input type="text" value="30"/> minutes

Figura 51 - Listagem dos projetos existentes no servidor.

9. Agora ele aparecerá listado na sessão “Applications”.

Instalação do *Client* do Banco de Dados

Para conexão com o banco de dados é necessário que o computador tenha o cliente de conexão com o DB2, o banco utilizado pelo SIE.

Caso não tenha instalado, obtenha o programa instalador com o suporte e execute os seguintes passos:

1. Execute o instalador.



Figura 52 - Tela de instalação do Cliente DB2.

2. Selecione a opção instalar um produto.

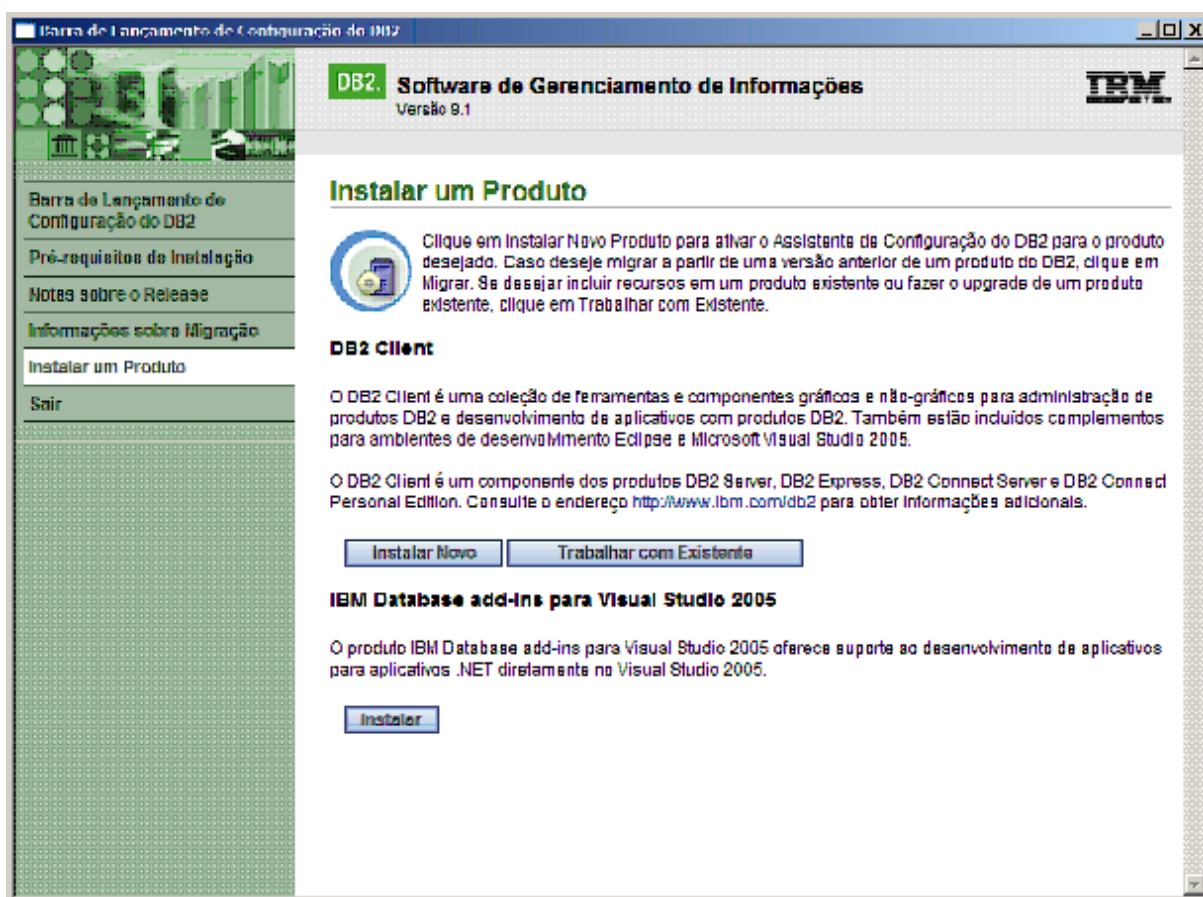


Figura 53 - Segunda tela de instalação do Cliente DB2.

