

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA
TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

BARBARA HOFFMANN ROCHA DUARTE DE OLIVEIRA

**CONTROLE DO TRABALHO DE CONCLUSÃO DE CURSO PARA O
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA DA UTFPR**

TRABALHO DE CONCLUSÃO DE CURSO

PONTA GROSSA

2018

BARBARA HOFFMANN ROCHA DUARTE DE OLIVEIRA

**CONTROLE DO TRABALHO DE CONCLUSÃO DE CURSO PARA O
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA DA UTFPR**

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas, do Departamento Acadêmico de Informática, da Universidade Tecnológica Federal do Paraná, Campus Ponta Grossa.

Orientador: Prof^a. Dra. Helyane Bronoski Borges

PONTA GROSSA

2018



TERMO DE APROVAÇÃO

CONTROLE DO TRABALHO DE CONCLUSÃO DE CURSO PARA O DEPARTAMENTO ACADÊMICO DE INFORMÁTICA

por

BARBARA HOFFMANN ROCHA DUARTE DE OLIVEIRA

Este Trabalho de Conclusão de Curso (TCC) foi apresentado em 20 de novembro de 2018 como requisito parcial para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas. A candidata foi arguida pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Prof^a Dra Helyane Bronoski Borges
Orientadora

Prof^a Dra Simone de Almeida
Membro titular

Prof MSc Geraldo Ranthum
Membro titular

Prof^a Dra Helyane Bronoski Borges
Responsável pelo Trabalho de Conclusão
de Curso

Prof. Dr. André Pinz Borges
Coordenador do curso

Dedico este trabalho à minha filha
Catarina.

AGRADECIMENTOS

Primeiramente agradeço imensamente minha família, que são as minhas raízes que me dão força e sustentação para seguir em frente, especialmente minha mãe e meu pai. Obrigada por todo apoio, paciência e preocupação que sempre tiveram comigo. Todas as vezes que pensei em desistir tive o apoio incondicional e motivos para continuar, sempre serão as pessoas que de todas as formas acreditam em meu potencial e me fazem ter a certeza que vou conseguir tudo que almejo, faltam palavras para expressar o imenso amor que sinto por vocês. Muito obrigada.

Agradeço também minha orientadora Professora Helyane Bronoski Borges, a qual sempre disponibilizou seu tempo. Pela disposição a dúvidas, ensinar, e passar da melhor forma todo conhecimento, sem o qual não teria também conseguido finalizar esse trabalho. Muito obrigada por ser exemplo de professora, orientadora e pessoa.

Agradeço também ao corpo docente da Universidade Tecnológica Federal do Paraná e também a todos os professores que passaram pela minha trajetória durante esses anos, pelos ensinamentos e por todas as vezes que precisei não medindo esforços a prestar auxílio e apoio.

E finalmente, expresso que sou eternamente grata pelos bons amigos que fiz durante a graduação, sem os quais não seria capaz de completar a trajetória, especialmente pelo companheirismo, disposição, eternas discussões e experiências.

Na maioria dos casos, forças e fraquezas
são dois lados da mesma moeda.

Uma força em uma situação é uma
fraqueza em outra, mas frequentemente
as pessoas não conseguem trocar as
marchas. É uma coisa muito sutil falar
sobre forças e fraquezas porque elas
sempre são a mesma coisa.

(JOBS, Steve)

RESUMO

HOFFMANN ROCHA DUARTE DE OLIVEIRA, Barbara. **Controle do Trabalho de Conclusão de Curso para o Departamento Acadêmico de Informática da UTFPR**. 2018. 48 f. Trabalho de Conclusão de Curso (Tecnologia em Análise e Desenvolvimento de Sistemas) - Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2018.

O Controle do Trabalho de Conclusão de Curso para o Departamento Acadêmico de Informática foi desenvolvido para a Universidade Tecnológica Federal do Paraná de Ponta Grossa. Podendo ser manipulado em qualquer dispositivo móvel, em virtude do sistema ser responsivo e se adaptar a qualquer tamanho de tela. O objetivo principal do sistema é automatizar a disciplina de TCC da Instituição, fornecendo um controle *Web* para o responsável pela disciplina. O responsável poderá cadastrar departamentos, cursos, professores, alunos, projetos, defesas, avisos, datas, modelos e perguntas frequentes. Além disso terá uma listagem de todos os itens citados, podendo realizar a alteração e exclusão. Inicialmente será explicado a metodologia que foi utilizada juntamente com a construção do sistema. E por fim a modelagem de dados e as figuras do sistema.

Palavras-chave: Controle de TCC. Sistema Web. Painel de Controle. Bootstrap. CodeIgniter.

ABSTRACT

HOFFMANN ROCHA DUARTE DE OLIVEIRA, Barbara. **Work of Conclusion Course System for the Informatics Academic Departament from UTFPR.** 2018. 48 p. Work of Conclusion Course (Technology in Analysis and Development Systems) - Federal Technology University - Paraná. Ponta Grossa, 2018.

The Work of Conclusion Course System for the Informatics Academic Departament was developed to the Federal Technology University of Paraná of Ponta Grossa. It can be manipulated on any mobile device due to be responsive system. The main objective of the system is to automate the Institution's course conclusion work discipline, providing a web control. This manner it's possible to book departments, courses, teachers, students, projects, evaluations, important dates, documents model and frequently asked questions. Initially, is explained the methodology that is used together with the system construction.

Keywords: Control Panel. Bootstrap. CodeIgniter. MVC. HTML5. CSS3. PHP7. MySQL.

LISTA DE ILUSTRAÇÕES

Figura 1 - Exemplo de <i>Media Queries</i>	16
Figura 2 - Sistema de Grade	18
Figura 3 - Pilha de Fontes Nativas	19
Figura 4 - Fluxograma do <i>CodeIgniter</i>	20
Figura 5 - Tipo de Tabela no MySQL	21
Figura 6 - Caso de Uso Sistema do Controle de TCC.....	25
Figura 11 - Diagrama de Sequência Cadastrar Defesa.....	29
Figura 12 - Modelo Relacional.....	30
Figura 13 - Painel de Controle: Home	31
Figura 14 - Painel de Controle: Datas	32
Figura 15 - Painel de Controle: Avisos	33
Figura 16 - Painel de Controle: Modelos	34
Figura 17 - Painel de Controle: FAQ's.....	35
Figura 18 - Painel de Controle: Login.....	35
Figura 19 - Painel de Controle: <i>Admin</i> – Login Responsável	36
Figura 21 - Painel de Controle: Lista Departamento	38
Figura 22 - Painel de Controle: Cadastro Departamento	38
Figura 23 - Painel de Controle: Lista Curso.....	39
Figura 24 - Painel de Controle: Cadastro Curso.....	39
Figura 25 - Painel de Controle: Lista Professor.....	40
Figura 26 - Painel de Controle: Cadastro Professor.....	40
Figura 27 - Painel de Controle: Lista Aluno.....	41
Figura 28 - Painel de Controle: Cadastro Aluno	42
Figura 31 - Painel de Controle: Lista Defesa.....	44
Figura 32 - Painel de Controle: Cadastro Defesa.....	44
Figura 33 - Painel de Controle: Justificativa Defesa.....	45
Figura 34 - Painel de Controle: Finalizar Defesa como Admin.....	46
Figura 35 - Painel de Controle: Finalizar Defesa como Professor.....	46

LISTA DE SIGLAS E ACRÔNIMOS

ANSI	<i>American National Standards Institute</i>
CSS	<i>Cascading Style Sheets</i>
CRUD	<i>Create Read Update Delete</i>
DAINF	Departamento Acadêmico de Informática
DER	Diagrama Entidade Relacionamento
FAQ	<i>Frequently Asked Questions</i>
HTML	<i>Hypertext Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
ISO	<i>International Organization for Standardization</i>
JS	<i>JavaScript</i>
MVC	<i>Model-View-Controller</i>
PHP	<i>Hypertext Preprocessor</i>
SGBD	Sistema de Gerenciamento de Banco de Dados
SQL	<i>Structured Query Language</i>
UML	<i>Unified Modeling Language</i>
URL	<i>Uniform Resource Locator</i>
UTFPR	Universidade Tecnológica Federal do Paraná

SUMÁRIO

1 INTRODUÇÃO	13
1.1 MOTIVAÇÃO	14
1.2 OBJETIVOS.....	14
1.2.1 Objetivo Geral	14
1.2.2 Objetivos Específicos.....	15
1.3 ESTRUTURA DO TRABALHO	15
2 FUNDAMENTAÇÃO TEÓRICA	16
2.1 BOOTSTRAP	16
2.2 CODEIGNITER	19
2.3 MYSQL	20
3 METODOLOGIA	23
3.1 ESTUDO DO PROCESSO ATUAL	23
3.2 LEVANTAMENTO DE REQUISITOS.....	24
3.3 MODELAGEM DO SISTEMA.....	24
3.3.1 Diagrama de Casos de Uso	24
3.3.2 Diagrama de Classes.....	25
3.3.3 Diagrama de Sequência: Cadastrar Datas	26
3.3.4 Diagrama de Sequência: Cadastros do Responsável.....	27
3.3.5 Diagrama de Sequência: Cadastrar Projeto	28
3.3.6 Diagrama de Sequência: Cadastrar Defesa	28
3.3.7 Diagrama Entidade Relacionamento	29
3.4 FERRAMENTAS UTILIZADAS NO DESENVOLVIMENTO	29
4 SISTEMA DE CONTROLE DO TCC	31
4.1 SISTEMA DE CONTROLE DO TCC.....	31
4.1.1 Home	31
4.1.2 Datas	32
4.1.3 Avisos	32
4.1.4 Modelos	33
4.1.5 Faq's	34
4.2 SISTEMA DE CONTROLE DO TCC LOGADO	36
4.2.1 Cadastro e Lista de Departamentos	37
4.2.2 Cadastro e Lista de Cursos	38
4.2.3 Cadastro e Lista de Professores.....	40
4.2.4 Cadastro e Lista de Alunos	41
4.2.5 Cadastro e Lista de Projetos.....	42
4.2.6 Cadastro e Lista de Defesas.....	43
4.2.7 Finalizar Defesa	45
5 CONCLUSÃO	47

REFERÊNCIAS.....	48
-------------------------	-----------

1 INTRODUÇÃO

A Universidade Tecnológica Federal do Paraná (UTFPR), Campus Ponta Grossa, tem como um dos requisitos para a conclusão de um curso de graduação o Trabalho de Conclusão de Curso (TCC). O TCC é uma atividade obrigatória, constituída por duas disciplinas curriculares: Trabalho de Conclusão de Curso 1 (TCC1) e Trabalho de Conclusão de Curso 2 (TCC2).

A disciplina de TCC1 consiste na elaboração de proposta de trabalho científico e/ou tecnológico envolvendo temas abrangidos pelo curso, bem como o início do desenvolvimento do trabalho proposto (Instrução Normativa 01/07 – PROGRAD). Já a disciplina de TCC2 versa na execução e a conclusão do trabalho proposto.

Cada curso de graduação da UTFPR possui um professor designado, por meio de portaria, a gerenciar todas as atividades relacionadas ao TCC, como por exemplo, controlar alunos que possuem projetos de pesquisa em andamento, alunos que irão defender o TCC, alunos que irão poder prorrogar o desenvolvimento do projeto por mais um semestre letivo, entre outras.

A UTFPR possui um módulo no sistema acadêmico denominado de Trabalho de Conclusão que permite, que o responsável pelo TCC do curso, cadastre o projeto do aluno quando este se matricula na disciplina de TCC2. Da mesma forma, quando o aluno conclui seu trabalho o professor responsável lança a nota do aluno e finaliza o projeto gerando uma declaração de orientação para o orientador do trabalho. Ademais, o restante do gerenciamento do TCC, dos cursos de graduação, do Departamento Acadêmico de Informática (DAINF) é realizado por meio de planilhas eletrônicas.

Sendo assim, faz-se necessário o desenvolvimento de um sistema *Web* que centralize as informações relacionadas ao TCC e faça o controle dos projetos dos alunos desde o seu início na disciplina de TCC1 até a sua conclusão em TCC2. O sistema *Web* será responsivo, ou seja, o *layout* do painel de controle se adequará a qualquer tamanho de tela e constituirá em uma contribuição para o Departamento Acadêmico de Informática (DAINF), auxiliando o docente responsável pela disciplina de TCC, bem como o aluno e orientador.

