

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS**

**PEDRO HENRIQUE LIBRELATO
WELLISON VICTOR BELUSSO**

SISTEMA WEB PARA INSCRIÇÃO EM ATIVIDADES ACADÊMICAS

TRABALHO DE CONCLUSÃO DE CURSO

**PATO BRANCO
2013**

**PEDRO HENRIQUE LIBRELATO
WELLISON VICTOR BELUSSO**

SISTEMA WEB PARA INSCRIÇÃO EM ATIVIDADES ACADÊMICAS

Trabalho de Conclusão de Curso de graduação, apresentado à disciplina de Trabalho de Diplomação, do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, da Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco, como requisito parcial para obtenção do título de Tecnólogo.


Orientador: Profa. Beatriz Terezinha Borsoi

**PATO BRANCO
2013**


ATA Nº: 238

DEFESA PÚBLICA DO TRABALHO DE DIPLOMAÇÃO DO ALUNO PEDRO HENRIQUE LIBRELATO.

Às 17:00 hrs do dia 27 de fevereiro de 2014, Bloco V da UTFPR, Câmpus Pato Branco, reuniu-se a banca avaliadora composta pelos professores Beatriz Terezinha Borsoi (Orientadora), Vinicius Pegorini (Convidado) e Adriano Serckumecka (Convidado), para avaliar o Trabalho de Diplomação do aluno Pedro Henrique Librelato, matrícula 1116738, sob o título **Sistema Web para Realizar Inscrição em Atividades Acadêmicas**; como requisito final para a conclusão da disciplina Trabalho de Diplomação do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, COADS. Após a apresentação o candidato foi entrevistado pela banca examinadora, e a palavra foi aberta ao público. Em seguida, a banca reuniu-se para deliberar considerando o trabalho **APROVADO**. Às 16:30 hrs foi encerrada a sessão.



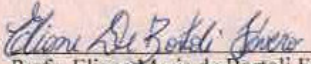
Prof. Beatriz Terezinha Borsoi, Dr.
Orientadora



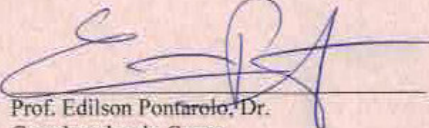
Prof. Vinicius Pegorini, Esp.
Convidado



Prof. Adriano Serckumecka, M.Sc.
Convidado



Prof. Eliane Maria de Bortoli Fávero, M.Sc.
Coordenador do Trabalho de Diplomação



Prof. Edilson Pontarolo, Dr.
Coordenador do Curso

ATA Nº: 239

DEFESA PÚBLICA DO TRABALHO DE DIPLOMAÇÃO DO ALUNO WELLISON VICTOR BELUSSO.

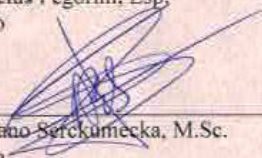
Às 17:00 hrs do dia 27 de fevereiro de 2014, Bloco V da UTFPR, Câmpus Pato Branco, reuniu-se a banca avaliadora composta pelos professores Beatriz Terezinha Borsoi (Orientadora), Vinicius Pegorini (Convidado) e Adriano Serckumecka (Convidado), para avaliar o Trabalho de Diplomação do aluno Wellison Victor Belusso, matrícula 1209418, sob o título **Sistema Web para Realizar Inscrição em Atividades Acadêmicas**; como requisito final para a conclusão da disciplina Trabalho de Diplomação do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, COADS. Após a apresentação o candidato foi entrevistado pela banca examinadora, e a palavra foi aberta ao público. Em seguida, a banca reuniu-se para deliberar considerando o trabalho **APROVADO**. Às 16:30 hrs foi encerrada a sessão.



Prof. Beatriz Terezinha Borsoi, Dr.
Orientadora



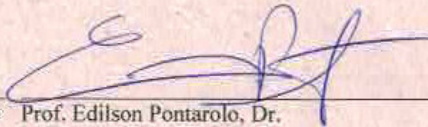
Prof. Vinicius Pegorini, Esp.
Convidado



Prof. Adriano Serckumecka, M.Sc.
Convidado



Prof. Eliane Maria de Bortoli Fávero, M.Sc.
Coordenador do Trabalho de Diplomação



Prof. Edilson Pontarolo, Dr.
Coordenador do Curso

RESUMO

LIBRELATO, Pedro Henrique; BELUSSO, Wellison Victor. Sistema web para inscrição em atividades acadêmicas. 2013. 58f. Monografia (Trabalho de Conclusão de Curso) - Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná. Pato Branco, 2013.

O Departamento Acadêmico de Informática (DAINF) e outros departamentos da Universidade Tecnológica Federal do Paraná, assim como ocorre com outras universidades e instituições de ensino, realiza atividades de extensão. Essas atividades são cursos de curta e de longa duração, palestras, oficinas e outros realizados para a comunidade externa e interna, embora a denominação extensão se aplique costumeiramente à comunidade externa. Essas atividades são divulgadas e os interessados realizam inscrição. No DAINF a inscrição é realizada pessoalmente, por *email* e mais recentemente por formulários GoogleDocs. Visando facilitar esse processo de inscrição foi implementado um sistema para realizar a inscrição nesse tipo de atividades, bem como registrar presença e aproveitamento para gerar a listagem para emissão de certificados. Para facilitar o acesso para a realização de inscrição, o sistema foi desenvolvido para *web* utilizando a linguagem de programação PHP.

Palavras-chave: Controle de inscrição em atividades. Aplicações web. PHP.

ABSTRACT

LIBRELATO, Pedro Henrique; BELUSSO, Wellison Victor. Web system to perform registration in academic activities. 2013. 58f. Monografia (Trabalho de Conclusão de Curso) - Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná. Pato Branco, 2013.

The Academic Computing Department (Departamento Acadêmico de Informática - DAINF) and other departments of the Federal Technological University of Paraná, as with other universities and educational institutions, conducts extension activities. These activities are courses of short and long term, lectures, workshops and others that are made for external and internal community; though the term extension is customarily apply to the external community. These activities are publicized and stakeholders perform registration. In DAINF registration is conducted in person, by email and more recently by forms GoogleDocs. Order to facilitate the process to sign up a system has been implemented to perform registration in such activities as well as document the presence and use to generate the listing for issuing certificates. To facilitate access to the achievement of enrollment, the system was developed for the web using PHP programming language.

Palavras-chave: Registration of academic events. Web application. PHP.

LISTA DE FIGURAS

FIGURA 1 – USUÁRIO WEB - SOLICITAÇÃO E RESPOSTA PARA UMA PÁGINA WEB	15
FIGURA 2 – REPRESENTAÇÃO DE UMA ARQUITETURA WEB TRÊS CAMADAS	16
FIGURA 3 – DIAGRAMA DE CASOS DE USO DO SISTEMA	24
FIGURA 4 – DIAGRAMA DE CLASSES	30
FIGURA 5 – DIAGRAMA DE ENTIDADES E RELACIONAMENTOS DO BANCO DE DADOS	35
FIGURA 6 – TELA DE LOGIN	39
FIGURA 7 – MENU DE NAVEGAÇÃO	40
FIGURA 8 – CADASTROS	40
FIGURA 9 – TELA DE CADASTRO DE USUÁRIOS	41
FIGURA 10 – JANELA PARA CONFIRMAÇÃO DE EXCLUSÃO DE REGISTRO	41
FIGURA 11 – PÁGINA CARREGADA PELA OPÇÃO 'NOVO'	42
FIGURA 12 – PÁGINA CARREGADA PELA OPÇÃO 'EDITAR'	43
FIGURA 13 – PÁGINA DE CADASTRO DE 'CURSOS'	44
FIGURA 14 – PÁGINA DE CADASTRO DE 'TURMAS'	45
FIGURA 15 – PÁGINA DE PROCURA DE 'AMBIENTES'	46
FIGURA 16 – PÁGINA DE INCLUSÃO DE 'RESPONSÁVEIS'	46
FIGURA 17 – PÁGINA DE INFORMAÇÃO DE 'CPF'	47
FIGURA 18 – PÁGINA DE CADASTRO DE 'ALUNO'	48
FIGURA 19 – PÁGINA DE INSCRIÇÃO.	49

LISTA DE QUADROS

QUADRO 1 – FERRAMENTAS E TECNOLOGIAS UTILIZADAS	20
QUADRO 2 – REQUISITOS FUNCIONAIS.....	23
QUADRO 3 – REQUISITOS NÃO FUNCIONAIS	24
QUADRO 4 – CASO DE USO MANTER CURSO	25
QUADRO 5 – CAMPOS DE ENTRADA DO CADASTRO DE CURSOS	25
QUADRO 6 – CAMPOS DE ENTRADA DO CADASTRO DE CURSOS	26
QUADRO 7 – CAMPOS DE ENTRADA DO CADASTRO DE RESPONSÁVEIS	26
QUADRO 8 – CAMPOS DE ENTRADA DO CADASTRO DE ALUNOS.....	27
QUADRO 9 – CAMPOS DE ENTRADA DO CADASTRO DE AMBIENTES	27
QUADRO 10 – CAMPOS DE ENTRADA DO CADASTRO DE MATRÍCULAS	27
QUADRO 11 – CAMPOS DE ENTRADA DO CADASTRO DE CONTROLES.....	28
QUADRO 12 – CAMPOS DE ENTRADA DO CADASTRO DE CATEGORIAS.....	28
QUADRO 13 – CAMPOS DE ENTRADA DO CADASTRO DE ÁREAS	28
QUADRO 14 – CAMPOS DE ENTRADA DO CADASTRO DE CONCEITOS.....	28
QUADRO 15 – CAMPOS DE ENTRADA DO CADASTRO DE PAPÉIS.....	28
QUADRO 16 – CAMPOS DE ENTRADA DO CADASTRO DE PESSOAS	29
QUADRO 17 – CAMPOS DE ENTRADA DO CADASTRO DE USUARIOS	29
QUADRO 18 – CASO DE USO ALTERAR DADOS PESSOAIS.....	29
QUADRO 19 – DESCRIÇÃO DA CLASSE PESSOA	30
QUADRO 20 – DESCRIÇÃO DA CLASSE USUARIO	31
QUADRO 21 – DESCRIÇÃO DA CLASSE CATEGORIA	31
QUADRO 22 – DESCRIÇÃO DA CLASSE AREA	31
QUADRO 23 – DESCRIÇÃO DA CLASSE CURSO.....	32
QUADRO 24 – DESCRIÇÃO DA CLASSE AMBIENTE	32
QUADRO 25 – DESCRIÇÃO DA CLASSE TURMA.....	33
QUADRO 26 – DESCRIÇÃO DA CLASSE PAPEL.....	33
QUADRO 27 – DESCRIÇÃO DA CLASSE CONCEITO	33
QUADRO 28 – DESCRIÇÃO DA CLASSE ALUNO	34
QUADRO 29 – DESCRIÇÃO DA CLASSE MATRICULA	34
QUADRO 30 – DESCRIÇÃO DA CLASSE CONTROLE.....	34
QUADRO 31 – TABELA PESSOAS	35
QUADRO 32 – TABELA CURSOS.....	36
QUADRO 33 – TABELA CATEGORIAS	36
QUADRO 34 – TABELA AREAS	36
QUADRO 35 – TABELA PAPEIS	36
QUADRO 36 – TABELA AMBIENTES	37
QUADRO 37 – TABELA RESPONSÁVEIS TUMACURSO	37
QUADRO 38 – TABELA TURMAS.....	38
QUADRO 39 – TABELA CONCEITOS	38
QUADRO 40 – TABELA CONTROLES.....	38
QUADRO 41 – TABELA ALUNOS	38
QUADRO 42 – TABELA MATRICULAS	39
QUADRO 43 – TABELA USUARIOS	39

LISTAGENS DE CÓDIGO

LISTAGEM 1 – MYSQL.....	50
LISTAGEM 2 – ALUNOCONTROLLER: CADASTRO E LISTAGEM DE ALUNOS	51
LISTAGEM 3 – CLASSE ALUNO	53
LISTAGEM 4 – PROCEDURE SP_ALUNOS.....	54
LISTAGEM 5 – EXEMPLO FORMULÁRIO DE VALIDAÇÃO.....	54
LISTAGEM 6 – EXEMPLO DE USO DE CLASSES DE VALIDAÇÃO.....	54
LISTAGEM 7 – EXEMPLO DE VALIDAÇÃO DE CAMPO.....	54

LISTA DE ABREVIATURAS E SIGLAS

CGI	<i>Common Gateway Interface</i>
CFP	Cadastro de Pessoa Física
CRUD	<i>Create, Retrieve, Update, Delete</i>
DAINF	Departamento Acadêmico de Informática
HTML	<i>HyperText Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IDE	<i>Integrated Development Environment</i>
IIS	<i>Internet Information Services</i>
MVC	<i>Model-View-Controller</i>
SGBD	Sistema de Gerenciamento de Banco de Dados
RIA	<i>Rich Internet Application</i>
SQL	<i>Struct Query Language</i>
UTFPR	Universidade Tecnológica Federal do Paraná

SUMÁRIO

1 INTRODUÇÃO	11
1.1 CONSIDERAÇÕES INICIAIS	11
1.2 OBJETIVOS	12
1.2.1 Objetivo Geral.....	12
1.2.2 Objetivos Específicos	12
1.3 JUSTIFICATIVA	12
1.4 ESTRUTURA DO TRABALHO	13
2 DESENVOLVIMENTO DE APLICAÇÕES PARA AMBIENTE INTERNET	14
2.1 APLICAÇÕES WEB	14
2.1.1 ARQUITETURA TRÊS CAMADAS.....	16
2.1.2 MODEL-VIEW-CONTROLLER	17
2.2 FRAMEWORKS PARA DESENVOLVIMENTO WEB.....	18
2.3 DESENVOLVIMENTO DE APLICAÇÕES WEB COM PHP	19
3 MATERIAIS E MÉTODO.....	20
3.1 MATERIAIS.....	20
3.2 MÉTODO	21
4 RESULTADO	22
4.1 ESCOPO DO SISTEMA.....	22
4.2 MODELAGEM DO SISTEMA.....	22
4.3 DESCRIÇÃO DO SISTEMA.....	39
4.4 IMPLEMENTAÇÃO DO SISTEMA	49
5 CONCLUSÃO	56
REFERÊNCIAS.....	57

1 INTRODUÇÃO

Este capítulo apresenta as considerações iniciais, os objetivos e a justificativa da realização deste trabalho. No final do capítulo está a organização do texto por meio de uma breve apresentação dos seus capítulos.

