

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

THIAGO AKIRA KAMIDA

**APLICAÇÃO *WEB* PARA AUXILIAR ALUNOS DA UNIVERSIDADE
TECNOLÓGICA FEDERAL DO PARANÁ NA ESCOLHA DE
ORIENTADOR PARA TRABALHO DE CONCLUSÃO DE CURSO**

TRABALHO DE CONCLUSÃO DE CURSO

PONTA GROSSA

2017

THIAGO AKIRA KAMIDA

**APLICAÇÃO *WEB* PARA AUXILIAR ALUNOS DA UNIVERSIDADE
TECNOLÓGICA FEDERAL DO PARANÁ NA ESCOLHA DE
ORIENTADOR PARA TRABALHO DE CONCLUSÃO DE CURSO**

Trabalho de Conclusão de Curso
apresentado como requisito parcial à
obtenção do título de Bacharel em Ciência
da Computação, do Departamento
Acadêmico de Informática, da
Universidade Tecnológica Federal do
Paraná.

Orientador: Prof. MSc. Vinícius Camargo
Andrade

PONTA GROSSA

2017



TERMO DE APROVAÇÃO

APLICAÇÃO WEB PARA AUXILIAR ALUNOS DA UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ NA ESCOLHA DE ORIENTADOR PARA TRABALHO DE CONCLUSÃO DE CURSO

por

THIAGO AKIRA KAMIDA

Este Trabalho de Conclusão de Curso (TCC) foi apresentado em 23 de maio de 2017 como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Prof. MSc. Vinícius Camargo Andrade
Orientador(a)

Richard Duarte Ribeiro
Membro titular

Rafael dos Passos Canteri
Membro titular

Prof. Dr. Ionildo José Sanches
Responsável pelo Trabalho de Conclusão
de Curso

Prof. Dr. Erikson Freitas de Moraes
Coordenador do curso

- A Folha de Aprovação assinada encontra-se arquivada na Secretaria Acadêmica –

AGRADECIMENTOS

Agradeço aos meus pais por todo o apoio dado ao longo desses anos na universidade.

Agradeço também ao Professor Vinícius Camargo Andrade pela orientação durante este trabalho.

Aos professores e amigos da Universidade Tecnológica Federal do Paraná, Campus Ponta Grossa, pela contribuição na minha formação profissional.

RESUMO

KAMIDA, Thiago Akira. **Aplicação Web para Auxiliar Alunos da Universidade Tecnológica Federal do Paraná na Escolha de Orientador para Trabalho de Conclusão de Curso**. 2017. 66 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2017.

Atualmente os alunos dos cursos de Tecnologia em Análise e Desenvolvimento de Sistemas e Bacharelado em Ciência da Computação, ambos do Departamento Acadêmico de Informática da Universidade Tecnológica Federal do Paraná, Câmpus Ponta Grossa, enfrentam dificuldades em encontrar um professor para orientá-los no Trabalho de Conclusão de Curso, pois as informações relativas a cada docente, como por exemplo, áreas de atuação, pesquisas, desenvolvimentos, nem sempre se encontram disponíveis e claras. Sendo assim, tanto a escolha por um orientador na área de maior interesse do discente quanto os projetos de pesquisas de professores são prejudicados. Este trabalho propõe o desenvolvimento de uma aplicação *web* para facilitar o acesso a informações sobre Trabalhos de Conclusão de Curso além de fornecer acesso aos trabalhos já defendidos com o intuito de facilitar ao aluno a busca das áreas de atuação de cada professor, podendo inclusive dar continuidade a trabalhos orientados pelo mesmo. O desenvolvimento utiliza a metodologia de gerenciamento *Kanban* para o gerenciamento do projeto e a metodologia de desenvolvimento *eXtreme Programming* para a implementação do mesmo, além do *Framework* Django para a codificação, fazendo uso das linguagens de programação Python e JavaScript em conjunto com as linguagens de marcação HTML e CSS.

Palavras-chave: Website. Trabalho de Conclusão de Curso. Orientação.

ABSTRACT

KAMIDA, Thiago Akira. **Web Application to Assist Students of the Federal Technological University of Paraná in Choosing an Advisor to Senior Project.** 2017. 66 p. Work of Conclusion Course (Graduation in Computer Science) - Federal University of Technology - Paraná. Ponta Grossa, 2017.

Nowadays, students of Analysis and System Development and students of Computer Science bachelor degree in the Academic Department of Informatics at Federal University of Technology - Paraná, campus Ponta Grossa face difficulties finding a professor to guide them through the Senior Project. One of the reasons is that the information about the professors such as, area of knowledge, researches, and work in development are not always available or easy to access. Therefore, the choice of a advisor by the common area of interest of the student and the professor's research projects are affected for that lack of access to information. This paper presents the development of a *web* application to facilitate the access to information related to Senior Projects. This *web* application also aims to provide access to works presented in the previous years in order to help students searching for the areas of expertise of each professor and students who are looking for continuing the development of previous projects. This project will be developed using *Kanban* as management methodology and Extreme Programming (XP) as development methodology. In addition, for the development of this project it was used Django as a *Framework* to develop the application, Python and JavaScript as programming languages, and HTML and CSS as mark-up languages.

Keywords: *Website*. Senior Project. Advisor.

LISTA DE ILUSTRAÇÕES

Figura 1 – Quadro de <i>Kanban</i>	16
Figura 2 – Diagrama de Caso de Uso UML.....	17
Figura 3 – Aspectos Estáticos.....	18
Figura 4 – Exemplo de uma Classe.....	19
Figura 5 – Exemplo diagrama de Atividade Simples.....	20
Figura 6 – Exemplo de Diagrama de Componentes.....	21
Figura 7 – Diagrama de Implantação da Virtual LTDA.....	22
Figura 8 – Diagrama de Caso de Uso Aluno.....	28
Figura 9 – Diagrama de Caso de Uso Professor TCC.....	29
Figura 10 – Diagrama de Caso de Uso Professor.....	30
Figura 11 – Diagrama de Atividades <i>Login</i>	31
Figura 12 – Diagrama de Atividades Editar Perfil.....	32
Figura 13 – Diagrama de Atividades Buscar Professor.....	32
Figura 14 – Diagrama de Atividades Buscar Trabalhos.....	33
Figura 15 – Diagrama de Classe.....	34
Figura 16 – Diagrama de Componentes.....	35
Figura 17 – Diagrama de Implantação.....	35
Figura 18 – Quadro de <i>Kanban</i> Fase Inicial.....	36
Figura 19 – Quadro de <i>Kanban</i> Primeira Etapa.....	37
Figura 20 – Modelo Conceitual Inicial.....	38
Figura 21 – Modelo Lógico Inicial.....	38
Figura 22 – Quadro de <i>Kanban</i> ao Início da Segunda Fase.....	39
Figura 23 – Configuração de URLs.....	40
Figura 24 – Modelo Conceitual Perfil.....	40
Figura 25 – Modelo Lógico Perfil.....	41
Figura 26 – Exemplo autenticação.....	41
Figura 27 – Terceira Etapa <i>Kanban</i>	42
Figura 28 – Admin.....	43
Figura 29 – Logout.....	43
Figura 30 – Gravatar.....	43
Figura 31 – Quarta Etapa <i>Kanban</i>	44
Figura 32 – Quinta Etapa <i>Kanban</i>	45
Figura 33 – Sexta Etapa <i>Kanban</i>	46
Figura 34 – Sétima Etapa <i>Kanban</i>	47
Figura 35 – Página Inicial.....	48
Figura 36 – Página Professores.....	49
Figura 37 – Página Áreas.....	50
Figura 38 – Página de Detalhes da Área.....	50

Figura 39 – Página de Detalhes da Disciplina.....	51
Figura 40 – Página de <i>Posts</i> com Usuário com Permissões de Post.....	52
Figura 41 – Página de <i>Posts</i> com Usuário Sem Permissão de Post.....	52
Figura 42 – Página Professor/Área.	53
Figura 43 – Página de <i>Login</i>	54
Figura 44 – Página de Perfil com Usuário Autenticado	54
Figura 45 – Página de Perfil com Usuário não Autenticado	55
Figura 46 – Página de Postagens do Professor Autenticado	56
Figura 47 – Página de Postagens do Professor Não Autenticado.....	56
Figura 48 – Página de TCCs do Professor Autenticado	57
Figura 49 – Página de TCCs do Professor Não Autenticado	58
Figura 50 – Página de Contato.....	58
Figura 51 – Página de <i>Login</i> Ambiente Administrativo.....	59
Figura 52 – Página Inicial do Ambiente Administrativo	60

LISTA DE SIGLAS

ANSI	<i>American National Standards Institute</i>
CSS	<i>Cascading Style Sheets</i>
HTML	<i>Hypertext Markup Language</i>
ISO	<i>International Organization for Standardization</i>
MTC	<i>Model-Template-Controller</i>
MVC	<i>Model-View-Controller</i>
NICBr	Núcleo de Informação e Coordenação do Ponto BR
SGBD	Sistema de Gerenciamento de Banco de Dados
SQL	<i>Structured Query Language</i>
TCC	Trabalho de Conclusão de Curso
UML	Unified Modeling Language
UTFPR	Universidade Tecnológica Federal do Paraná
XP	<i>Extreme Programming</i>

SUMÁRIO

1 INTRODUÇÃO	11
1.1 PROBLEMA	13
1.2 OBJETIVOS.....	13
1.2.1 Objetivo Geral.....	14
1.2.2 Objetivos Específicos.....	14
1.3 ORGANIZAÇÃO DO TRABALHO.....	14
2 REVISÃO BIBLIOGRÁFICA	15
2.1 METODOLOGIAS DE GERENCIAMENTO E DESENVOLVIMENTO	15
2.1.1 Quadro de Kanban.....	15
2.2 LINGUAGEM UNIFICADA DE MODELAGEM – UML.....	16
2.2.1 Diagrama de Caso De Uso	17
2.2.2 Diagramas de Classe.....	18
2.2.2.1 Classificadores.....	18
2.2.2.2 Relacionamentos	19
2.2.3 Diagrama de Atividade.....	19
2.2.4 Diagrama de Componente	20
2.2.5 Diagrama de Implantação	21
2.2.5.1 Nós.....	21
2.3 FERRAMENTAS DE DESENVOLVIMENTO	22
2.3.1 Linguagem Python.....	23
2.3.2 Framework Django e Arquitetura MVC	23
2.3.3 HTML	24
2.3.4 CSS	25
2.3.5 JavaScript	25
2.4 SISTEMA DE GERENCIAMENTO DE BANCO DE DADOS (SGBD).....	25
2.4.1 Banco De Dados Relacional	26
2.4.2 Linguagem de Consulta Estruturada – SQL	26
3 DESENVOLVIMENTO.....	27
3.1 MODELAGEM.....	28
3.1.1 Diagrama de Caso de Uso.....	28
3.1.2 Diagrama de Atividades.....	30
3.1.3 Diagrama de Classe	33
3.1.4 Diagrama de Componente.....	34
3.1.5 Diagrama de Implantação	35
3.2 QUADRO DE KANBAN.....	36
3.2.1 Primeira Etapa de Desenvolvimento.....	36
3.2.2 Segunda Etapa de Desenvolvimento.....	39
3.2.3 Terceira Etapa de Desenvolvimento	42

3.2.4 Quarta Etapa de Desenvolvimento	44
3.2.5 Quinta Etapa de Desenvolvimento.....	45
3.2.6 Sexta Etapa de Desenvolvimento	46
3.2.7 Sétima Etapa de Desenvolvimento	47
4 RESULTADO	48
4.1 PÁGINA INICIAL.....	48
4.2 PÁGINA DE PROFESSORES	49
4.3 PÁGINA DE ÁREAS	49
4.4 PÁGINA DE DETALHES DA ÁREA.....	50
4.5 PÁGINA DE DETALHES DA DISCIPLINA.....	51
4.6 PÁGINA DE POSTS	51
4.7 PÁGINA DE PROFESSOR/ÁREA	53
4.8 PÁGINA DE LOGIN	53
4.9 PÁGINA DE PERFIL DO PROFESSOR	54
4.10 PÁGINA DE POSTAGENS.....	55
4.11 PÁGINA TCCS	57
4.12 PÁGINA DE CONTATO	58
4.13 LOGIN AMBIENTE ADMINISTRATIVO.....	59
4.14 PÁGINA INICIAL DO AMBIENTE ADMINISTRATIVO	59
5 CONCLUSÃO.....	61
5.1 TRABALHOS FUTUROS	61
REFERÊNCIAS.....	63

1 INTRODUÇÃO

Atualmente o uso de computadores e da Internet facilitam o trabalho e lazer das pessoas. Nos Estados Unidos, no ano de 2011, aproximadamente 71% dos adultos possuíam Internet em casa e 75% possuíam um computador. Atualmente, esta média já está em 89% para adultos com acesso à Internet. Situação um pouco diferente a nível global, em que a porcentagem não ultrapassa 67% (POUSHTER, 2016).

Segundo Poushter (2016), no Brasil, a Internet é utilizada por 60% dos adultos. Este acesso não ocorre apenas por computadores, mas também pela utilização de *smartphones* e dos *tablets*. Nos Estados Unidos, a quantidade de adultos que possuíam um *smartphone* em 2011 era de 35%, e em 2015, esse número subiu para 68%. O número de usuários de *tablets* em 2010 que era apenas de 3%, em 2015 chegou a 45% (ANDERSON, 2015).

Este amplo uso da Internet, está muitas vezes relacionado a alguns fatores sociais, como por exemplo, a idade das pessoas, o nível de educação e a renda. Em países desenvolvidos, como os Estados Unidos, o uso da Internet por adultos com idades entre 18 e 34 anos é de 99%, no Canadá e Itália, 100%. Em contrapartida, para adultos acima de 35 anos, nos Estados Unidos é de 85%, no Canadá 87% e na Itália 65% (POUSHTER, 2016).

No Brasil, entretanto, a porcentagem de adultos usando a Internet é relativamente menor do que em países mais desenvolvidos, e está em aproximadamente 82% para adultos com idades entre 18 a 34 anos e apenas 44% para adultos acima de 35 anos (POUSHTER, 2016). Em relação ao nível de escolaridade, de acordo com o Núcleo de Informação e Coordenação do Ponto BR (NIC.br), no Brasil, adultos analfabetos ou com educação infantil que usam a Internet não somam 5% e a porcentagem de adultos com nível fundamental que acessam a Internet é de apenas 44%. Em contrapartida, adultos com nível superior de escolaridade, somam 97%. A renda familiar também é um fator que influencia o uso da Internet. No país apenas 17% dos adultos com renda de até um salário mínimo acessam a rede e 76% para os adultos com renda familiar média que varia entre três salários mínimos a cinco salários mínimos. Já adultos com renda maior de dez salários mínimos somam 95% de acesso à Internet (NIC.br, 2016).

A partir dos anos 90, quando emergiu a *web 2.0*, a Internet tem se tornado uma plataforma colaborativa, em que companhias e usuários podem tirar proveito de dados que foram disponibilizados na rede por eles mesmos (KAMBIL, 2008). Conseqüentemente, estudantes, desenvolvedores, companhias e pesquisadores procuram estudar, criar e gerenciar tecnologias e aplicações de uma forma que estas continuem a ajudar no crescimento da Internet, bem como facilitar o uso da mesma para as pessoas.

No Brasil, as universidades também oferecem acesso à Internet e a muitos outros meios de informação para seus alunos. Além disto, os alunos também têm acesso a qualificações necessárias para a construção de um profissional que, por sua vez, tem que ser capaz de enfrentar os desafios da sociedade através de toda teoria e de toda prática vistas em sua vida acadêmica (BENITO-OSORIO, 2013).

Trabalhos, teses e artigos são criados e escritos em universidades e instituições de ensino ao redor do mundo. Diariamente, alunos, professores e orientadores trabalham e estudam para a elaboração de trabalhos acadêmicos nas mais variadas áreas do conhecimento. Entre estes, destaca-se o Trabalho de Conclusão de Curso (TCC), que é desenvolvido por um ou mais acadêmicos sob a orientação de um professor.

De acordo com o Regulamento do Trabalho de Conclusão de Curso para os Cursos de Graduação da UTFPR, o TCC tem como alguns de seus objetivos: desenvolver a capacidade de aplicação dos conceitos e teorias adquiridas durante o curso, desenvolver a capacidade de planejamento e também a disciplina para resolver problemas dentro das diversas áreas de formação, despertar o interesse pela pesquisa como meio para resolução de problemas e estimular a inovação tecnológica (UTFPR, 2016).