1.1 MOTIVAÇÃO

Para a realização do Trabalho de Conclusão de Curso (TCC) o aluno precisa seguir alguns passos para a entrega de documentos à qual é realizada de forma manual e necessita de algumas etapas onde o responsável pela disciplina precisa de um maior controle e autonomia.

- a) Na disciplina de TCC 1 o aluno define um tema e um orientador para o desenvolvimento do trabalho, apresentando uma proposta;
- b) Após a conclusão da disciplina de TCC 1 o aluno precisa se matricular na disciplina de TCC 2;
- c) Na disciplina de TCC 2 o aluno desenvolve o projeto e defende no final do semestre.

Um dos problemas existentes consiste em controlar os alunos aprovados e reprovados na disciplina, bem como os alunos que podem solicitar a prorrogação da disciplina de TCC2. Além disso, é importante controlar quais os professores possuem trabalhos em orientação. Outra questão refere-se à entrega da documentação ao término do TCC, como o cronograma de reuniões e seus relatórios, que atualmente são entregues no formato impresso.

O sistema, portanto, será desenvolvido para automatizar o controle do TCC no sistema acadêmico. Sendo assim, os usuários do sistema, professor responsável pela disciplina e orientadores, terão uma maior clareza, visto que o mesmo estará estruturado nesse painel de controle, com: datas, projetos, notas, orientadores, orientandos e banca.

1.2 OBJETIVOS

Os objetivos deste trabalho estão divididos em geral e específicos.

1.2.1 Objetivo Geral

Desenvolver um sistema *Web* para gerenciar a disciplina de Trabalho de Conclusão de Curso, dos cursos de graduação, do Departamento Acadêmico de Informática da UTFPR.

1.2.2 Objetivos Específicos

Como objetivos específicos têm-se:

- Fazer o levantamento de requisitos;
- Realizar a modelagem do sistema de TCC;
- Implementar o sistema em uma linguagem de programação;
- Validar o sistema com o professor responsável pelo TCC do DAINF;

1.3 ESTRUTURA DO TRABALHO

O trabalho está dividido em 5 Capítulos. O Capítulo 2 apresenta a Fundamentação Teórica utilizada no desenvolvimento do trabalho. O Capítulo 3 apresenta o Levantamento de Requisitos, a Modelagem utilizada na construção do trabalho, bem como as Ferramentas utilizadas para o desenvolvimento do sistema. O Capítulo 4 apresenta o Sistema de Controle do TCC por meio de interfaces. E por fim, o Capítulo 5 descreve a conclusão do trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Neste Capítulo é apresentada a Fundamentação Teórica para o desenvolvimento do projeto. A Seção 2.1 aborda, portanto, a documentação do *framework Bootstrap*, acompanhada pela Seção 2.2, a qual apresenta a documentação do *framework CodeIgniter*. Por último, a Seção 2.3 exhibe e discute a documentação do gerenciamento de banco de dados MySQL.

2.1 BOOTSTRAP

O *Bootstrap*, atualmente, é o principal *framework front-end* para desenvolvimento *Web* responsivo, ou seja, por fazer uso do *Hypertext Markup Language* (HTML), *Cascading Style Sheets* (CSS) e *Javascript* (JS), reage ao tamanho da tela do dispositivo e aos recursos (BOOTSTRAP, 2011).

O W3C (*World Wide Web Consortium*) organização internacional que desenvolve padrões, protocolos e diretrizes que garantam o crescimento da *Web* (W3C, 2018), implementou o que se chama de *media queries* como parte da especificação CSS3, a qual concedeu essas características para definir formatações para diversos dispositivos. Desta forma, o navegador *Web* interpreta as expressões definidas na *query*, destacando que, caso o dispositivo se adeque às condições estabelecidas, o CSS será aplicado (TABLELESS, 2019).

Figura 1 - Exemplo de *Media Queries*

```
1 /* Telefones em paisagem e abaixo */
2 @media (max-width: 480px) {
3     /* estilos aqui */
4 }
5
6 /* Telefones em paisagem a tablet em retrato */
7 @media (max-width: 767px) {
8     /* estilos aqui */
9 }
10
11 /* Tablet em retrato a paisagem e desktop */
12 @media (min-width: 768px) and (max-width: 979px) {
13     /* estilos aqui */
14 }
15
16 /* Desktop grande */
17 @media (min-width: 1200px) {
18     /* estilos aqui */
19 }
20
```

Fonte: Autoria Própria

Entende-se, portanto, que as *media queries* são manipuladas dentro do código CSS, como mostrado na Figura 1. Nesta figura pode-se ver um código de *breakpoints*, no qual o atributo utilizado é o *width* (largura) que unido com o prefixo *min/max* delimita a largura máxima e mínima do *viewport* (tamanho da janela do dispositivo).

Observa-se, ainda, que as *medias queries* podem ser importadas, também, por meio do *@import*, porém nos dois casos o dispositivo é testado. Isso por que as *media queries* não se limitam a alterações de *layout* em navegadores modernos, pois no geral, são consideradas “comentários condicionais” (MARCOTTE, 2010).

Além das *media queries* e dos *breakpoints*, auxiliares no *mobile first*, é preciso desenvolver o *layout*, inicialmente para as telas menores, mas também para as telas médias e por último para as telas maiores. Significa que o *framework* assume que a tela é um dispositivo móvel, para depois verificar seu tamanho real (BOOTSTRAP, 2010).

O *container* é um componente invisível do *framework*. No entanto, que tal componente é de estrutura, o que garante o alinhamento do *layout* e do conteúdo, podendo ser qualquer *tag* semântica do HTML5. Segundo a Equipe do *Bootstrap* (2010), existem dois tipos de *containers* que podem ser utilizados: o simples e o fluído. O *container* simples é padrão, centralizado e de largura fixa. Já o *container* fluído ocupa toda a tela.

Como pode-se observar no Quadro 1 se o *container* estiver sujeito ao dispositivo e a orientação de tela, é alterado seu tamanho. *Bootstrap* (2010) demonstra como a página deve ter no mínimo um *container*, conhecido como *wrapper*, que abrange todo o escopo. Por meio deste conceito observa-se, portanto, que o *grid system* se constitui de *containers*, linhas e colunas, é responsivo e construído com *flexbox*.

Quadro 1 - Largura de Contêineres de acordo com o tamanho da tela do dispositivo

Largura de Contêineres de acordo com o tamanho da tela do dispositivo				
Dispositivo	Celulares	Tablets	Desktops	Telas grandes
Largura	Automática	máx. de 750px	máx. de 970px	máx. de 1170px

Fonte: Bootstrap: Criando Layouts e Grids – BOOTSTRAP (2010)

Torna-se imprescindível destacar que as *rows* (linhas), definem as divisões horizontais do *layout* e são encontradas dentro do *container*, podendo ser

empregados a qualquer *tag* que defina uma estrutura, como `<div>`, `<header>` e `<footer>`. Logo as *columns* (colunas), definem as divisões verticais das linhas, sendo encontradas dentro das linhas, definindo as áreas (BOOTSTRAP, 2010).

Entende-se, assim, que o *grid system* é uma matriz que utiliza *divs* e classes para calcular as posições dos itens na tela, as quais contém um padrão de 12 colunas para criar o *container*.

No exemplo, mostrado na Figura 2, tem-se a primeira linha com doze colunas de tamanho 1 ($12 * 1 = 12$), a segunda linha com três colunas de tamanho 4 ($3 * 4 = 12$), a terceira linha com duas colunas de tamanho 4 e 8 ($4 + 8 = 12$), a quarta linha com duas colunas de tamanho 6 ($6 + 6 = 12$), e a última linha com uma coluna de tamanho total.

Figura 2 - Sistema de Grade

span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1
span 4				span 4				span 4			
span 4				span 8							
span 6						span 6					
span 12											

Fonte: Bootstrap Grid System – BOOTSTRAP (2010)

Além disso, as colunas possuem prefixos para indicar onde a coluna ficará posicionada de acordo com a tela, conforme ilustrados no Quadro 2. Destaca-se que os asteriscos devem ser substituídos por números.

Quadro 2 - Prefixos das Colunas no Bootstrap

Prefixos				
Dispositivo	celulares	Tablets	desktops	telas grandes
Prefixo	.col-xs-*	.col-sm-*	.col-md-*	.col-lg-*

Fonte: Bootstrap: Criando Layouts e Grids – BOOTSTRAP (2010)

Avançando, tem-se o *reboot*, do inglês reinicie, que é uma biblioteca CSS pré-definida para o HTML. As vantagens em utilizar o *reboot* estão em customização e padronização, visto que o `<body>` configura automaticamente alguns elementos, tais como, cabeçalhos, parágrafos, listas, textos, tabelas, formulários (MARCOTTE, 2010).

Uma curiosidade do *Bootstrap* está no uso de uma pilha de fontes nativas, que poderão ser observadas na Figura 3. Para otimizar a renderização de texto, o *browser* seleciona a melhor fonte para o sistema operacional do dispositivo e, caso seja necessário a troca da fonte, é preciso atualizar a variável *\$font-family-base* e recompilar o código.

Figura 3 - Pilha de Fontes Nativas

```
$font-family-sans-serif:
  // Safari para OS X e iOS (San Francisco)
  -apple-system,
  // Chrome < 56 para OS X (San Francisco)
  BlinkMacSystemFont,
  // Windows
  "Segoe UI",
  // Android
  "Roboto",
  // Fallback de fonts web básicas
  "Helvetica Neue", Arial, sans-serif,
  // Fonts de Emoji
  "Apple Color Emoji", "Segoe UI Emoji", "Segoe UI Symbol" !default;
```

Fonte: Get Bootstrap (2018)

2.2 CODEIGNITER

O *CodeIgniter* é um *framework* de aplicações, ou seja, um conjunto de bibliotecas com o intuito de desenvolver um projeto de interface simples e estrutura lógica com ferramentas PHP a disposição, diminuindo a quantidade de códigos (*CODEIGNITER*, 2016).

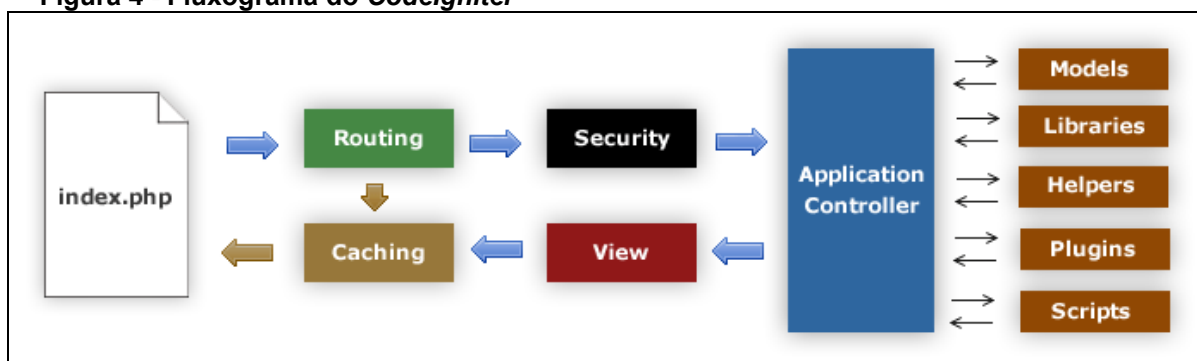
Existem inúmeras vantagens de utilizar este *framework*, sendo que as principais delas consistem em: ser gratuito, poucas configurações, sem regras restritas de configuração, abordagem MVC (Modelo-Visão-Controle, do inglês *Model-View-Controller*), URLs (do inglês, *Uniform Resource Locator*) amigáveis aos motores de busca, soluções simples, documentação robusta, entre outros inúmeros recursos.

O MVC, citado acima, é um padrão de desenvolvimento de *software* onde a lógica (*back-end*) é separada da apresentação (*front-end*), sendo a principal definição dessa arquitetura a separação de conceitos. Nela, os objetos protegem

suas informações e como são manipuladas, apresentando apenas a sua interface, como é realizado na programação orientada a objetos (CODEIGNITER, 2016).

A Figura 4 apresenta como ocorre o fluxo de dados da aplicação por meio do sistema. Nota-se que *index.php* é o ponto inicial da aplicação, encadeando os recursos necessários. O roteador (*routing*) examina a requisição HTTP para ordenar o próximo passo. Caso exista um arquivo em *cache* (*caching*), ele é enviado para o navegador ignorando a execução normal.