1.1 CONSIDERAÇÕES INICIAIS

Na Universidade Tecnológica Federal do Paraná (UTFPR), para realizar atividades acadêmicas - como cursos, palestras e oficinas que são ofertados como extensão - geralmente é necessário realizar inscrição. Atualmente é comum que as inscrições nessas atividades de extensão sejam realizadas por formulário impresso e mais recentemente, mas com pouca incidência, por formulários elaborados no GoogleDocs. A elaboração por meio de formulários GoogleDocs facilita a inscrição, mas não oferece mecanismos de gerenciamento. Os dados do formulário de inscrição são armazenados como arquivo de uma planilha de cálculos.

O Departamento Acadêmico de Informática (DAINF) da UTFPR, Câmpus Pato Branco, tem realizado uma grande quantidade de atividades de extensão que são ofertadas para a comunidade interna e externa. Considera-se como atividade de extensão: cursos, palestras, oficinas e outras que são ofertadas para os alunos (como complementares às atividades vinculadas às disciplinas dos cursos), aos servidores (como formativas e informativas) e para comunidade externa (como capacitação e informação).

Um aplicativo computacional utilizado para realizar a inscrição facilita a atividade de quem está realizando a inscrição (o aluno) e da pessoa responsável por gerenciar essas inscrições e/ou mesmo ministrar as atividades, facilitando também, o controle de presença e de aproveitamento. Esses itens podem ser utilizados para a emissão de comprovante de participação (certificado).

Verificou-se, assim, a oportunidade de desenvolver um aplicativo *web* que permita a realização de inscrição em atividades de extensão promovidas pelo Departamento Acadêmico de Informática e mesmo por outros departamentos e áreas da universidade. O aplicativo foi desenvolvido tendo como base o interesse do Departamento Acadêmico de Informática, mas pode ser utilizado por outros departamentos e mesmo por outras instituições que possam beneficiar-se de um sistema que permita inscrição em atividades acadêmicas de extensão por

meio da Internet.

1.2 OBJETIVOS

O objetivo geral está relacionado ao resultado principal que é esperado com a realização deste trabalho e os objetivos específicos o complementam.

1.2.1 Objetivo Geral

Implementar um aplicativo *web* para realização de inscrições em atividades de extensão ofertados pelo Departamento Acadêmico de Informática.

1.2.2 Objetivos Específicos

Facilitar o processo de inscrição, por parte da comunidade externa e interna, em atividades de extensão que são ofertadas pelo Departamento Acadêmico de Informática.

Agilizar o processo de inscrição em atividades de extensão, possibilitando que a inscrição seja realizada por meio de um sistema *web*.

Facilitar o gerenciamento e o acompanhamento das inscrições realizadas em atividades de extensão.

Facilitar o controle de aproveitamento e presença dos participantes em atividades de extensão.

1.3 JUSTIFICATIVA

O DAINF, assim, como outros departamentos da UTFPR, realiza atividades de extensão como cursos e palestras para os acadêmicos e servidores e para a comunidade externa. É comum que seja necessário fazer a inscrição para poder participar dessas atividades. Atualmente a inscrição nessas atividades tem sido realizada, geralmente, de forma presencial pelo interessado ou por meio de *email*. Para algumas atividades mais recentes, no

ano de 2013, o DAINF adotou formulários no GoogleDocs para realizar a inscrição em alguns dos cursos que ofereceu. Esses formulários, embora sejam bem mais interessantes que a inscrição por *email* e em formulários impressos, não fornecem mecanismos para o gerenciamento do número inscritos e para o controle de acompanhamento dos cursos, por exemplo.

Assim, uma das principais justificativas da realização deste trabalho de conclusão de curso é fornecer uma maneira de facilitar a realização de inscrição em atividades de extensão promovidas pelo DAINF. O sistema desenvolvido como resultado deste trabalho além de possibilitar a realização da inscrição, permitirá o gerenciamento dessas inscrições e a realização do controle do aproveitamento dos participantes.

A justificativa da escolha de um sistema para Internet decorre da facilidade de acesso, seja no ambiente interno da universidade, seja pela comunidade externa. Auxiliando, assim, no processo de divulgação, inscrição e gerenciamento das atividades de extensão promovidas pelo DAINF e mesmo pela UTFPR.

1.4 ESTRUTURA DO TRABALHO

Este texto está organizado em capítulos. Este é o primeiro e apresenta as considerações, o objetivo e a justificativa do trabalho.

O Capítulo 2 apresenta o referencial teórico e está centrado em desenvolvimento de aplicações para Internet porque o aplicativo desenvolvido como resultado deste trabalho é para *web*.

No Capítulo 3 estão os materiais e o método. Os materiais apresentam as tecnologias e ferramentas utilizadas para a modelagem e a implementação do sistema. O método contém as principais atividades realizadas para o desenvolvimento do trabalho.

Os resultados da realização deste trabalho são apresentados no Capítulo 4. Esses resultados estão centrados na modelagem e na implementação do sistema.

Por fim está a conclusão seguida das referências bibliográficas.

2 DESENVOLVIMENTO DE APLICAÇÕES PARA AMBIENTE INTERNET

Este capítulo apresenta o referencial teórico que fundamenta o aplicativo desenvolvido como resultado deste trabalho.

2.1 APLICAÇÕES WEB

Uma aplicação *web* é caracterizada como é a que é acessada por um navegador ou *browser web*. Essas aplicações executam em um servidor e o cliente é uma página *web* por meio do qual o usuário interage com o aplicativo. As diversas tecnologias existentes para implementar essas aplicações, sejam linguagens de programação, *frameworks* de componentes para interface e técnicas como Ajax, permitem desenvolver aplicativos tanto *web* tradicionais quanto semelhantes às aplicações *desktop*.

Tradicionais são as aplicações implementadas como conjuntos de páginas desenvolvidas com *HyperText Markup Language* (HTML) e que são vinculadas por meio de *hiperlinks*. Essas aplicações possuem formulários com componentes muito simples, baseados em *Common Gateway Interface* (CGI) (CHO et al., 1997).

As aplicações *web* semelhantes às *desktop* são as que possuem recursos de interação utilizados nas aplicações *desktop*, como os efeitos de arrastar e soltar, validações realizadas em tempo de execução e menus e botões diferenciados (SHAN; HUA, 2006). Esses recursos fazem com que essas aplicações sejam denominadas como ricas, as *Rich Internet Application* (RIA). As RIAs provêm interface com recursos que facilitam a representação de processos e dados ao mesmo tempo em que reduzem o tráfego entre cliente e servidor (FUKUDA; YAMAMOTO, 2008).

Uma aplicação *web* deve ser desenvolvida de forma que possa ser acessada por meio de um navegador *web*, seja ela mais tradicional ou com recursos que a caracterizam como *Rich Internet Application* (MELIÁ et al., 2010). O servidor que hospeda a aplicação deve possuir recursos que permitam essa interação com o usuário. Esse tipo de servidor também é caracterizado como servidor *web*.

As RIAs minimizam a comunicação com o servidor. Ao invés de carregar uma página HTML inteira a cada interação do usuário, o evento gerado pode ser tratado localmente (como a validação de um dado) ou apenas parte da página ser atualizada, ou seja, a que é afetada

pelo evento gerado (SHAN; HUA, 2006).

A função do cliente *web* é permitir fazer solicitações ao servidor, exibindo o resultado do pedido. O navegador *web* (*browser*) é o software que se comunica com o servidor fazendo a comunicação entre o usuário e o servidor. O navegador é o cliente *web* do sistema (PALMEIRA, 2013). A Figura 1 apresenta de forma esquemática esse comportamento.

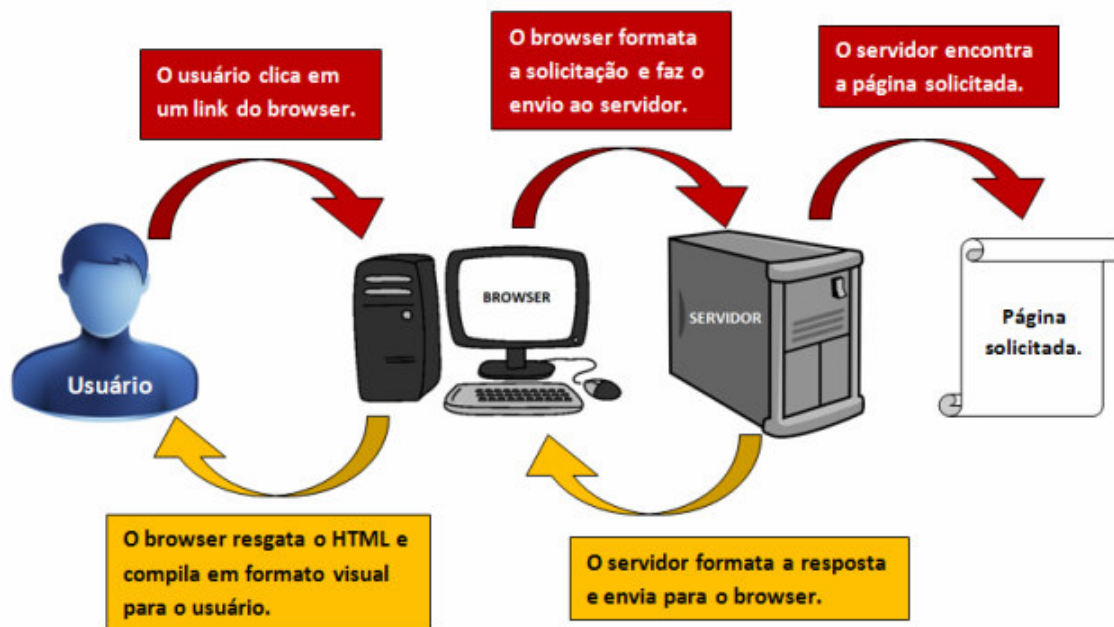


Figura 1 – Usuário Web - Solicitação e resposta para uma página web
Fonte: Palmeira (2013).

De acordo com a representação da Figura 1, o usuário, por meio de um navegador *web*, acessa o aplicativo que formata a informação e a envia para o servidor. O servidor encontra a página solicitada, realiza as operações necessárias (consulta a banco de dados, realização de regras de negócio, acesso a outros aplicativos, dentre outros), formata a resposta e a envia para o navegador. Esse apresenta a informação que está formatada em HTML ou outras tecnologias, ao usuário.

O desenvolvimento de uma aplicação *web* pode ser organizado em camadas, no sentido de agrupar os componentes de acordo com funcionalidades. Uma aplicação *web* típica ou que atende ao modelo cliente/servidor padrão, normalmente é desenvolvida em duas camadas: cliente e servidor. Esse é o modelo padrão de aplicações em rede, mas sem uso de tecnologias *web* era comum antes do uso da Internet como plataforma de execução de aplicações (BATTISTI, 2013). Nesse modelo:

- Cliente – é responsável pela a lógica de negócio que define como os dados serão processados e armazenados e pelas funcionalidades de acesso aos dados que

estariam armazenados em um servidor de banco de dados. O cliente também é responsável pela apresentação da aplicação, a interface de interação do usuário.

- b) Servidor – armazena os dados que são manipulados pelo cliente e as regras de negócio que manipulam esses dados.

Uma evolução do modelo de duas camadas é a separação da apresentação, da lógica de negócio e dos dados em uma arquitetura três camadas, o que caracteriza o *Model-View-Controller* (MVC) (MCHEICK; QI, 2011).

2.1.1 ARQUITETURA TRÊS CAMADAS

Considerando o navegador *web* como a camada cliente, nas aplicações *web* o modelo de arquitetura em três camadas que organiza o código em elementos de apresentação, lógica de negócio e persistência passa a ser visto como um modelo quatro camadas: cliente, apresentação, lógica de negócio e persistência (dados). Ressalta-se que não necessariamente cada camada está localizada fisicamente em computadores distintos e que uma mesma camada pode estar em computadores diferentes quando há, por exemplo, diversos servidores de banco de dados.

A Figura 2 apresenta de forma esquemática o funcionamento de uma aplicação *web* de acordo com o modelo de arquitetura em três camadas.

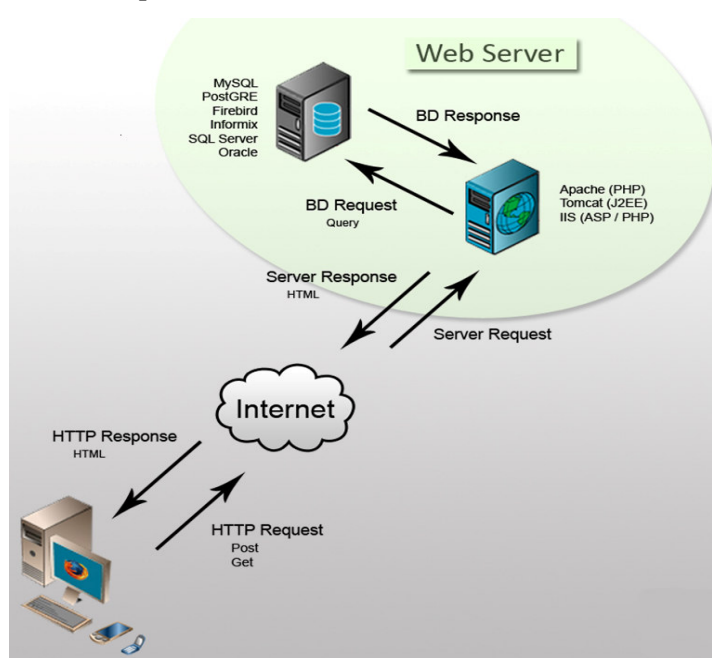


Figura 2 – Representação de uma arquitetura web três camadas
Fonte: França (2010, p. 1).

De acordo com a representação da Figura 2, *web server* é o conjunto formado pelo servidor de aplicação e o servidor de banco de dados e as interações que ocorrem entre eles. Em uma aplicação *web*, na troca de mensagens entre um computador cliente e um *web server*, o usuário acessa uma página por meio de um navegador que envia uma requisição, utilizando o protocolo *Hypertext Transfer Protocol* (HTTP), para o servidor *web*. Essa requisição é processada pelo servidor e pode ser um *script* ou um arquivo a ser baixado pelo usuário. Se for um *script*, o servidor de aplicação o processa utilizando o interpretador da linguagem. Se necessário, o *script* em execução faz conexão com o banco de dados e envia uma requisição, que pode ser uma instrução *Struct Query Language* (SQL). A requisição é processada no servidor de banco de dados e o resultado do processamento é devolvido para o servidor da aplicação. Após finalizada a execução do *script*, os dados resultantes do processamento são mesclados com *tags* de formação HTML. O documento HTML é enviado para o cliente por meio de uma mensagem HTTP. Na camada cliente, o documento HTML é renderizado pelo navegador gerando uma página *web* (FRANÇA, 2010).