Por ser um trabalho tão importante, muitas vezes os alunos possuem dúvidas sobre o que desenvolver e/ou muitas vezes não sabem como definir um orientador. Atualmente, no Câmpus Ponta Grossa, os alunos de Bacharelado em Ciência da Computação e Tecnologia em Análise e Desenvolvimento de Sistemas não possuem uma aplicação que facilite o acesso a informações dos professores em conjunto com informações sobre trabalhos de conclusão de curso.

Este trabalho propõe o desenvolvimento de um *website*, para facilitar a escolha de um professor orientador por parte dos alunos considerando as áreas de interesse de ambos. Além disso, este sistema permitirá acesso aos Trabalhos de

Conclusão de Curso autorizados e disponibilizados pelo professor para que os alunos possam fazer o *download* dos mesmos e utilizá-los como referência para a elaboração de novos trabalhos dando continuidade aos já existentes.

1.1 PROBLEMA

Atualmente na Universidade Tecnológica Federal do Paraná, Câmpus Ponta Grossa, quando um aluno possui interesse em escolher um orientador para o Trabalho de Conclusão de Curso, ele precisa enviar um e-mail ou encontrar outra maneira de se comunicar para verificar a disponibilidade do professor para orientá-lo.

Além disso, informações como áreas de conhecimento e áreas de interesse de cada professor não estão concentradas em uma mesma aplicação, o que ocasiona em uma maior dificuldade por parte dos alunos em encontrar estas informações.

Neste contexto, uma aplicação *web* aproximaria alunos e professores. A partir desta aplicação os alunos encontrariam informações sobre a área de atuação de seus professores bem como os trabalhos de conclusão de curso orientados e disponibilizados por eles. Nesta mesma ferramenta, os alunos poderiam enviar uma breve descrição do projeto que possuem interesse em desenvolver como trabalho de conclusão de curso e os professores, por sua vez, tomariam nota dos potenciais orientandos e seus respectivos projetos, podendo, inclusive, dar sugestões ou indicar um novo projeto já em andamento.

Neste trabalho, foi utilizada a metodologia ágil para gestão e planejamento de projetos de *software Kanban* para elaborar o projeto e implementá-lo. No desenvolvimento foram empregadas as tecnologias de desenvolvimento *web*, tais como: HTML, CSS, JavaScript e Python com o *framework* de desenvolvimento *web Django*.

1.2 OBJETIVOS

Os objetivos deste trabalho são descritos a seguir.

1.2.1 Objetivo Geral

Desenvolver uma aplicação *web* utilizando metodologias ágeis para gestão e planejamento de projetos de *software* com a finalidade de relacionar alunos que procuram temas de TCC com os professores da instituição (UTFPR-PG).

1.2.2 Objetivos Específicos

- Estruturar e aplicar a metodologia *Kanban*;
- Modelar o sistema utilizando UML;
- Estruturar e aplicar a metodologia *eXtreme Programming*;
- Implementar o sistema;

1.3 ORGANIZAÇÃO DO TRABALHO

Este trabalho está dividido em cinco capítulos. O Capítulo 2 apresenta a revisão bibliográfica do trabalho. O Capítulo 3 descreve o desenvolvimento da aplicação *web* e a modelagem dos diagramas. O Capítulo 4 exhibe os resultados obtidos por este trabalho. E, por fim, o Capítulo 5 apresenta as conclusões e propostas para trabalhos futuros.

2 REVISÃO BIBLIOGRÁFICA

Este Capítulo aborda os tópicos referentes à revisão bibliográfica deste trabalho. A seção 2.1 apresenta a metodologia de gerenciamento e desenvolvimento de *software* utilizadas na implementação da aplicação. A seção 2.2 disserta sobre Linguagem Unificada de Modelagem (UML). As ferramentas de desenvolvimento são descritas na seção 2.3. E, por fim, a seção 2.4 relata sobre sistema gerenciador de banco de dados.

2.1 METODOLOGIAS DE GERENCIAMENTO E DESENVOLVIMENTO

A seção 2.1.1 apresenta a metodologia de controle de fluxos de produção/desenvolvimento *Kanban*. A seção 2.1.2 descreve a metodologia de desenvolvimento de *software* eXtreme Programming (XP). Adicionalmente, serão apresentados alguns critérios básicos da metodologia.

2.1.1 Quadro de *Kanban*

Segundo Ikonen et al. (2011), *Kanban* é uma palavra de origem japonesa que se refere a um quadro de avisos. Ao ser usado como gerenciador de produção esta palavra é referenciada como um sistema de agendamento que informa "o que", "quando" e "quanto" a ser produzido.

A utilização do *Kanban* em ambientes de produção possui vários benefícios, como por exemplo: a melhora do fluxo de desenvolvimento, prevenção de produção exagerada, controle de níveis de operação, visualização do cronograma e o gerenciamento de processos (IKONEN et al., 2011). Além disso, a principal característica do *Kanban* é diminuir o trabalho em progresso (WIP, em inglês *Work in Progress*), ou seja, a soma de todas as tarefas que estão sendo desenvolvidas naquele momento (AHMAD et al., 2013).

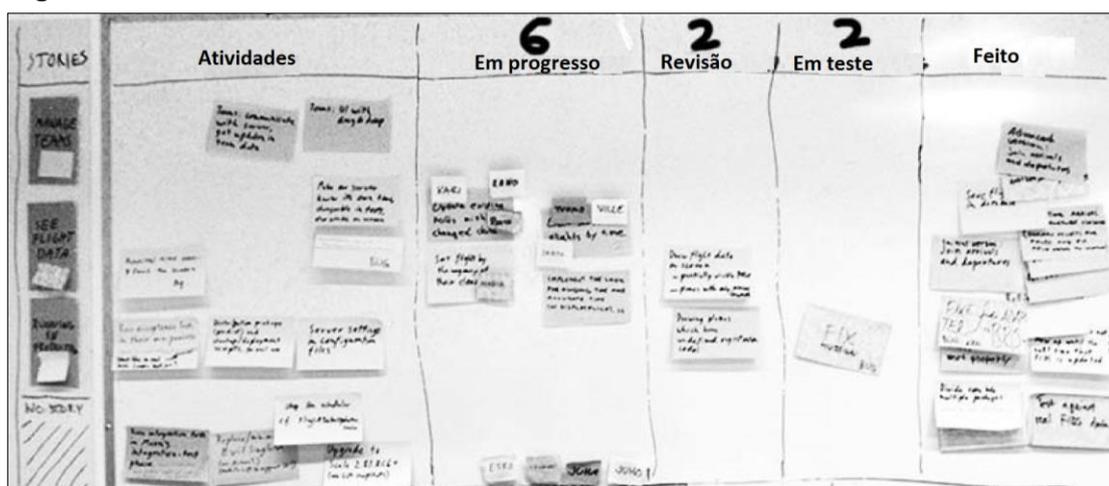
O uso do *Kanban* no desenvolvimento de *software* originou-se em 2004 quando David J. Anderson estava ajudando um pequeno grupo de TI na Microsoft e percebeu que as tarefas estavam sendo mal executadas. Neste contexto, o *Kanban* solucionou o problema de gerenciamento de projeto do time, visto que em ambientes

de projetos de TI, a metodologia preza por constantes entregas de tarefas produzidas, rápidos ciclos de *feedbacks* e a falta de interações obrigatórias (AHMAD et al., 2013).

No quadro de *Kanban*, cada atividade do projeto a ser desenvolvida é colocada em um cartão, o qual será utilizado para o controle de fluxo nas diferentes fases da produção. Conseqüentemente, com o uso do quadro de *Kanban* possibilita uma melhor visualização das tarefas, do fluxo de trabalho e do ciclo de criação (IKONEN et al., 2011).

A Figura 1 ilustra um exemplo de utilização do quadro de *Kanban*. Neste exemplo, as atividades são identificadas pelos *post-its*, por sua vez, os estados de produção são distinguidos pelas colunas: Atividades, Em progresso, Revisão, Em teste, Feito.

Figura 1 – Quadro de *Kanban*



Fonte: Traduzido de Ikonen et al. (2011).

2.2 LINGUAGEM UNIFICADA DE MODELAGEM – UML

De acordo com Berardi, Calvanese e De Giacomo (2005), a Linguagem Unificada de Modelagem (UML, em inglês *Unified Modeling Language*) é um formalismo padrão para análise e *design* de *software*. Em termos simples, pode-se dizer que UML é uma maneira formal de documentar o desenvolvimento de um *software*.

A UML foi proposta pelo OMG (*Object Management Group*) para a modelagem de sistemas e se tornou um padrão de uso por sua facilidade de uso e notação intuitiva (BORGES; MOTA, 2007).

De acordo com Borges e Mota (2007), o modelo UML representa a descrição de um conjunto de objetos e suas respectivas interações em uma aplicação. Sendo assim, um modelo UML pode ser visto como a representação do sistema em uma maneira formal.

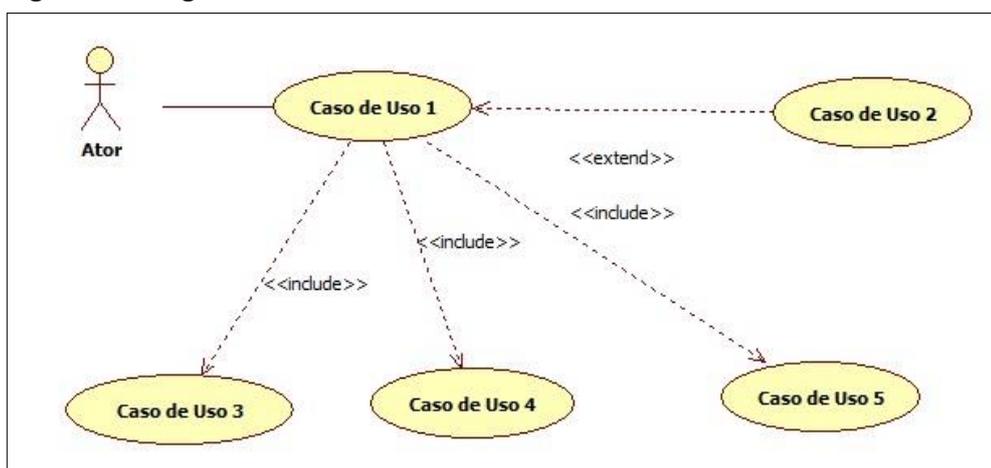
2.2.1 Diagrama de Caso De Uso

De acordo com Booch, Rumbaugh e Jacobson (2006), um Diagrama de Caso de Uso é uma construção de modelagem orientada a objetos que é usada para definir o comportamento de um sistema. Esta representação se dá pelo conjunto de casos de usos e atores e seus relacionamentos (SENGUPTA; BHATTACHARYA, 2006).

Segundo Sengupta e Bhattacharya (2006), as interações entre o usuário e o sistema são descritas através de um protótipo de ação claro junto com um possível conjunto de ações alternativas. Conseqüentemente, pode-se dizer que o diagrama de caso de uso é usado para mostrar as interações entre usuário e o sistema.

A Figura 2 ilustra um diagrama UML de caso de uso que mostra o relacionamento entre um ator e casos de uso.

Figura 2 – Diagrama de Caso de Uso UML

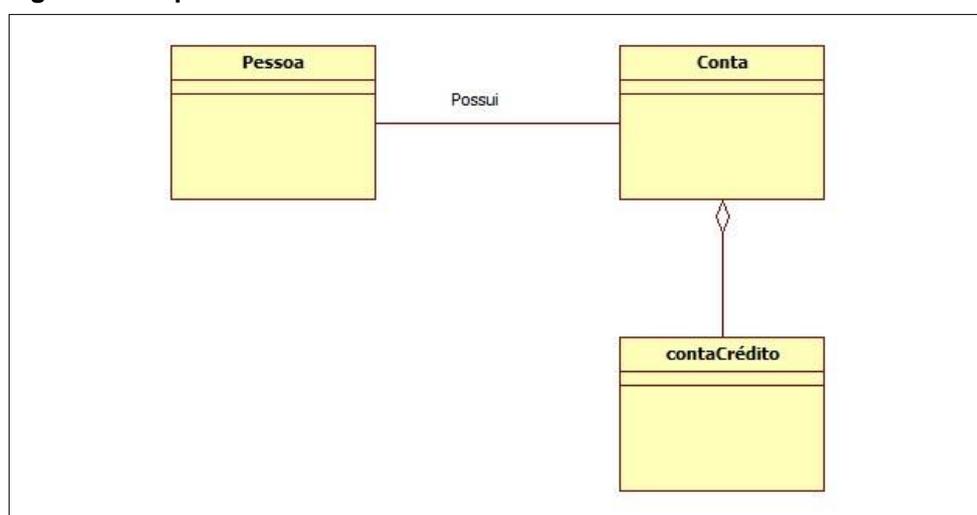


Fonte: Traduzido de SENGUPTA e BHATTACHARYA (2006).

2.2.2 Diagramas de Classe

Segundo Booch, Rumbaugh e Jacobson (2006), os diagramas de classe são usados para fazer a modelagem da visão estática do projeto de um sistema. Estes diagramas modelam conceitos do domínio da aplicação e também aspectos estruturais do sistema usando classificadores e seus relacionamentos (BORGES; MOTA, 2007). A Figura 3 exibe um diagrama de classes e seus aspectos estáticos.

Figura 3 – Aspectos Estáticos



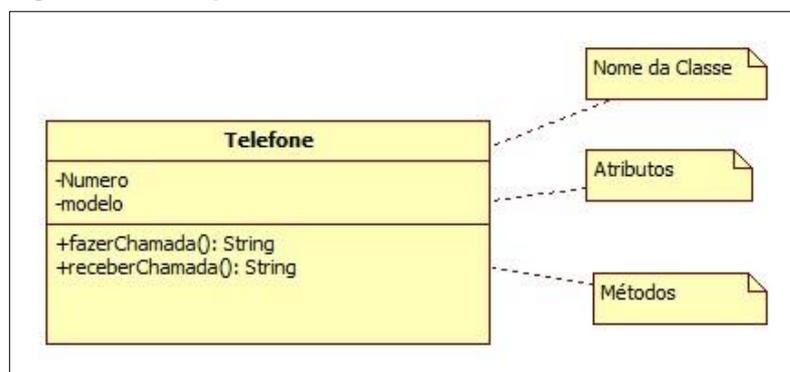
Fonte: Traduzido de BORGES e MOTA (2007).

2.2.2.1 Classificadores

Um classificador é uma meta-classe que qualifica todas as construções que ordenam valores. Seu propósito é introduzir entidades nos diagramas (BORGES; MOTA, 2007). Dentre estes classificadores pode-se destacar as classes e os tipos de dados.

A Figura 4 ilustra um exemplo de uma classe chamada Telefone, que tem alguns atributos, como por exemplo, número e modelo, ambos do tipo String, e métodos, tais como: fazerChamada() e receberChamada(), definidos pelos seus tipos específicos de argumentos e valores de retorno.

Figura 4 – Exemplo de uma Classe



Fonte: Traduzido de BERARDI, CALVANESE, e De GIACOMO (2005).

2.2.2.2 Relacionamentos

Relacionamentos denotam conexões semânticas entre os elementos do modelo (BORGES; MOTA, 2007). Adicionalmente, pode-se dizer que os diagramas UML fornecem maneiras de criar e exibir conexões entre uma ou mais classes. As conexões em UML podem ser expressas de diferentes maneiras, mas pode-se destacar as duas principais que são de associação e generalização (BORGES; MOTA, 2007).