Figura 4 - Fluxograma do CodeIgniter



Fonte: CODEIGNITER (2016)

Observa-se ainda na Figura 4 que, antes do *controller* ser solicitado, a requisição HTTP e os dados são verificados pela segurança. Após a verificação, o *controller* carrega os recursos necessários para atender à solicitação específica: *models*, bibliotecas (*libraries*), ajudas (*helpers*), *plug-ins* e os *scripts*. Na sequência a *view* é finalizada, renderizada e enviada para o navegador. E, por fim, se o *cache* estiver habilitado, a *view* é cacheada para atender as próximas requisições.

2.3 MYSQL

O MySQL é um Sistema de Gerenciamento de Banco de Dados (SGBD), o qual usa a linguagem SQL (Linguagem de Consulta Estruturada, do inglês *Structure Query Language*) como interface. Tornou-se o sistema padrão do *American National Standards Institute* (ANSI) em 1986 e da *International Organization for Standardization* (ISO) em 1987 (W3SCHOOLS, 2018).

Atualmente, devido a sua confiabilidade e desempenho, tornou-se a principal escolha de banco de dados para projetos baseados na web. Além disso, o MySQL é um software *open-source* (OOS), ou seja, possui código aberto garantindo a sua

acessibilidade, perpetuidade e interoperabilidade. Entre as suas principais características encontram-se: portabilidade, compatibilidade, estabilidade, controle transacional, além de suportar *triggers*, *cursors*, *stored procedures*, *functions*, entre outros (MILANI, 2012).

Uma característica incomum do *MySQL* consiste na escolha do tipo de tabela no momento de criação da mesma. A definição do tipo de tabela é realizada por meio do comando *CREATE TABLE*, como é mostrado na Figura 5. Observa-se que a tabela é do tipo *MyISAM*, que é o valor *default* caso não seja mencionado o *TYPE*. É apropriado para tabelas que os dados não mudam com regularidade. Um exemplo deste fato seria uma tabela de estados onde os valores seriam apenas para leitura (MILANI, 2012).

Figura 5 - Tipo de Tabela no MySQL

```
CREATE TABLE teste (  
    id INT NOT NULL,  
    texto CHAR(30) NOT NULL,  
    PRIMARY KEY (id)  
    ) TYPE=MyISAM;
```

Fonte: DEV MEDIA (2006)

No que se refere ao comando demonstrado na Figura 5, André Milani (2012) descreve os tipos de tabelas suportados no *MySQL*:

a) *MyISAM*

- a.1 São armazenadas em três arquivos, separando a definição os dados e os índices;
- a.2 São tabelas de grande desempenho para consulta, extremamente rápidas devido a arquitetura simplificada;
- a.3 Possuem operação atômica o que não permitem controle de transações (*commit* e *rollback*);
- a.4 Não possuem integridade referencial, ou seja, as restrições da chave estrangeira não são respeitadas.

b) *HEAP*

- b.1 Extremamente rápidas, pois são armazenadas em memória;
- b.2 Conteúdo volátil, por não ser gravada em disco;

- b.3 São tabelas que são consultadas com muita frequência, mas que não sofrem muitas alterações;
 - b.4 É necessário criar um processo para realizar a execução dos dados quando o servidor iniciar.
- c) MERGE
- c.1 São coleções de tabelas *MyISAM*, permite a divisão de uma tabela maior em partes menores;
 - c.2 Contudo, permitindo acesso simultâneo as partes menores como se fosse uma única parte;
 - c.3 Permite superar as limitações do SO em relação ao tamanho dos arquivos;
 - c.4 A desvantagem nessa tabela é que só é permitido realizar o MERGE de tabelas *MyISAM* idênticas.
- d) BDB
- d.1 Possuem controle de transações (*commit* e *rollback*);
 - d.2 Além disso, realiza a recuperação de dados automática caso haja uma queda no sistema.
- e) InnoDB
- e.1 Tabela transacional com integridade referencial, ou seja, possui chave estrangeira;
 - e.2 Permissão de *constraints* (*SET*, *NULL*, *SET DEFAULT*, *RESTRICT* e *CASCADE*);
 - e.3 Ferramenta de *backup online*;
 - e.4 É armazenamento de dados em *tablespace*.

Percebe-se que a utilização de tipos diferentes de tabelas é muito importante para se obter um maior desempenho, dependendo dos requisitos da aplicação. Vale ressaltar que é possível a utilização de vários tipos de tabelas em um mesmo banco de dados, lembrando que a tabela do tipo *MyISAM* não suporta transações (MILANI, 2012).

3 METODOLOGIA

Neste Capítulo, será apresentada a metodologia empregada no desenvolvimento do projeto. A Seção 3.1 descreve o Estudo do Processo Atual. A Seção 3.2 aborda o Levantamento de Requisitos. A Seção 2.3 apresenta a Modelagem do Sistema por meio de diagramas e por fim, a Seção 3.4 cita as ferramentas utilizadas para o desenvolvimento deste projeto.

3.1 ESTUDO DO PROCESSO ATUAL

Conforme o regulamento do TCC para os Cursos de Graduação da UTFPR (UTFPR, 2006), o TCC é dividido em duas disciplinas, TCC 1 e TCC 2. O Professor Responsável pelo TCC é o encarregado de administrar todos os procedimentos relacionados as duas disciplinas.

Na disciplina de TCC 1 o aluno deve escolher um orientador, apresentar uma proposta do projeto, para uma banca examinadora, bem como iniciar o desenvolvimento do projeto. Se o aluno conseguir aprovação nesta disciplina ele poderá matricular-se na disciplina de TCC 2.

A disciplina de TCC 2 consiste na execução e conclusão do projeto apresentado no TCC 1. Para a aprovação do aluno nessa disciplina são necessárias a elaboração de um trabalho no formato de monografia e a realização de uma defesa oral para uma banca examinadora a qual é formada pelo professor orientador e, no mínimo, dois professores.

Caso o aluno não tenha concluído o projeto durante o semestre, o professor orientador juntamente com seu orientado poderá solicitar a prorrogação, para a conclusão do trabalho, por mais um semestre letivo. Se o aluno não concluir o trabalho, neste período, o mesmo será reprovado na disciplina de TCC2 e terá que apresentar uma nova proposta de TCC.

3.2 LEVANTAMENTO DE REQUISITOS

Diante das reuniões com a responsável pelos TCC's, dos cursos de Tecnologia em Análise de Sistemas e Bacharelado em Ciência da Computação, foram observados os recursos e as funcionalidades deveria possuir.

Com essas informações foi definido que o sistema deveria ser *Web* e capaz de suportar dois tipos de usuários e que cada usuário teria acesso a diferentes tipos de funcionalidades no sistema.

O usuário *admin* é o responsável pela disciplina de TCC e responsável por todos os cadastros existentes no sistema. Inclusive o cadastro dos usuários. Os usuários cadastrados pelo *admin* serão os orientadores que terão acesso ao projeto e a defesa, podendo editar e finalizar somente o projeto e a defesa em que estará definido como orientador.

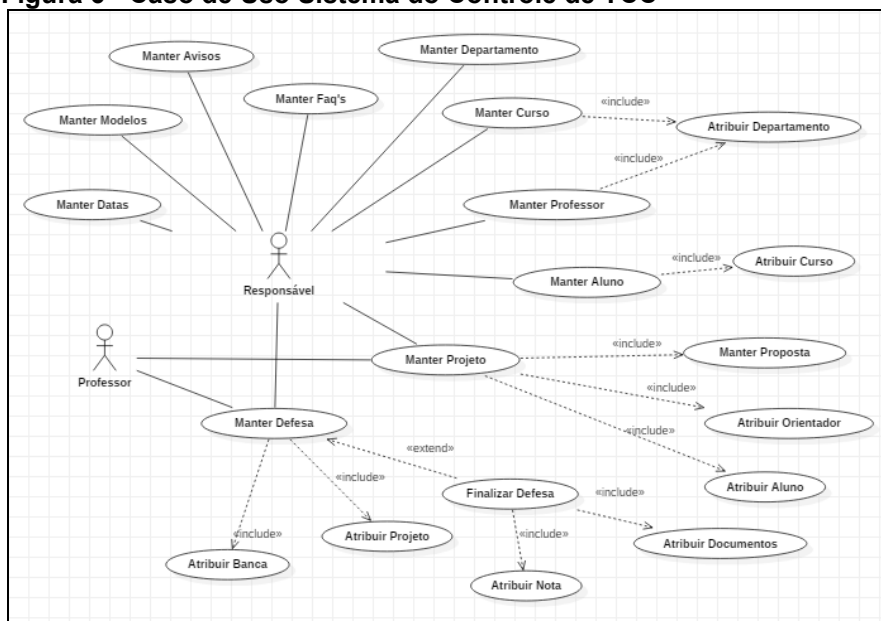
3.3 MODELAGEM DO SISTEMA

Nessa subseção foi efetuada a diagramação e modelagem do sistema. Foram gerados os diagramas de: Casos de Uso, Classes, Sequência e o Relacional de Banco de Dados. Essa modelagem trouxe melhorias no entendimento da arquitetura e no desenvolvimento das funcionalidades do projeto.

3.3.1 Diagrama de Casos de Uso

A Figura 6 demonstra o Diagrama de Caso de Uso geral do sistema, onde o ator principal nomeado como Responsável pela disciplina de TCC. Esse é responsável pelos cadastros de departamentos, cursos, professores, alunos, projetos, defesas, datas, avisos, modelos e *faq's*. Já o ator Professor é o responsável por cadastrar a defesa onde ele é orientador.

Figura 6 - Caso de Uso Sistema do Controle de TCC

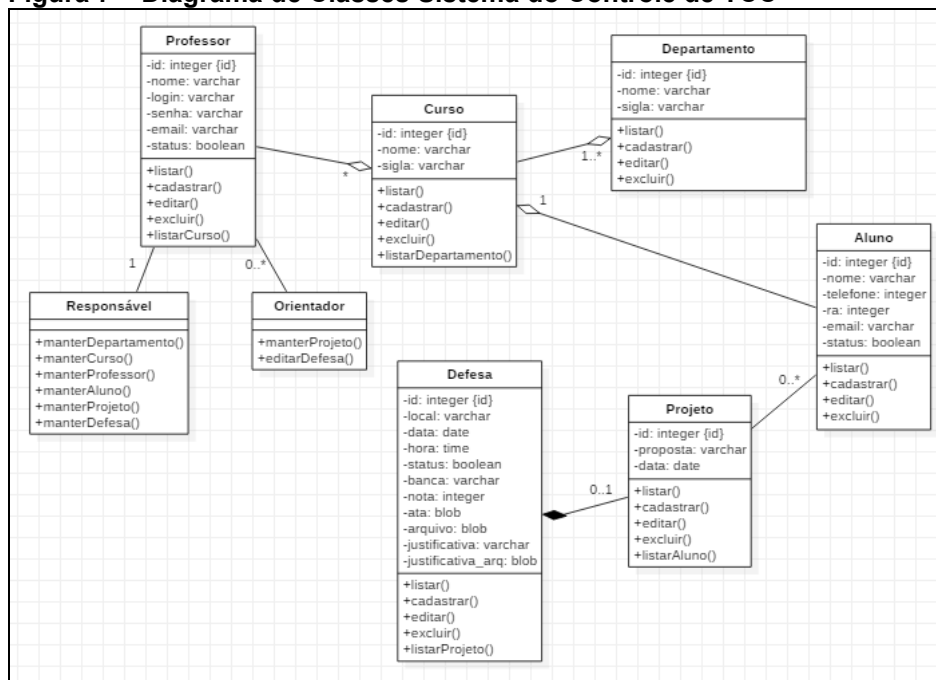


Fonte: Autoria própria

3.3.2 Diagrama de Classes

A Figura 7 demonstra o Diagrama de Classes do sistema, é importante ressaltar que as *functions* citadas no diagrama, encontram-se no *model* de cada objeto do sistema, são eles: manter, listar, cadastrar, editar e excluir.