Nesse modelo em uma aplicação *web*, a camada de apresentação é o navegador *web* que contém a interface de interação com a aplicação. Na camada de persistência está o servidor de banco de dados que armazena os dados da aplicação e conterá o Sistema de Gerenciamento de Banco de Dados (SGBD). A camada de lógica de negócio é o servidor executará a aplicação. Esse servidor deve conter o interpretador da linguagem como, por exemplo, Apache, *Internet Information Services* (IIS) ou Tomcat. Essas partes podem ser organizadas por um padrão arquitetural como o MVC, por exemplo.

2.1.2 MODEL-VIEW-CONTROLLER

MVC é um padrão de projeto arquitetural. Esse tipo de padrão define os elementos, as relações e as regras a serem seguidas que já tiveram sua utilidade avaliada em soluções de problemas passados (GERMOGLIO, 2010). Nesse modelo as regras de negócio ficam em um servidor de aplicação. Essas regras manipulam os dados que estão em um servidor de dados. E a apresentação, a interface de interação com o sistema, fica no cliente.

O MVC provê separação de objetos em três partes (GAMMA et al., 2000; GONÇALVES et al., 2005; SWEAT, 2005; HANSEN; FOSSUM, 2005): modelo, visão e controle. Por meio dessa separação, múltiplas visões e controles podem interagir com um mesmo modelo de dados. Assim, os dados podem ser apresentados de formas distintas e

lógicas de negócio distintas também podem ser aplicadas ao mesmo modelo de dados. A seguir uma breve descrição dessas três partes.

a) Modelo (*model*) - o modelo encapsula os dados do aplicativo e os métodos para acessar e manipular esses dados. A camada de dados contém os componentes que representam e armazenam a informação que é apresentada. Essa camada contém as tabelas de uma base de dados, por exemplo, e as regras de negócio que se aplicam aos dados. O modelo é responsável por manter o estado da aplicação.

b) Visão (*view*) - a visão é responsável por exibir os dados para o usuário. A visão extrai dados do modelo e os formata para apresentação.

c) Controle (*control*) – o controle representa a lógica de negócio da aplicação, sendo responsável pelo processamento de dados e pela atualização do modelo e da visão. O controle direciona o fluxo do aplicativo e recebe as interações com a aplicação.

O MVC tem como principal objetivo organizar os dados e a lógica de negócio (*Model*), o fluxo da aplicação (*Controller*) e a interface com usuário (*View*). No a visão é apresentada no cliente, mas o modelo e o controle podem ser segmentados de diversas maneiras entre cliente e servidor (LEFF; RAYFIELD, 2001). E a forma como o acoplamento entre as três partes desse modelo é realizada impacta no desenvolvimento, na manutenção e no reuso do código (HANSEN; FOSSUM, 2005).

Para a implementação do MVC e mesmo para organizar e agilizar o desenvolvimento de aplicações *web* podem ser utilizados *frameworks*.

2.2 FRAMEWORKS PARA DESENVOLVIMENTO WEB

Os *frameworks* encapsulam operações como, por exemplo, inclusão, seleção, atualização e exclusão de dados em banco de dados (CUI et al, 2009). Assim, as operações de banco de dados tornam a implementação mais simplificada e conseqüentemente o foco do desenvolvimento fica centrado na lógica de negócio em vez de estar centrada na elaboração de instruções SQL, por exemplo.

A categoria de software mais popular que melhora a produtividade de desenvolvedores de software é chamada *framework* (JOHNSON, 1997). Um *framework* é um conjunto de componentes integrados que colaboram para produzir uma arquitetura reusável para uma família de aplicações e oferecem os seguintes benefícios (OKANOVIC; MATELJAN, 2011):

- a) Modularidade – *frameworks* encapsulam detalhes de implementação e a modularidade auxilia a melhorar a qualidade do software pela localização do impacto das mudanças no projeto e na implementação.
- b) Reusabilidade – uma interface estável provida pelos *frameworks* aumenta o reuso pela definição de componentes genéricos que podem ser utilizado para criar novas aplicações. Reuso pode produzir melhorias substanciais na produtividade do programador e na qualidade do software.
- c) Extensibilidade – um *framework* melhora a extensibilidade por prover pontos de extensão que permitem às aplicações serem estendidas a partir de suas interfaces.
- d) Inversão de controle – uma arquitetura em tempo de execução de um *framework* é caracterizado por uma inversão de controle. Quando um evento ocorre, o expedidor (*dispatcher*) de eventos reage por meio da invocação de um método em um ponto de extensão que realiza o processamento de um evento específico.

2.3 DESENVOLVIMENTO DE APLICAÇÕES WEB COM PHP

O crescimento rápido da Internet vinculado ao desenvolvimento de aplicações *web* trouxe demanda para alta eficiência, confiança, manutenibilidade e escalabilidade dessas aplicações (CUI et al., 2009). Esses autores destacam que PHP possui como características a intuitividade e a facilidade de manipulação, execução rápida, multi-plataforma e código fonte aberto, tornando-se uma das mais importantes linguagens de desenvolvimento *web*.

PHP é uma linguagem de programação originalmente criada para a implementação de conteúdo dinâmico em páginas *web* (PHP, 2013). Essa linguagem executa no lado servidor com uso eficiente de recursos de processamento e de memória em diversas plataformas (SANTOS; MENDONÇA; MARTINS, 2008).

Na forma padrão de desenvolvimento com PHP há uma mescla de código que acessa os dados com o código que processa a lógica de negócio e o que faz a apresentação (interface) *web* do sistema. O uso do padrão de projeto MVC provê uma maneira mais efetiva de geração de aplicações modulares. Com o uso de MVC há a separação da aplicação em camadas que podem ser definidas e mesmo implementadas separadamente. Pelo desacoplamento de modelos e visões, o MVC auxilia a reduzir a complexidade de um projeto arquitetural e a incrementar a flexibilidade e o reuso de código (CUI, 2009).

3 MATERIAIS E MÉTODO

Este capítulo apresenta os materiais e o método utilizados para a realização deste trabalho. Os materiais estão relacionados às tecnologias e ferramentas utilizadas e o método apresenta a sequência dos passos, com as principais atividades realizadas.

3.1 MATERIAIS

O Quadro 1 apresenta as ferramentas e as tecnologias que foram utilizadas para modelar e implementar o sistema.

Ferramenta / Tecnologia	Versão	Finalidade	Referência
Astah Community	6.2.1	Modelagem do diagrama com a visão geral do sistema e diagrama de casos de uso	http://astah.net/editions/community
Case Studio 2	2.25.0	Modelagem do diagrama de entidades e relacionamentos do banco de dados	http://www.casestudio.com/enu/default.aspx
Notepad++	6.3.2	Ambiente de desenvolvimento.	http://notepad-plus-plus.org/
PHP	5.3	Linguagem de programação	http://php.net/
Apache Tomcat	6.0	Servidor <i>web</i>	http://tomcat.apache.org/
MySQL	5.5	Gerenciador de banco de dados	http://www.mysql.com/
HTML	5	Linguagem de marcação utilizada para produzir páginas na <i>web</i>	http://www.w3schools.com/html/
CSS	3	Linguagem de estilo utilizada para definir a apresentação de documentos escritos em uma linguagem de marcação	http://www.w3schools.com/css/
JavaScript	5	Linguagem de programação interpretada	http://www.w3schools.com/js/
JQuery	1.9.1	Biblioteca JavaScript de código aberto	http://jquery.com/
PhpMyAdmin	4.0.8	Para gerenciamento do banco de dados MySQL	http://www.phpmyadmin.net/home_page/index.php

Quadro 1 – Ferramentas e tecnologias utilizadas

3.2 MÉTODO

A modelagem do aplicativo foi realizada como trabalho de estágio pelos autores deste trabalho de conclusão de curso. O estudo das tecnologias, representado pela implementação das operações de inclusão, exclusão, consulta e alteração de um cadastro, também foi realizado como trabalho de estágio. Os requisitos e a modelagem foram revistos, mas a ênfase deste trabalho está na implementação do sistema. E a implementação teve como base o modelo sequencial linear como descrito em Pressman (2008). As fases ou etapas definidas estão descritas a seguir. Foi utilizado o modelo sequencial linear porque os requisitos do sistema estavam bem definidos. Contudo, ressalta-se que pequenas mudanças foram realizadas nos requisitos e na modelagem durante a fase de implementação.

a) Requisitos

A revisão dos requisitos foi realizada juntamente com a professora orientadora, a solicitante do sistema. Na fase de revisão foi discutida a melhor maneira de realizar a inscrição nas atividades. Conclui-se que o aluno (quem faz a inscrição em uma atividade) deveria estar previamente cadastrado.

Também ficou definido que o usuário com perfil de professor (quem ministra as atividades, embora exerça papéis distintos, como instrutor, auxiliar e coordenador, por exemplo) tem acesso somente às atividades das quais ele faz parte, ou seja, está vinculado.

b) Análise e projeto do sistema

Na fase de análise e projeto a modelagem foi revista e os ajustes necessários foram realizados. Campos das tabelas que armazenam dados de inscrições e de cursos foram revistos. Não houve alterações significativas no banco de dados, apenas inclusão e exclusão de campos em tabelas.

c) Implementação

A implementação foi realizada utilizando a *Integrated Development Environment* (IDE) Notepad++.

d) Testes

Foram realizados testes para verificação do código e de interface e usabilidade do sistema pelos autores deste trabalho.

4 RESULTADO

Este capítulo apresenta o resultado da realização deste trabalho que é o desenvolvimento de um aplicativo *web* para inscrições e gerenciamento dessas inscrições de atividades acadêmicas de extensão.

4.1 ESCOPO DO SISTEMA

O aplicativo se destina à realização de inscrição em atividades de extensão promovidas pela UTFPR. O aplicativo será utilizado, inicialmente, pelo Departamento Acadêmico de Informática, mas o mesmo poderá ser utilizado por outros departamentos e mesmo outras instituições que realizam esse tipo de atividade.

Além de permitir a inscrição em atividades de extensão (cursos, palestras, oficinas e outros), o aplicativo permitirá o registro de presença e aproveitamento da atividade possibilitando emitir listagens de participantes do curso que podem receber comprovante ou certificado. A emissão de comprovante pode ser dependente de o aluno atender a determinados requisitos como presença e aproveitamento.

Os cursos são instanciados por meio de atividades vinculadas ao mesmo. Cada curso pertence a uma área e uma categoria. Papéis (que representam funções desempenhadas por pessoas) vinculam pessoas aos cursos. Os alunos que realizam as atividades são organizados em turmas, definindo uma lista de alunos. E cada turma está relacionada a uma atividade. O aluno recebe comprovante de realização da atividade, se atender aos critérios de aproveitamento e presença definidos para a respectiva atividade.

4.2 MODELAGEM DO SISTEMA

A listagem do Quadro 2 apresenta os requisitos funcionais identificados para o sistema.

Identificação	Nome	Descrição
RF001	Manter usuário	Cadastrar usuário do sistema.
RF002	Manter categorias de cursos	O cadastro de categorias será utilizado no cadastro de cursos. Incluir categorias com: nome e descrição. Excluir categoria cadastrada.

		Editar, com a possibilidade de alteração.
RF003	Manter cursos	Incluir cursos com: nome, descrição, requisitos, observadores e categoria. Excluir cursos cadastrados. Editar, com possibilidade de alteração e reutilização.
RF004	Manter turmas de cursos	Inclusão de turma para os cursos com: curso, nome da turma, data do início do curso, data do fim do curso, horário, carga horária, data do início da inscrição, data do fim da inscrição, limite de vagas, observações, ambiente da realização, local e contato. Excluir turma cadastrada. Editar, com possibilidade de alteração.
RF005	Inscrição em curso	Incluir inscrição com: turma, nome, <i>email</i> , ordem de inscrição, instituição e telefone. Excluir uma inscrição cadastrada. Editar, com a possibilidade de alteração, informações de uma inscrição cadastrada.
RF006	Cadastro de frequência e aproveitamento	Incluir controle de frequência e aproveitamento com: conceito, inscrição, turma, faltas, nota e resultado. Editar, com possibilidade de alteração. Consultar dados de frequência e aproveitamento.
RF007	Cadastro de conceitos	Incluir conceitos (aprovado, reprovado, por exemplo) com: nome e descrição. Excluir conceito cadastrado. Editar conceito com possibilidade de alteração.
RF008	Cadastro de pessoas	Incluir pessoas com: nome, <i>email</i> , <i>email</i> alternativos e telefone. Excluir uma pessoa cadastrada. Editar cadastro da pessoa com possibilidade de alteração de dados.
RF009	Cadastro de papéis	Incluir papéis (instrutor, monitor, coordenador, responsável, etc.) com: nome e descrição. Editar papel com possibilidade de descrição.
RF010	Cadastro de responsáveis por curso	Incluir responsáveis (vínculo entre cadastro de papéis e pessoas) com: pessoa, papel e turma. Excluir responsável cadastrado. Alterar dados de um responsável cadastrado.

Quadro 2 – Requisitos funcionais

A listagem do Quadro 3 apresenta os requisitos não-funcionais identificados para o sistema. Os requisitos não funcionais explicitam regras de negócio, restrições como as de acesso, aspectos de qualidade, desempenho, segurança e outros.

Identificação	Nome	Descrição
RFN01	Inscrição	Cada curso possui um número máximo de inscrições permitidas. Ao ser alcançado o limite de inscrições, se ainda dentro do prazo estabelecido, a inscrição será permitida, mas deve ser apresentada mensagem que é para lista de espera.
RNF02	Validação inscrições	Deverá ser apresentada a listagem de todos os alunos inscritos em uma determinada turma.
RNF03	Frequência e aproveitamento	O sistema permitirá o registro das frequências (total, se não permitir controle por chamada) e do aproveitamento, a nota. O sistema permitirá definir os parâmetros de valor para

		frequência e nota para emitir a listagem dos que obterão certificado. O professor responsável pelo curso editará a listagem dos alunos, com frequência e aproveitamento, e atribuirá o conceito ao mesmo.
RNF04	Acesso ao sistema	Haverá três níveis de acesso: a) Administrador com acesso a todas as funcionalidades do sistema b) Professor com permissão para cadastrar cursos e disponibilizar cursos que ele cadastrou. c) Aluno com permissão para fazer inscrição em cursos. O aluno não precisa estar cadastrado para ter acesso para realizar inscrição em cursos.
RNF05	Cadastro de cursos	Possibilitar a utilização de um mesmo curso várias vezes. Uma turma define uma instância de um determinado curso.