3. Clique no botão “Instalar Novo”.
4. Avance o assistente de instalação com as configurações padrão até o fim.

Configuração do Apache

Para o bom funcionamento do sistema é necessário que o Apache seja configurado para utilizar arquivos “.htaccess”. Para isso, abra o arquivo de configuração (httpd.conf) e localize a seguinte linha:

```
#LoadModule rewrite_module modules/mod_rewrite.so
```

Substitua por:

```
LoadModule rewrite_module modules/mod_rewrite.so
```

Em seguida, reinicie o servidor.

Configuração do PHP

Além da configuração anterior do apache, é preciso que alguns módulos sejam habilitados e alguns recursos sejam ativados no PHP.

Das configurações, os seguintes recursos devem ser ativados:

- `magic_quotes_gpc`
- `magic_quotes_runtime`
- `magic_quotes_sybase`
- `short_open_tag`

Para ativá-los localize as linhas seguintes linhas:

```
short_open_tag = Off  
magic_quotes_gpc = Off  
magic_quotes_runtime = Off  
magic_quotes_sybase = Off
```

E substitua por estas, respectivamente:

```
short_open_tag = On  
magic_quotes_gpc = On  
magic_quotes_runtime = On  
magic_quotes_sybase = On
```

Os seguintes módulos devem ser ativados:

- `php_dba`
- `php_dbase`
- `php_mcrypt`
- `php_mhash`
- `php_mysql`
- `php_mssql`

- php_openssl
- php_pdo_firebird
- php_pdo_mssql
- php_pdo_odbc
- php_pdo_pgsql
- php_pdo_sqlite
- php_pgsql
- php_sockets
- php_sqlite
- php_ibm_db2
- php_http
- php_iisfunc
- php_pdo_ibm
- php_java

Para habilitar um módulo adicione uma linha segundo o padrão a seguir dentro da sessão [PHP] do arquivo “php.ini”:

extension={nome da extensão}.dll

Por exemplo, para habilitar o módulo “php_sockets” adicione a linha “extension=php_sockets.dll”.

Antes de adicionar a linha, verifique se o módulo já está habilitado e se a sua respectiva DLL existe na pasta de extensões do PHP.

Instalando a aplicação

Para instalar a aplicação, basta copiar o a pasta com os códigos-fontes dentro da pasta raiz do servidor web.

Problemas comuns

Configuração do Client DB2

Caso o sistema não passe da tela de login alegando que o client do DB2 não foi instalado, confira se as variáveis de ambiente do DB2 foram registradas corretamente. Em caso de erro, corrija e reinicie o sistema.

Verifique também se o seu antivírus não está bloqueando a utilização de alguma DLL do DB2. Caso isso aconteça, adicione a pasta onde a referida DLL está localizada como exceção de verificação do seu aplicativo antivírus.

Versão do PHP

Se algum módulo ou configuração não foi encontrado(a) no arquivo “php.ini” e na pasta de extensões do PHP, verifique se está sendo utilizada a versão correta.

ANEXO 4 - INSTALAÇÃO DO PHPUNIT

Pré requisitos:

- PHP 5.1.4 ou superior instalado
- Xdebug 2.0.5 ou superior

1º passo: Instalando o PEAR -> Vá até o diretório onde o php está instalado em sua máquina e execute o arquivo “go-pear.bat”.