Figura 7 – Diagrama de Classes Sistema do Controle de TCC



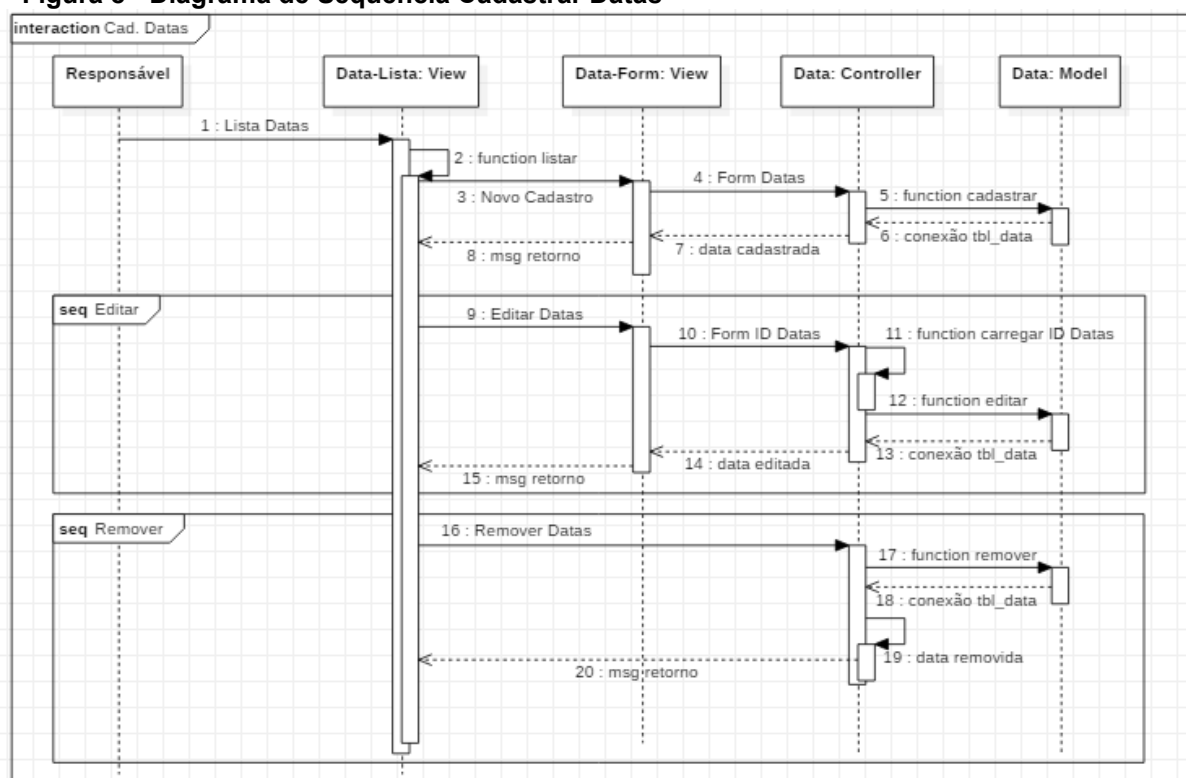
Fonte: Autoria própria

3.3.3 Diagrama de Sequência: Cadastrar Datas

A Figura 8 demonstra o Diagrama de Sequência Cadastrar Datas, o Diagrama de Sequência de Avisos, Modelos e FAQ's seguem a mesma lógica. Pois, são dados cadastrados pelo responsável no *admin*, os quais ficarão visíveis para o usuário sem *Login* na *home* e listados no *admin* para edição e remoção dos mesmos.

Observa-se que o objeto *datas* possui duas linhas de vida que correspondem a *view* do sistema, são elas: *listaDatas* e *formDatas*. Essas *views* foram padronizadas para todos os objetos do sistema, a primeira sendo a *view* de listagem e a segunda a *view* de formulário de cadastro, respectivamente. E as outras linhas de vidas apresentadas, é o *controller* e o *model* responsáveis pela comunicação entre as camadas e pela interação com o banco de dados, respectivamente.

Figura 8 - Diagrama de Sequência Cadastrar Datas

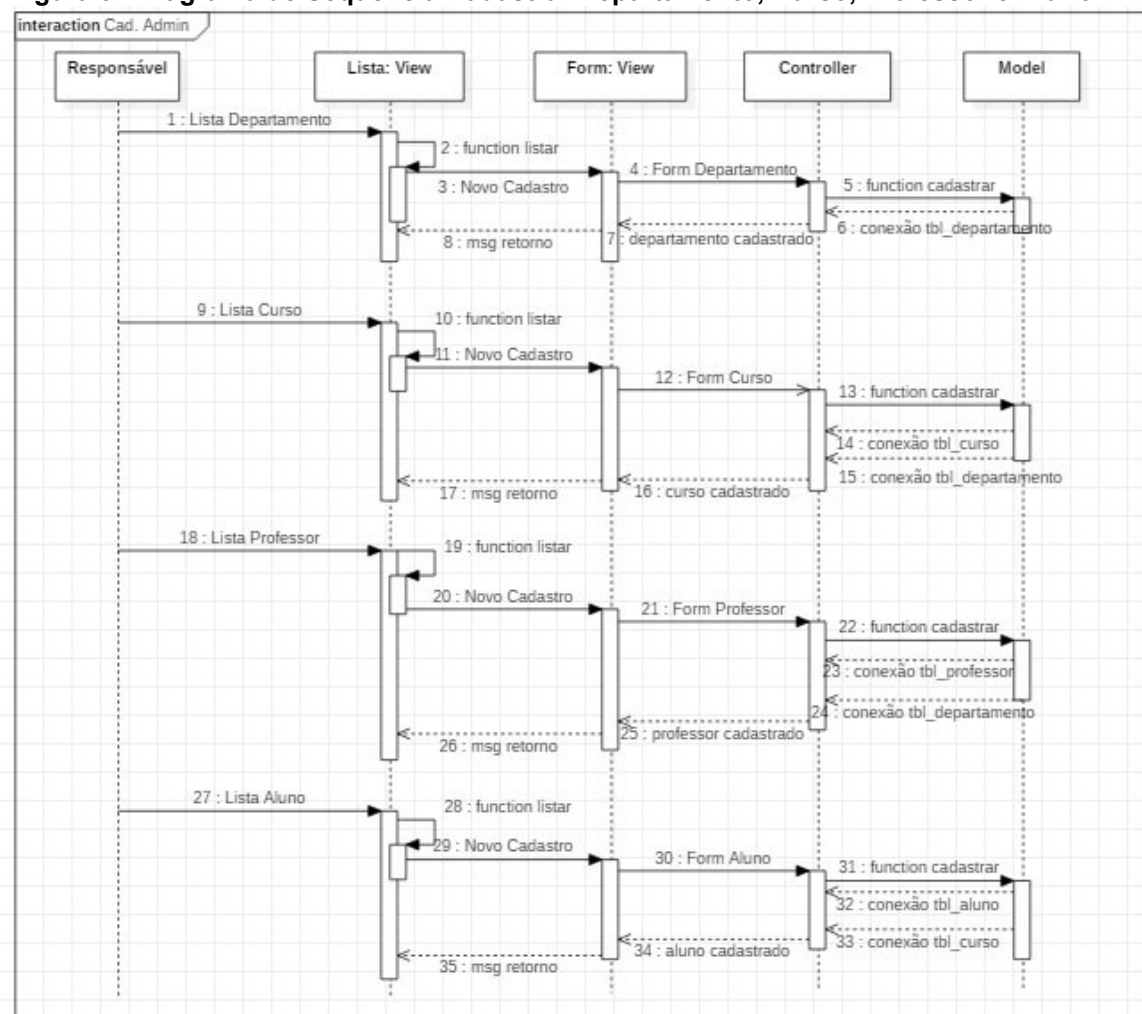


Fonte: Autoria própria

3.3.4 Diagrama de Sequência: Cadastros do Responsável

A Figura 9 demonstra o Diagrama de Sequência de Cadastro de Departamento, Curso, Professor e Aluno. Esses cadastros também possuem o *CRUD*, e são cadastros efetuados pelo responsável. O *CRUD*, do inglês *Create, Read, Update and Delete*, é composto pelas quatro operações básicas de manipulação de dados.

Figura 9 - Diagrama de Sequência Cadastrar Departamento, Curso, Professor e Aluno



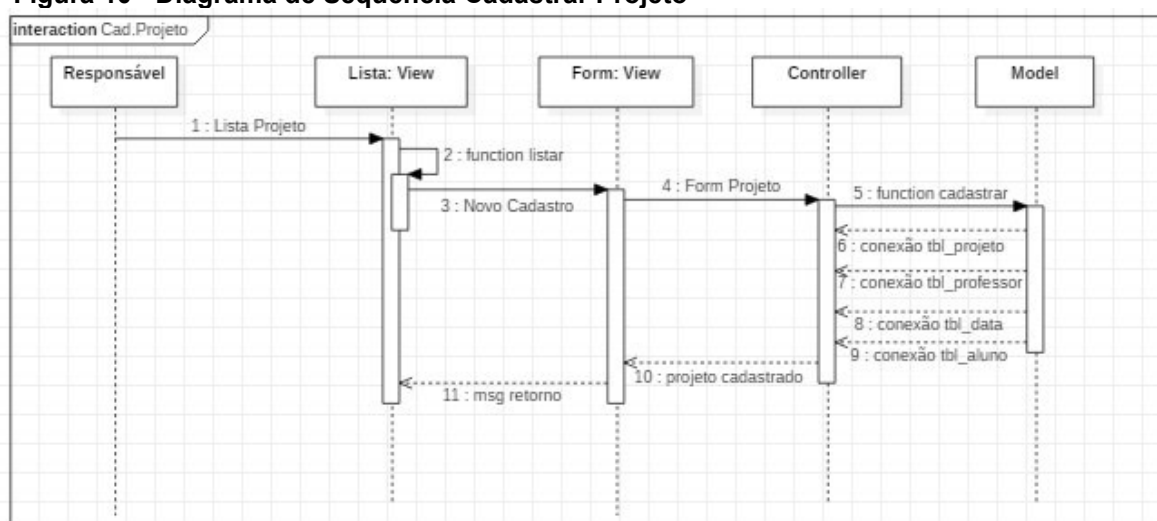
Fonte: Autoria própria

Observa-se no diagrama apresentado que primeiro se cadastra um departamento para depois cadastrar um curso e um professor e conseqüentemente atribuir um departamento para um curso e para um professor. O mesmo ocorre com curso, primeiro se cadastra um curso para depois cadastrar um aluno, e conseqüentemente atribuir um curso a um aluno.

3.3.5 Diagrama de Sequência: Cadastrar Projeto

A Figura 10 representa a funcionalidade de cadastrar o projeto em que primeiro se atribui um orientador e de um até três alunos. Além disso, é possível realizar a inserção de um coorientador. Nesse formulário de cadastro do projeto, é possível delimitar o prazo de desenvolvimento, como também definir o *status* em: aprovado, reprovado ou em andamento

Figura 10 - Diagrama de Sequência Cadastrar Projeto



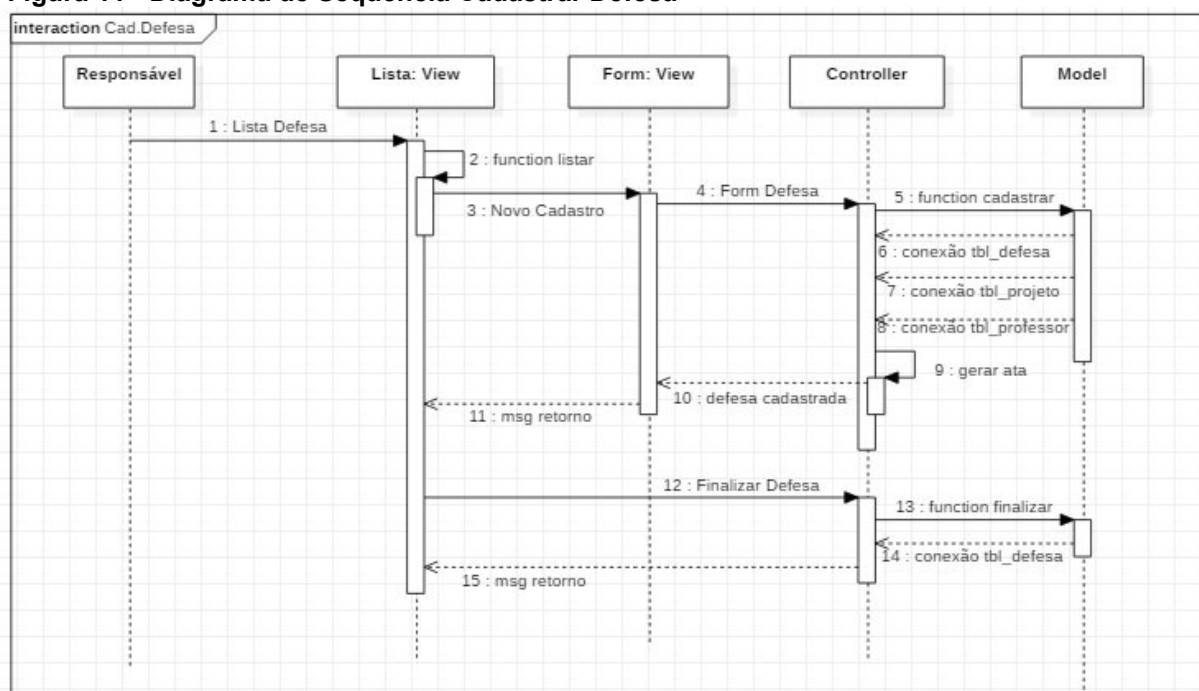
Fonte: Autoria própria

O Cadastro do Projeto é o responsável por nortear a compreender quem são os orientadores, os orientandos e as propostas em andamento antes do momento da defesa. Nota-se que o formulário do cadastro de projeto possui mais de uma chave estrangeira, logo que ele realiza conexão com as tabelas: professor, data e aluno.