Quadro 3 – Requisitos não funcionais

A partir dos requisitos foram definidos os casos de uso apresentados na Figura 3.

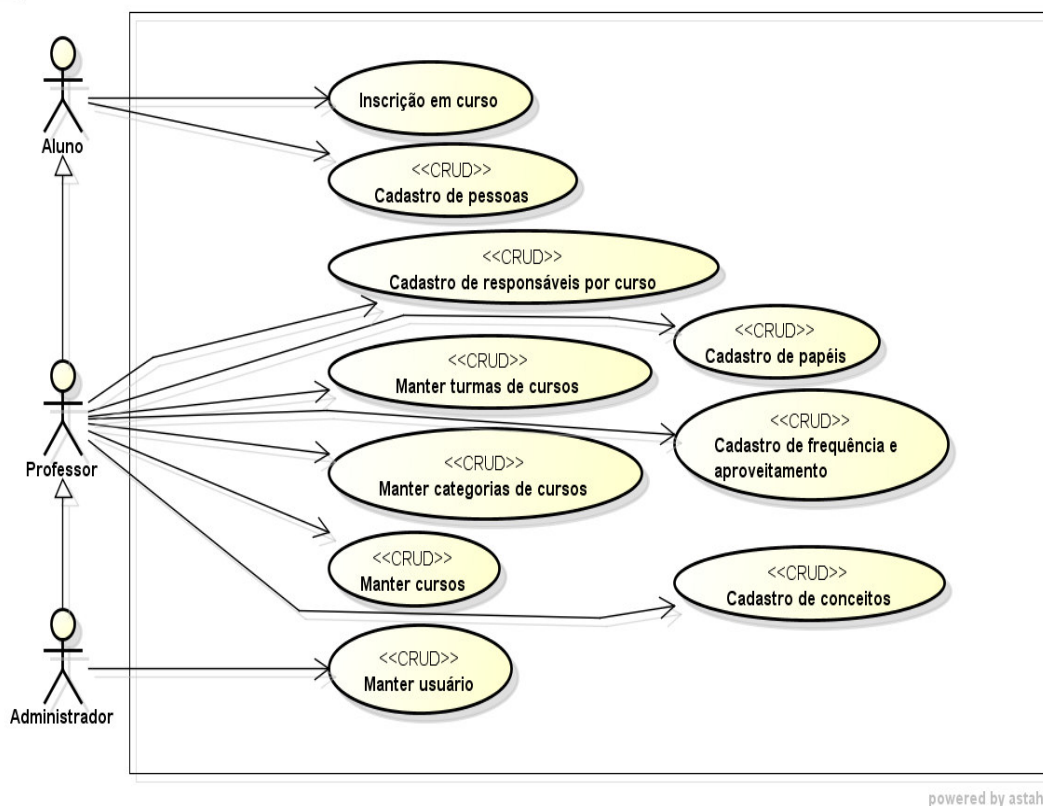


Figura 3 – Diagrama de casos de uso do sistema

A seguir, um caso de uso de cadastro está documentado. Esses casos de uso são identificados com o estereótipo CRUD (*Create, Retrieve, Update, Delete*) na Figura 3 e representam as operações realizadas com dados armazenados em tabelas. Esse caso de uso é o de “Manter curso” e é utilizado para exemplificar como são descritos os casos de uso tipo

CRUD. Todos os casos de uso de cadastro seguem o mesmo padrão, considerando que pode ou não haver necessidade de dados provenientes de outros cadastros.

<p>1.1 Identificador do caso de uso: Manter curso.</p> <p>Descrição: Cadastro de curso aos quais estarão categorias e turmas vinculadas.</p> <p>Evento Iniciador: O usuário solicita a inclusão de um curso no sistema.</p> <p>Atores: Professor, Administrador</p> <p>Pré-condição: A categoria deve estar cadastrada.</p> <p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. Ator Professor/Administrador acessa a tela para cadastro de um novo curso e inclui as informações necessárias. A categoria, um dos campos de entrada, deve estar cadastrada e é escolhida a partir de uma listagem apresentada. 2. O sistema insere os dados no banco de dados, verificando se o nome do curso está descrito e informa ao usuário que o referido curso foi incluído. <p>Pós-Condição: Curso inserido no banco de dados.</p> <p>Extensões: Cadastrar categoria.</p>	
Nome do fluxo alternativo (extensão)	Descrição
1.1 Cadastro de categoria.	1.1 O ator professor ou administrador acessa a tela do sistema para cadastrar a categoria pretendida e inclui as informações solicitadas. 1.2 Sistema inclui informações no banco de dados.

Quadro 4 – Caso de uso manter curso

A seguir são apresentadas as listagens de campos para cada um dos casos de uso identificados como o estereótipo CRUD na Figura 3.

Listagem dos campos de entrada do cadastro de curso (Quadro 5).

Dado	Descrição	Tipo	Obrigatório
Nome do curso	Identificação do curso.	Texto	Sim
Descrição	Descrição do curso	Texto	Não
Categoria	Escolher uma categoria cadastrada	Numérico	Sim
Área	Escolher uma área cadastrada	Numérico	Sim
Requisitos	Os requisitos necessários para realizar o curso.	Texto	Não
Observações	Campo para inclusão de informações adicionais sobre o curso	Texto	Não

Quadro 5 – Campos de entrada do cadastro de cursos

A listagem dos campos de entrada do cadastro de turmas é apresentada no Quadro 6.

Dado	Descrição	Tipo	Obrigatório
Nome da turma	Identificação da turma	Texto	Sim
Curso	Campo para seleção do curso.	Numérico/Caractere	Sim
Ambiente	Campo para seleção do ambiente.	Numérico/Caractere	Sim
Data de início do curso	Campo para informação da data de início do curso	Data	Sim
Data de finalização do curso	Campo para informação da data de finalização do curso	Data	Sim
Horário	Campo para informação do horário das aulas.	Hora	Sim
Carga Horária	Campo para a informação da carga horário do curso	Numérico	Sim
Data de início das inscrições	Campo para informação da data de início das inscrições para o curso	Data	Sim
Data fim das inscrições	Campo para informação da data de finalização das inscrições para o curso	Data	Sim
Limite de Vagas	Campo para a informação do número de limite de vagas	Numérico	Sim
Observações	Campo para inclusão de informações referentes à turma	Texto	Não
Contato	Campo para informação do nome do contato responsável pela turma	Texto	Não
Email Contato	Campo para informação do <i>email</i> do contato responsável pela turma	Texto	Não
Dias da semana	Dias da semana nas quais o curso será realizado	Texto	Não

Quadro 6 – Campos de entrada do cadastro de cursos

No Quadro 7 está a listagem dos campos de entrada do cadastro de responsáveis.

Dado	Descrição	Tipo	Obrigatório
Responsável	Campo para seleção do responsável.	Numérico	Sim
Pessoa	Campo para seleção da pessoa.	Numérico	Sim
Papel	Campo para seleção do papel.	Numérico	Sim
Turma	Campo para seleção da turma.	Numérico	Sim

Quadro 7 – Campos de entrada do cadastro de responsáveis

Os campos de entrada do cadastro de alunos no Quadro 8. Os alunos realizam inscrições em turmas de cursos.

Dado	Descrição	Tipo	Obrigatório
Nome do aluno	Identificação do aluno.	Texto	Sim
Telefone	Telefone de contato.	Texto	Não
Celular	Telefone celular de contato.	Texto	Não
Email	<i>Email</i> do aluno.	Texto	Não
Cidade	Cidade de residência do aluno.	Texto	Não
Instituição	Local de trabalho do aluno ou instituição a que o mesmo está vinculado.	Texto	Não
Login	O <i>login</i> do aluno é o número do Cadastro de Pessoa Física (CPF) do aluno.	Texto	Sim
Senha	Senha cadastrada pelo aluno para acesso ao sistema.	Texto	Sim

Quadro 8 – Campos de entrada do cadastro de alunos

Os campos de entrada do cadastro de ambientes (Quadro 9) são para o cadastro dos ambientes nos quais as atividades de extensão são realizadas.

Dado	Descrição	Tipo	Obrigatório
Nome do ambiente	Identificação do ambiente.	Texto	Sim
Descrição	Descrição do ambiente.	Texto	Não
Localização	Local da realização do curso	Texto	Não

Quadro 9 – Campos de entrada do cadastro de ambientes

Os campos de entrada do cadastro de matrículas em uma turma de um curso são apresentados no Quadro 10.

Dado	Descrição	Tipo	Obrigatório
Turma	Campo para seleção da turma desejada.	Numérico	Sim
Aluno	Aluno já cadastrado, quem está realizando a inscrição.	Numérico	Sim
Ordem de Inscrição	Campo responsável pela listagem dos alunos inscritos no curso. Campo oculto.	Auto-incremento	Sim

Quadro 10 – Campos de entrada do cadastro de matrículas

No Quadro 11 estão os campos de entrada do cadastro de controles. Os controles são usados para definir se os alunos de um determinado curso recebem ou não certificado.

Dado	Descrição	Tipo	Obrigatório
Conceito	Conceito atribuído a determinado aluno. Os conceitos são previamente cadastrados.	Numérico	Sim
Matrícula	Identificação do aluno e do curso que o mesmo realizou. Um curso é instanciado por uma turma.	Numérico	Sim
Faltas	Quantidade de faltas que cada	Numérico	Não

	aluno teve no curso.		
Nota	Resultado do aproveitamento do aluno no curso.	Numérico	Não
Conclusão	Indica se o aluno conclui o curso com aproveitamento e frequência, isto é, se pode receber comprovante.	Numérico	Não

Quadro 11 – Campos de entrada do cadastro de controles

A listagem do Quadro 12 apresenta os campos de entrada do cadastro de categorias.

Dado	Descrição	Tipo	Obrigatório
Nome da categoria	Identificação da categoria.	Texto	Sim
Descrição	Descrição da categoria. Exemplos: Redes, programação, análise, banco de dados, multimídia, etc.	Texto	Não

Quadro 12 – Campos de entrada do cadastro de categorias

No Quadro 13 está a listagem dos campos de entrada do cadastro de áreas.

Dado	Descrição	Tipo	Obrigatório
Nome da área	Identificação da área.	Texto	Sim
Descrição	Descrição da área.	Texto	Não

Quadro 13 – Campos de entrada do cadastro de áreas

A listagem dos campos de entrada do cadastro de conceitos é apresentada no Quadro 14.

Dado	Descrição	Tipo	Obrigatório
Nome do conceito	Identificação do conceito.	Texto	Sim
Descrição	Descrição do conceito. Aprovado ou Reprovado.	Texto	Não

Quadro 14 – Campos de entrada do cadastro de conceitos

A listagem dos campos de entrada do cadastro de papéis é apresentada no Quadro 15. Papéis são as funções exercidas pelas pessoas envolvidas na realização do curso.

Dado	Descrição	Tipo	Obrigatório
Nome do papel	Identificação do papel: coordenador, organizador, ministrante, palestrante.	Texto	Sim
Descrição	Descrição do papel como complemento ao nome.	Texto	Não

Quadro 15 – Campos de entrada do cadastro de papéis

Os campos de entrada do cadastro de pessoas no Quadro 16. Pessoas são responsáveis pela realização dos cursos e exercem papéis.

Dado	Descrição	Tipo	Obrigatório
Nome da pessoa	Identificação da pessoa.	Texto	Sim
Telefone	Telefone de contato.	Texto	Não
Email	<i>Email</i> do aluno.	Texto	Não
Email alternativo	Segundo <i>email</i> para contato	Texto	Não
Login	O <i>login</i> para acesso ao sistema é o número do CPF.	Texto	Sim
Senha	Senha cadastrada pelo aluno para acesso ao sistema.	Texto	Sim

Quadro 16 – Campos de entrada do cadastro de pessoas

Os campos do cadastro de usuários (Quadro 17). Esse cadastro armazena os dados do usuário com permissões de administrador.

Dado	Descrição	Tipo	Obrigatório
Nome do usuário	Identificação do usuário.	Texto	Sim
Senha	Senha cadastrada para acesso ao sistema.	Texto	Sim

Quadro 17 – Campos de entrada do cadastro de usuários

No Quadro 18 está a descrição dos casos de uso que não se referem especificamente aos cadastros com operações padrão de inclusão, exclusão, consulta e alteração.

<p>Identificador do caso de uso: Alterar dados pessoais.</p> <p>Descrição: O usuário pode alterar os seus dados pessoais incluindo <i>login</i> e senha, após ter sido cadastrado pelo administrador com o <i>email</i> como <i>login</i> e uma senha padrão.</p> <p>Evento Iniciador: O usuário acessa a tela de alteração de dados pessoais. Para isso o usuário deve estar logado ao sistema.</p> <p>Atores: Professor, Administrador</p> <p>Pré-condição: O usuário deve já ter sido cadastrado pelo administrador.</p> <p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. Ator acessa tela para cadastro de um novo projeto e incluindo as informações necessárias. O tipo de projeto, um dos campos de entrada, deve estar cadastrado e é escolhido de uma listagem apresentada. 2. O sistema insere os dados no banco de dados, verificando se o nome do projeto está descrito e informa o usuário que o referido projeto foi incluído. <p>Pós-Condição: Projeto inserido no banco de dados.</p> <p>Extensões: Cadastrar curso.</p>
--

Quadro 18 – Caso de uso alterar dados pessoais

Na Figura 4 é apresentado o diagrama de classes. Nesse diagrama não constam os métodos para facilitar a visualização. Os métodos e outros elementos que compõem as classes são apresentados na sequência à figura.