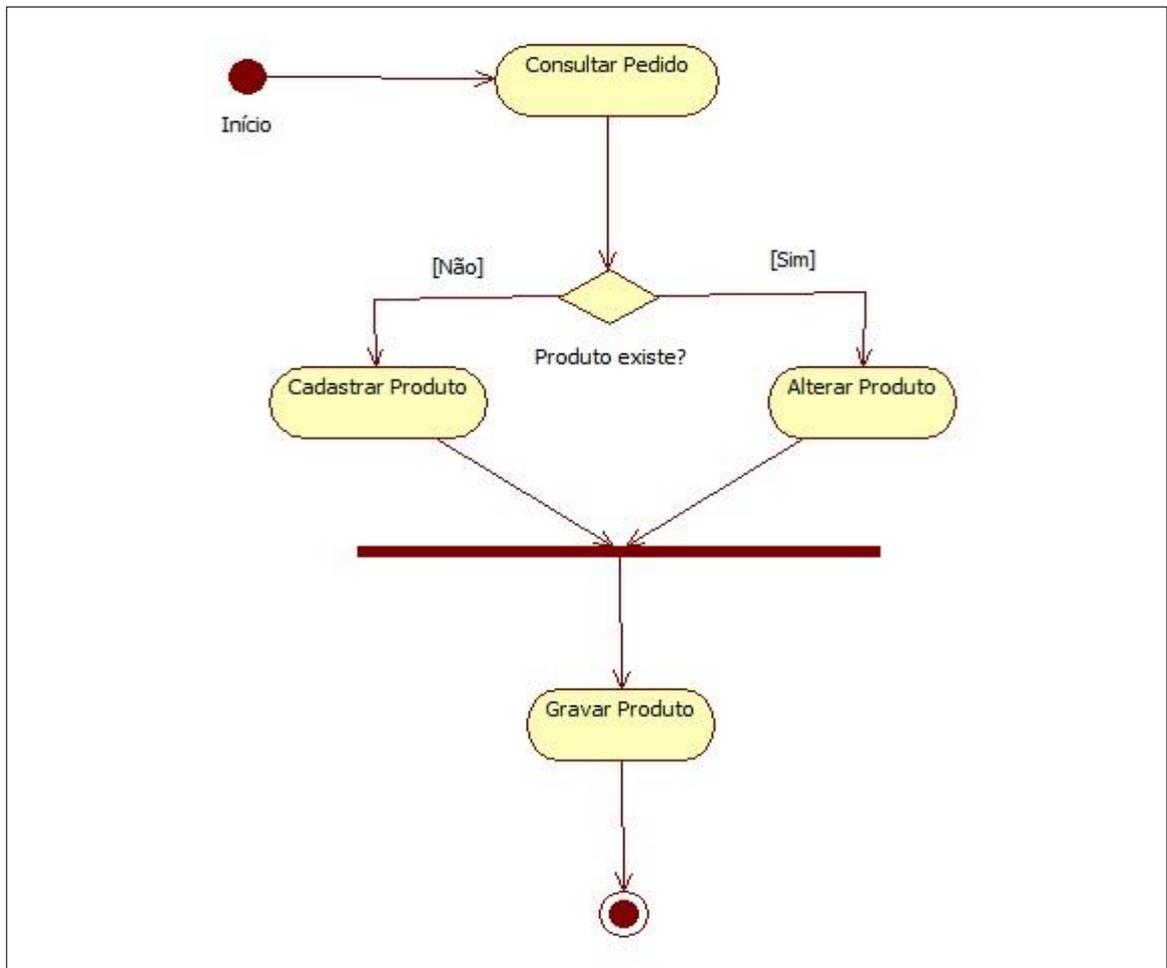
2.2.3 Diagrama de Atividade

Segundo Booch, Rumbaugh e Jacobson (2006), um diagrama de atividade é essencialmente um gráfico de fluxo, mostrando o fluxo de controle de uma atividade para outra. O diagrama de atividades é adequado para descrever interações entre os objetos do sistema (KIM, 2007). Resumidamente, pode-se dizer que o diagrama de atividades exibe um fluxo sequencial de como o sistema funciona.

A Figura 5 ilustra um diagrama de atividades no qual são exibidas as atividades disponíveis neste sistema que controla produtos. Neste sistema, uma das atividades possíveis é a consulta de um produto, que está ilustrada por “Consultar Produto”, é possível cadastrar um produto e alterar um produto que já esteja cadastrado, ambas atividades representadas respectivamente por “Cadastrar Produto” e “Alterar Produto”.

O diagrama da Figura 5 exibe todo o funcionamento do sistema que inicia com a consulta de um produto. Caso este exista, é possível fazer alterações, caso não exista é possível cadastrá-lo no sistema, ao final as alterações são gravadas no sistema e o diagrama termina.

Figura 5 – Exemplo diagrama de Atividade Simples



Fonte: Traduzido de MOBILITEC.com.br (2012).

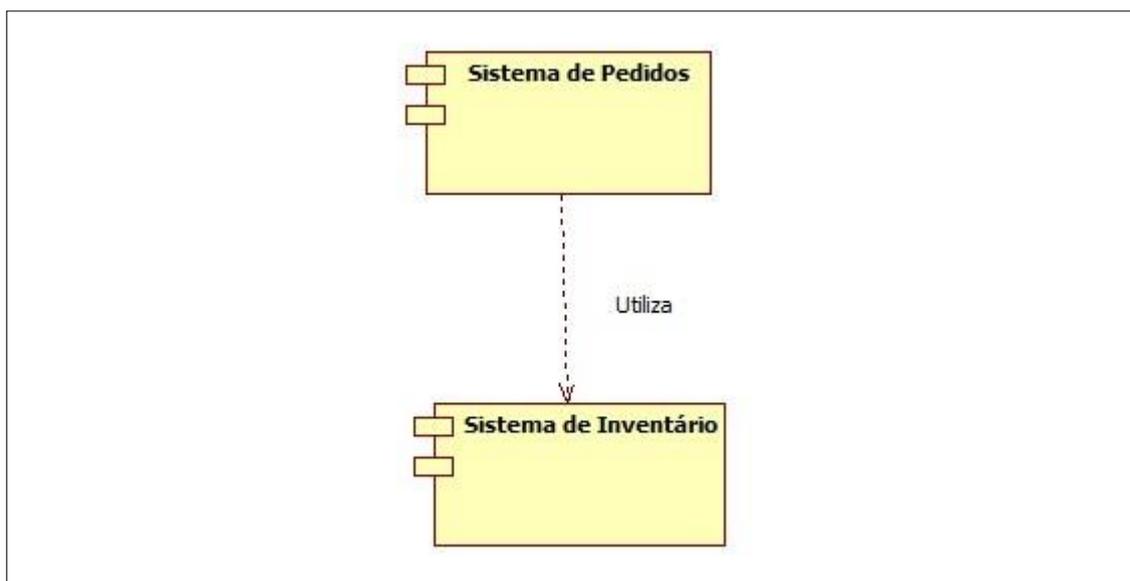
2.2.4 Diagrama de Componente

De acordo com Bell (2004), o principal propósito do diagrama de componente é mostrar os relacionamentos estruturais entre os componentes de um sistema. Os componentes são empregados para a modelagem de componentes físicos que podem residir em um nó, como executáveis, bibliotecas, tabelas, arquivos e documentos (BOOCH; RUMBAUGH; JACOBSON, 2006).

Segundo Booch, Rumbaugh e Jacobson (2006), um componente é a parte física e substituível de um sistema ao qual se adapta e fornece a realização de um conjunto de interfaces. Sendo assim, pode-se dizer que o diagrama de componentes representa a estrutura física de um sistema.

A Figura 6 ilustra um simples diagrama de componentes que mostra o relacionamento entre dois componentes, neste diagrama um sistema de inventário utiliza um sistema de pedidos.

Figura 6 – Exemplo de Diagrama de Componentes



Fonte: Traduzido de BELL (2004).

2.2.5 Diagrama de Implantação

Um diagrama de implantação mostra um conjunto de nós e seus relacionamentos (BOOCH; RUMBAUGH; JACOBSON, 2006). Sendo assim, este diagrama irá exibir o relacionamento dos nós onde o sistema será implementado.

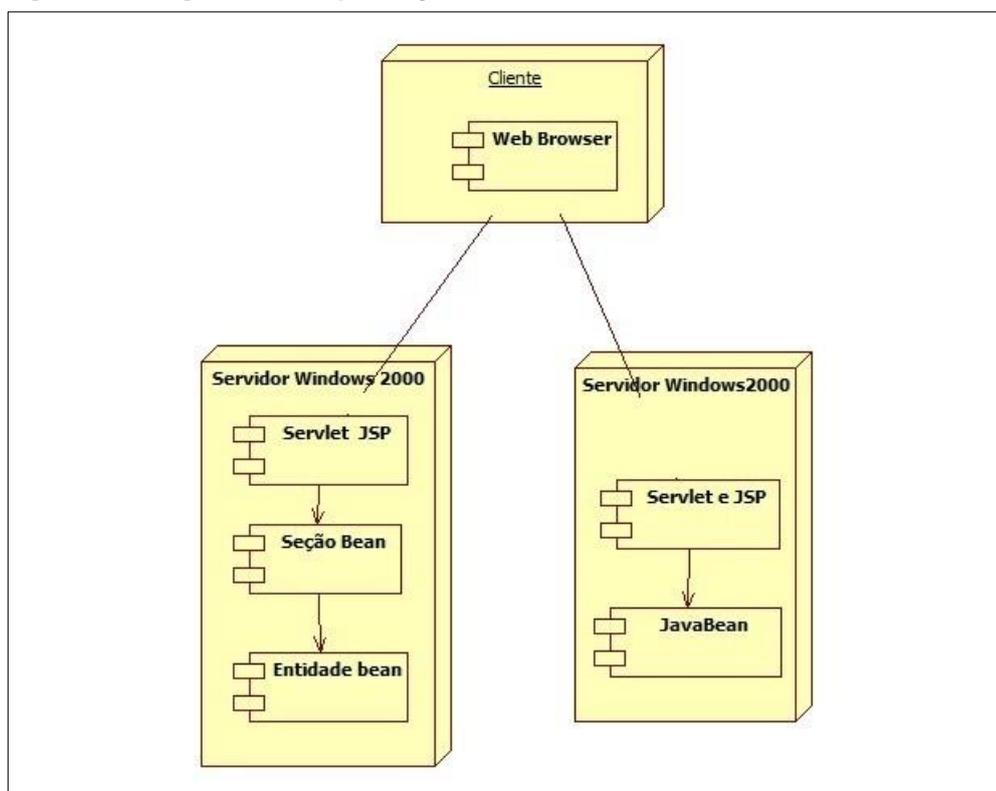
2.2.5.1 Nós

De acordo com Booch, Rumbaugh e Jacobson (2006), os nós são empregados para a modelagem da topologia do *hardware* em que o sistema é

executado. Sendo assim, um nó pode representar algum dispositivo do mundo real onde o sistema será implementado.

A Figura 7 exibe um diagrama de implantação para uma empresa chamada Virtual LTDA. Neste diagrama há 3 (três) nós, que são: Cliente e 2 (dois) Servidores, além da relação entre eles. Também é possível notar a relação dos componentes dos nós apresentados.

Figura 7 – Diagrama de Implantação da Virtual LTDA



Fonte: Traduzido de RIBEIRO (2016).

2.3 FERRAMENTAS DE DESENVOLVIMENTO

Nesta seção serão apresentadas as ferramentas de desenvolvimento utilizadas para a elaboração do presente trabalho. A subseção 2.3.1 apresenta a linguagem de programação Python. As informações sobre o *Framework* para desenvolvimento *web* Django são definidas na subseção 2.3.2. Na subseção 2.3.3 é conceituada a linguagem HTML. A linguagem CSS é descrita na subseção 2.3.4. Por fim, a seção 2.3.5 apresenta a linguagem JavaScript.

2.3.1 Linguagem Python

Segundo Sanner (1999), Python é uma linguagem de programação *open source*, interpretada, interativa e orientada a objetos que fornece variadas estruturas de dados de alto nível. Um programa em Python é compilado automaticamente por um interpretador em código de bytes, que por sua vez são independentes de plataforma.

Python é uma linguagem modular, a qual tem um *kernel*¹ pequeno e pode ser estendido a partir da importação de módulos (SANNER,1999). A distribuição do Python também inclui bibliotecas para extensões para a linguagem. Estas extensões podem ser para manipulações de *Strings* ou para geradores de interface de usuário que também podem estar relacionadas a *web* (SANNER,1999).

Há atualmente duas versões do Python disponíveis para uso, a versão 2.x e a 3.x (PYTHON.ORG, 2017). Na versão 2.x o último lançamento foi o Python 2.7 e para a versão 3.x o Python 3.6. Apesar da última versão do Python 2.x ter sido lançada em 2010, o Python 2.x ainda é a versão que vem instalada como padrão nos sistemas operacionais Linux e Mac OS (PYTHON.ORG, 2017). O Python 3.x já está completamente pronto para substituir o Python 2.x, mas devido à grande comunidade ainda ativa para o Python 2.x o suporte para esta versão foi estendida fazendo com que o Python tenha duas versões ativas para uso (PYTHON.ORG, 2017).

2.3.2 *Framework* Django e Arquitetura MVC

Django é um *Framework* para desenvolvimento *Web* baseado na linguagem Python (HOLOVATY; KAPLAN-MOSS, 2009). Sendo assim, o Django, como todo *Framework* para desenvolvimento, oferece facilidades para a criação de aplicações, neste caso, aplicações *web*.

O Django como a grande maioria dos *frameworks* para desenvolvimento *web* utiliza a arquitetura *Model-View-Controller* (MVC). MVC é um conceito que está disponível para uso há um bom tempo, mas desde o surgimento da Internet o seu uso

¹ O *kernel* é uma série de arquivos que constituem o núcleo, o centro de todas as atividades desempenhadas pelo sistema operacional. (dos SANTOS PIRES, 2007).

crece exponencialmente devido a facilidade de desenvolver aplicações cliente-servidor (HOLOVATY; KAPLAN-MOSS, 2009).

O *Model* do MVC é a representação dos dados da aplicação. A *View* é o que pode ser visto na aplicação pelo navegador e é a partir dela que é exibida uma interface para coletar as entradas do usuário. O *Controller* controla o fluxo de informação entre o *Model* e a *View*, ou seja, é a partir dele que determinadas informações são retiradas do banco de dados pelo *Model* e exibidas na *View* (HOLOVATY; KAPLAN-MOSS, 2009).

Apesar do Django seguir o padrão MVC, há algumas diferenças na forma que esta arquitetura foi implementada neste *framework*. No Django foi implementado uma arquitetura que pode ser referida como *Model-Template-View* (MTV), onde o *Model* é a camada de acesso aos dados, o *Template* é a camada de apresentação, ou seja, como os dados serão apresentados na página *web* e a *View* é a camada que contém a lógica que faz a relação do *Model* com o *Template* (HOLOVATY; KAPLAN-MOSS, 2009).

2.3.3 HTML

HTML, que significa Linguagem de Marcação Hipertexto, é o elemento mais básico de uma página *web* e é usada para criar e representar visualmente uma página *web* (W3C, 2016b). De acordo com Marcondes (2000), os comandos de HTML são chamados de *tags*, que significa etiquetas em inglês.

A partir do uso de *tags*, é possível a criação de documentos na linguagem HTML. As *tags* informam ao navegador o que são os elementos que estão na página (MARCONDES, 2000).

O HTML nasceu para unificar elementos fragmentados que antes eram distribuídos pelos seus autores, como por exemplo, fotos, sons e textos (KENNEDY; MUSCIANO, 2006). Além disto, o HTML também permitiu que documentos e páginas fossem automaticamente referenciados a outros documentos localizados em outros lugares através de *links* (KENNEDY; MUSCIANO, 2006).

2.3.4 CSS

CSS (*Cascading Style Sheets*) é uma linguagem *stylesheet* que descreve a apresentação de um documento HTML ou XML (MOZILLA, 2016). Com o CSS é possível controlar fontes, margens, cores e até a posição de elementos na tela, ou seja, o CSS apresenta um documento com uma aparência melhor do que o HTML (MEYER, 2006). De acordo com W3C (2016a), o CSS permite a apresentação do conteúdo de páginas *web* em diferentes tipos de dispositivos, podendo estes serem de telas grandes, telas pequenas e ou serem impressoras.

2.3.5 JavaScript

JavaScript é uma linguagem de programação leve, interpretada e com funções (MOZILLA, 2016). Seja no computador, nos *smartphones* ou *tablets*, todos os navegadores são capazes de exibir aplicações criadas nesta linguagem, conseqüentemente, segundo Flanagan (2011), o JavaScript é a linguagem de programação mais ubíqua da história.

Além disso, JavaScript é amplamente utilizada para aumentar a interatividade das páginas da *Web* (NEGRINO; SMITH, 2001), como por exemplo, certificar de que os usuários introduzam informações válidas nos formulários (NEGRINO; SMITH, 2001). Pelo fato de ser uma linguagem de script é considerada robusta e eficiente (FLANAGAN, 2011).

2.4 SISTEMA DE GERENCIAMENTO DE BANCO DE DADOS (SGBD)

Um sistema de Gerenciamento de Banco de Dados (SGBD) é um sistema responsável pelo gerenciamento permanente e auto descritivo de um depósito de dados (RUMBAUGH; et al., 1994). Sendo que, estes depósitos são popularmente chamados de banco de dados.