3.3.6 Diagrama de Sequência: Cadastrar Defesa

A Figura 11 demonstra o Diagrama de Sequência Cadastrar Defesa, responsável por agendar a defesa da proposta, denominar a banca e atribuir a ata de defesa. No momento do cadastro da defesa o formulário disponibilizará o cadastro de no mínimo dois até quatro professores para participar da banca, como também um cadastro de pessoa externa e de no mínimo um e no máximo três alunos se o trabalho foi desenvolvido em grupo.

Figura 11 - Diagrama de Sequência Cadastrar Defesa



Fonte: Autoria própria

Nota-se que a função Finalizar Defesa da Figura 11, responsável por abrir um modal e pedir o *upload* do Cronograma de Desenvolvimento e do Relatório de Acompanhamento.

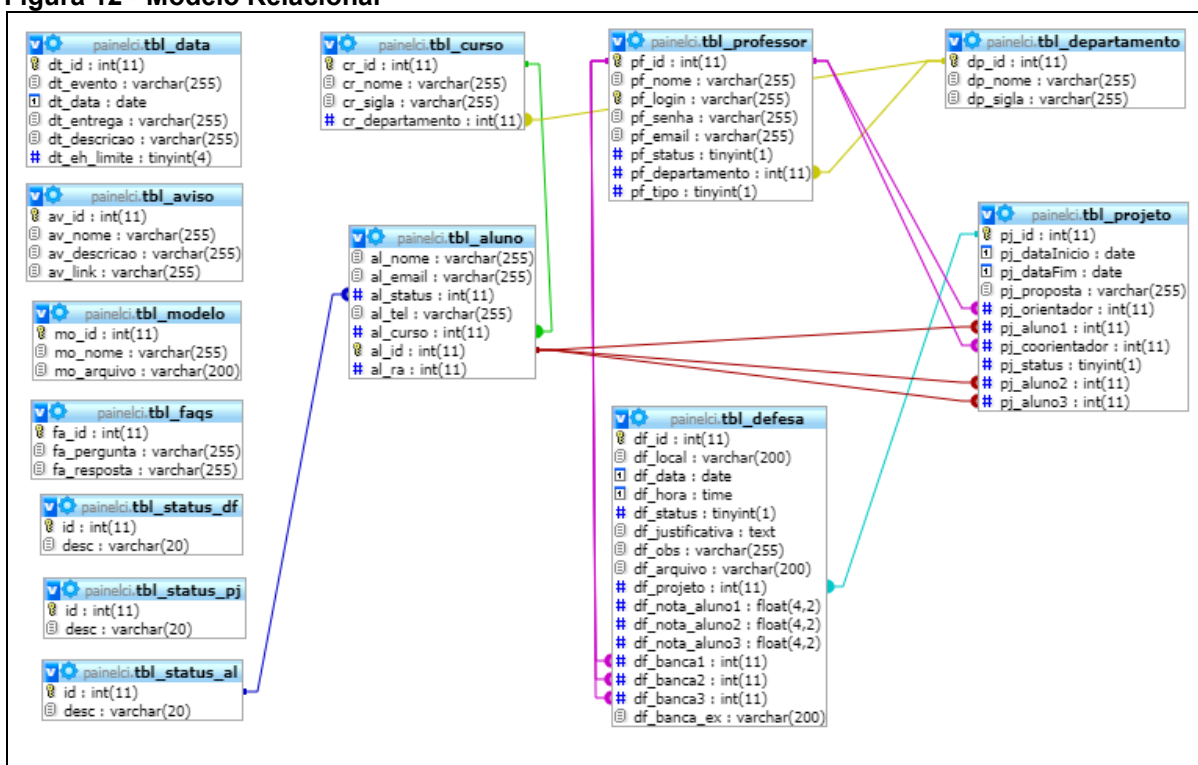
3.3.7 Diagrama Entidade Relacionamento

A Figura 12 ilustra o Modelo Relacional criado para o desenvolvimento do sistema a partir da ferramenta *phpMyAdmin*, *software* livre para manipulação de *MySQL* pela *internet*.

3.4 FERRAMENTAS UTILIZADAS NO DESENVOLVIMENTO

Para o desenvolvimento do sistema proposto neste trabalho de conclusão de curso foi utilizado o ambiente *WampServer* o qual permite a criação de aplicações com *Apache2*, *PHP* e o banco de dados *MySQL* juntamente com o *PhpMyAdmin*, que permite gerenciar o banco de dados.

Figura 12 - Modelo Relacional



Fonte: Autoria própria

Para o desenvolvimento do software, no padrão de arquitetura MVC separando o domínio da apresentação, foi realizada a integração do *HTML5*, *CSS3* e *Javascript*.

Para a padronização optou-se pela utilização do *framework open-source Bootstrap*, o qual possui elementos *HTML*, *CSS* e *Javascript*, ademais *plugins jQuery*, o que facilitou a criação responsiva por meio de *grids* flexíveis, que permitem definir conjuntos de *grids* para resoluções de tela diferentes. Desempenho definido por meio das classes *xs* (dispositivos muito pequenos), *sm* (dispositivos pequenos), *md* (dispositivos médios) e *lg* (dispositivos grandes).

O *framework web PHP* escolhido foi o *CodeIgniter*, o qual possui a documentação toda inclusa para orientação e com maior fórum de dúvidas nacional na internet. Ele incentiva *MVC*, mas não força o seu uso, já que grande parte da configuração é feita por opção própria, por exemplo, colocar modelos em uma pasta "modelos".

4 SISTEMA DE CONTROLE DO TCC

Este Capítulo é responsável pela apresentação das telas do Sistema. Na Seção 4.1 apresenta o Sistema de Controle de TCC deslogado. E por fim a Seção 4.2 apresenta o Sistema de Controle de TCC logado.

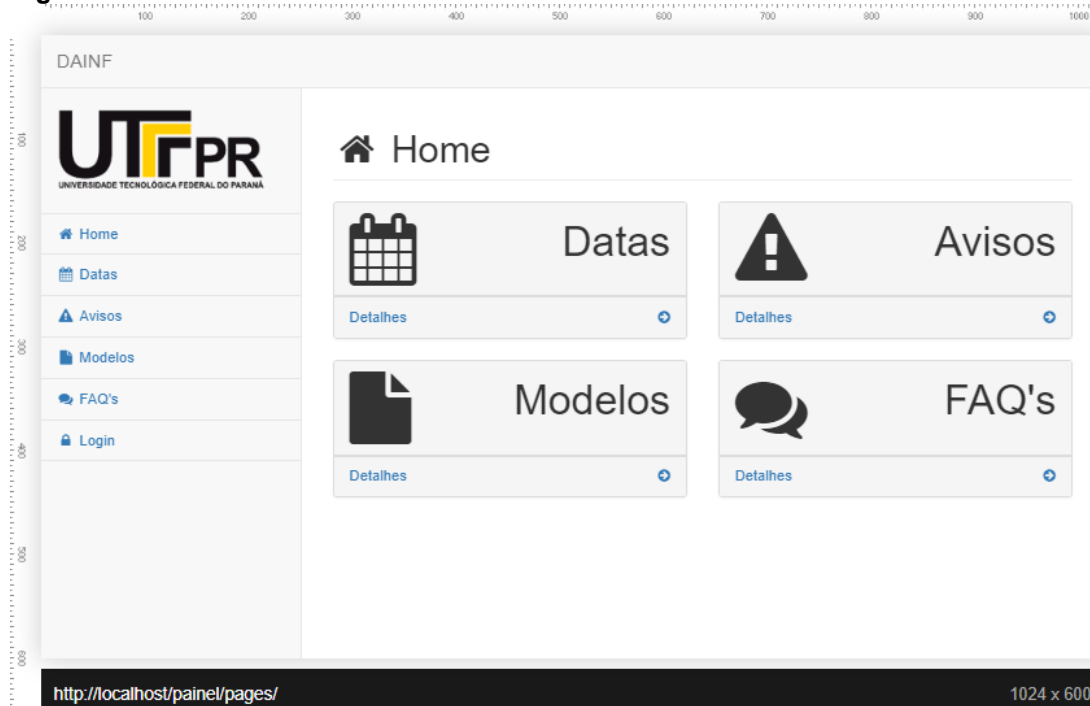
4.1 SISTEMA DE CONTROLE DO TCC

Nesta Seção são apresentadas as interfaces do Sistema de Controle de TCC antes de efetuar o *login*.

4.1.1 Home

A Figura 13 corresponde com a *home*, ou seja, a página inicial, o teste foi realizado em uma tela de 1024x600 *pixels*. A página inicial sem a realização do *login*, ou seja, a *home* deve listar as datas, avisos, modelos e perguntas frequentes.

Figura 13 - Painel de Controle: Home



Fonte: Autoria Própria

Para a apresentação das telas da interface sem *login* foi utilizado a ferramenta *Quirk Tools Screenfly*, a qual fornece testes de tela responsivo para monitores, *smartphones*, *tablets*, entre outros.

4.1.2 Datas

A Figura 14 refere-se as datas, o teste foi realizado em uma tela de notebook 1024x600 *pixels*. No cadastro de datas, os campos obrigatórios são eventos, descrição, data e entrega.

Figura 74 - Painel de Controle: Datas

The screenshot shows a web interface for 'DAINF' (Universidade Tecnológica Federal do Paraná). The main heading is 'Datas'. Below it, there is a table titled 'Datas Importantes' with the following data:

Item	Data	Evento	Descrição	Entrega
1	03/09	Proposta	Descrição Proposta	Prof. Helyane Borges
2	10/09	Relatório	Descrição Relatório	Prof. Helyane Borges
3	24/09	Cronograma	Descrição Cronograma	Prof. Helyane Borges
4	03/09	Proposta	Descrição Proposta	Prof. Helyane Borges
5	10/09	Relatório	Descrição Relatório	Prof. Helyane Borges
6	24/09	Cronograma	Descrição Cronograma	Prof. Helyane Borges

The interface also features a sidebar menu with the following items: Home, Datas, Avisos, Modelos, FAQ's, and Login. The footer of the page displays the URL 'http://localhost/painel/pages/datas.html' and the resolution '1024 x 600'.

Fonte: Autoria Própria

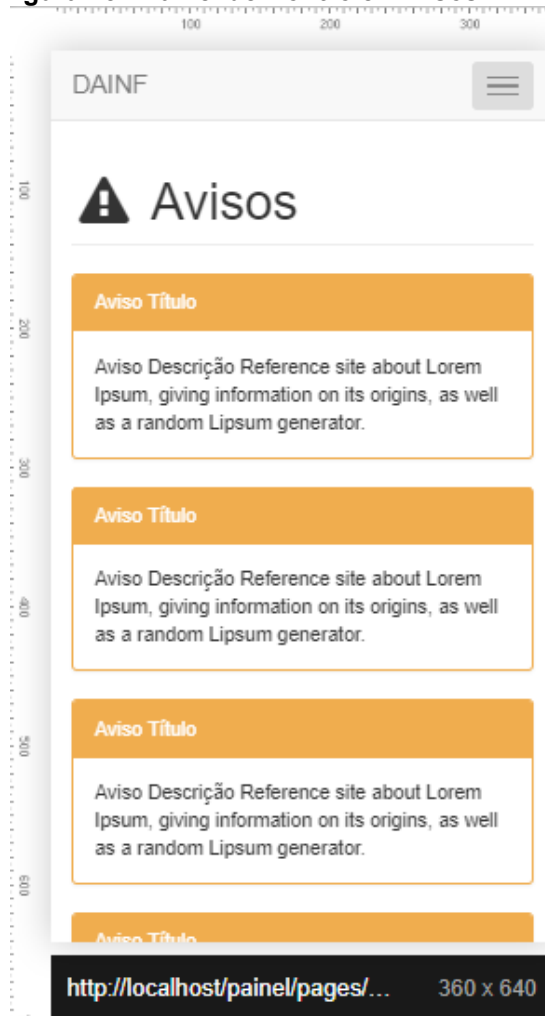
4.1.3 Avisos

A Figura 15 refere-se aos avisos, o teste foi realizado em uma tela de notebook 360x640 *pixels*. No cadastro de avisos, os campos obrigatórios são título, descrição e *link*.