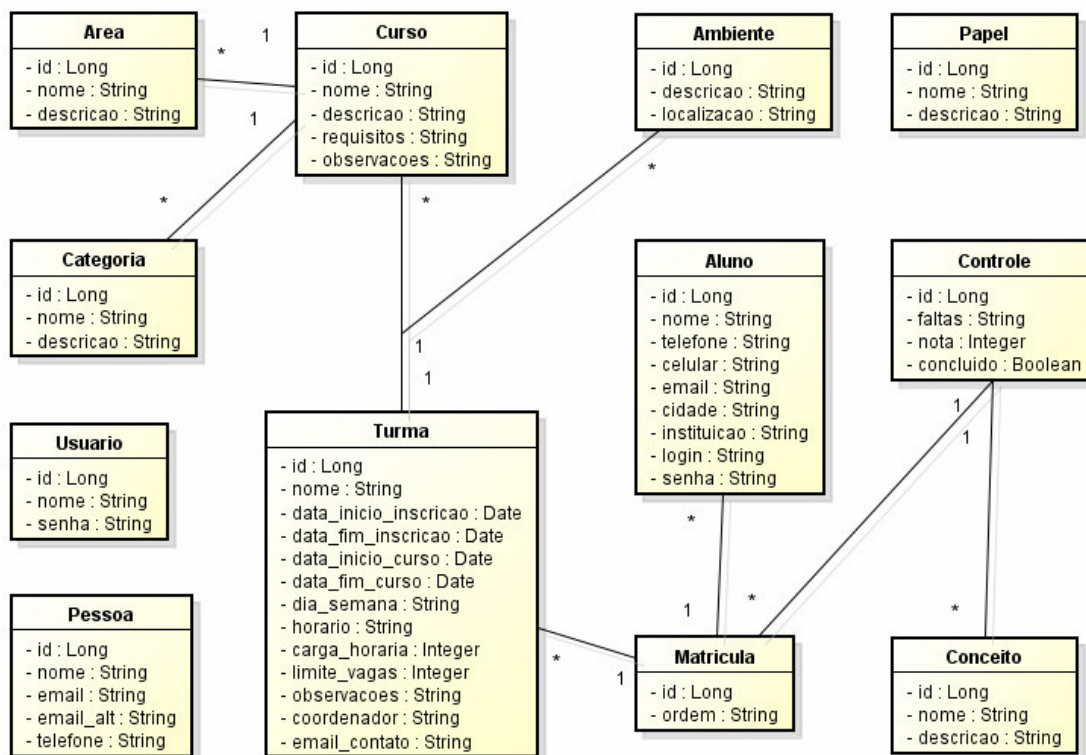


Figura 4 – Diagrama de classes

As classes apresentadas no diagrama da Figura 3 estão documentadas a seguir. O Quadro 19 apresenta a classe Pessoa.

Identificação:	Pessoa
Descrição:	Pessoa que terá um cadastro no site.
Requisitos:	RF03
Atributos:	id (número): campo chave primária no banco de dados da respectiva tabela. nome (string): nome da pessoa. email (string): email de contato da pessoa. email_alt (string): email alternativo para contato. telefone (string): telefone de contato da pessoa. login (string): login para acesso da pessoa. senha (string): senha para acesso da pessoa.
Métodos:	Boolean salvar(); Void carregar(ID); Boolean deletar(ID); Array getAll();

Quadro 19 – Descrição da classe Pessoa

No Quadro 20 está a apresentação da classe Usuario.

Identificação:	Usuário
Descrição:	Usuário é a pessoa que possui acesso ao sistema, podendo usar as funções do site.
Requisitos:	RF015
Atributos:	id (número): campo chave primária no banco de dados da respectiva tabela. nome (string): nome do usuário. senha (string): para acesso ao sistema.
Métodos:	Boolean salvar(); Void carregar(ID); Boolean deletar(ID); Array getAll();

Quadro 20 – Descrição da classe Usuario

A classe Categoria é apresentada no Quadro 21.

Identificação:	Categoria
Descrição:	Categoria é relacionada ao curso, definindo uma especificação para o mesmo.
Requisitos:	RF016
Atributos:	id (número): campo chave primária no banco de dados da respectiva tabela. nome (string): nome da categoria. descricao (string): descrição da categoria.
Métodos:	Boolean salvar(); Void carregar(ID); Boolean deletar(ID); Array getAll();

Quadro 21 – Descrição da classe Categoria

No Quadro 22 é apresentada a classe Area.

Identificação:	Area
Descrição:	Área está relacionada ao curso, definindo uma especificação para o mesmo.
Requisitos:	RF02
Atributos:	id (número): campo chave primária no banco de dados da respectiva tabela. nome (string): nome da área. descricao (string): descrição da área.
Métodos:	Boolean salvar(); Void carregar(ID); Boolean deletar(ID); Array getAll();

Quadro 22 – Descrição da classe Area

No Quadro 23 é apresentada a descrição da classe dos Cursos.

Identificação:	Curso
Descrição:	Curso é relacionado à Turma, permitindo o controle posterior do aproveitamento.
Requisitos:	RF03, RF04.
Atributos:	id (número): campo chave primária no banco de dados da respectiva tabela. nome (string): nome do curso. descricao (string): descrição do curso. requisitos (string): requisitos do curso. observacoes (string): observações do curso
Métodos:	Boolean salvar(); Void carregar(ID); Boolean deletar(ID); Array getAll();

Quadro 23 – Descrição da classe Curso

No Quadro 24 está a apresentação da classe Ambiente.

Identificação:	Ambiente
Descrição:	Ambiente no qual será realizado o curso. Classe relacionada a classe Turma.
Requisitos:	RF16
Atributos:	id (número): campo chave primária no banco de dados da respectiva tabela. descricao (string): descrição do ambiente. localizacao (string): localização referente ao ambiente.
Métodos:	Boolean salvar(); Void carregar(ID); Boolean deletar(ID); Array getAll();

Quadro 24 – Descrição da classe Ambiente

No Quadro 25 está a apresentação da classe Turma.

Identificação:	Turma
Descrição:	Turma é a classe responsável pelo controle dos cursos e ambientes.
Requisitos:	RF06, RF07
Atributos:	id (número): campo chave primária no banco de dados da respectiva tabela. nome (string): nome da turma. data_inicio_inscricao (date): data de início das inscrições. data_fim_inscricao (date): data de término das inscrições. data_inicio_curso (date): data de início do curso. data_fim_curso (date): data de término do curso. dia_semana (string): dia(s) da semana em que ocorrerá o curso. horario (string): horario em que o curso ocorrerá. carga_horaria (integer): carga horária do curso. limite_vagas (integer): limite de vagas para a turma.

	observacoes (string): observações relativas à turma. coordenador (string): coordenador da turma. email_contato (string): email de contato do coordenador. nome_curso (string): nome referente ao curso. nome_ambiente (string): nome referente ao ambiente. responsaveis (string): nome do responsável.
Métodos:	Boolean salvar(); Void carregar(ID); Boolean deletar(ID); Array getAll();

Quadro 25 – Descrição da classe Turma

No Quadro 26 está a apresentação da classe Papel.

Identificação:	Papel
Descrição:	O Papel de uma pessoa no controle de Turmas
Requisitos:	RF17
Atributos:	id (número): campo chave primária no banco de dados da respectiva tabela. nome (string): nome do papel. descricao (string): descrição do papel.
Métodos:	Boolean salvar(); Void carregar(ID); Boolean deletar(ID); Array getAll();

Quadro 26 – Descrição da classe Papel

No Quadro 27 está a apresentação da classe Conceito.

Identificação:	Conceito
Descrição:	Conceitos utilizados para indicar o aproveitamento de alunos das turmas.
Requisitos:	RF07
Atributos:	id (número): campo chave primária no banco de dados da respectiva tabela. nome (string): nome do conceito. descricao (string): descrição do conceito.
Métodos:	Boolean salvar(); Void carregar(ID); Boolean deletar(ID); Array getAll();

Quadro 27 – Descrição da classe Conceito

No Quadro 28 está a apresentação da classe Aluno.

Identificação:	Aluno
Descrição:	O cadastro do aluno.
Requisitos:	RF07
Atributos:	id (número): campo chave primária no banco de dados da respectiva tabela.

	nome (string): nome do aluno. telefone (string): telefone do aluno. celular (string): celular do aluno. email (string): <i>email</i> para contato com o aluno. cidade (string): cidade onde reside o aluno. instituição (string): instituição do aluno. login (string): login para acesso do aluno. senha (string): senha para acesso do aluno.
Métodos:	Boolean salvar(); Void carregar(ID); Boolean deletar(ID); Array getAll();

Quadro 28 – Descrição da classe Aluno

No Quadro 29 está a apresentação da classe Matricula.

Identificação:	Matricula
Descrição:	O cadastro da matricula.
Requisitos:	RF07
Atributos:	id (número): campo chave primária no banco de dados da respectiva tabela. ordem (string): ordem de inscrição.
Métodos:	Boolean salvar(); Void carregar(ID); Boolean deletar(ID); Array getAll();

Quadro 29 – Descrição da classe Matricula

No Quadro 30 está a apresentação da classe Controle.

Identificação:	Controle
Descrição:	O controle das turmas.
Requisitos:	RF07
Atributos:	id (número): campo chave primária no banco de dados da respectiva tabela. faltas (string): campo para informação do número de faltas do aluno. nota (integer): campo para informação da nota do aluno. Concluído (boolean): campo para definir se o aluno concluiu o curso ou não.
Métodos:	Boolean salvar(); Void carregar(ID); Boolean deletar(ID); Array getAll();

Quadro 30 – Descrição da classe Controle

A Figura 5 apresenta o diagrama de entidades e relacionamentos do Banco de Dados.

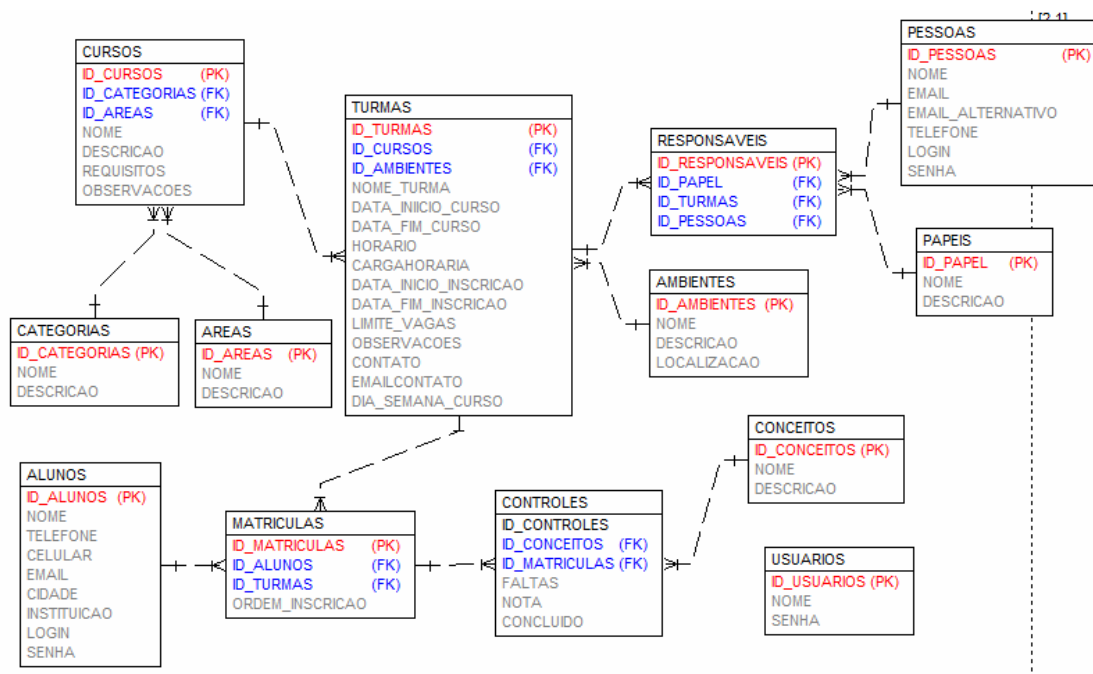


Figura 5 – Diagrama de entidades e relacionamentos do banco de dados

A seguir está a descrição das tabelas que compõem o banco de dados, conforme expõe a Figura 5.

Tabela Pessoas (Quadro 31) - uma pessoa possui *login* e senha e nesse caso ele tem acesso ao sistema. Um usuário sem *login* ou senha pode possuir papéis relacionados às atividades. Pessoas realizam as atividades. Uma pessoa tem acesso somente às atividades que é responsável.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
ID_PESSOAS	Número	Não	Sim	Não	
NOME	Texto	Não	Não	Não	
EMAIL	Texto	Não	Não	Não	
EMAIL_ALT	Texto	Sim	Não	Não	
TELEFONE	Texto	Sim	Não	Não	
LOGIN	Texto	Não	Não	Não	
SENHA	Texto	Não	Não	Não	

Quadro 31 – Tabela pessoas

Tabela Cursos (Quadro 32) - os cursos serão utilizados para gerar as turmas.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
ID_CURSOS	Número	Não	Sim	Não	
ID_CATEGORIAS	Número	Não	Não	Sim	Vem da tabela Categorias
ID_AREAS	Número	Não	Não	Sim	Vem da tabela Areas
NOME	Texto	Não	Não	Não	
DESCRICAÇÃO	Texto	Não	Não	Não	
REQUISITOS	Texto	Sim	Não	Não	
OBSERVACOES	Texto	Sim	Não	Não	

Quadro 32 – Tabela cursos

Tabela Categorias (Quadro 33) - categorias vinculadas aos cursos. As categorias permitem organizar os cursos por tipos.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
ID_CATEGORIA	Número	Não	Sim	Não	
NOME	Texto	Não	Não	Não	
DESCRICAÇÃO	Texto	Sim	Não	Não	

Quadro 33 – Tabela categorias

Tabela Areas (Quadro 34) - áreas vinculadas aos cursos.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
ID_AREAS	Número	Não	Sim	Não	
NOME	Texto	Não	Não	Não	
DESCRICAÇÃO	Texto	Sim	Não	Não	

Quadro 34 – Tabela areas

Tabela Papeis (Quadro 35) – cadastro de papéis para serem vinculados com o cadastro de pessoas. Os papéis definem as atribuições das pessoas às atividades.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
ID_PESSOAS	Número	Não	Sim	Não	
NOME	Texto	Não	Não	Não	
DESCRICAÇÃO	Texto	Não	Não	Não	

Quadro 35 – Tabela papeis

Tabela Ambientes (Quadro 36) – cadastro de ambientes para informar as pessoas sobre a localização dos cursos.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
ID_AMBIENTES	Número	Não	Sim	Não	
NOME	Texto	Não	Não	Não	
DESCRICAO	Texto	Não	Não	Não	
LOCALIZACAO	Texto	Não	Não	Não	

Quadro 36 – Tabela ambientes

Tabela ResponsaveisTumaCurso (Quadro 37) – cadastro que associa os responsáveis pelas atividades com os papéis que elas realizam em turmas de cursos.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
ID_RESPONSAVEIS	Número	Não	Sim	Não	
ID_PESSOAS	Número	Não	Não	Sim	Vem da tabela Pessoas
ID_PAPEIS	Número	Não	Não	Sim	Vem da tabela Papeis
ID_TURMAS	Número	Não	Não	Sim	Vem da tabela Turmas