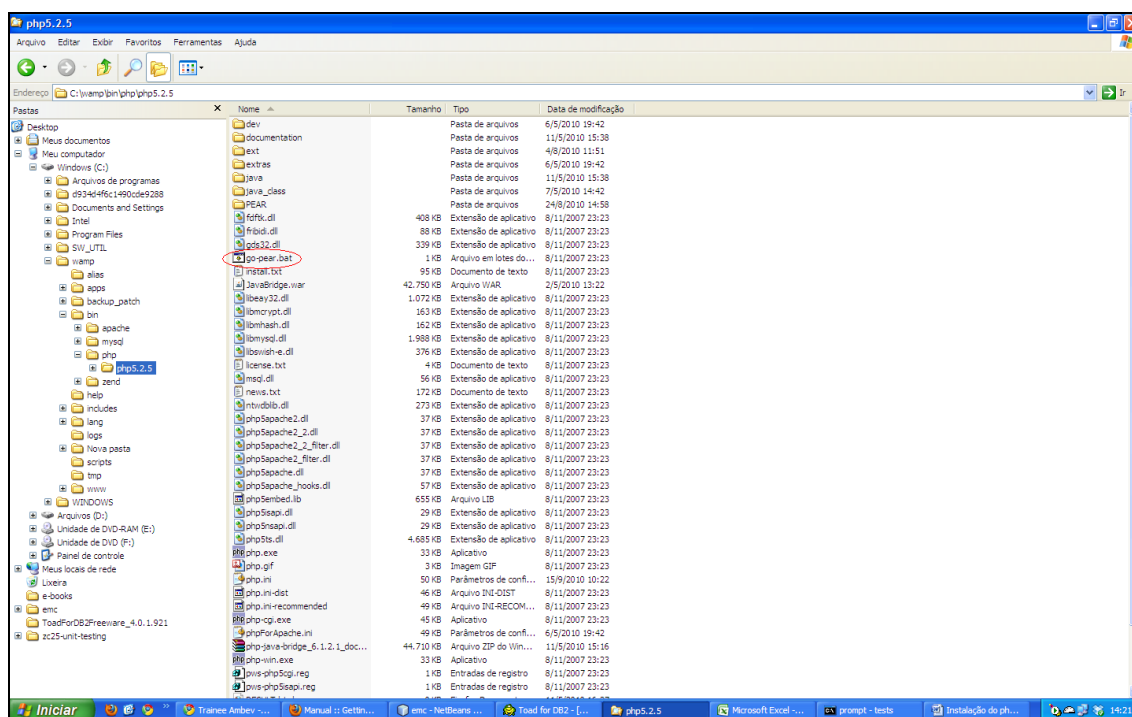


Figura 54 - Arquivo “.bat” usado para instalar o “PEAR”.

A instalação fará algumas perguntas e o PEAR será instalado no caminho especificado durante a instalação.

2º passo: Adicionar o caminho da instalação nas variáveis de ambiente. Existem 2 formas de fazer isso, manualmente (meu computador->propriedades->avançado->variáveis de ambiente)