Como definido por Rumbaugh (1994), existem muitas razões pelas quais os desenvolvedores utilizam os SGBDs. São elas:

- Recuperação de desastres: O banco de dados fica protegido de falhas;

- Compartilhamento entre usuários: Banco de dados compartilhado entre usuários com acessos simultâneos;
- Compartilhamento entre aplicações: Banco de dados compartilhado por aplicações simultaneamente;
- Segurança: Dados protegidos contra acessos indesejados;
- Integridade: A partir de regras definidas um SGBD pode controlar seus dados e o que é oferecido a aplicações;
- Extensibilidade: Dados são adicionados e reorganizados no banco de dados sem afetar outros programas que o utilizam;
- Distribuição de dados: O banco de dados pode ser subdividido em várias partes, em diferentes instalações e plataformas;

2.4.1 Banco De Dados Relacional

De acordo com Rumbaugh (1994), um SGBD Relacional (SGBDR) é um programa de computador para gerenciamento de tabelas. Sendo que estas são formadas por dados que são exibidos em forma de linhas e colunas. As colunas das tabelas são denominadas atributos e as linhas são denominadas tuplas e correspondem às instâncias dos objetos e ligações.

2.4.2 Linguagem de Consulta Estruturada – SQL

A Linguagem de Consulta Estruturada (SQL, em inglês *Structured Query Language*) se tornou a linguagem mais popular para aplicações em SGBDR e é um padrão para o ANSI (*American National Standards Institute*) e a ISO (*International Standardization Organization*) (RUMBAUGH; et al., 1994). Segundo Melton (1996), SQL é uma sub-linguagem de dados de banco de dados relacional.

O conceito fundamental da SQL é a tabela, com uma ou mais colunas contendo um determinado tipo de dado (MELTON, 1996). Estes tipos de dados nas tabelas podem ser numéricos, caracteres, de tempo, mídia, datas, entre outros.

3 DESENVOLVIMENTO

Neste capítulo, são apresentados os passos realizados no desenvolvimento de um sistema *web* que irá auxiliar os alunos a encontrarem informações para a realização do TCC.

O desenvolvimento do sistema foi feito utilizando a metodologia de gerenciamento *Kanban* (IKONEN et al., 2011) em conjunto com a metodologia de desenvolvimento ágil *eXtreme Programming XP* (BECK, 2009). Para a criação do quadro de *Kanban* (IKONEN et al., 2011) foi utilizada uma aplicação *Web* chamada *KanbanFlow* (KANBAN FLOW, 2017), na qual é possível criar e gerenciar as tarefas do projeto.

No desenvolvimento desta aplicação *web* também foram utilizadas outras ferramentas para auxiliar no desenvolvimento, como por exemplo:

- a) BRModelo (brModelo 2.0, 2007): para a criação do modelo do banco de dados;
- b) StarUML (StarUML 2.8.0, 2016): para a criação dos modelos da Linguagem Unificada de Modelagem (UML);
- c) Python 2.7.6 (SANNER, 1999): linguagem de programação;
- d) *Framework Django* (HOLOVATY; KAPLAN-MOSS, 2009): para a criação da aplicação *Web* versão 1.10;
- e) SQLite3 (HOLOVATY; KAPLAN-MOSS, 2009): base de dados usada com o *framework* para a construção da aplicação;
- f) Visual Studio Code (MICROSOFT, 2017a): editor de texto e IDE responsável pela criação da aplicação;
- g) Windows Bash (MICROSOFT, 2017b): terminal utilizado para rodar o servidor e o *framework Django* (HOLOVATY; KAPLAN-MOSS, 2009) no sistema operacional Windows 10 com o Ubuntu versão 14.04.5 LTS;
- h) *Virtual Environment* (PYTHON.ORG, 2017): responsável pela criação de um ambiente virtual para a aplicação;
- i) Gravatar (AUTOMATTIC, 2017): utilizado para exibição de fotos no perfil;

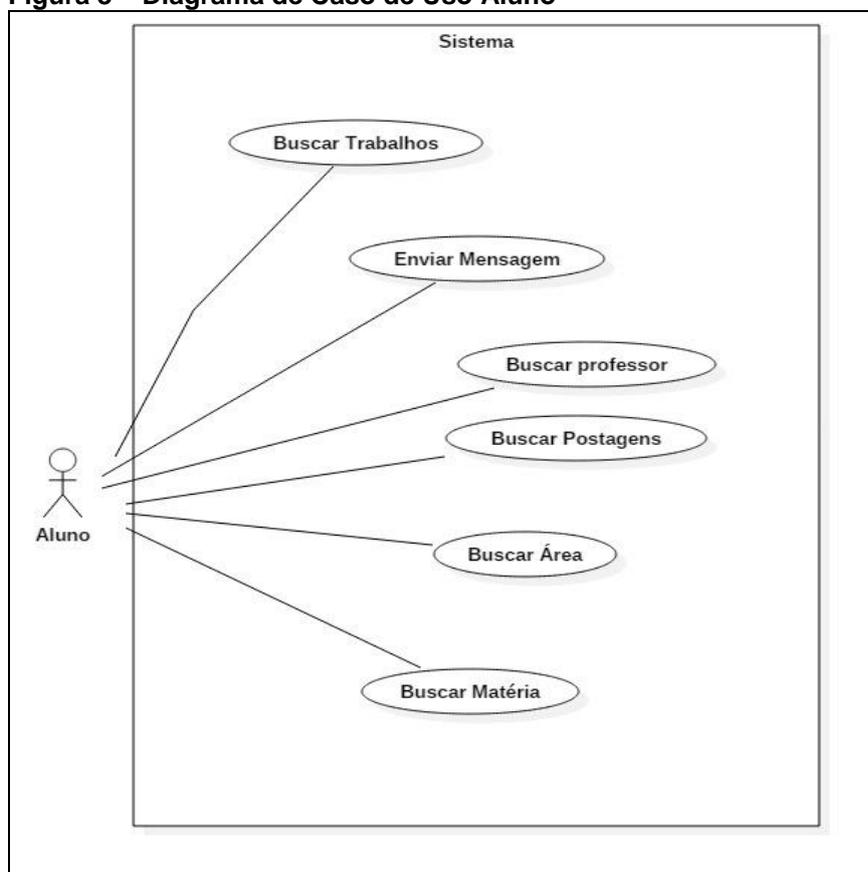
3.1 MODELAGEM

Nesta seção será abordada a modelagem da aplicação desenvolvida. Na subseção 3.1.1 é apresentado o diagrama de caso de uso. A subseção 3.1.2 descreve os diagramas de atividades. A subseção 3.1.3 discorre sobre o diagrama de classe. A subseção 3.1.4 aborda o diagrama de componentes. E, por fim, a subseção 3.1.5 apresenta o diagrama de implantação.

3.1.1 Diagrama de Caso de Uso

A Figura 8 apresenta o diagrama de caso de uso do ator Aluno. Neste, o ator “Aluno” pode realiza as seguintes tarefas: Buscar trabalhos, Enviar Mensagem, Buscar Professor, Buscar Postagens, Buscar Área e Buscar Matéria.

Figura 8 – Diagrama de Caso de Uso Aluno

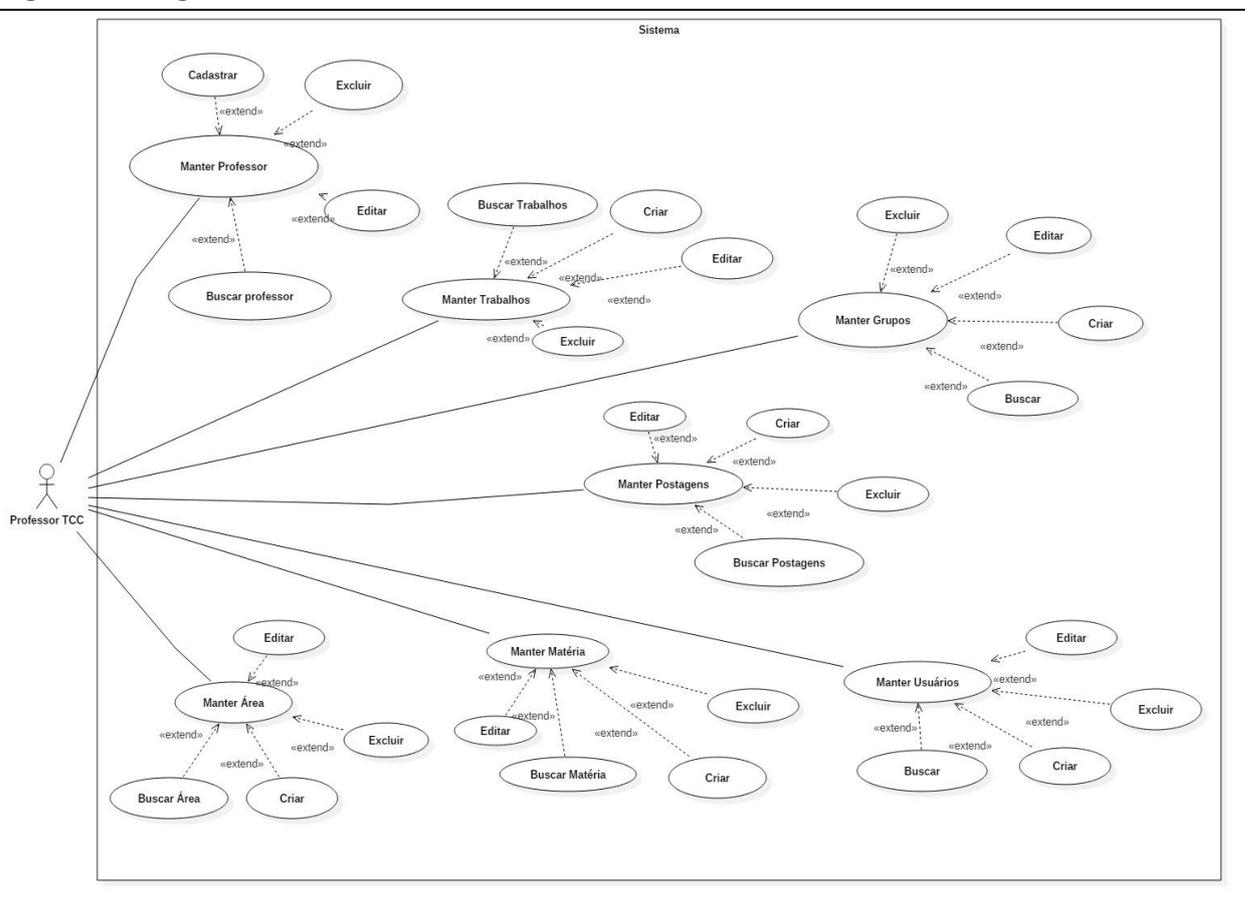


Fonte: Autoria Própria.

O ator “Aluno” é um usuário do sistema, porém este não precisará fazer a autenticação, ou seja, não terá um cadastro no sistema. Como funções disponíveis ao Aluno, o mesmo apenas poderá consultar informações e enviar mensagens.

A Figura 9 apresenta o diagrama de caso de uso do ator Professor TCC. O “Professor TCC” tem função administrativa no sistema, ou seja, ele tem acesso a criação, edição, exclusão e busca de Trabalhos, Áreas, Matérias, Postagens, Professores, Usuários e Grupos. Conseqüentemente, ele se relaciona aos Casos de Usos de: Manter Trabalhos, Manter Área, Manter Matéria, Manter Postagens, Manter Professor, Manter Usuários, Manter Grupos, além de poder Enviar Avisos.

Figura 9 – Diagrama de Caso de Uso Professor TCC

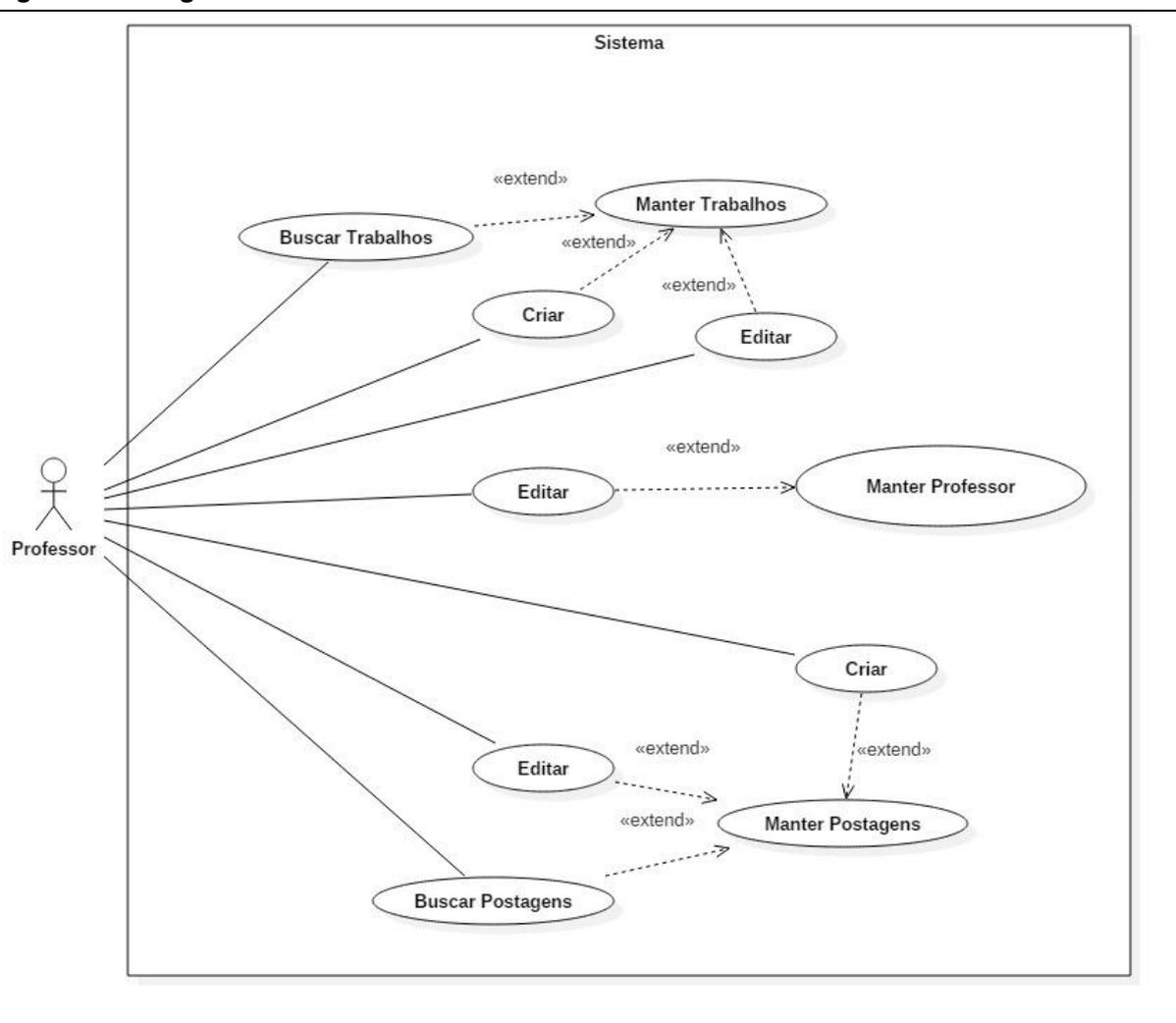


Fonte: Autoria Própria.

A Figura 10 apresenta o diagrama de caso de uso do ator Professor. O ator “Professor” é um usuário do sistema que pode criar e editar trabalhos e postagens e também pode editar informações de seu próprio perfil, sendo assim, ele se relaciona com os casos de usos ligados a estas funções. Adicionalmente, o Professor também

se relaciona com Exibir Trabalhos e Exibir Postagens que são funções em que o mesmo pode obter acesso as suas postagens além dos trabalhos que criar.