Nota-se que o menu lateral nesse tamanho de tela especificado, assumiu um comportamento responsivo, alternando a navegação, pois se encontra na parte

superior alinhado à esquerda em formato de ícone. Ao clicar no ícone, o menu irá se estender pelo motivo de que o *nav* contém a classe *collapse*.

Figura 15 - Painel de Controle: Avisos

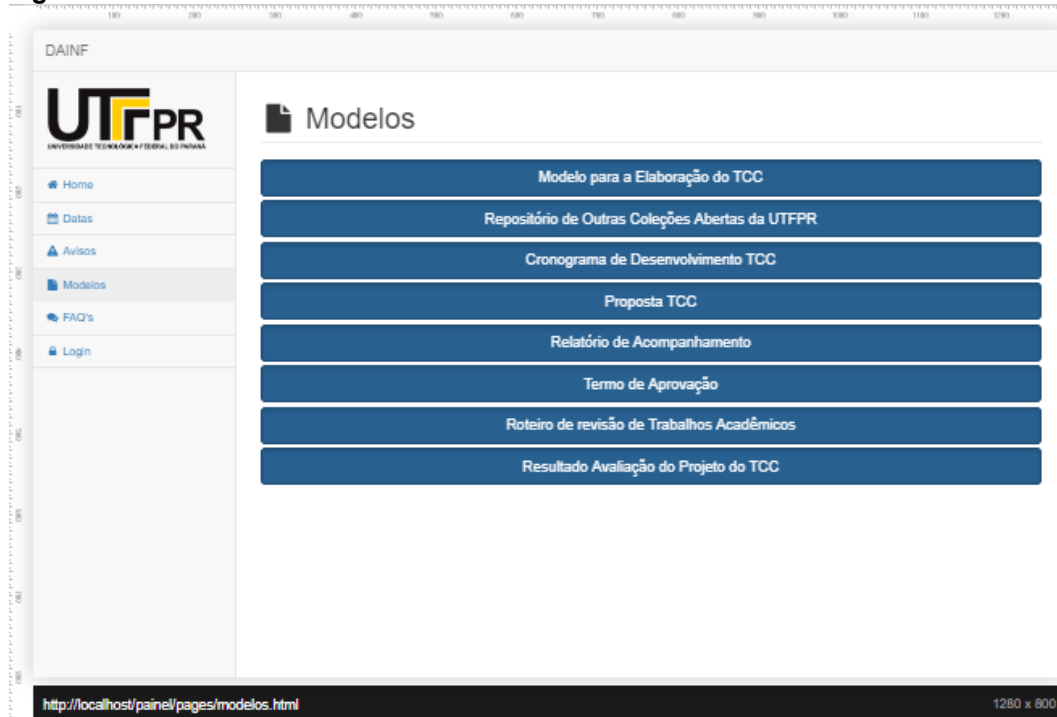


Fonte: Autoria Própria

4.1.4 Modelos

A Figura 16 refere-se aos modelos, o teste foi realizado em uma tela de 1280x800 *pixels*. No cadastro de modelos, os campos obrigatórios são nome e arquivo, o qual será salvo em um diretório raiz.

Figura 16 - Painel de Controle: Modelos



Fonte: Autoria Própria

4.1.5 Faq's

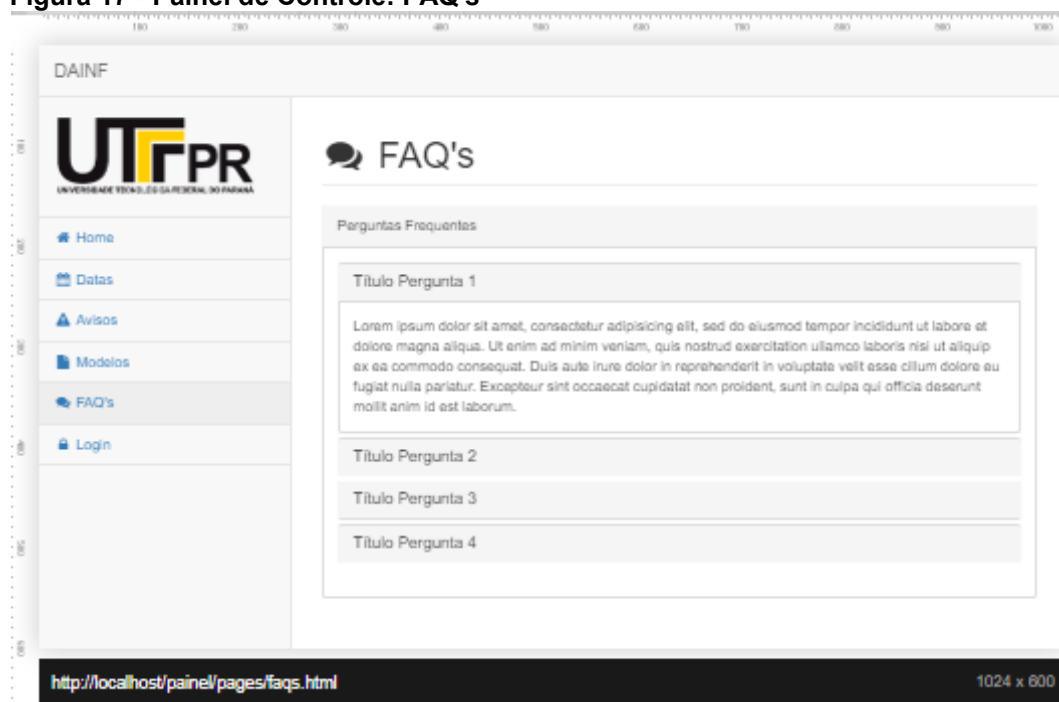
A Figura 17 refere-se as perguntas frequentes, o teste foi realizado em uma tela de 1024x600 *pixels*. No cadastro de perguntas frequentes, os campos obrigatórios são pergunta e resposta.

Essa página foi criada com o componente *collapse* do *Bootstrap*, ele é responsável por alternar a visibilidade do conteúdo com CSS e JS. O *collapse* foi criado no modelo *accordion*, que estende o padrão para criar um acordeão.

4.1.6 Login

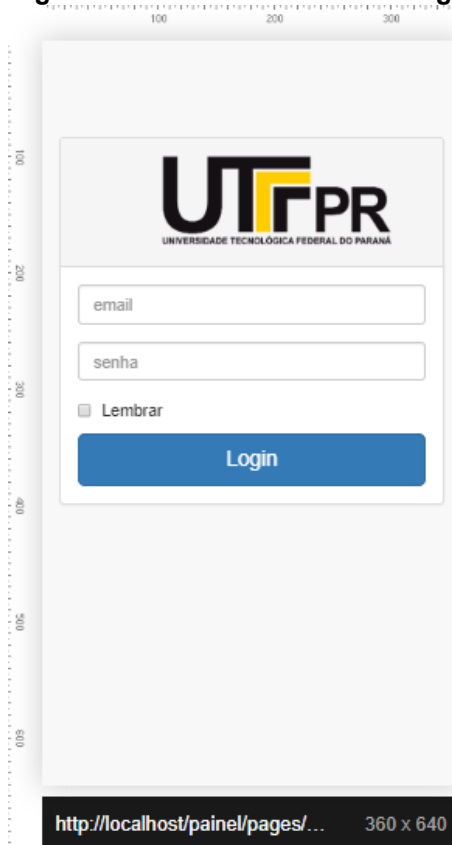
A Figura 18 refere-se a tela de *login* do painel de controle, o teste foi realizado em uma tela de 360x640 *pixels*. O *CodeIgniter* possui um *script* de criptografia de senha, que para fazer uso dessa ferramenta é preciso alterar algumas configurações de aplicação do *CodeIgniter* na pasta raiz.

Figura 17 - Painel de Controle: FAQ's



Fonte: Autoria Própria

Figura 18 - Painel de Controle: Login



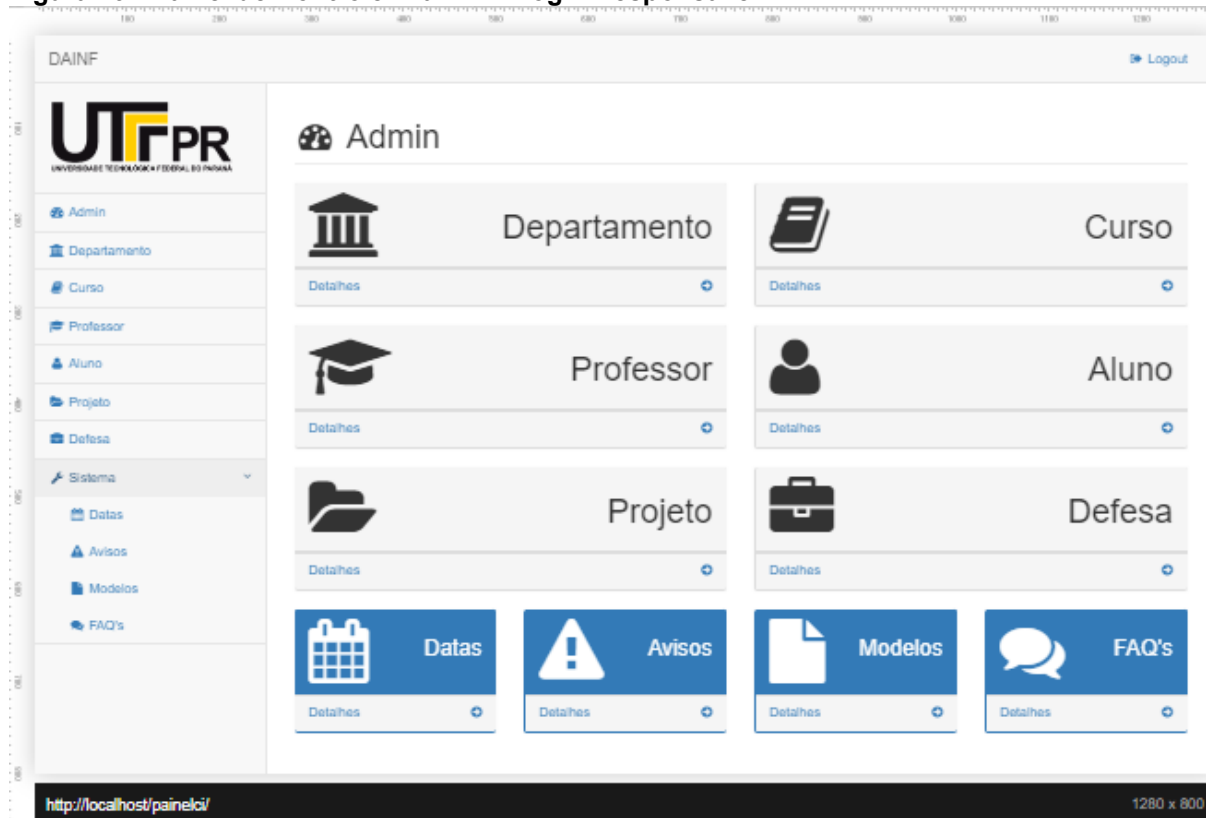
Fonte: Autoria Própria

4.2 SISTEMA DE CONTROLE DO TCC LOGADO

Nessa seção é apresentado a interface do sistema após a realização do Login. Após a efetivação do *Login*, o usuário é encaminhado para a página inicial do sistema logado, ou seja, o *admin*. O responsável deverá cadastrar departamentos, cursos, professores, alunos, projetos, defesas, como também os itens da *home*. O sistema é composto por cadastros e listagens.

A Figura 19 refere-se a tela *Admin*, ou seja, é a tela inicial do painel de controle logado, o teste foi realizado em uma tela de 1280x800 *pixels*. No menu lateral da *home* existe a seção Sistema, responsável pelos cadastros dos dados que ficam visíveis para o usuário sem *Login*, na tela *Home*.