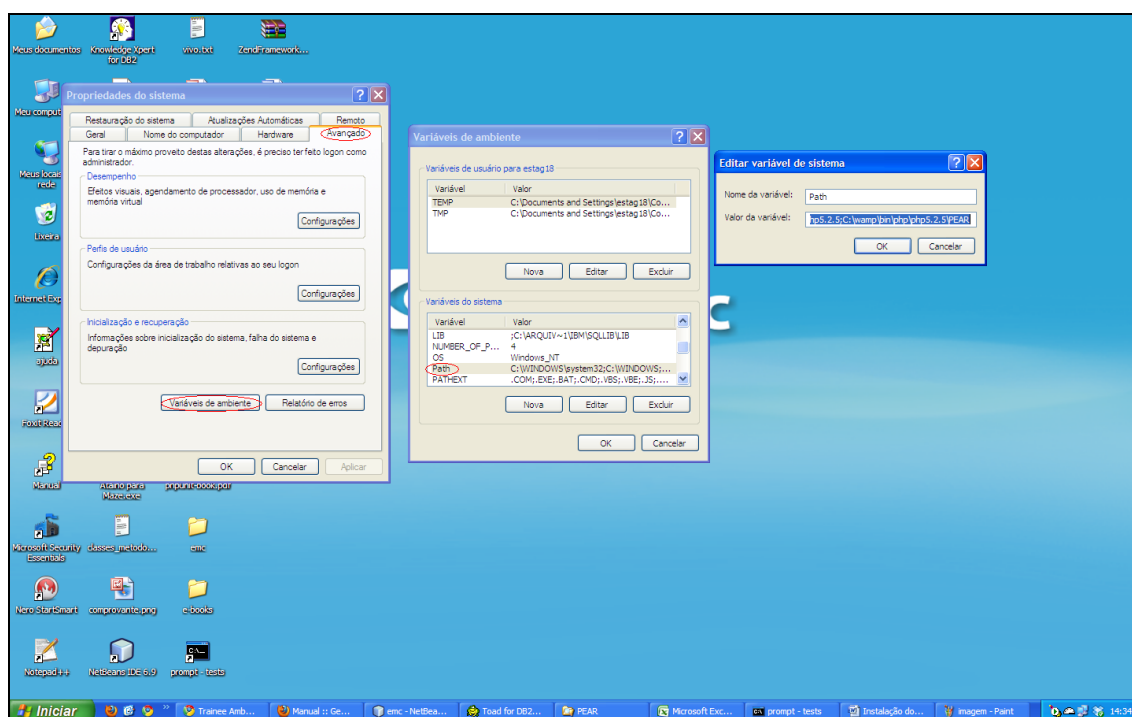


Figura 55 - Adicionando o caminho da instalação do “PEAR” nas variáveis de ambiente manualmente ou executando o arquivo “PEAR_ENV.reg” que foi gerado na instalação do “PEAR” e se encontra dentro da pasta “php/PEAR”.

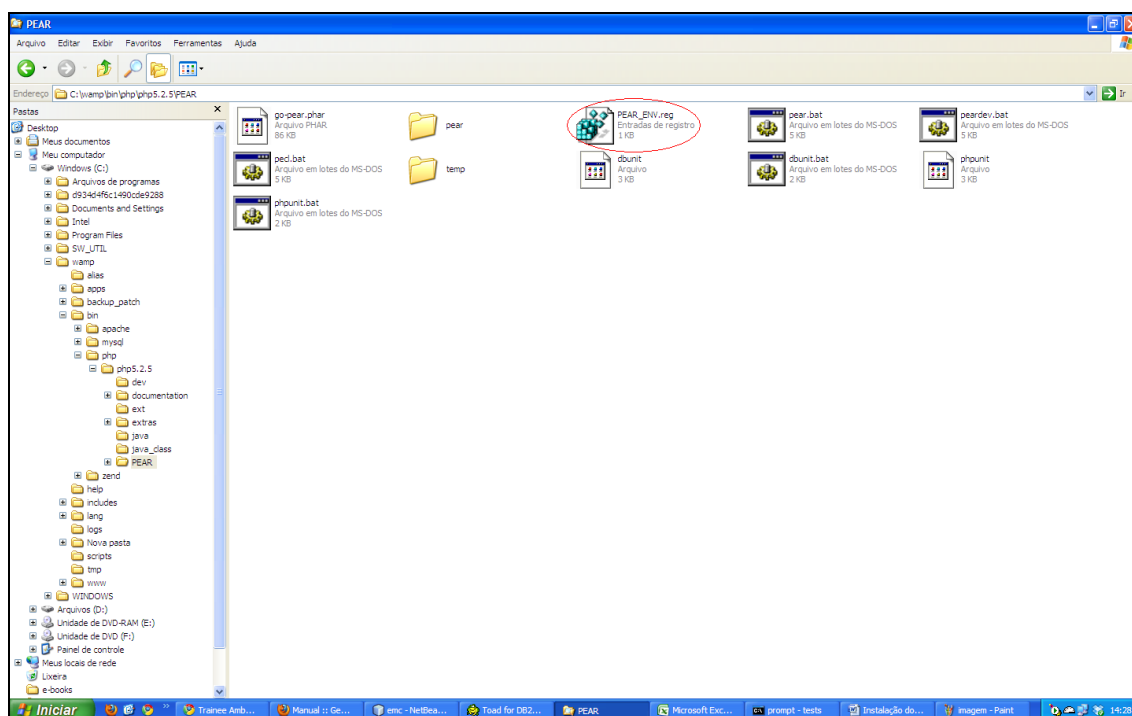


Figura 56 - Arquivo .reg utilizado para adicionar o caminho de instalação do PEAR nas variáveis de ambiente.

Obs.: Para atualizar sua instalação do “PEAR”, baixe o arquivo go-pear.php em <http://pear.php.net/go-pear> e na linha de comando digite “php go-pear.php”.