Figura 10 – Diagrama de Caso de Uso Professor



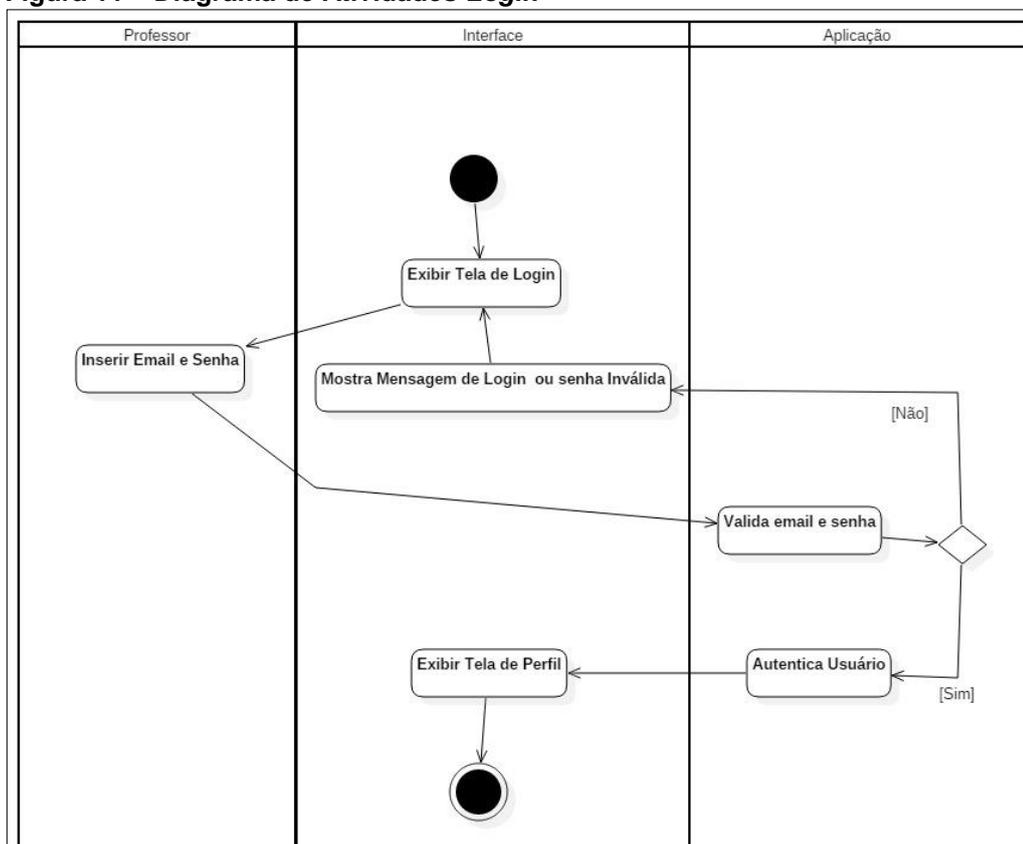
Fonte: Autoria Própria.

3.1.2 Diagrama de Atividades

Nesta subseção, serão apresentados os diagramas de atividades da aplicação *Web*. A Figura 11 ilustra o diagrama de atividades para a autenticação do Professor na aplicação de Perfil. Inicialmente o sistema exibe a tela de *login* para o usuário, o qual precisará entrar com o seu endereço de e-mail e senha. Após a inserção dos dados, o sistema faz a validação dos mesmos. Caso a validação falhe, o sistema exibe uma mensagem de erro de *Login* ou senha inválida e novamente será mostrada a tela de *login* para uma nova tentativa. Caso o usuário realize o *login*

corretamente, a tela de perfil será exibida com as respectivas permissões para o usuário e o fluxo desta atividade termina.

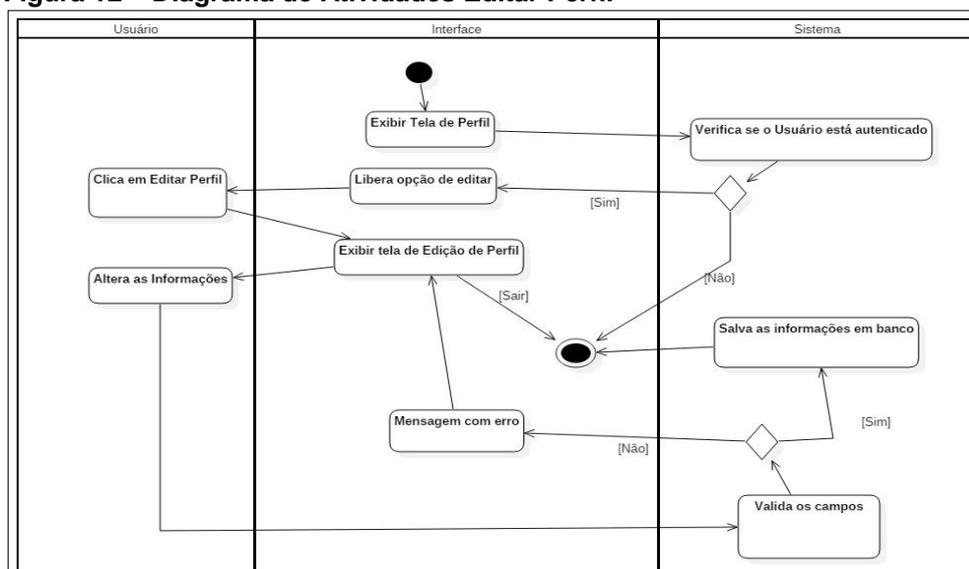
Figura 11 – Diagrama de Atividades *Login*



Fonte: Autoria Própria.

O diagrama de atividade para a edição das informações no perfil de um professor é ilustrado na Figura 12. Inicialmente é exibida a tela de perfil do professor e caso o mesmo esteja autenticado será exibida a opção de editar perfil. Caso o professor decida editar o seu perfil, ao clicar no botão para edição ele será redirecionado a uma nova página onde estarão as suas informações nos campos de um formulário para que o mesmo possa fazer as edições e salvá-las.

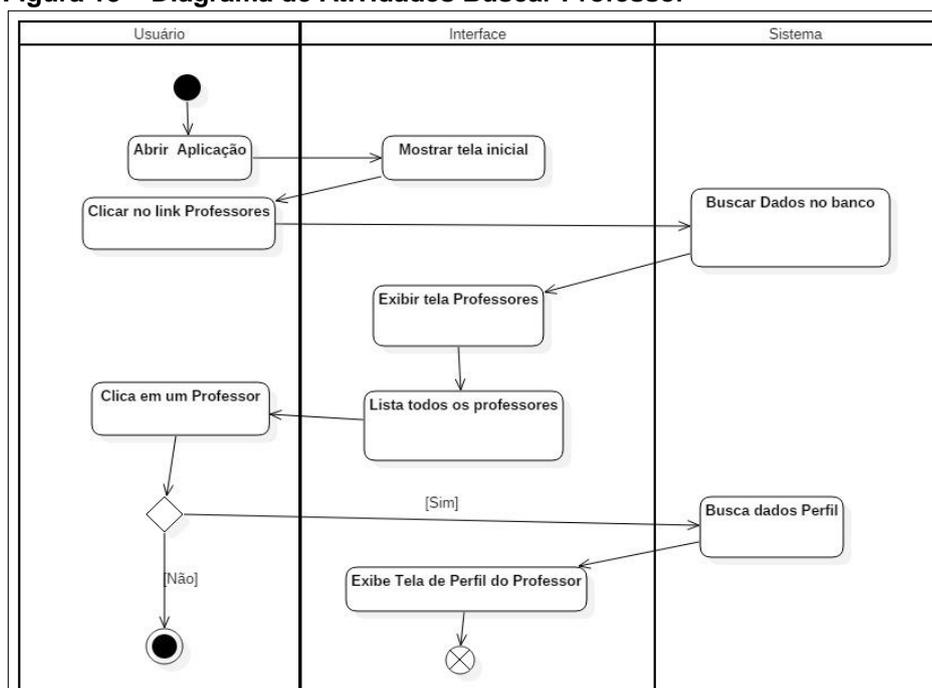
Figura 12 – Diagrama de Atividades Editar Perfil



Fonte: Autoria Própria.

O diagrama de atividade representado na Figura 13 refere-se a busca de informações de um professor. Esta inicia-se com o usuário abrindo a aplicação, após esta etapa, é exibida a tela inicial. Ao clicar no *link* de professores o sistema apresenta ao usuário os dados de todos os professores cadastrados, possibilitando ao aluno escolher um dos professores para obter mais detalhes.

Figura 13 – Diagrama de Atividades Buscar Professor

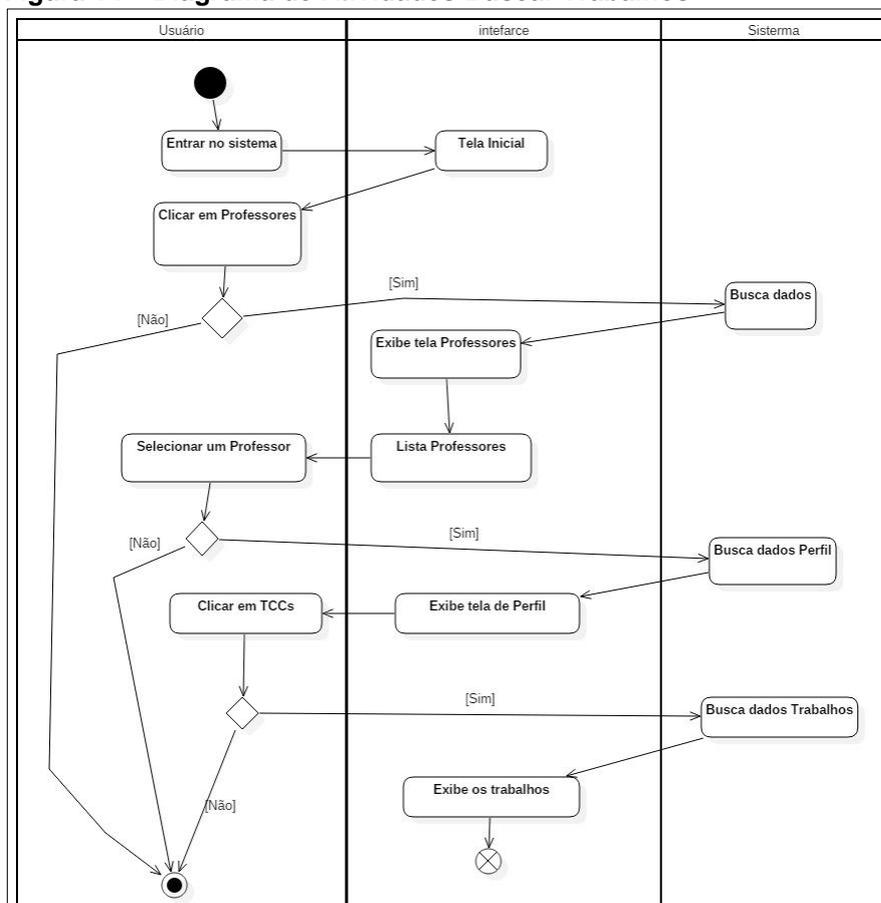


Fonte: Autoria Própria.

A Figura 14 apresenta o diagrama de atividades de Buscar Trabalhos. O fluxo inicia-se com o usuário acessando o sistema. Ao clicar em Professores, o sistema fará uma busca no banco de dados para apresentar as informações referentes aos professores cadastrados. Caso o aluno escolha um dos professores listados, o sistema buscará os dados do perfil deste professor específico.

A página do professor contém informações profissionais de cada docente, bem como suas publicações e TCC's orientados.

Figura 14 – Diagrama de Atividades Buscar Trabalhos



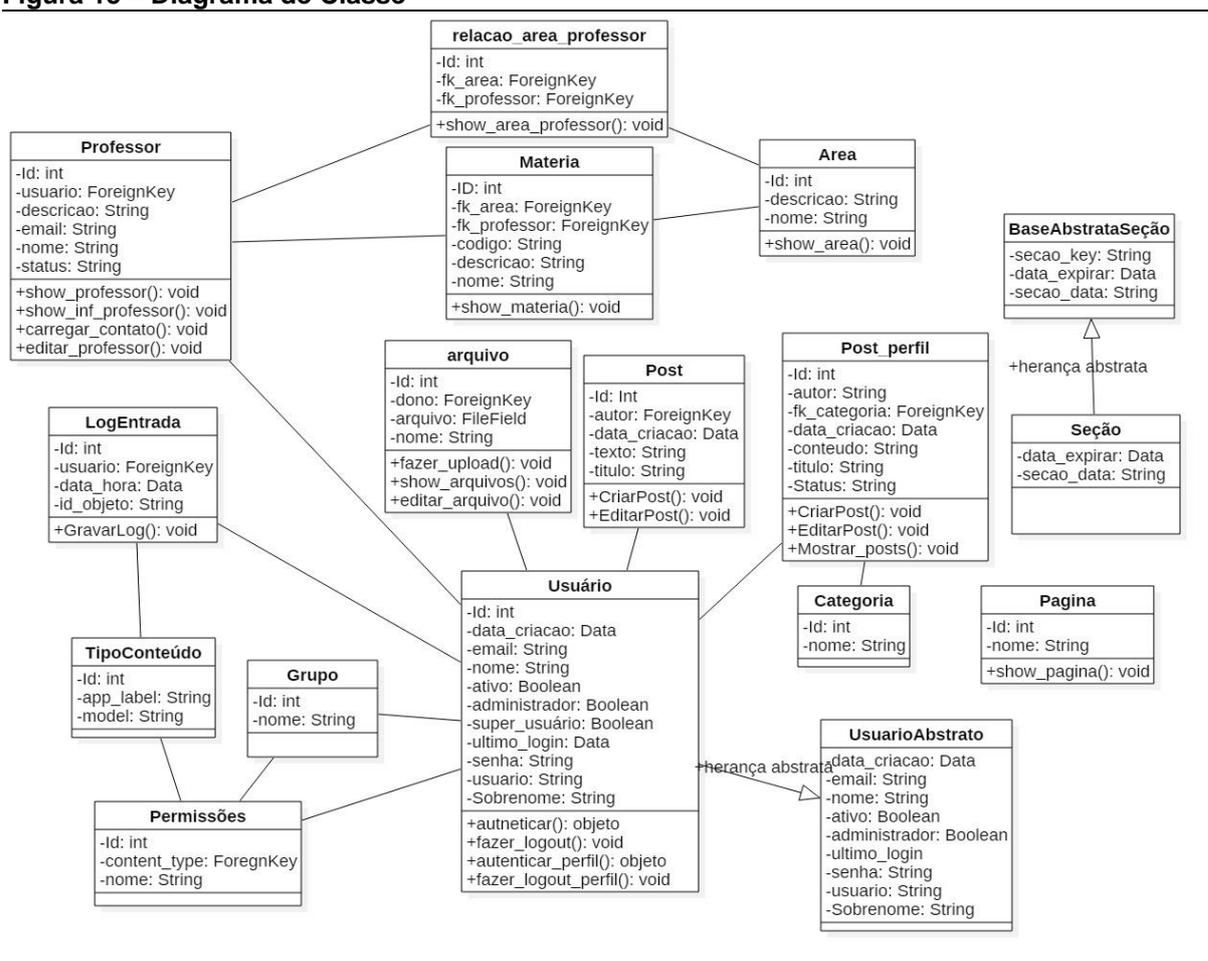
Fonte: Autoria Própria.

3.1.3 Diagrama de Classe

O diagrama de classes contém todas as informações das classes do sistema *web*. Adicionalmente, também estão neste diagrama algumas classes que estavam implementadas no *framework* Django (IKONEN et al., 2011) e que foram utilizadas na aplicação, sendo elas as classes: Usuário, Grupo, Permissões, TipoConteúdo,

LogEntrada, UsuárioAbstrato, Seção e BaseAbstrataSeção. Na classe Usuário, mesmo a classe já estando implementada no *framework*, os métodos exibidos no diagrama foram codificados durante a fase de desenvolvimento da aplicação. O diagrama de classe é ilustrado na Figura 15.

Figura 15 – Diagrama de Classe



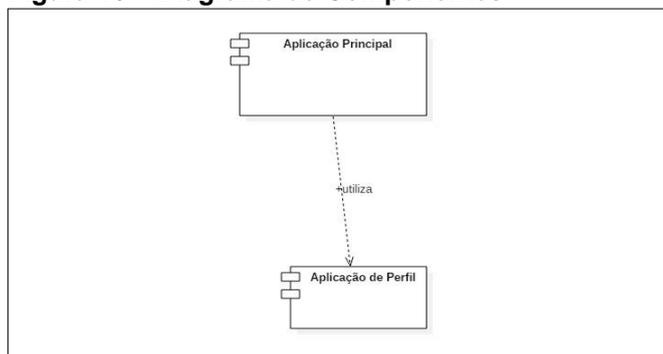
Fonte: Autoria Própria.

3.1.4 Diagrama de Componente

O sistema é composto por duas aplicações, são elas: Aplicação Principal e Aplicação de Perfil. A primeira é responsável por fazer a listagem de professores, áreas e matérias disponíveis. A segunda tem a função de controlar um perfil para um usuário Professor, ou seja, através desta os professores poderão criar postagens e fazer *upload* de arquivos.

Contudo, para que o sistema funcione corretamente, há uma relação de dependência por parte da aplicação principal com a aplicação de perfil. A Figura 16 ilustra o diagrama de Componentes da aplicação *Web*.