Quadro 37 – Tabela responsaveistumacurso

Tabela Turmas (Quadro 38) – cadastro de turmas vinculadas aos cursos cadastrados.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
ID_TURMAS	Número	Não	Sim	Não	
ID_CURSOS	Número	Não	Não	Sim	Vem da tabela Cursos
ID_AMBIENTES	Número	Não	Não	Sim	Vem da tabela Ambientes
NOME_TURMA	Texto	Não	Não	Não	Vem da tabela Turmas
DATA_INICIO_CURSO	Data	Não	Não	Não	
DATA_FIM_CURSO	Data	Não	Não	Não	
HORARIO	Número	Não	Não	Não	
CARGA_HORARIA	Número	Não	Não	Não	
DATA_INICIO_INSCRICAO	Data	Não	Não	Não	
DATA_FIM_INSCRICAO	Data	Não	Não	Não	
LIMITE_VAGAS	Número	Não	Não	Não	
OBSERVACOES	Texto	Não	Não	Não	
CONTATO	Texto	Não	Não	Não	

EMAIL_CONTATO	Texto	Não	Não	Não	
DIAS_SEMANA_CURSO	Texto	Não	Não	Não	

Quadro 38 – Tabela turmas

Tabela Conceitos (Quadro 39) – cadastro de conceitos relacionados aos dados do aluno na finalização do curso.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
ID_CONCEITOS	Número	Não	Sim	Não	
NOME	Texto	Não	Não	Não	
DESCRICAO	Texto	Não	Não	Não	

Quadro 39 – Tabela conceitos

Tabela Controles (Quadro 40) – cadastro de controle sobre a frequência e aproveitamento do curso.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
ID_CONTROLE	Número	Não	Sim	Não	
ID_CONCEITOS	Número	Não	Não	Sim	Vem da tabela Conceitos
ID_MATRICULAS	Número	Não	Não	Sim	Vem da tabela Matriculas
FALTAS	Número	Sim	Não	Não	
NOTA	Número	Sim	Não	Não	
CONCLUIDO	Número	Sim	Não	Não	

Quadro 40 – Tabela controles

Tabela Alunos (Quadro 41) – os alunos que podem realizar inscrições em cursos.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
ID_ALUNOS	Número	Não	Sim	Não	
NOME	Texto	Não	Não	Não	
TELEFONE	Texto	Sim	Não	Não	
CELULAR	Texto	Sim	Não	Não	
EMAIL	Texto	Não	Não	Não	
CIDADE	Texto	Sim	Não	Não	
INSTITUICAO	Texto	Sim	Não	Não	
LOGIN	Texto	Não	Não	Não	
SENHA	Texto	Não	Não	Não	

Quadro 41 – Tabela alunos

Tabela Matriculas (Quadro 42) – são os alunos matriculados em determinada turma de um curso.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
ID_MATRICULAS	Número	Não	Sim	Não	
ID_ALUNOS	Numérico	Não	Não	Sim	Vem da tabela Alunos
ID_TURMAS	Numérico	Não	Não	Sim	Vem da tabela Turmas
ORDEM_INSCRICAO	Numérico	Não	Não	Não	

Quadro 42 – Tabela matriculas

Tabela Usuarios (Quadro 43) – é o cadastro de usuários com permissão de administrador.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
ID_USUARIO	Número	Não	Sim	Não	
NOME	Texto	Não	Não	Não	
SENHA	Texto	Não	Não	Não	

Quadro 43 – Tabela usuarios

4.3 DESCRIÇÃO DO SISTEMA

A Figura 6 apresenta a tela de *login*. Para acessar o sistema é necessário informar nome de *login* e senha.

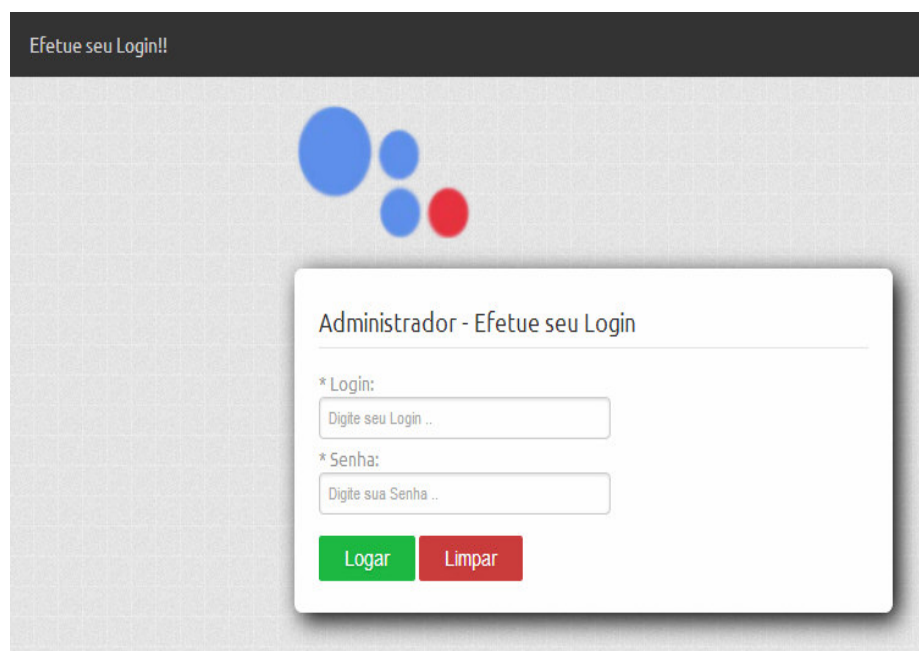


Figura 6 – Tela de login

A Figura 7 apresenta a tela inicial do sistema (que é apresentada após o *login* do usuário). Nessa tela está o menu de navegação.

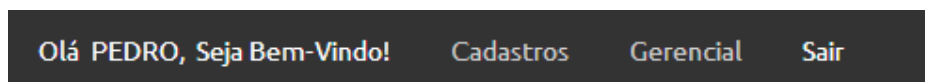


Figura 7 – Menu de Navegação

No cabeçalho da página estão as opções 'Cadastros' e 'Gerencial' responsáveis por apresentar o restante das opções. Ao selecionar a opção Cadastro são listados todos os cadastros do sistema, conforme apresenta a Figura 8.

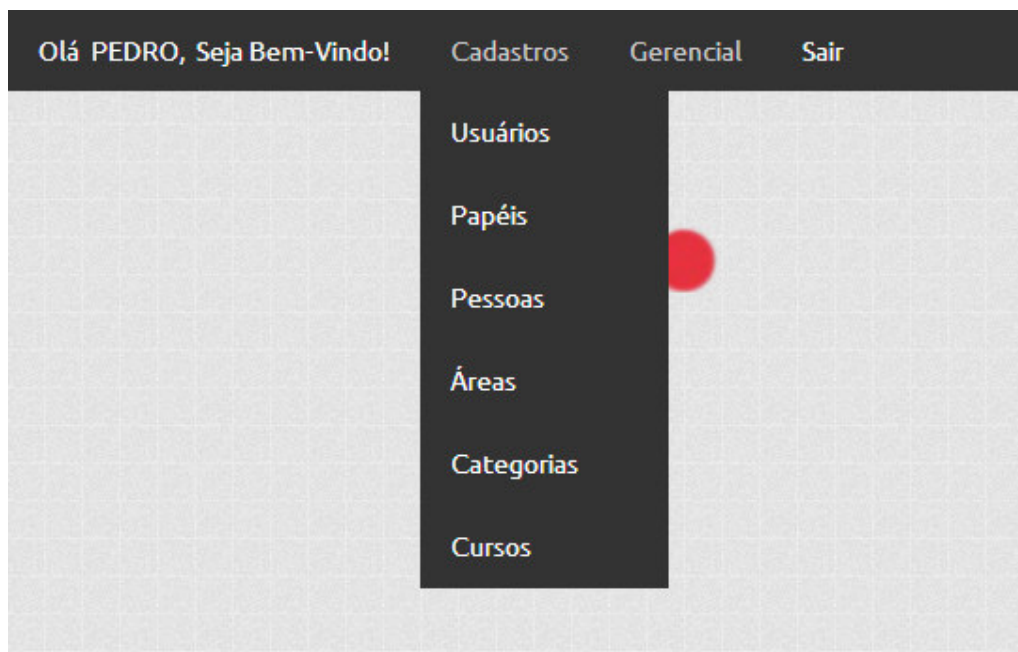


Figura 8 – Cadastros

O sistema conterá vários *grids*. A Figura 9 apresenta um desses *grids*. Os *grids* serão responsáveis para exibir dados cadastrados. A imagem da Figura 9 mostra os usuários cadastrados no sistema e são apresentadas na listagem apenas o código e *login*, a senha não é apresentada. Na parte superior do *grid* há uma lista de opções, referente a cada tipo de dado. As opções 'Novo, Alterar e Excluir' estarão praticamente em todos os *grids* do sistema. 'Novo' carregará a página, responsável pela inserção de novos alunos. 'Alterar' carregará a página de cadastro de aluno e também os dados do aluno selecionado. 'Excluir' excluirá o aluno selecionado.



Figura 9 – Tela de cadastro de usuários

Caso um usuário tenha sido selecionado, ao pressionar o botão 'Excluir' o mesmo será excluído do banco de dados. Uma janela é apresentada questionando o usuário se o mesmo tem certeza que deseja excluir o registro selecionado. A ação será realizada somente se o usuário informar afirmativamente, pressionando o botão 'OK'. Porém, se o cadastro estiver ativo, ele não poderá ser excluído, apresentando a mensagem que está destacada na Figura 10.

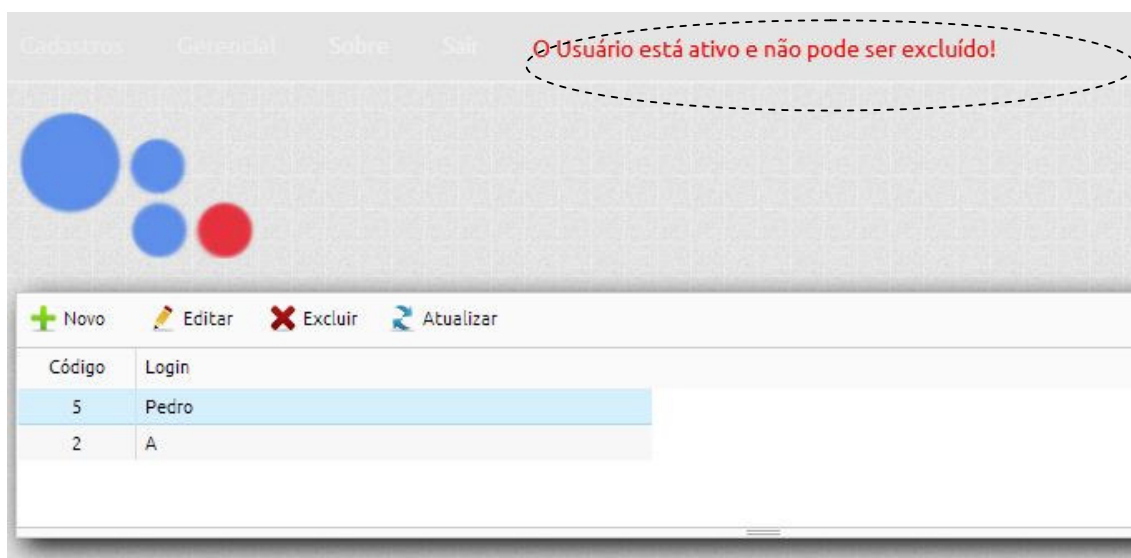


Figura 10 – Janela para confirmação de exclusão de registro

Clicando em 'Novo' ou 'Editar', o usuário será direcionado para a tela de cadastro de usuários (Figura 11). Se for selecionada a opção 'Novo' será aberto um formulário para inclusão de um novo cadastro de usuário. Se for selecionada a opção 'Editar' haverá o direcionamento para mesma página do 'Novo', mas com 'Login' já preenchido.



The image shows a web application interface with a dark header containing the menu items 'Cadastros', 'Gerencial', and 'Sair'. Below the header is a decorative graphic of four circles (three blue, one red) on a light gray grid background. The main content is a white modal form titled 'Usuário - Cadastro'. The form contains three required input fields: 'Login' (placeholder: 'Digite seu Login ..'), 'Senha' (placeholder: 'Digite sua Senha ..'), and 'Confirme sua Senha:' (placeholder: 'Confirme sua Senha ..'). At the bottom of the form are two buttons: a green 'Salvar' button and a red 'Voltar' button.

Figura 11 – Página carregada pela opção 'Novo'

A Figura 12 apresenta o cadastro de usuário carregado quando selecionada a opção 'Editar'.