3º passo: Instalando o Xdebug -> Na linha de comando digite:

“pecl install xdebug”

Em seguida vá até seu arquivo php.ini e adicione a seguinte linha

zend_extension=" c:\wamp\bin\php\php5.2.5\ext\php_xdebug-2.1.0-5.2-vc6.dll"

Não se esqueça de mudar o caminho e o nome do arquivo de acordo com o local onde foi instalado e a versão do “xdebug” usada!

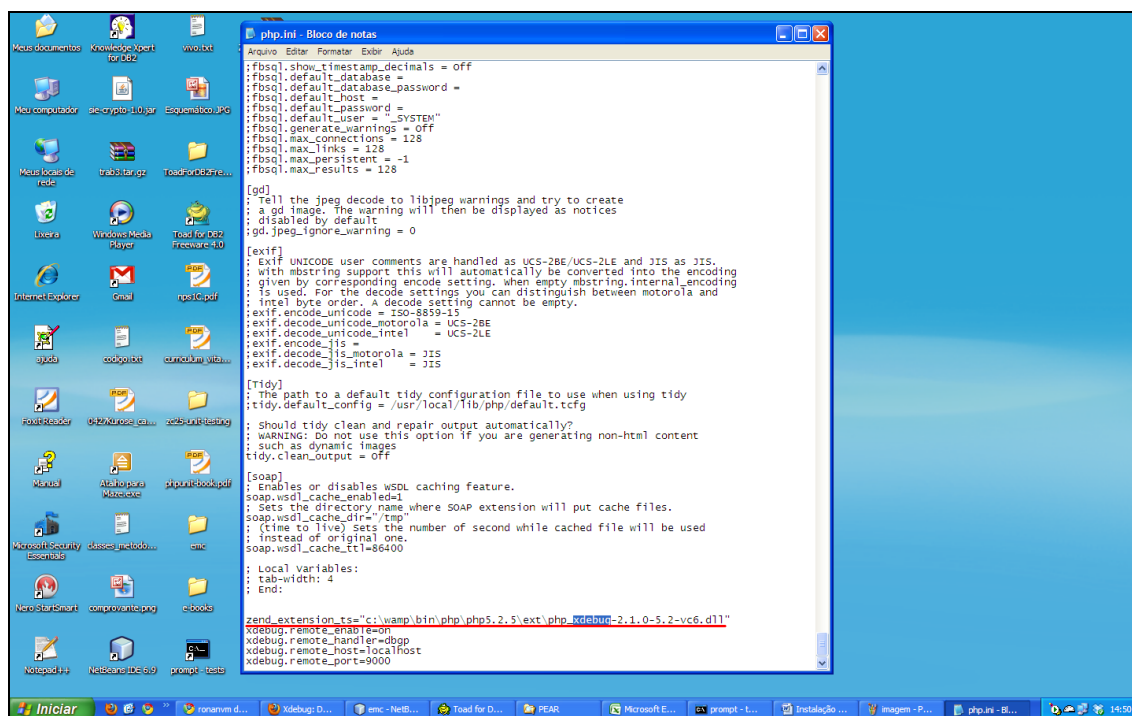


Figura 57 – Local onde o caminho deve ser mudado.

4º passo: Uma vez instalados o “PEAR” e o “xdebug”, vamos tratar da instalação de “PHPUnit”. Para isso o canal PEAR (pear.phpunit.de) que é usado para distribuir o

“PHPUnit” precisa ser registrado com o ambiente “PEAR” local. Além do mais, um componente que o “PHPUnit” necessita está hospedado no canal *Symfony Components* PEAR (pear.symfony-project.com).

Digite as seguintes linhas na linha de comando

```
pear channel-discover pear.phpunit.de  
pear channel-discover pear.symfony-project.com
```

Isso tem que ser feito apenas uma vez. Agora o instalador do “PEAR” pode ser usado para instalar os pacotes a partir do canal “PHPUnit”.

```
pear install phpunit/PHPUnit
```

Depois da instalação o “PHPUnit” pode ser achado dentro da pasta php/PEAR.