Figura 16 – Diagrama de Componentes

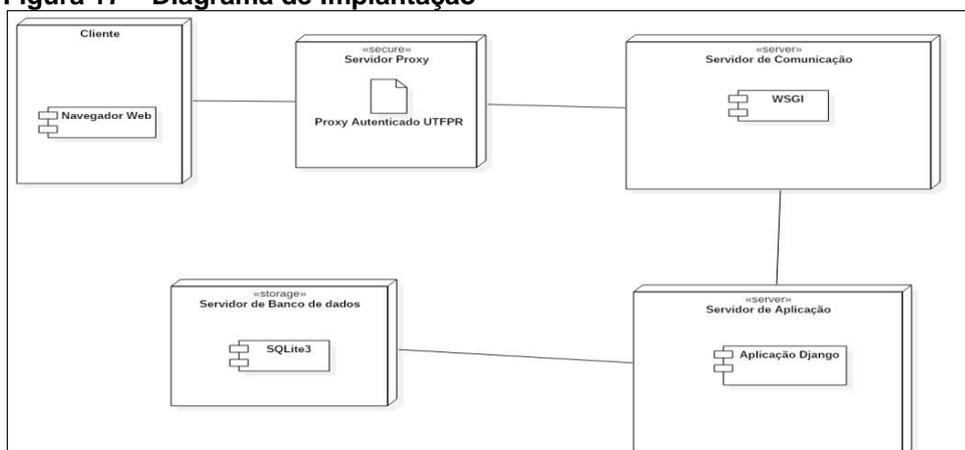


Fonte: Autoria Própria.

3.1.5 Diagrama de Implantação

A Figura 17 apresenta o diagrama de implantação da aplicação *Web* desenvolvida. Esta é hospedada em um servidor com o auxílio do WSGI² (*Web Server Gateway Interface*) e as aplicações disponibilizadas por ele podem ser acessadas remotamente por um cliente via *Navegador Web*.

Figura 17 – Diagrama de Implantação



Fonte: Autoria Própria.

² WSGI é uma interface para comunicação com o servidor web, que descreve como o servidor se comunica com as aplicações web. WSGI é um padrão utilizado pela linguagem de programação Python (WSGI.ORG).

3.2 QUADRO DE KANBAN

Nesta seção são apresentadas as etapas de desenvolvimento utilizando o quadro de *Kanban* (IKONEN et al., 2011) para a gestão do desenvolvimento da aplicação. Como o quadro de *Kanban* (IKONEN et al., 2011) tem um dos princípios de limitar a quantidade de atividades em desenvolvimento, para este projeto foi limitado o número de atividades simultaneamente em andamento para três. Consequentemente, em cada etapa de desenvolvimento serão apresentadas três atividades desenvolvidas.

Na Figura 18 é possível ver algumas das atividades listadas no quadro de *Kanban* (IKONEN et al., 2011) em sua etapa inicial, onde as tarefas estão na coluna “A fazer”.

Figura 18 – Quadro de *Kanban* Fase Inicial

Á fazer	Em andamento 0 / 3	Feito
<div style="background-color: #e0ffe0; padding: 5px; margin-bottom: 5px;">Instalar o ambiente</div> <div style="background-color: #e0e0ff; padding: 5px; margin-bottom: 5px;">Criar o banco de dados da aplicação principal</div> <div style="background-color: #e0e0ff; padding: 5px; margin-bottom: 5px;">Criar as primeiras views (Controllers)</div> <div style="background-color: #e0e0ff; padding: 5px; margin-bottom: 5px;">Criar os primeiros templates</div> <div style="background-color: #ffe0e0; padding: 5px; margin-bottom: 5px;">Configurar as URLs para as views criadas</div> <div style="background-color: #ffe0e0; padding: 5px;">Perfil dos professores</div>		

Fonte: Autoria Própria.

A subseção 3.2.1 exibe a primeira etapa de desenvolvimento. A segunda etapa é apresentada na subseção 3.2.2, e assim sucessivamente até a última etapa que será descrita na subseção 3.2.7.

3.2.1 Primeira Etapa de Desenvolvimento

Na primeira fase do desenvolvimento, criou-se algumas das atividades da aplicação: Criar o banco de dados da aplicação principal, criar as primeiras *views* da

aplicação no *Framework* Django (HOLOVATY; KAPLAN-MOSS, 2009) e criar os primeiros *templates*. A Figura 19 exhibe o Quadro de *Kanban* (IKONEN et al., 2011) desta primeira fase de desenvolvimento.

Figura 19 – Quadro de *Kanban* Primeira Etapa

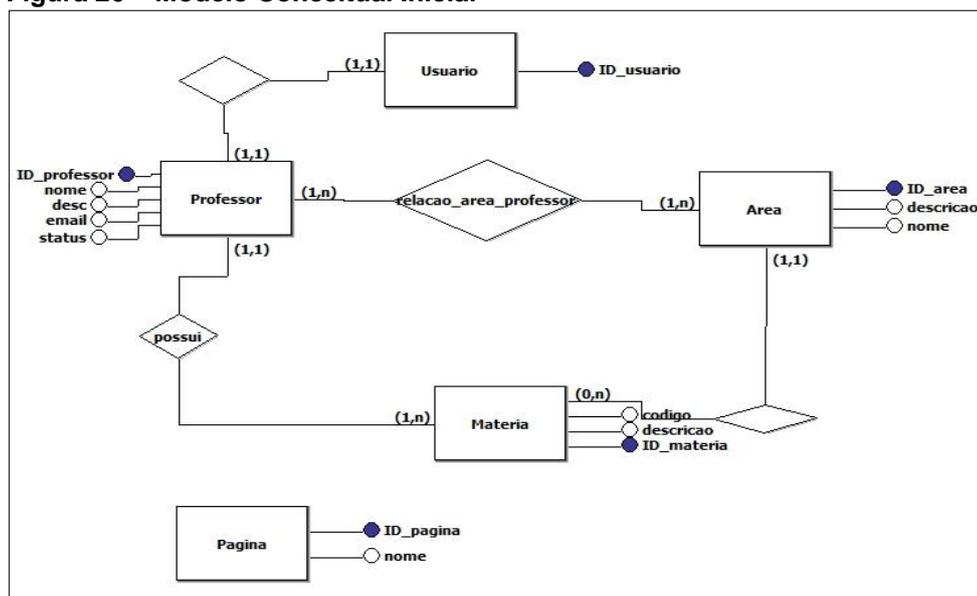


Fonte: Autoria Própria.

A primeira atividade a ser concluída foi a de instalar o ambiente, ou seja, a instalação dos componentes necessários para que o ambiente de desenvolvimento estivesse preparado. Sendo que, nesta primeira atividade, realizou-se a instalação do interpretador da linguagem Python, do *Framework* de desenvolvimento *web* Django (HOLOVATY; KAPLAN-MOSS, 2009) e a instalação do Virtual Environment (PYTHON.ORG, 2017).

Dentre as atividades com o status de “Em andamento”, a primeira a ser desenvolvida foi a criação do banco de dados inicial, no qual identificou-se as primeiras tabelas a serem inseridas: Professor, Area, Materia, relacao_area_professor e Pagina. A Figura 20 exhibe o modelo conceitual do banco de dados modelado na ferramenta BrModelo (BRMODELO 2.0, 2007).

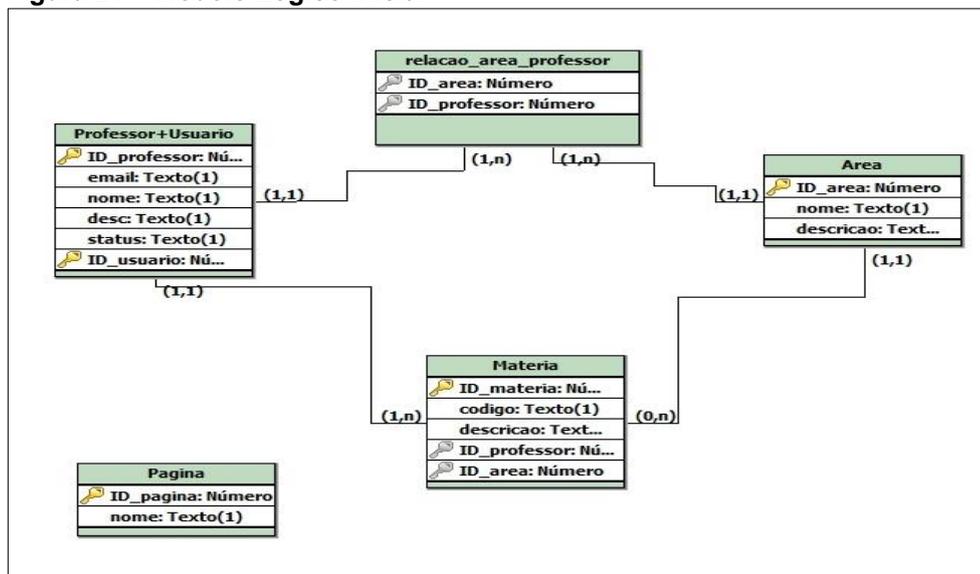
Figura 20 – Modelo Conceitual Inicial



Fonte: Autoria Própria.

A partir da criação do modelo conceitual gerou-se o modelo lógico para a criação das tabelas do banco de dados, para esta atividade utilizou-se a ferramenta BrModelo (BRMODELO 2.0, 2007). O modelo lógico é ilustrado na Figura 21.

Figura 21 – Modelo Lógico Inicial



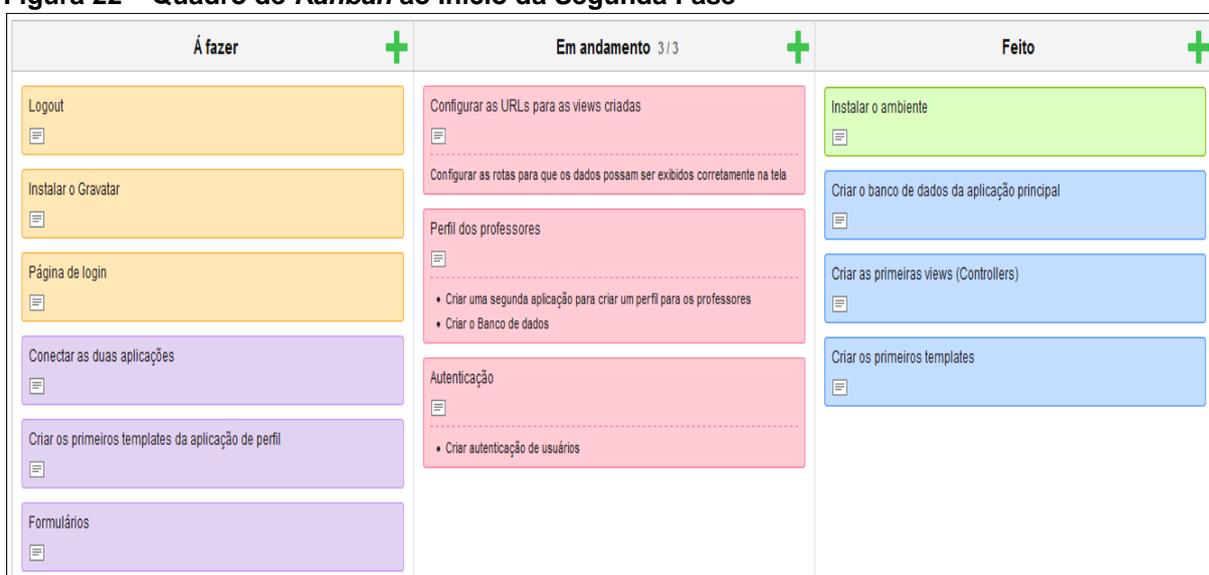
Fonte: Autoria Própria.

A partir dos modelos apresentados acima, criou-se as tabelas no banco de dados através do *Model* no *Framework Django* (HOLOVATY; KAPLAN-MOSS, 2009), no qual, cada *Model* a ser criado é uma classe da linguagem Python.

3.2.2 Segunda Etapa de Desenvolvimento

Nesta etapa desenvolveu-se as seguintes atividades: Configurar as URLs no *template* para ligar as *views (controllers)* com os *templates*, criou-se uma segunda aplicação que será utilizada para a criação do perfil dos professores e também desenvolveu-se a autenticação de usuários. A Figura 22 ilustra as atividades no quadro de *Kanban* no início desta segunda etapa de produção da aplicação *web*.

Figura 22 – Quadro de *Kanban* ao Início da Segunda Fase



Fonte: Autoria Própria.

Dentre estas atividades, a primeira a ser finalizada foi a de Configurar as URLs para conectar as *views* do *framework* com os *templates*. O *Django* (HOLOVATY; KAPLAN-MOSS, 2009) possui um arquivo no qual podem ser configuradas as URLs para criar as rotas da aplicação. Este arquivo sempre precisará ser alterado quando um novo *template* e uma *view* forem inseridas na aplicação.

A Figura 23 ilustra o arquivo de URLs da aplicação, neste há um vetor de URLs em que estas se utilizam de uma função chamada “url” para fazer a criação das rotas. A função “url” encontra-se pré-implementada no *framework* e admite três parâmetros: o primeiro é uma expressão regular para validar uma URL; o segundo é a *view* que processará os dados para o *template*; e o terceiro é o nome desta url para que ela possa ser utilizada no *template*.

Figura 23 – Configuração de URLs

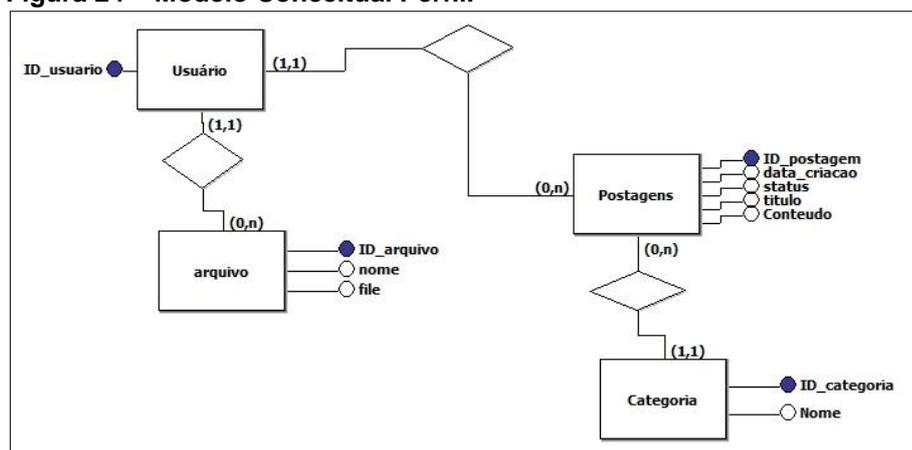
```
urlpatterns = [
    url(r'^$', views.index, name='index'),
    url(r'^auth', views.auth, name='blog.auth'),
    url(r'^logout$', views.logout_view, name='blog.logout'),
    url(r'^blog/news$', views.post_create, name='blog.post_create'),
    url(r'^blog/areaProf$', views.areaProf, name='blog.areaProf'),
    url(r'^pagina/(?P<pagina_id>\d+)', views.show_paginas, name='blog.paginas_id'),
    url(r'^perfil/(?P<professor_id>\d+)', perfil.show_professor, name='blog.professor_id'),
    url(r'^materia/(?P<materia_id>\d+)', views.show_materia, name='blog.materia_id'),
    url(r'^area/(?P<area_id>\d+)', views.show_area, name='blog.area_id'),
    url(r'^blog/news/edit(?P<post_id>\d+)', views.show_edit, name='blog.post_edit'),
]
```

Fonte: Autoria Própria.

Outra atividade desenvolvida foi a de criação de uma aplicação para gerenciar o perfil dos professores. Esta aplicação estará dentro da aplicação *web*, porém trabalhará como uma aplicação separada. Sendo assim, para esta nova aplicação também foi criado o banco de dados além de suas *views*.

A Figura 24 exibe o modelo conceitual criado para modelar o banco de dados desta aplicação. Algumas tabelas utilizam uma chave estrangeira que é o *ID_usuario*, que também foi utilizada na aplicação principal.