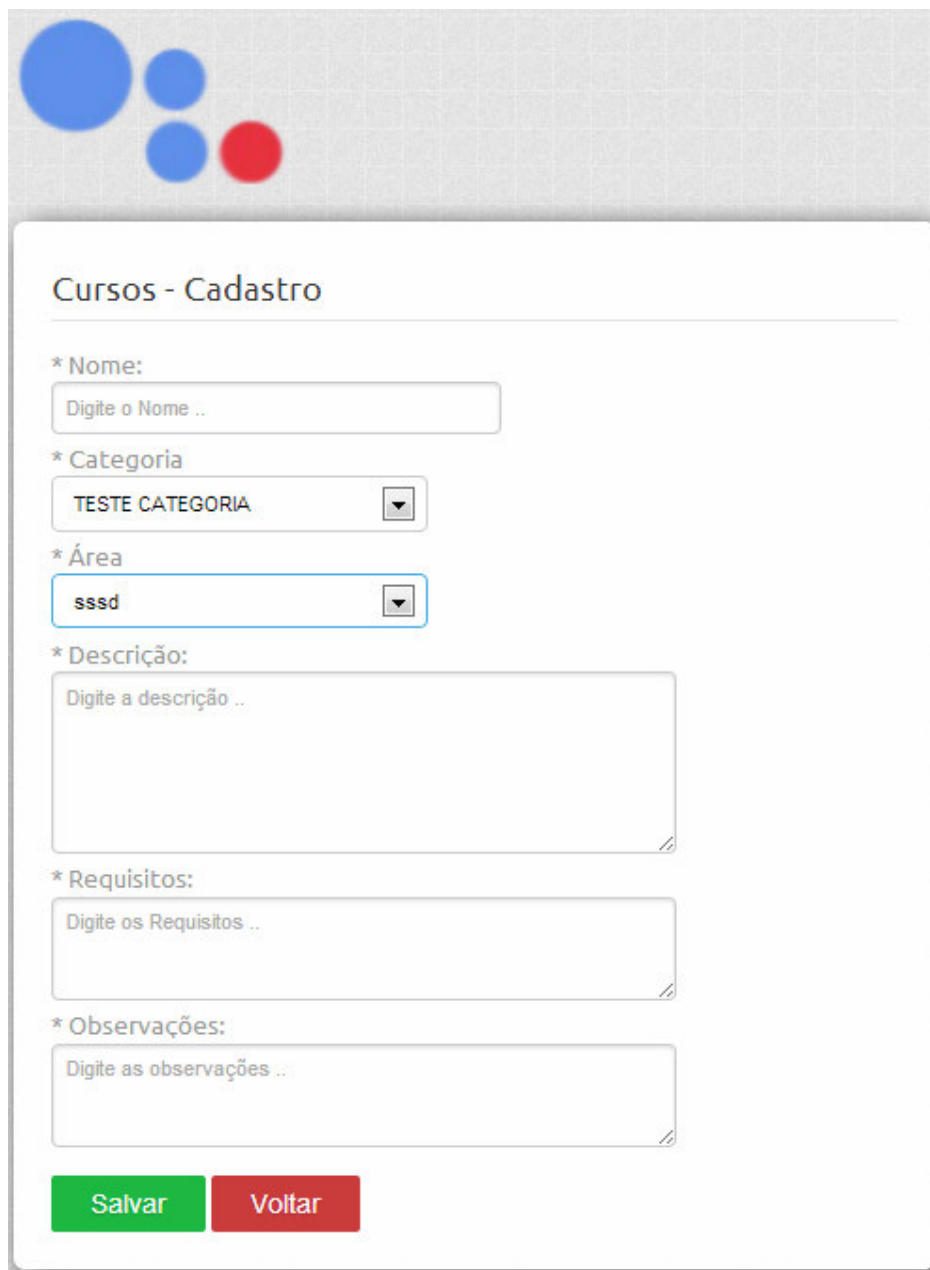


The image shows a web application interface with a dark header containing the menu items 'Cadastros', 'Gerencial', and 'Sair'. Below the header is a logo consisting of four circles: one large blue circle, two smaller blue circles, and one red circle. The main content area features a white modal window titled 'Usuário - Cadastro'. Inside this modal, there are three required input fields: '* Login:' with the value 'Pedro', '* Senha:' with the placeholder 'Digite sua Senha ..', and '* Confirme sua Senha:' with the placeholder 'Confirme sua Senha ..'. At the bottom of the modal are two buttons: a green 'Salvar' button and a red 'Voltar' button.

Figura 12 – Página carregada pela opção 'Editar'

Quando selecionada a opção 'Editar' (Figura 12), os dados do formulário são preenchidos, possibilitando a alteração do cadastro. Na parte inferior da tela de cadastro estão os botões, 'Salvar' e 'Voltar'. Esse último ao ser clicado retornará à página do cadastro de usuários. Já o botão 'Salvar' armazenará os dados inseridos nos campos e também atualizará os dados, no caso de alteração.

A Figura 13 apresenta o cadastro de Cursos. Para o cadastro de cursos são apresentadas as categorias e as áreas já cadastradas.



The image shows a web form titled "Cursos - Cadastro". At the top left, there is a logo consisting of four circles: one large blue circle, one medium blue circle, one small blue circle, and one small red circle. The form itself is white with a thin border. It contains the following elements:

- * Nome:** A text input field with the placeholder text "Digite o Nome ..".
- * Categoria:** A dropdown menu with the selected value "TESTE CATEGORIA".
- * Área:** A dropdown menu with the selected value "sssd".
- * Descrição:** A large text area with the placeholder text "Digite a descrição ..".
- * Requisitos:** A text area with the placeholder text "Digite os Requisitos ..".
- * Observações:** A text area with the placeholder text "Digite as observações ..".
- At the bottom, there are two buttons: a green "Salvar" button and a red "Voltar" button.

Figura 13 – Página de cadastro de ‘Cursos’

Uma das funcionalidades principais do sistema é o cadastro de turmas (a tela é apresentada na Figura 14). Nessa tela são carregados os cursos e os ambientes já cadastrados. Uma turma representa um evento de realização de um curso. Por exemplo, o curso de Redes Básico pode ser ministrado em períodos distintos e cada realização do mesmo está associada a uma turma.

Turmas - Cadastro

* Curso:
Teste categoria

* Ambiente:
ambiente informatica

* Nome:
Digite o Nome ..

Período Matrícula

Data Inicial: Data Final:
Data Inicial .. Data Final ..

Período Curso

Data Inicial: Data Final:
Data Inicial .. Data Final ..

Figura 14 – Página de cadastro de ‘Turmas’

Ao clicar no botão ‘...’ (três pontos) (áreas destacadas da Figura 14), o sistema carrega uma tela, conforme apresenta a Figura 15 para a busca de registros. Na Figura 15 está a tela para a ‘Procura de Ambientes’.

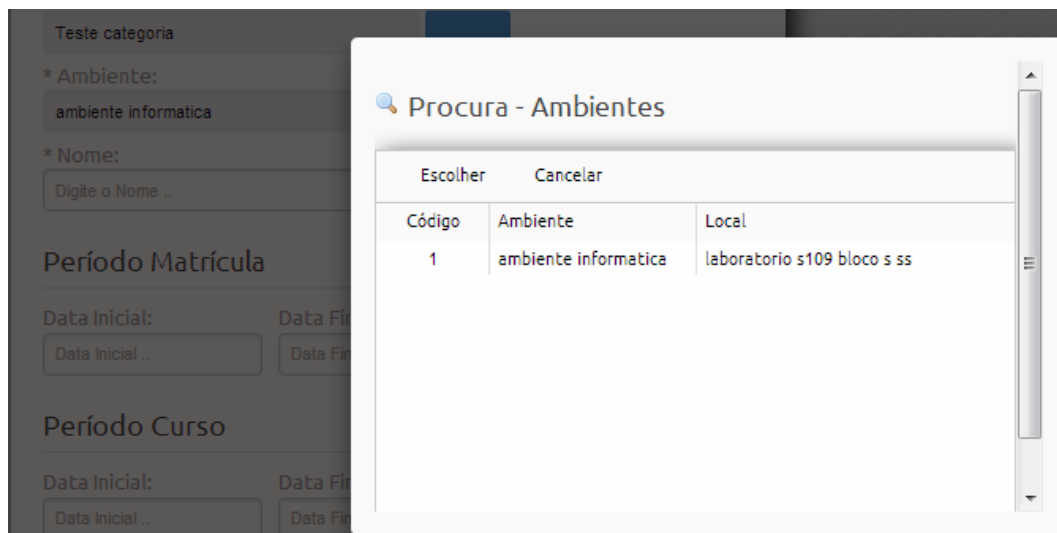


Figura 15 – Página de procura de ‘Ambientes’

Ainda na mesma tela da Figura 14 está a opção de inclusão de responsáveis por ministrar o curso, conforme é apresentado na Figura 16.



Figura 16 – Página de inclusão de ‘Responsáveis’

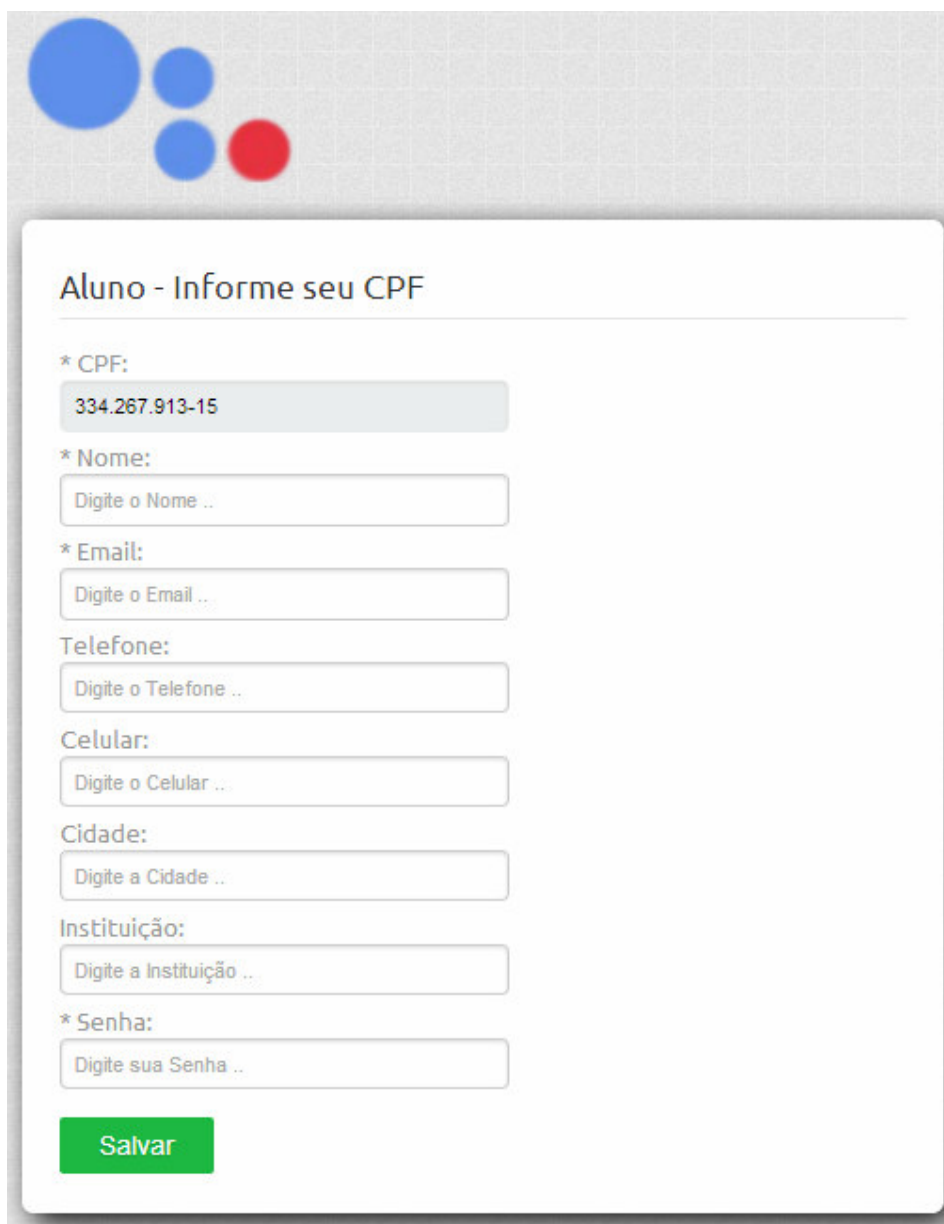
Quando selecionada a opção ‘Incluir’ (Figura 16), os campos ‘Pessoa’ e ‘Papel’ são transferidos para a listagem que vai sendo composta abaixo do botão ‘Incluir’. Nessa lista está a opção ‘Excluir’. Na parte inferior da tela de cadastro estão os botões, ‘Salvar’ e ‘Voltar’. Esse último ao ser clicado retornará à página do cadastro de Turmas. Já o botão ‘Salvar’ armazenará os dados inseridos nos campos e também atualizará os dados, no caso de alteração.

Na Figura 17 está o primeiro acesso de um aluno. Para acessar o sistema é necessário informar o número do Cadastro de Pessoa Física (CPF).

The image shows a web form with a light gray background. In the top left corner, there are four circles: one large blue circle, one medium blue circle, one small blue circle, and one small red circle. The form itself is a white rounded rectangle with a thin gray border. At the top of the form, the text 'Aluno - Informe seu CPF' is displayed in a dark gray font. Below this, there is a label '* CPF:' followed by a text input field with the placeholder text 'Digite seu CPF ..'. At the bottom of the form, there is a green button with the text 'Continuar' in white.

Figura 17 – Página de informação de ‘CPF’

Ao preencher o CPF corretamente, se o aluno não estiver cadastrado (primeiro acesso) será apresentada a tela da Figura 18.



The image shows a web form for student registration. At the top left, there is a logo consisting of four circles: one large blue circle, one medium blue circle, one small blue circle, and one small red circle. Below the logo, the title of the form is "Aluno - Informe seu CPF". The form contains several input fields, each with a label and a placeholder text. The labels are: "* CPF:", "* Nome:", "* Email:", "Telefone:", "Celular:", "Cidade:", "Instituição:", and "* Senha:". The CPF field contains the value "334.267.913-15". The other fields have placeholder text: "Digite o Nome ..", "Digite o Email ..", "Digite o Telefone ..", "Digite o Celular ..", "Digite a Cidade ..", "Digite a Instituição ..", and "Digite sua Senha ..". At the bottom of the form, there is a green button labeled "Salvar".

Figura 18 – Página de cadastro de ‘Aluno’

Ao preencher os campos obrigatórios (que são precedidos de um * (asterisco)) e clicar em ‘Salvar’, o aluno será redirecionado para a tela que representa a Figura 19.



Figura 19 – Página de inscrição.

Nesta tela da Figura 19 são apresentados todos os cursos listados, cada um com a sua descrição, seu período, horário e carga horária. São apresentados os cursos que possuem período de inscrição ativo.

4.4 IMPLEMENTAÇÃO DO SISTEMA

Nesta seção são apresentadas algumas das funcionalidades implementadas visando agilizar e facilitar a implementação do sistema. Também são exemplificadas as operações realizadas em cadastros. A Listagem 1 apresenta a classe MySQL para realizar a conexão.

```
<?php
class MySQL
{
    private static $server = "mysql.c4team.com";
    private static $user = "belutto";
    private static $pass = "teste";
    private static $db = "belutto";

    public static $handle;

    public static function connect()
    {
        MySQL::$handle = mysql_connect(MySQL::$server, MySQL::$user, MySQL::$pass) or die("Não foi Possível Conectar no Servidor::$server");
        mysql_select_db(MySQL::$db, MySQL::$handle) or die("Não foi Possível Selecionar o Banco::$db");
    }
}
```

```

public static function disconnect()
{
    mysql_close(MySQL::$handle);
}
}

```

Listagem 1 – MySQL

MySQL é a classe responsável pela conexão com o banco de dados. Essa classe contém uma variável ‘private’ chamada ‘connect’ que também será ‘static’, pois a conexão será compartilhada com todo o projeto. Assim, não será necessário instanciá-la toda vez, mas somente na primeira vez que o projeto for compilado ou postado em algum servidor. Trabalhar com banco de dados, abrindo e fechando conexão pode reduzir o desempenho das transações, por isso uma conexão será mantida sempre ativa. Essa mesma classe conterá um método para acessar a variável ‘connect’.