APÊNDICES

APÊNDICE A – CLASSES DO ZEND FRAMEWORK UTILIZADAS NO PROJETO

A seguir, estão listadas as classes do ZF utilizadas nesse trabalho e sua respectiva utilidade [18]:

- **Zend_Acl** – Permite que seja implementado o controle de acesso, leve e flexível, e com gerência de permissões/privilégios. A especificação de regras de controle de acesso tem suporte à herança.
- **Zend_Auth** – Disponibiliza uma API simples e escalável, que trabalha com autenticação de usuários, fazendo uso direto de tabelas do banco de dados e validando dados da forma que for necessário. Abstrai o armazenamento automático da identidade, facilitando a customização.
- **Zend_Config** – Torna simples a utilização de dados de configuração por aplicações web e também suporta herança de uma sessão para outra.
- **Zend_Controller** – Controla toda a lógica de negócios da aplicação criada. Além disso, permite a criação de um sistema utilizando o padrão de projeto MVC.
- **Zend_Db** – Camada de acesso ao banco de dados, que oferece interface para o PDO (*PHP Data Objects*), usado nesse projeto, além de outras extensões de bases de dados. Traz também adaptadores para cada *driver* de banco e uma API para construção de *queries*.
- **Zend_Dojo** – Faz a efetiva integração entre o *Dojo Toolkit* e o ZF. Essa classe dá suporte ao protocolo JSON-RPC¹², traz compatibilidade com dados da biblioteca *Dojo* e também fornece decoradores a elementos do tipo *Zend_Form*, tornando a visualização de dados mais agradável ao usuário final.
- **Zend_Filter** – Componente que fornece um leque de opções mais comuns de filtros e validação de dados. Disponibiliza também um mecanismo de filtros encadeados que podem ser aplicados em um único dado definido pelo desenvolvedor. Suporta também a personalização de mensagens de erros que serão exibidas.

¹² Protocolo bidirecional semelhante ao XML-RPC, projetado para ser simples [21].

- **Zend_Form** – Simplifica a criação e a manipulação de formulários na aplicação criada. Fornece as seguintes opções: filtro e validação de dados, ordenação de elementos, processamento de formulários, agrupamento de componentes e configuração de níveis de formulários.
- **Zend_Loader** – Carrega arquivos, classes e recursos dinamicamente em sua aplicação. Disponibiliza o carregamento automático da biblioteca padrão SPL¹³ e fornece o mecanismo de tratamento de exceções.
- **Zend_Locale** – Identifica os padrões de cada país. Esse componente, automaticamente mascara os dados da forma que o programador definir. É nessa classe que se define a localização (L10n) e a internacionalização (I18n) da sua aplicação. A localização supre as necessidades específicas desses sistemas e desses usuários, tal como: plurais, nomes, símbolos, ordenação, triagem e também detalhes de números, datas e horas, entre outros. Já a internacionalização se refere no suporte ao uso de sistemas às especificidades de diversos tipos de usuários no tocante a convenções de formatos de números, dados financeiros, padrões de data e hora, etc. Envolve as classes:
 - *Zend_Currency* – Formatos monetários.
 - *Zend_Date* – Formatos de datas.
 - *Zend_Measure* – Formatos de medidas.
 - *Zend_TimeSync* – Sincronização de horários com diferentes servidores, independente do servidor que hospeda a aplicação.
 - *Zend_Translate* – Tradução de dados.
- **Zend_Registry** – O registro é um container que armazena objetos e valores no espaço da aplicação. Ao armazenar o valor em um registro o mesmo objeto estará sempre disponível para que seja acessado em toda a aplicação. Essa é uma alternativa ao uso de armazenamento global.
- **Zend_Session** – Este componente auxilia no gerenciamento e na manutenção dos dados da sessão, ao longo de múltiplas requisições de página feitas pelo mesmo cliente. Fornece uma interface OO de acesso às informações e uma segurança adicional aos dados de sessão.

¹³ Da língua inglesa: *Standard PHP Library*.

- **Zend_View** – Classe que fornece uma máquina de templates, que será usada na camada de exibição de dados. E, juntamente com o Zend_Controller, fornece a infra-estrutura para o uso de MVC.