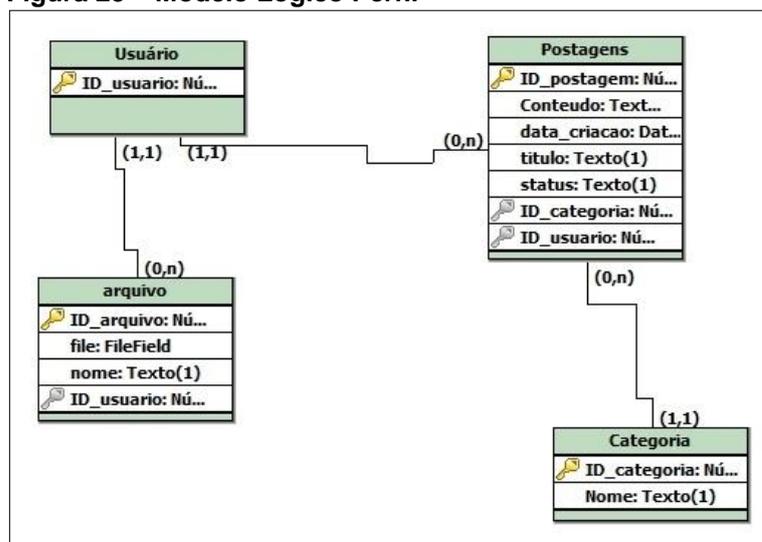
Figura 24 – Modelo Conceitual Perfil.



Fonte: Autoria Própria

Após a criação do modelo conceitual, gerou-se o modelo lógico para o banco de dados da aplicação de perfil, ilustrado na Figura 25.

Figura 25 – Modelo Lógico Perfil



Fonte: Autoria Própria.

Após a criação do banco de dados e *views* para a aplicação de perfil, implementou-se a autenticação dos usuários, para que os mesmos pudessem interagir com o sistema.

A Figura 26 exibe um exemplo da *view* responsável pela autenticação do usuário no sistema. Esta recebe como parâmetro um *request* do *template* e verifica se o método do *form* criado é um POST. Caso seja, realiza-se a autenticação do usuário a partir dos dados – usuário e senha – fornecidos nos campos do *form* no *template* a partir do uso da função *authenticate* do *Framework* Django (HOLOVATY; KAPLAN-MOSS, 2009).

Figura 26 – Exemplo autenticação

```
def auth(request):
    error = False
    paginas = Pagina.objects.all()
    if request.method == 'POST':
        username = request.POST.get('username')
        password = request.POST.get('password')
        user = authenticate(username=username, password=password)
        if user is not None:
            error = False
            login(request, user)
        else:
            error = True

    context = {
        'paginas': paginas,
        'error': error
    }
    return render(request, 'blog/auth.html', context)
```

Fonte: Autoria Própria.

3.2.3 Terceira Etapa de Desenvolvimento

Desenvolveu-se as seguintes atividades na terceira etapa de desenvolvimento: Configuração do ambiente administrativo, criação do *logout* e instalação do Gravatar (AUTOMATTIC, 2017) que será utilizado para a exibição de fotos no perfil dos professores. A Figura 27 ilustra o quadro de *Kanban* (IKONEN et al., 2011) no início da terceira etapa de desenvolvimento.

Figura 27 – Terceira Etapa *Kanban*



Fonte: Autoria Própria.

Como primeira atividade, desenvolveu-se a configuração do ambiente administrativo. O *framework* Django (HOLOVATY; KAPLAN-MOSS, 2009) possui por padrão um ambiente administrativo próprio, sendo assim, para se obter um ambiente administrativo funcional para este sistema foi necessário alterar suas configurações.

A partir do sistema administrativo é possível fazer a inserção e edição de dados, cadastro e edição de usuários e grupos além da exclusão dos mesmos. A Figura 28 ilustra a configuração da tabela de professor no arquivo de configuração do *admin* da aplicação. Nesta Figura os campos *id*, *nome*, *status*, *email*, *usuario* e *desc* foram listados no ambiente administrativo. Os campos *id*, *nome* e *email* são usados para pesquisa, e o campo *status* utilizado como filtro para a listagem dos campos. Após isto, é realizado o registro desta tabela no ambiente administrativo através do uso de da função *register* do *Framework* Django (HOLOVATY; KAPLAN-MOSS, 2009).

Figura 28 – Admin

```
class professorAdmin(admin.ModelAdmin):
    list_display = ('id','nome','status','email','usuario','desc')
    search_fields =['id','nome','email']
    list_filter =['status']

admin.site.register(Professor, professorAdmin)
```

Fonte: Autoria Própria.

Uma vez configurado o ambiente administrativo, desenvolveu-se o *logout* de um usuário do sistema. Para isto, criou-se uma função na *view* e uma url que invocasse esta função assim que fosse clicado no *link* de *logout* no *template*. A Figura 29 ilustra a implementação da função *logout_view* para a realização do *logout*.

Figura 29 – Logout

```
def logout_view(request):
    logout(request)
    return redirect('/')
```

Fonte: Autoria Própria.

Como última atividade, desenvolveu-se a instalação do serviço Gravatar na aplicação. Este serviço possibilita que através de um e-mail cadastrado, uma fotografia seja relacionada a este. Sendo assim, caso o usuário utilize este e-mail cadastrado no Gravatar em outras aplicações que também utilizem este serviço, a foto que foi cadastrada será exibida automaticamente. A Figura 30 ilustra a utilização do Gravatar (AUTOMATTIC, 2017).

Figura 30 – Gravatar

```
<center>

```

Fonte: Autoria Própria.

Como é possível ver na figura 30 acima, através do uso do Gravatar (AUTOMATTIC, 2017) é possível criar uma URL que seja usada para colocar a imagem de perfil no *template*. Esta URL chama uma função do Gravatar (AUTOMATTIC, 2017) na qual passa como parâmetro o e-mail e a função retorna o

link para a imagem relacionada a aquele e-mail. Conseqüentemente, ao colocar o retorno desta função no campo 'src' da tag *img* no html, esta imagem estará referenciando a imagem colocada na aplicação do Gravatar (AUTOMATTIC, 2017).

3.2.4 Quarta Etapa de Desenvolvimento

Nesta etapa focou-se no desenvolvimento dos *templates* para a aplicação de perfil e também na criação de *links* entre as duas aplicações. Desenvolveu-se as seguintes atividades nesta etapa: desenvolvimento de uma página de *login* para a aplicação de perfil; criação de *links* entre as aplicações (de perfil e a principal) e; criação de *templates* para a aplicação de perfil. O quadro de *Kanban* (IKONEN et al., 2011) referente ao início da quarta etapa é ilustrado na Figura 31.

Figura 31 – Quarta Etapa Kanban

Á fazer	Em andamento 3/3	Feito
Formulários	Página de login	Instalar o Gravatar
Criar os models para o Banco de dados dos Forms	Conectar as duas aplicações	Logout
Configurar a aplicação para permitir upload de arquivos	Criar os primeiros templates da aplicação de perfil	Ambiente administrativo

Fonte: Autoria Própria.

A criação da página de *login* que será utilizada na aplicação de perfil foi a primeira atividade que se desenvolveu na quarta etapa. Para isso, criou-se o *template* para a página de *login*. Sendo assim, quando um professor optar por realizar o *login*, o mesmo poderá fazer em qualquer uma das aplicações.

Desenvolveu-se como segunda atividade a conexão entre as duas aplicações. Como citado anteriormente, há uma aplicação principal e outra que irá fazer o gerenciamento do perfil dos professores que será executada em conjunto com a aplicação principal. Para que ambas se conectem, criaram-se *links* entre elas e também se realizou a inserção das rotas nos arquivos de URLs para trabalharem em conjunto.

Como última atividade, desenvolveu-se a criação dos *templates* utilizados na aplicação de perfil. O primeiro *template* refere-se ao de perfil o qual exibe a foto do professor e informações a seu respeito, o segundo, diz respeito ao local em que serão exibidas as postagens feitas pelo professor, o terceiro, é responsável pelo contato com o docente, ou seja, a página em que será possível enviar uma mensagem ao mesmo e, por fim, o último *template*, refere-se aos TCCs. Neste *template* são exibidos os trabalhos de conclusão de curso de um determinado professor.

3.2.5 Quinta Etapa de Desenvolvimento

Na quinta etapa desenvolveu-se as seguintes atividades: criação dos formulários utilizados nas aplicações; criação do banco de dados para armazenar o conteúdo de alguns formulários, tais como: formulário de postagens e de *upload* de arquivos; criação do mecanismo para fazer *upload* de arquivos na aplicação. A Figura 32 ilustra o quadro de *Kanban* (IKONEN et al., 2011) no início da quinta etapa de desenvolvimento.

Figura 32 – Quinta Etapa Kanban



Fonte: Autoria Própria.

A primeira atividade que se desenvolveu no início desta fase foi a de criação dos formulários para serem utilizados nos *templates*. As páginas da aplicação que utilizam formulários são: *Post* na aplicação principal e as páginas de Postagens, Contato e TCCs na aplicação de perfil.

A segunda atividade realizada foi a de criação das tabelas para a inserção de dados de formulários. Criou-se estas tabelas com o intuito de armazenar os *Posts* na aplicação principal, as *Postagens* e os *TCCs* na aplicação de perfil.

Os *Posts* na aplicação principal e as *Postagens* nos perfis dos professores são iguais, sendo estes apenas mensagens criadas pelos próprios docentes para exibição nas aplicações. Os *TCCs* são os trabalhos ou arquivos que serão colocados à disposição pelo professor, ou seja, caso o professor queira disponibilizar algum trabalho ou arquivo estes serão salvos no banco de dados.

A terceira atividade realizada nesta etapa foi a de desenvolvimento de uma página para fazer o *upload* de arquivos no sistema, visto que é uma das principais funcionalidades do mesmo.

3.2.6 Sexta Etapa de Desenvolvimento

Desenvolveu-se na sexta etapa as seguintes atividades: criação de uma maneira de editar informações pela aplicação; desenvolvimento da Página de Contato no perfil dos professores; e criação da página Professor/Área na aplicação principal. A Figura 33 ilustra a situação do quadro de *Kanban* (IKONEN et al., 2011) no início desta etapa.

Figura 33 – Sexta Etapa *Kanban*



Fonte: Autoria Própria.

A primeira atividade desenvolvida nesta etapa foi a de criação de uma página para editar informações nas páginas de postagens e perfil dos professores. Para isto, criou-se funções nas *views* das aplicações para que os dados pudessem ser atualizados após suas edições.

A criação de um formulário para o envio de mensagens por e-mail através da aplicação foi a segunda etapa desenvolvida. Para isto, desenvolveu-se a página de contato na aplicação de perfil.

A última atividade desenvolvida nesta etapa foi a de criação de uma página a qual relaciona os professores com as suas áreas de atuação, ou seja, caso um professor tenha várias áreas de atuação, estas estarão listadas nesta página.

3.2.7 Sétima Etapa de Desenvolvimento

Na etapa final realizou-se testes unitários da aplicação a fim de validá-la. Sendo assim, foram cadastrados usuários, grupos, professores, disciplinas, áreas e postagens para garantir que todas as funcionalidades do sistema *web* estavam funcionando corretamente. Adicionalmente, também foram verificadas as permissões de diferentes usuários: os professores, os administradores e os alunos que serão usuários anônimos. A Figura 34 ilustra o quadro de *Kanban* (IKONEN et al., 2011) na última etapa de desenvolvimento, onde todas as atividades estão na coluna “Feito”.

Figura 34 – Sétima Etapa *Kanban*



Fonte: Autoria Própria.

4 RESULTADO

Este Capítulo aborda os tópicos referentes ao resultado obtido ao final do desenvolvimento da aplicação.

A seção 4.1 apresenta a página inicial da aplicação *web*. A seção 4.2 exibe a página de professores. A página de áreas é abordada na seção 4.3. A seção 4.4 detalha a página de detalhes da área. A seção 4.5 discorre sobre a página de detalhes da matéria. A página de *posts* é apresentada na seção 4.6. Na seção 4.7 exibe a página de professor/área da aplicação. A página de *login* é detalhada na seção 4.8. A seção 4.9 aborda a página de perfil do professor. A página de postagens é apresentada na página 4.10. Na seção 4.11 discorre sobre a página TCCs. A página de contato é apresentada na seção 4.12. Na seção 4.13 exibe a página de *login* no ambiente administrativo. E, por fim, a seção 4.14 apresenta a página inicial do ambiente administrativo.

4.1 PÁGINA INICIAL

Considerada como página de apresentação. Nesta, exibe-se o logo utilizado e uma breve descrição sobre a aplicação. Além disso, na parte superior da página encontram-se os *links* para as demais com o intuito de facilitar a navegação entre as mesmas. A Figura 35 ilustra a página inicial do sistema.

Figura 35 – Página Inicial

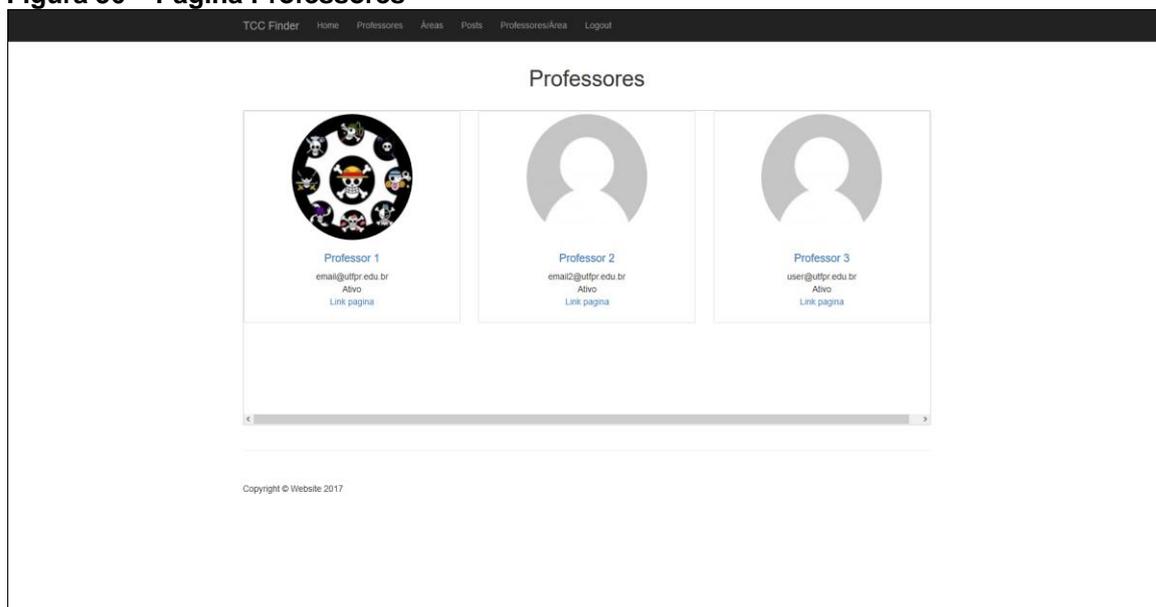


Fonte: Autoria Própria.

4.2 PÁGINA DE PROFESSORES

A página de Professores tem como intuito a exibição de todos os professores cadastrados no sistema. Nesta serão exibidos os docentes com suas respectivas imagens de perfil, seus e-mails, status e o *link* para a página de perfil dos mesmos. A Figura 36 exibe a página de professores da aplicação *web* desenvolvida.

Figura 36 – Página Professores



Fonte: Autoria Própria.

4.3 PÁGINA DE ÁREAS

A lista de todas as áreas cadastradas no sistema é exibida na página de Áreas. Adicionalmente, nesta página também são exibidas as matérias respectivas a cada área bem como seus professores. A Figura 37 ilustra a página de áreas da aplicação.

Figura 37 – Página Áreas

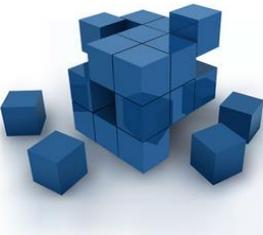
Nome	Matérias relacionadas	Professores das matérias
Redes	Redes Redes 2	Professor 1 Professor 1
Banco de dados	Banco de dados	Professor 2
Teoria da computação	LFAC Compiladores Análise de algoritmos	Professor 1 Professor 1 Professor 1
Estrutura de dados	Análise de algoritmos	Professor 3

Fonte: Autoria Própria.

4.4 PÁGINA DE DETALHES DA ÁREA

Esta página tem como o intuito mostrar informações adicionais sobre alguma área. Sendo que, estas informações são cadastradas no sistema na hora da criação da mesma. Dentre estas informações estão o nome e uma breve descrição dela. A Figura 38 exibe a página de detalhes da área.