Figura 19 - Painel de Controle: Admin – Login Responsável



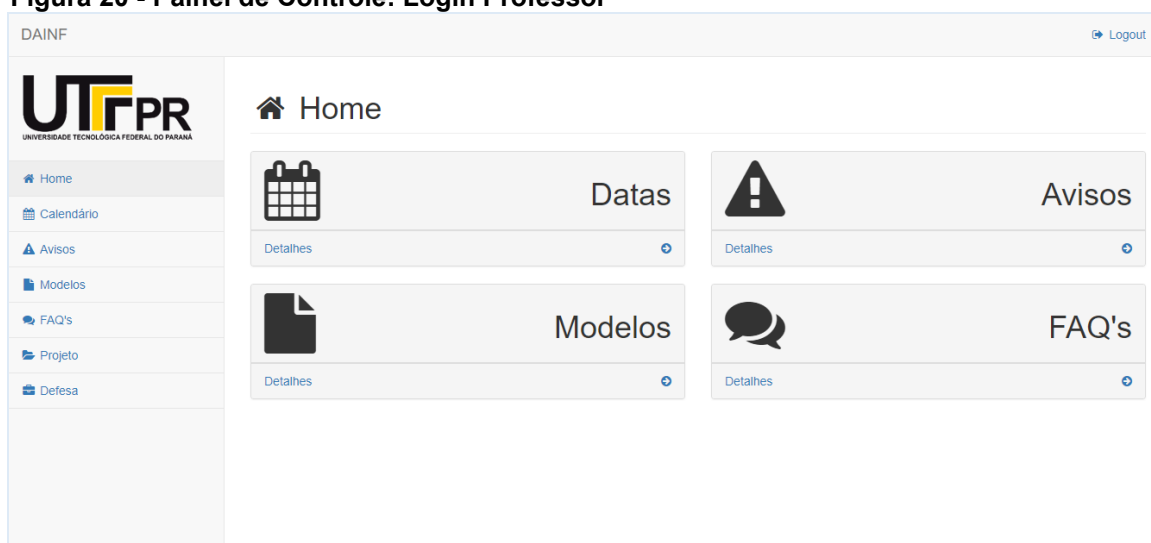
Fonte: Autoria Própria

Existem dois tipos de usuários, o responsável e o professor. O *admin* do responsável, deve listar os departamentos, cursos, professores, alunos, projetos, defesas, como também os itens da *home*. O *admin* do professor, deve listar os itens

da *home* apenas para visualização e listar os projetos e as defesas onde ele é orientador para permissão de edição.

A Figura 20 refere-se a tela inicial do painel de controle logado, por um usuário professor orientador. Os menus datas, avisos, modelos e *faqs* são apenas para visualização apresentando apenas uma listagem. E os menus projeto e defesa são apenas para edição, não podendo realizar a exclusão.

Figura 20 - Painel de Controle: Login Professor



Fonte: Autoria Própria

4.2.1 Cadastro e Lista de Departamentos

A Figura 21 refere-se a Lista de Departamentos cadastrados, observe que nessa tabela contém a coluna ações. Em cada linha dessa coluna é apresentado dois ícones, o editar e o remover. Acima dessa lista contém o botão Novo Cadastro, para cadastrar um novo Departamento.

A Figura 22 refere-se ao Formulário de Departamento, a qual refere-se ao formulário para Cadastro de Departamento. No cadastro de departamento, os campos obrigatórios são nome e sigla.

Figura 21 - Painel de Controle: Lista Departamento

DAINF Logout

UTPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Admin
Departamento
Curso
Professor
Aluno
Projeto
Defesa
Sistema

Lista Departamento

Novo Cadastro

Departamentos Cadastrados no Sistema

ID	Nome	Sigla	Ações
1	Departamento Acadêmico de Informática	DAINF	
2	Departamento Acadêmico de Informática	DAINF	
3	Departamento Acadêmico de Informática	DAINF	

Fonte: Autoria Própria

Figura 22 - Painel de Controle: Cadastro Departamento

DAINF Logout

UTPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Admin
Departamento
Curso
Professor
Aluno
Projeto
Defesa
Sistema

Cadastro Departamento

Lista / Cadastro

Todos os campos são obrigatórios.

Preencha o Formulário

Nome

Sigla

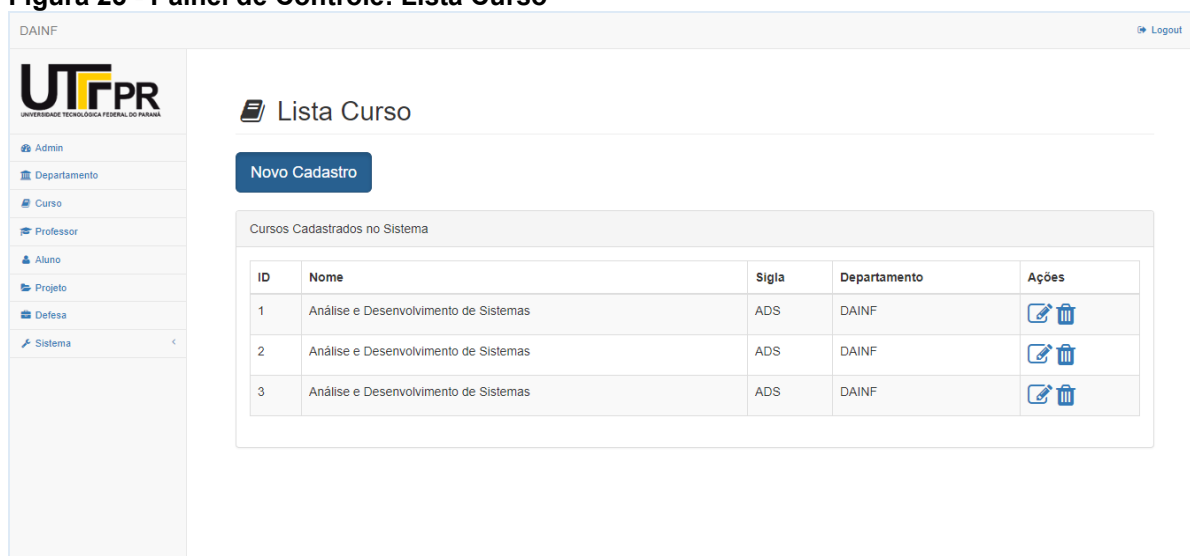
Cadastrar

Fonte: Autoria Própria

4.2.2 Cadastro e Lista de Cursos

A Figura 23 refere-se a Lista de Curso cadastrados, observe que nessa tabela contém a coluna Departamento.

Figura 23 - Painel de Controle: Lista Curso



DAINF Logout

UT FPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

- Admin
- Departamento
- Curso
- Professor
- Aluno
- Projeto
- Defesa
- Sistema

Lista Curso

[Novo Cadastro](#)

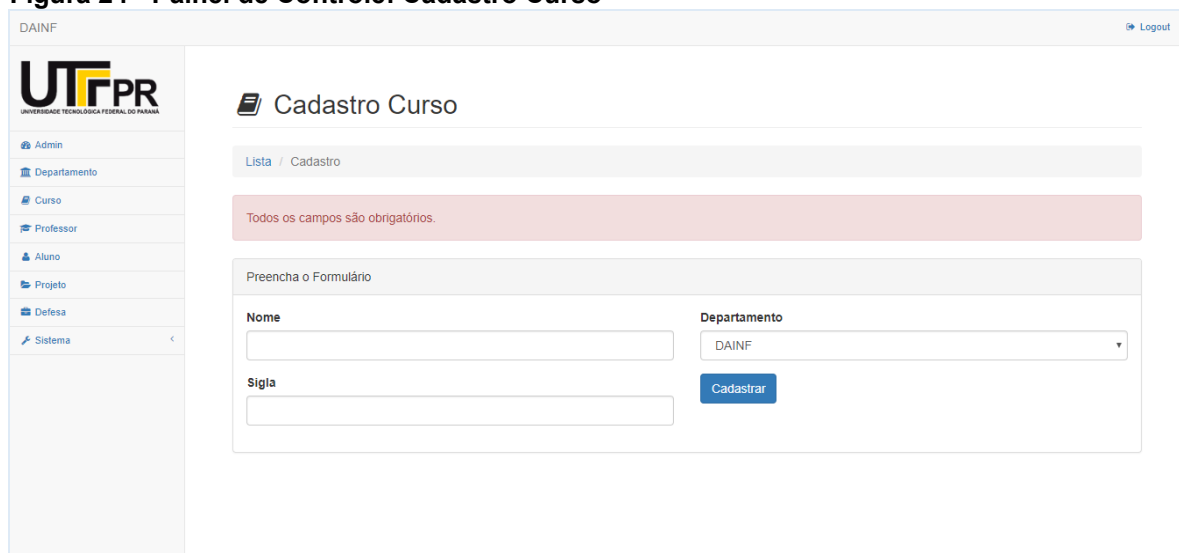
Cursos Cadastrados no Sistema

ID	Nome	Sigla	Departamento	Ações
1	Análise e Desenvolvimento de Sistemas	ADS	DAINF	✎ 🗑️
2	Análise e Desenvolvimento de Sistemas	ADS	DAINF	✎ 🗑️
3	Análise e Desenvolvimento de Sistemas	ADS	DAINF	✎ 🗑️

Fonte: Autoria Própria

A Figura 24 refere-se ao Formulário de Curso, a qual refere-se ao formulário para Cadastro de Curso, contém o *input* Departamento, com os Departamentos anteriormente cadastrados, para a atribuição ao Curso. No cadastro de curso, os campos obrigatórios são departamento, nome e sigla.

Figura 24 - Painel de Controle: Cadastro Curso



DAINF Logout

UT FPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

- Admin
- Departamento
- Curso
- Professor
- Aluno
- Projeto
- Defesa
- Sistema

Cadastro Curso

Lista / Cadastro

Todos os campos são obrigatórios.

Preencha o Formulário

Nome

Departamento

Sigla

[Cadastrar](#)

Fonte: Autoria Própria

4.2.3 Cadastro e Lista de Professores

A Figura 25 refere-se a Lista de Professor cadastrado, observe que nessa tabela contém a coluna Curso e a coluna *Status*.

Figura 25 - Painel de Controle: Lista Professor

ID	Curso	Nome	Email	Login	Senha	Status	Ações
1	ADS	Helyane	helyane@utfpr.edu.br	admin	criptorafada	Ativo	
2	ADS	Helyane	helyane@utfpr.edu.br	admin	criptorafada	Ativo	
3	ADS	Helyane	helyane@utfpr.edu.br	admin	criptorafada	Ativo	

Fonte: Autoria Própria

A Figura 26 refere-se ao Formulário de Professor, a qual refere-se ao formulário para Cadastro de Professor que contém o *input* Curso, com os Cursos anteriormente cadastrados, para atribuição ao Professor.

Figura 26 - Painel de Controle: Cadastro Professor

Fonte: Autoria Própria

Além disso, possui os campos *radio* para informar se esse professor está Ativo ou Inativo, ou seja, apto ou não para ser orientador ou banca. No cadastro de professor, os campos obrigatórios são departamento, nome, *email*, *login*, senha, repetir senha e *status*. Existe um campo *admin* para informar se ele é responsável ou professor. O campo *status* do cadastro de professor é para delimitar se ele está ativo ou inativo para ser orientador e/ou banca.

4.2.4 Cadastro e Lista de Alunos

A Figura 27 refere-se a Lista de Aluno cadastrado, observe que nessa tabela contém a coluna *Status*.

Figura 27 - Painel de Controle: Lista Aluno

RA	Nome	Email	Telefone	Status	Ações
1429116	Barbara	barbara@gmail.com	(42)99807-8211	TCC 1	
1429116	Barbara	barbara@gmail.com	(42)99807-8211	TCC 2	
1429116	Barbara	barbara@gmail.com	(42)99807-8211	TCC 1	

Fonte: Autoria Própria

A Figura 28 refere-se ao Formulário de Aluno, a qual refere-se ao formulário para Cadastro de Aluno que contém os campos *radio* para informar se esse aluno está inscrito no TCC1 ou no TCC2. No cadastro de aluno, os campos obrigatórios são registro acadêmico, nome, curso, *email* e *status*. O campo *status* do cadastro de aluno é para delimitar se com relação a disciplina de TCC 2 está sendo cursada ou foi interrompida.