A Listagem 2 apresenta o código do AlunoController. Essa classe contém métodos públicos e permitem acesso às variáveis. O conjunto em si caracteriza exatamente todos os campos contidos em uma tabela no banco de dados. Todas as telas conterão um *controler*, exatamente igual a sua quantidade de campos.

```

<?php
require_once("Aluno.php");

class AlunoController extends Controller
{
    function __construct() {
        $this->model = new Aluno();
    }

    public function salvar($id, $nome, $telefone, $celular, $email, $cidade,
    $instituicao, $login, $senha) {

        $this->model->id = $id;
        $this->model->nome = $nome;
        $this->model->telefone = $telefone;
        $this->model->celular = $celular;
        $this->model->email = $email;
        $this->model->cidade = $cidade;
        $this->model->instituicao = $instituicao;
        $this->model->login = $login;
        $this->model->senha = $senha;

        $this->model->salvar();

        if($this->model->id > 0) {
            $this->model->toSession();
        }
    }
}

```

```

        return $this->model;
    }else{
        return false;
    }
}

public function carregar() {
    $this->model->carregar($_SESSION['ALUNO']['ID']);
    return $this->model;
}

public function logar($login, $senha) {
    $id = $this->model->logar($login, $senha);
    if($id > 0) {
        $this->model->carregar($id);
        $this->model->toSession();
        return $this->model;
    }else{
        return false;
    }
}

public function exists($cpf) {
    $this->model->login = $cpf;
    return $this->model->exists();
}

public function logged() {
    return alunoLogado();
}

public function logout() {
    $this->model->logout();
    return true;
}
}

```

Listagem 2 – AlunoController: cadastro e listagem de alunos

Aluno (Listagem 3) é a classe responsável por todas as interações com o banco de dados. Essa classe ficará na pasta ‘Model’ na qual existem várias outras classes, seguindo o mesmo padrão.

```

<?php

class Aluno extends Model{

    public $nome;
    public $telefone;
    public $celular;
}

```

```

public $email;
public $cidade;
public $instituicao;
public $login;
public $senha;

function __construct(){
    parent::__construct('ALUNOS');
}

public function carregarFromCPF($cpf){
    $rs = $this->queryOne("SELECT ID_ALUNOS, NOME, TELEFONE, CELULAR,
EMAIL,
                CIDADE, INSTITUICAO, LOGIN, SENHA FROM ALUNOS WHERE LOGIN =
$cpf ");

    $this->loadFields($rs);
}

public function carregar($id){
    $rs = $this->queryOne("SELECT ID_ALUNOS, NOME, TELEFONE, CELULAR,
EMAIL,
                CIDADE, INSTITUICAO, LOGIN, SENHA FROM ALUNOS WHERE ID_ALUNOS =
$id ");

    $this->loadFields($rs);
}

private function loadFields($rs){
    $this->id = $rs['ID_ALUNOS'];
    $this->nome = $rs['NOME'];
    $this->telefone = $rs['TELEFONE'];
    $this->celular = $rs['CELULAR'];
    $this->email = $rs['EMAIL'];
    $this->cidade = $rs['CIDADE'];
    $this->instituicao = $rs['INSTITUICAO'];
    $this->login = $rs['LOGIN'];
    $this->senha = $rs['SENHA'];
}

public function exists(){
    $rs = $this->queryOne("SELECT ID_ALUNOS FROM ALUNOS WHERE LOGIN =
'$this->login' LIMIT 1 ");
    return (((int)$rs['ID_ALUNOS']) > 0);
}

public function logar($login, $senha){
    $this->queryExec("CALL
SP_LOGIN_ALUNOS('$login', '$senha', @ID_ALUNOS_LOGIN) ");
    $rs = $this->queryOne("SELECT COALESCE(@ID_ALUNOS_LOGIN, 0) AS
ID_ALUNOS ");
    return $rs['ID_ALUNOS'];
}

public function salvar(){

```



```

        $this->queryExec("SET @ID_ALUNOS_CAD = $this->id ");
        $this->queryExec("CALL SP_ALUNOS (@ID_ALUNOS_CAD, '$this->nome', '$this->telefone',
        '$this->celular', '$this->email', '$this->cidade', '$this->instituicao',
        '$this->login', '$this->senha' )");
        $rs = $this->queryOne("SELECT COALESCE(@ID_ALUNOS_CAD ,0) AS
ID_ALUNOS ");
        $this->id = ((int)$rs['ID_ALUNOS'] > 0) ? (int)$rs['ID_ALUNOS'] : $this->id;
    }

    public function toSession(){
        $_SESSION['ALUNO'] = array();
        $_SESSION['ALUNO']['ID'] = $this->id;
        $_SESSION['ALUNO']['NOME'] = $this->nome;
        $_SESSION['ALUNO']['LOGIN'] = $this->login;
    }

    public function logout(){
        $_SESSION['ALUNO'] = null;
    }
}
?>

```

Listagem 3 – Classe Aluno

A *procedure* SP_ALUNOS (Listagem 4) é responsável por inserir e atualizar dados. O código dessa *procedure* está na Listagem 4. Essa *procedure* fará uma verificação se está sendo realizado um *insert* ou um *update*, em seguida validará o CPF informado. Se for uma inserção um *select* é realizado para verificar se o banco de dados já não contém o valor informando. Caso o dado já esteja armazenado é retornado um valor que é tratado na aplicação, informado que os dados não puderam ser armazenados, pois o CPF informado já está contido na base de dados. O mesmo acontece, quando a operação é de atualização. O valor não poderá existir em outro cadastro de aluno.

```

CREATE PROCEDURE SP_ALUNOS (
    INOUT PID_ALUNOS INT,
    IN PNOME VARCHAR( 50 ),
    IN PTELEFONE VARCHAR( 20 ),
    IN PCELULAR VARCHAR( 20 ),
    IN PEMAIL VARCHAR( 50 ),
    IN PCIDADE VARCHAR( 50 ),
    IN PINSTITUICAO VARCHAR( 50 ),
    IN PLOGIN VARCHAR( 50 ),
    IN PSENHA VARCHAR( 50 )

```

```

)BEGIN
    INSERT INTO ALUNOS VALUES
(PID_ALUNOS, PNOME, PTELEFONE, PCELULAR, PEMAIL, PCIDADE, PINSTITUICAO, PLOGIN, PSE
NHA )
    ON DUPLICATE KEY
    UPDATE NOME = PNOME, TELEFONE= PTELEFONE, CELULAR= PCELULAR, EMAIL=
PEMAIL, CIDADE = PCIDADE, INSTITUICAO = PINSTITUICAO, LOGIN = PLOGIN, SENHA
= PSENHA;
    SELECT LAST_INSERT_ID() INTO PID_ALUNOS ;
END

```

Listagem 4 – Procedure SP_ALUNOS

Para inicializar o *plugin* de validação para os campos de texto do formulário funcionarem de acordo com a *tag required*, é necessário informar o código apresentado na Listagem 5.

```
inputs = $("#formulario #frm :input[type=text]").validator();
```

Listagem 5 – Exemplo formulário de validação

No momento de salvar dados no banco de dados, uma estrutura de condição verifica se os campos são ou não válidos. Caso não seja válido, uma mensagem é exibida ao lado dos campos. Essa validação é apresentada na Listagem 6.

```
if(inputs.data("validator").checkValidity()){
}
```

Listagem 6 – Exemplo de uso de classes de validação

Na Listagem 7 está código que valida um campo.

```

<label for="txtNome"> * Nome: </label>
<input type="text" data-message="Nome n&atilde;o Informado."
placeholder = "Digite o Nome .." name="txtNome" size="40" maxlength=
"50" id= "txtNome"required> </input>

```

Listagem 7 – Exemplo de validação de campo

Explicando um pouco melhor os códigos das Listagens 6 e 7: a validação do formulário é realizada por meio de *JavaScript*, no caso de campos de texto com a *tag 'required'* informada será então verificado se o campo está vazio. No caso de campo de *email*, é necessário validar além de o campo não estar vazio se este é um *email* com estrutura válida.

O parâmetro '*data-message*' é referente à mensagem de erro, a ser exibida no caso de o valor não atingir as exigências do campo. Se o campo estiver em branco será apresentada uma mensagem no formato: "O campo 'Nome' do aluno não poderá ficar em branco".

5 CONCLUSÃO

O objetivo principal deste trabalho foi a implementação de um aplicativo para o gerenciamento de inscrições em atividades acadêmicas de extensão realizadas pelo DAINF. Porém, ressalta-se que o aplicativo pode ser utilizado por todos os departamentos da UTFPR e mesmo por outras Universidades. Com esse aplicativo, é possível identificar os cursos cadastrados e disponíveis para realizar inscrição, os professores envolvidos e as pessoas que realizam os cursos.

A opção por um sistema para Internet visa facilitar o acesso dos usuários do sistema, principalmente para quem fará a inscrição nas atividades e também para quem é responsável pela mesma, seja professor ou aluno da UTFPR.

No desenvolvimento em PHP foi optado pela não utilização de *frameworks*, para que fosse possível entender e aprimorar o conhecimento de implementação utilizando essa linguagem em um contexto mais específico da tecnologia. Considerou-se que o aprendizado seria mais efetivo desta forma, não ficando dependentes de uma técnica ou padronização já implementada. Porém, ressalta-se que no desenvolvimento de aplicativos em PHP com finalidades comerciais o uso de *frameworks*, como o Zend, no caso do PHP, é muito importante. Eles agilizam o desenvolvimento além de organizar a forma de implementação, pelo uso de padrões de projeto, por exemplo.

No desenvolvimento deste trabalho também foram elaborados procedimentos no banco de dados, aumentando assim a segurança e o desempenho nas operações relacionadas.

A implementação desde o princípio foi realizada com o intuito de manter a compatibilidade para os navegadores mais utilizados no momento (incluindo o Internet Explorer 8). Visando, assim, evitar problemas de usabilidade.

As funcionalidades essenciais do sistema foram implementadas e alguns requisitos adicionais ficam como implementação futura. Um desses requisitos é o envio de *email* para os alunos inscritos em determinado curso como forma de confirmação da efetivação da inscrição. Essa confirmação seria enviada para os alunos que possuem a sua inscrição validada e cuja inscrição está dentro do limite de vagas. *Email* também poderá ser enviado informando que o certificado de participação está disponível e as instruções para retirá-lo.

REFERÊNCIAS

- BATTISTI, Julio. **Criando aplicações em 3, 4 ou n camadas**. Disponível em: <<http://www.juliobattisti.com.br/artigos/ti/ncamadas.asp>>. Acesso em: 17 out. 2013.
- CHO, Euan S.; KIM, Soo D.; RHEW, Sung Y.; LEE, Sang D.; KIM, Chang G. **Object-oriented web application architectures and development strategies**. Asia Pacific Software Engineering Conference and International Computer Science Conference 1997 (APSEC '97/ICSC '97), 1997, p. 322 – 331.
- CUI, Wei; HUANG, Lin; LIANG, LiJing, LI, Jing. **The research of PHP development framework based on MVC pattern**. 2009 Fourth International Conference on Computer Sciences and Convergence Information Technology, p. 947-949, 2009.
- DUHL J., **Rich internet applications**, IDC white papers. Disponível em <<http://www.idc.com>>, 2003.
- FRANÇA, Flávio A. de. **Tutorial – Entendendo Java para web (parte 1)**. 2010. Disponível em: <<http://flavioaf.wordpress.com/2010/02/25/tutorial-entendendo-java-para-web-parte-1/>>. Acesso em: 14 set. 2013.
- FUKUDA, H., YAMAMOTO, Y. **A system for supporting development of large scaled rich internet applications**. In: 23rd IEEE/ACM International Conference on Automated Software Engineering (ASE 2008), 2008, p. 459-462.
- GAMMA, Erich; JOHNSON, Ralph, 1948-; HELM, Richard; VLISSIDES, John. **Padrões de projeto: soluções reutilizáveis de software orientado a objetos**. Porto Alegre: Bookman, 2004.
- GERMOGLIO, Guilherme. **Arquitetura de software**, Texas: Rice University, 2010.
- GONÇALVES, Rodrigo F.; GAVA, Vagner Luis; PESSÔA, Marcelo S.; SPINOLA, Mauro. M. **Uma proposta de processo de produção de aplicações web**. Revista Produção, vol. 15, n. 3, p. 376-389, Set./Dez. 2005.
- HANSEN, Stuart A.; FOSSUM, Tim V. **Refactoring Model-View-Controller**. Journal of Computing Sciences in Colleges, vol. 21, no. 1, October 2005, 2005, p. 120-129.
- JOHNSON, Ralf. **Frameworks home page**. Disponível em: <<http://st-www.cs.illinois.edu/users/johnson/frameworks.html>>. Acesso em: 13 set. 2013.
- LEFF, Avraham; RAYFIELD, James T. **Web-Application development using the modelniewlcontroller design pattern**. Proceedings of the 5th IEEE International Conference on Enterprise Distributed Object Computing, p. 118-127. 2001.
- MCHEICK, Hamid; YAN Qi. **Dependency of components in MVC distributed architecture**. 24th Canadian Conference on Electrical and Computer Engineering (CCECE), p. 691-694, 2011.

MELIÁ, Santiago; GÓMEZ, Jaime; PÉREZ, Sandy; DÍAZ, Oscar. Architectural and Technological Variability in Rich Internet Applications. **IEEE Internet Computing**, may/june 2010, 2010, p. 24-32.

OKANOVIC, Vensada; MATELJAN, Tadej. **Designing a new web application framework**. Proceedings of the 34th International Convention (MIPRO 2011), p. 1315 – 1318, 2011.

PALMEIRA, Thiago V. V. **Como funcionam as aplicações web**. Disponível em: <<http://www.devmedia.com.br/como-funcionam-as-aplicacoes-web/25888>>. Acesso em: 17 out. 2013.

PHP. **Php.net**. Disponível em: <<http://www.php.net>>. Acesso em: 18 set. 2013.

PRESSMAN, Roger. **Engenharia de software**, 5ª ed. 2002. Rio de Janeiro: McGrawHill.

SANTOS, João; MENDONÇA, José; MARTINS, João C. **Instrumentation remote control through internet with PHP**. IEEE International Conference on Virtual Environments, Human-Computer Interfaces, and Measurement Systems (VECIMS 2008), p. 41-44, 2008.

SHAN, Tony C, HUA, Winnie W. **Taxonomy of java web application frameworks**. IEEE International Conference on e-Business Engineering (ICEBE'06), p. 378 - 385, 2006.

SWEAT, Jason E. **PHP architect's: guide to PHP design patterns**. Marco Tabini and Associates. Canada, 2005.