BIBLIOGRAFIA

1. **Universidade Federal de Uberlândia.** Universidade Federal de Uberlândia. [Online] [Citado em: 01 de 06 de 2010.] <http://www.sie.ufu.br/node/31>.
2. **DataHouse.** Multi-tier Architecture. [Online] [Citado em: 11 de 06 de 2010.] <http://www.datahousecorp.com/eng/technology/n-tier.htm>.
3. **Unicamp.** Departamento de Engenharia de Computação e Automação Industrial. [Online] [Citado em: 22 de 06 de 2010.] http://www.dca.fee.unicamp.br/cursos/POO_CPP/node3.html.
4. **Universidade Federal de Uberlândia.** Universidade Federal de Uberlândia. [Online] [Citado em: 02 de 06 de 2010.] <http://www.sie.ufu.br/node/34>.
5. **Educação, Ministério da.** MEC. [Online] [Citado em: 20 de 08 de 2010.] <http://emec.mec.gov.br/mec/>.
6. —. MEC. [Online] [Citado em: 20 de 08 de 2010.] <http://www.inep.gov.br/superior/enade/default.asp>.
7. **Maria, Universidade Federal de Santa.** UFSM/RS. [Online] [Citado em: 20 de 08 de 2010.] http://w3.ufsm.br/pi/index.php?option=com_content&view=article&id=51&Itemid=48.
8. **Universidade Federal do Paraná.** *Universidade Federal do Paraná.* [Online] [Citado em: 16 de 07 de 2010.] <http://www.intranet.ufpr.br/sie/treinamento/ofertas/oferta.htm>.
9. **Universidade Federal de Santa Maria.** Documentação do Sistema de Informações para Ensino. *Oferta de Disciplinas.* Santa Maria : s.n., 2005.
10. **Teixeira, Renan.** Renan Teixeira - PROGRAD/UFES. [Online] [Citado em: 14 de 06 de 2010.] <http://renanteixeira.wordpress.com/category/duvidas/>.
11. **Microsoft.** COM. [Online] [Citado em: 10 de 06 de 2010.] <http://www.microsoft.com/com/default.mspx>.
12. **Ziff Davis, Inc.** DLL Hell. *PC Magazine.* [Online] [Citado em: 20 de 07 de 2010.] http://www.pcmag.com/encyclopedia_term/0,2542,t=DLL+hell&i=41591,00.asp.
13. **Microsoft.** MSDN. [Online] [Citado em: 20 de 07 de 2010.] <http://msdn.microsoft.com/en-us/library/kwdt6w2k%28VS.71%29.aspx>.
14. *Requirements engineering: a good practice guide.* **Sommerville, Ian e Sawyer, Pete.** Chichester, England : John Wiley and Sons, Inc., 1997.

15. **Minetto, Elton Luís.** *Frameworks para Desenvolvimento em PHP*. Chapecó : Novatec Editora Ltda., 2007.
16. **Open Source Initiative.** Open Source Initiative. [Online] [Citado em: 02 de 08 de 2010.] <http://www.opensource.org/licenses/bsd-license.php>.
17. **Pressman, Roger S.** *Engenharia de Software*. São Paulo : Mc Graw Hill, 2006. ISBN 85-86804-57-6.
18. **Zend Technologies.** Zend Framework: Documentation. *Reference Guide of Zend Framework*. [Online] Zend Technologies. [Citado em: 22 de 05 de 2010.] <http://framework.zend.com/manual/en/>.
19. **Wikipedia.** Orientação a Objetos. [Online] [Citado em: 14 de 07 de 2010.] http://pt.wikipedia.org/wiki/Orientação_a_objetos.
20. **Dasgupta, Sanjoy, Papadimitriou, Christos e Vazirani, Umesh.** *Algorithms*. Berkeley, CA : The McGraw-Hill Companies, 2006.
21. **Edgewall Software.** JSON-RPC. [Online] Edgewall Software. [Citado em: 30 de 07 de 2010.] <http://json-rpc.org/>.
22. **Microsoft.** COM. [Online] [Citado em: 10 de 06 de 2010.] <http://www.microsoft.com/com/default.mspix>.
23. **PHP.** php.net. *php.net*. [Online] [Citado em: 12 de 05 de 2010.] http://br2.php.net/manual/pt_BR/intro-what-is.php.