Figura 8 - Painel de Controle: Cadastro Aluno

DAINF Logout

UTPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Admin
Departamento
Curso
Professor
Aluno
Projeto
Defesa
Sistema

Cadastro Aluno

Lista / Cadastro

Todos os campos são obrigatórios.

Preencha o Formulário

RA: Email:

Nome: Telefone:

Curso:

Status:
 TCC 2 - Interrompido
 TCC 2 - Cursando

Copyright © 2018

Fonte: Autoria Própria

4.2.5 Cadastro e Lista de Projetos

A Figura 29 refere-se a Lista de Projeto cadastrado. Observa-se que nessa tabela contém a coluna Orientador, Coorientador e Alunos.

Figura 29 - Painel de Controle: Lista Projeto

DAINF Logout

UTPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Admin
Departamento
Curso
Professor
Aluno
Projeto
Defesa
Sistema

Lista Projeto

Projetos Cadastrados no Sistema

10 resultados por página Pesquisar

ID	Título Proposta	Orientador	Coorientador	Data Inicio	Data Fim	Aluno(s)	Ações
1	f	Inativo	Inativo	2018-10-17	2018-10-02	Barbara	<input type="button" value="Editar"/> <input type="button" value="Excluir"/>
2	dsdsdf	Inativo	Inativo	2018-10-31	2018-11-30	n	<input type="button" value="Editar"/> <input type="button" value="Excluir"/>

Mostrando de 1 até 1 de 1 registros

Anterior **1** Próximo

Fonte: Autoria Própria

A Figura 30 refere-se ao Formulário de Projeto, a qual refere-se ao formulário para Cadastro de Projeto, o qual contém o campo *input* Aluno1, Aluno2 e Aluno3 com os Alunos anteriormente cadastrados, sendo somente o Aluno1 um campo

obrigatório. Como também o campo *input* Orientador e o campo *input* Coorientador, com os Professores anteriormente cadastrados, sendo somente o Orientador um campo obrigatório.

Figura 30 - Painel de Controle: Cadastro Projeto

The screenshot shows the 'Cadastro Projeto' form in the DAINF system. The form is titled 'Cadastro Projeto' and includes a navigation sidebar on the left with options: Admin, Departamento, Curso, Professor, Aluno, Projeto, Defesa, and Sistema. The main form area contains the following fields and options:

- Título Proposta:** A text input field with an asterisk indicating it is required.
- Orientador:** A dropdown menu with the text 'Selecione uma opção' and an asterisk indicating it is required.
- Coorientador:** A dropdown menu with the text 'Selecione uma opção'.
- Data Início:** A text input field with the format 'dd/mm/aaaa' and an asterisk indicating it is required.
- Aluno 1, Aluno 2, Aluno 3:** Three dropdown menus, each with the text 'Selecione uma opção'.
- Status:** Three radio buttons labeled 'Prorrogação', 'Aprovado', and 'Reprovado'.
- Buttons:** Two buttons labeled 'Salvar' (green) and 'Limpar' (blue).

A red banner at the top of the form area states: 'Os campos com asterisco são obrigatórios.' (Fields with an asterisk are required.)

Fonte: Autoria Própria

No cadastro de projeto, os campos obrigatórios são proposta, orientador, data início, aluno 1 e *status*. Os campos não obrigatórios são coorientador, atribuição de mais alunos e se o TCC foi prorrogado ou não. O campo *status* do cadastro de projeto é para delimitar se ele está em prorrogação, aprovado ou reprovado. Após o *status* do projeto ser alterado de desenvolvimento para aprovado, ele poderá ser cadastrado em uma defesa.

4.2.6 Cadastro e Lista de Defesas

A Figura 31 refere-se a Lista de Defesa cadastrada, observe que nessa tabela contém a coluna Título da Proposta e Banca. Observa-se que na Lista Defesa na coluna Ações contém um ícone responsável pela finalização da defesa.

Figura 31 - Painel de Controle: Lista Defesa

Fonte: Autoria Própria

A Figura 32 refere-se ao Formulário de Defesa, conforme ilustrado essa interface possui um formulário que contém o campo *input* Projeto com os Projetos anteriormente cadastrados e aprovados. Além disso, há também o campo *input* Banca1, Banca2 e Banca3, com os Professores anteriormente cadastrados.

Figura 32 - Painel de Controle: Cadastro Defesa

Fonte: Autoria Própria

No Cadastro da Defesa se a data inserida ultrapassar o prazo estipulado, é apresentado um *textarea* e botão de upload de arquivo para a inserção da Justificativa, apresentado na Figura 33, se tornando um campo obrigatório. Essa

data é inserida no cadastro de datas, onde existe uma validação para o responsável selecionar se a data que está sendo inserida é a data do limite da defesa.

Figura 33 - Painel de Controle: Justificativa Defesa

DAINF Logout

UTPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Admin
Departamento
Curso
Professor
Aluno
Projeto
Defesa
Sistema

Cadastro Defesa

Lista / Cadastro

Para cadastrar uma data fora do intervalo é preciso apresentar uma justificativa.

Preencha o Formulário

* Projeto: Seleccione uma opção

* Data: 15/11/2018

* Banca: Professor 1 Seleccione uma opção

Justificativa: Descreva o motivo do projeto ser defendido após a data limite estipulada.

Banca: Professor 2 Seleccione uma opção

Escolher arquivo Nenhum arquivo selecionado

Banca: Professor 3 Seleccione uma opção

* Hora: --:--

* Local: Local

Salvar Limpar

Fonte: Autoria Própria

No cadastro de defesa, os campos obrigatórios são projeto, banca, alunos, data, hora, local e *status*. Os campos não obrigatórios são a atribuição de mais membros na banca. E a aceitação dessa defesa fora do prazo é de responsabilidade do *admin* que poderá deferir ou indeferir. O campo *status* do cadastro de defesa é para delimitar se ele está pendente, deferida ou indeferida.

4.2.7 Finalizar Defesa

Ao finalizar uma defesa, como usuário *admin*, o sistema direciona para o *modal* da Figura 34, onde é inserido a nota dos alunos envolvidos e nesse momento é gerada a ata da defesa que ficará disponível na lista para *download*.

Qualquer alteração do usuário professor em uma defesa que já esteja com o status deferida, é alterada para pendente e a ata é removida da lista para *download* ou somente visualização.

Figura 34 - Painel de Controle: Finalizar Defesa como Admin

The screenshot shows the DAINF system interface. A modal window titled "Finalizar Defesa" is open, allowing an Admin user to enter grades. The modal contains three input fields labeled "Nota 1", "Nota 2", and "Nota 3". Below the input fields are two buttons: "Sim" (blue) and "Cancelar" (grey). The background shows a sidebar with navigation options like "Admin", "Departamento", "Curso", "Professor", "Aluno", "Projeto", "Defesa", and "Sistema". The main content area displays a table of defense records with columns for "Titulo Proposta", "Banca", "Local", "Data", "Hora", "Documentos", "Ata", "Status", and "Ações".

Titulo Proposta	Banca	Local	Data	Hora	Documentos	Ata	Status	Ações
Projeto X	Professor 3, Professor 2, novo	C202	13:00	10/07/2018	Download	Download	Pendente	[Edit] [Check] [Delete]
Proposta 2	novo, Professor 3, Professor 1	sadasd	12:00	20/10/2020	Download	Download	Deferida	[Edit] [Check] [Delete]
Proposta x	Professor 3, Professor 2	c303	12:30	05/11/2018			Pendente	[Edit] [Check] [Delete]

Fonte: Autoria Própria

Ao finalizar uma defesa, como usuário professor, o sistema direciona para o *modal* da Figura 35, onde é inserido os arquivos: Cronograma de Desenvolvimento e Relatório de Acompanhamento.

Figura 35 - Painel de Controle: Finalizar Defesa como Professor

The screenshot shows the DAINF system interface. A modal window titled "Finalizar Defesa" is open, allowing a Professor user to upload files. The modal contains a file selection field labeled "Cronograma + Relatório" with a button "Escolher arquivo" and the text "Nenhum arquivo selecionado". Below the field is a "Download" button. The background shows the same sidebar and table as in Figure 34.

Titulo Proposta	Banca	Local	Data	Hora	Documentos	Ata	Status	Ações
Projeto X	Professor 3, Professor 2, novo	C202	13:00	10/07/2018	Download	Download	Pendente	[Edit] [Check] [Delete]
Proposta 2	novo, Professor 3, Professor 1	sadasd	12:00	20/10/2020	Download	Download	Deferida	[Edit] [Check] [Delete]
Proposta x	Professor 3, Professor 2	c303	12:30	05/11/2018			Pendente	[Edit] [Check] [Delete]

Fonte: Autoria Própria

5 CONCLUSÃO

O Trabalho de Conclusão de Curso (TCC) é uma disciplina obrigatória na maioria dos cursos de graduação. No Departamento Acadêmico de Informática (DAINF), da UTFPR Campus Ponta Grossa, dois cursos de graduação são ofertados: Tecnologia em Análise e Desenvolvimento de Sistemas e Bacharelado em Ciência da computação e ambos possuem essa disciplina em sua matriz curricular.

Um professor é selecionado para gerenciar todos os procedimentos relacionados ao TCC do curso. Atualmente, esse gerenciamento é realizado manualmente e por meio de planilhas eletrônicas. Dessa maneira, percebeu-se a necessidade de desenvolver um sistema que realizasse esse gerenciamento.

O Sistema de Controle de TCC foi desenvolvido para plataforma Web. As funcionalidades do sistema foram desenvolvidas conforme o fluxo do processo para a conclusão da disciplina de TCC. Com esse sistema o responsável pela disciplina terá maior visibilidade de quem são os orientadores e alunos que estão com projetos ativos e os que devem ser reprovados na disciplina naquele semestre letivo.

Para o desenvolvimento do sistema foi feito um levantamento requisitos, diagramação e a modelagem do banco de dados. Para a implementação do sistema foram utilizadas as linguagens *HTML5*, *CSS3*, *JavaScript* e *PHP* e para o banco de dados foi selecionado o *MySQL*. O sistema desenvolvido permite a realização de novas funcionalidades, pois foi realizado com *frameworks* e ambos contêm uma documentação robusta disponível.

5.1 TRABALHOS FUTUROS

Para trabalhos futuros, seria interessante qualquer usuário criar o seu *login* e senha por meio de uma confirmação de *email*, como professor ou aluno, e ter acesso ao projeto e defesa em que está envolvido. E após a confirmação da finalização da defesa criar uma sessão para geração de documentos, como: Declaração de Orientação e Declaração de Banca.

REFERÊNCIAS

BOOTSTRAP. **Bootstrap v4.1.3.** Disponível em: <<https://getbootstrap.com/docs/4.1/getting-started/introduction/>>. Acesso em: 14 nov. 2018.

CODEIGNITER. Guia do Usuário do CodeIgniter. 26 jul. 2016. Disponível em: <<https://cibr.github.io/User-Guide-CodeIgniter-PtBr/index.html>>. Acesso em: 12 set. 2018.

COEPP. **Resolução nº 120/06 – COEPP, de 07 de dezembro de 2006.** Regulamento do Trabalho de Conclusão de Curso para os cursos de graduação da UTFPR. Disponível em: <<http://portal.utfpr.edu.br/documentos/legislacao-da-utfpr/legislacao-dos-cursos-de-graduacao/regulamento-de-tcc-da-graduacao/view>>. Acesso em: 10 out. 2018.

EIS, Diego. Tableless. **Introdução sobre Media Queries.** 06 jul. 2009. Disponível em: <<https://tableless.com.br/introducao-sobre-media-queries/>>. Acesso em: 02 set. 2018.

GET BOOTSTRAP. ABOUT – Bootstrap. **The most popular HTML, CSS, and JS library in the world.** 19 ago. 2011. Disponível em: <<https://getbootstrap.com/docs/4.1/getting-started/introduction/>>. Acesso em: 11 set. 2018.

MARCOTTE, Ethan. A List Apart & Our Authors. **Responsive Web Design.** 25 mai. 2010. Disponível em: <<https://alistapart.com/article/responsive-web-design>>. Acesso em: 02 set. 2018.

MILANI, André. **Construindo aplicações web com PHP e MySQL.** 4. ed. São Paulo, SP: Novatec, 2012.

W3C. **W3C.** Disponível em: <<https://www.w3.org/>>. Acesso em: 02 set. 2018.

W3Schools. The World's Largest Web Developer Site. **W3Schools.** Disponível em: <<https://www.w3schools.com/>>. Acesso em: 11 set. 2018.