Figura 38 – Página de Detalhes da Área

TCC Finder	Home	Professores	Áreas	Posts	Professores/Área	Logout
<h2>Área</h2>  <h3>Estrutura de dados</h3> <p>Ham hock capicola porchetta biltong short ribs pork frankfurter. Pork loin chuck pork chop, bacon pork strip steak landjaeger. Chuck pork chop biltong pork loin, landjaeger jowl shank ham hock. Ground round tri-tip cow andouille picanha fatback.</p>						
Copyright © Website 2017						

Fonte: Autoria Própria.

4.5 PÁGINA DE DETALHES DA DISCIPLINA

As informações sobre uma determinada disciplina são acessadas através do *link* na página de áreas. Estas informações são cadastradas no momento da criação da matéria no sistema. A Figura 39 ilustra a página de detalhes da disciplina.

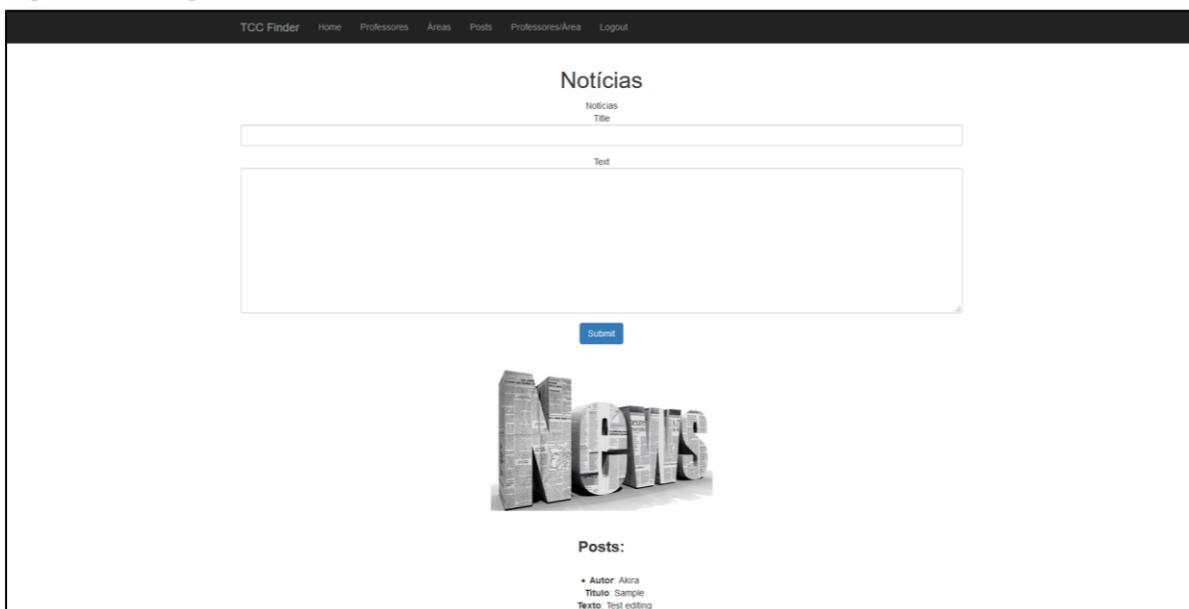
Figura 39 – Página de Detalhes da Disciplina



Fonte: Autoria Própria.

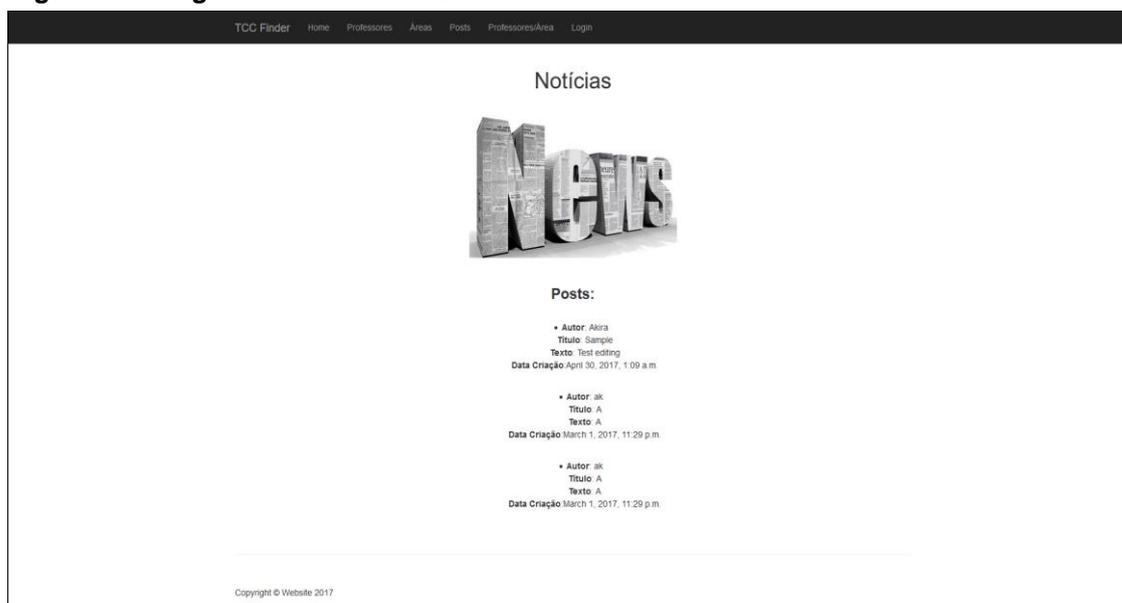
4.6 PÁGINA DE *POSTS*

A página de *posts* tem como intuito exibir postagens feitas pelo professor responsável pela aplicação. Nesta área, encontram-se informações, notícias e avisos para outros usuários visualizarem. A Figura 40 exibe a página de *posts* da maneira que o professor com as permissões necessárias vê.

Figura 40 – Página de Posts com Usuário com Permissões de Post

Fonte: Autoria Própria.

A Figura 41 ilustra a página de posts para um usuário que não tem as permissões necessárias para criar uma postagem na aplicação. Nesta, as informações são exibidas e não podem ser editadas.

Figura 41 – Página de Posts com Usuário Sem Permissão de Post

Fonte: Autoria Própria.

4.7 PÁGINA DE PROFESSOR/ÁREA

Na página de professor/área são exibidas as relações dos professores com suas áreas de interesse. A Figura 42 ilustra a página professor/área.

Figura 42 – Página Professor/Área.

TCC Finder Home Professores Áreas Posts Professores/Área Login	
<h3>Professores /Áreas de interesse</h3>	
Professor	Área
Professor_1	Redes
Professor_3	Banco de dados
Professor_2	Teoria da computação
Copyright © Website 2017	

Fonte: Autoria Própria.

4.8 PÁGINA DE LOGIN

A página de *login* tem como propósito fazer a autenticação do usuário no sistema. Caso o usuário deseje entrar no sistema ele deve clicar em “*login*”, localizado na parte superior da página. A Figura 43 exibe a página de *login* do sistema.

Figura 43 – Página de Login

Fonte: Autoria Própria.

4.9 PÁGINA DE PERFIL DO PROFESSOR

Na página de perfil do professor são exibidas informações sobre o mesmo, como, por exemplo, seu nome, e-mail, matérias que leciona, foto de perfil e uma breve descrição. A Figura 44 exibe a página de perfil do professor quando ele está autenticado no sistema. Neste caso, o sistema permite ao docente visualizar informações adicionais como o seu nome de usuário e botão que permite realizar alterações das informações de seu perfil.

Figura 44 – Página de Perfil com Usuário Autenticado

Fonte: Autoria Própria.

A Figura 45 ilustra a página de perfil do professor, porém visualizado por um usuário que não esteja autenticado no sistema. As informações apresentadas nesta página são de domínio público, ou seja, visíveis para qualquer usuário.

Figura 45 – Página de Perfil com Usuário não Autenticado

[TCC Finder](#) [Perfil](#) [Postagens](#) [TCCs](#) [Contato](#) [Login](#)



Professor 1

email@utfpr.edu.br
[Postagens](#) | [Contato](#) | [TCCs](#)

Matérias:

Redes
areaRedes

Redes 2
areaRedes

Linguagens Formais, Autômatos
areaTeoria da computação

Compiladores
areaTeoria da computação

Análise E Projetos De Algoritmos
areaTeoria da computação

Descrição

O professor 1 é Doutor em Ciência da Computação Pela UTFPR e tem mais de 10 anos de experiência na área.

Fonte: Autoria Própria.

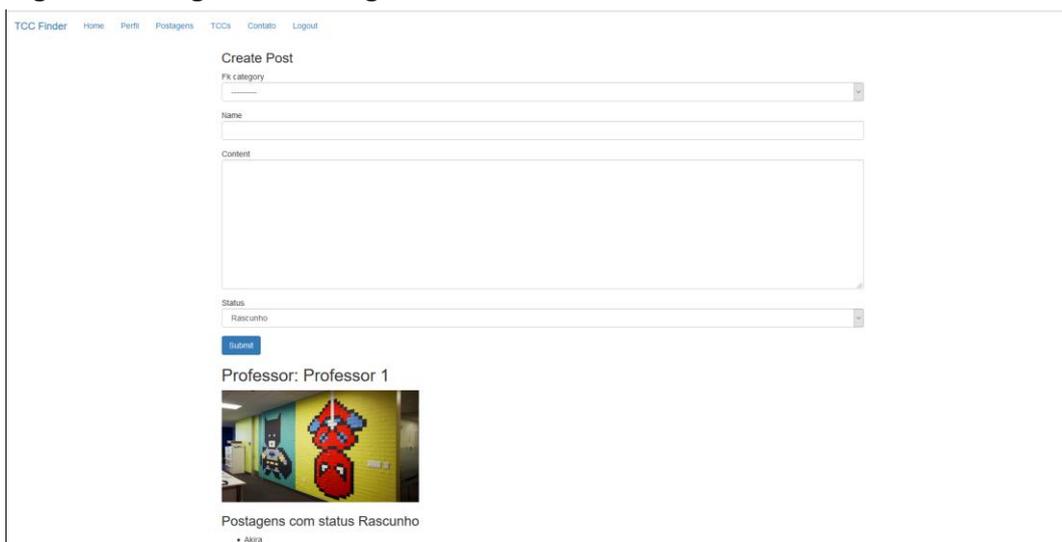
4.10 PÁGINA DE POSTAGENS

A página de postagens do professor permite que o mesmo escreva informações que são visíveis a outros usuários. A tarefa de criação de postagens é

somente permitida para professores que estejam autenticados e possuam permissões para tal. Caso contrário, exibem-se as postagens criadas.

A Figura 46 exibe a página de postagens do professor com o mesmo autenticado e com as devidas permissões para postar informações.

Figura 46 – Página de Postagens do Professor Autenticado

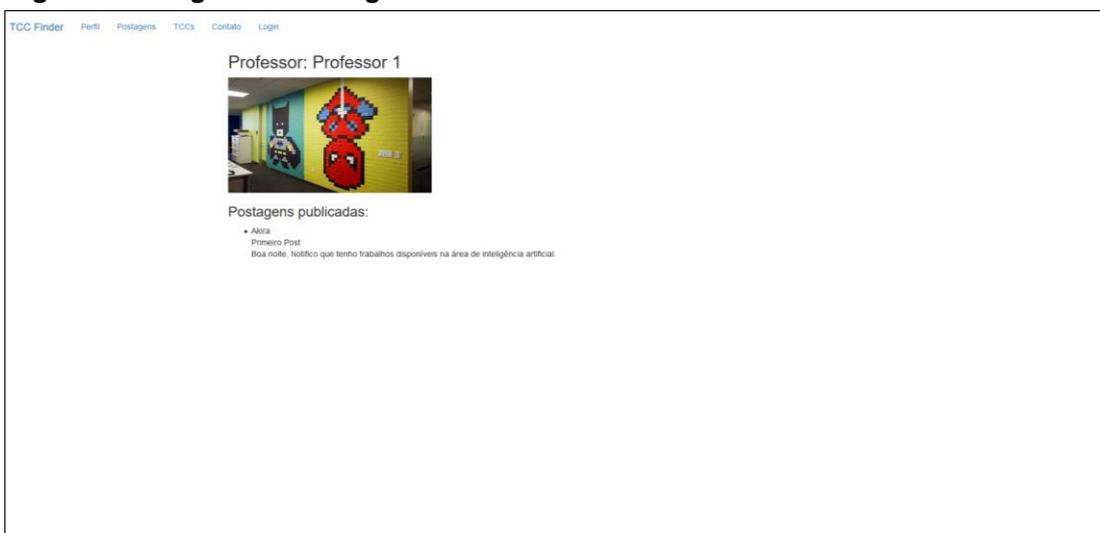


The screenshot shows a web interface for creating a post. At the top, there is a navigation bar with links: TCC Finder, Home, Perfil, Postagens, TCCs, Contato, and Logout. Below the navigation bar, the main content area is titled 'Create Post'. It contains several input fields: a dropdown menu for 'FK category', a text input for 'Name', a large text area for 'Content', and a dropdown menu for 'Status' with 'Rascunho' selected. Below these fields, there are two buttons: 'Student' (highlighted in blue) and 'Professor'. Underneath the buttons, the user's profile is displayed as 'Professor: Professor 1' with a profile picture of a red and blue character. Below the profile picture, it says 'Postagens com status Rascunho' and a small 'Alura' logo.

Fonte: Autoria Própria.

A Figura 47 exibe a página de postagens do professor para um usuário não autenticado ou que não possui permissão para criar uma postagem nesta página.

Figura 47 – Página de Postagens do Professor Não Autenticado



The screenshot shows a web interface for viewing published posts. At the top, there is a navigation bar with links: TCC Finder, Perfil, Postagens, TCCs, Contato, and Login. Below the navigation bar, the main content area is titled 'Professor: Professor 1' with a profile picture of a red and blue character. Below the profile picture, it says 'Postagens publicadas:'. There is a small 'Alura' logo and a list of posts. The first post is titled 'Primeiro Post' and has the text 'Boa noite, notifico que tenho trabalhos disponíveis na área de inteligência artificial.'

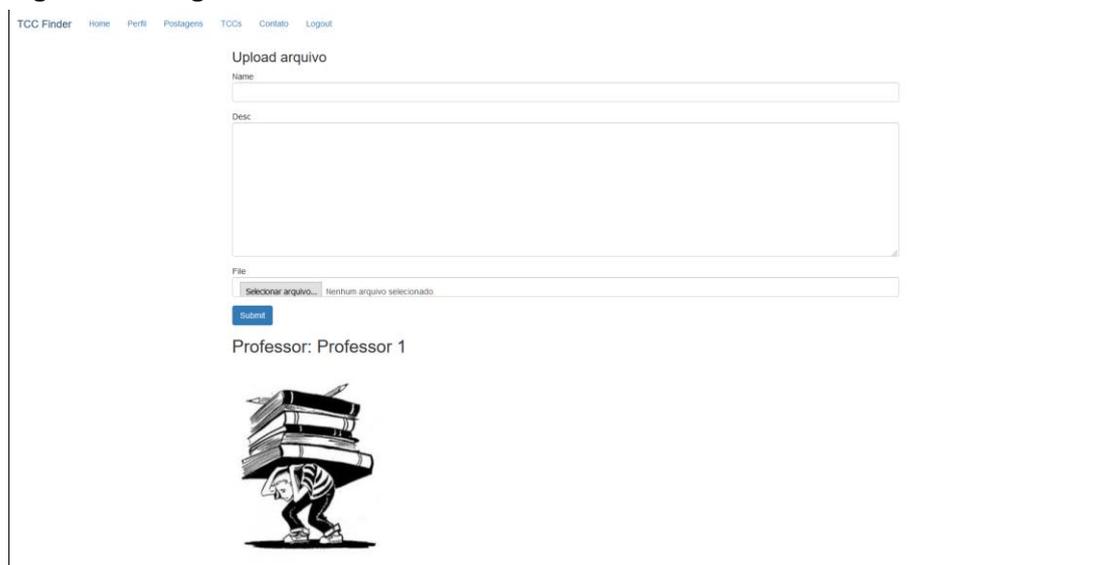
Fonte: Autoria Própria.

4.11 PÁGINA TCCS

A página de Trabalhos de Conclusão de Curso (TCC) exibe arquivos para auxiliar os alunos a encontrar ou escrever sobre um tema de TCC. Nesta, o professor pode fazer *upload* de arquivos para permitir o acesso aos mesmos por parte dos alunos, sendo que, dentre estes arquivos também podem ser disponibilizados trabalhos já orientados pelo mesmo.

A Figura 48 apresenta a página de TCCs visualizada pelo professor autenticado e com as permissões para fazer o *upload* de arquivos.

Figura 48 – Página de TCCs do Professor Autenticado

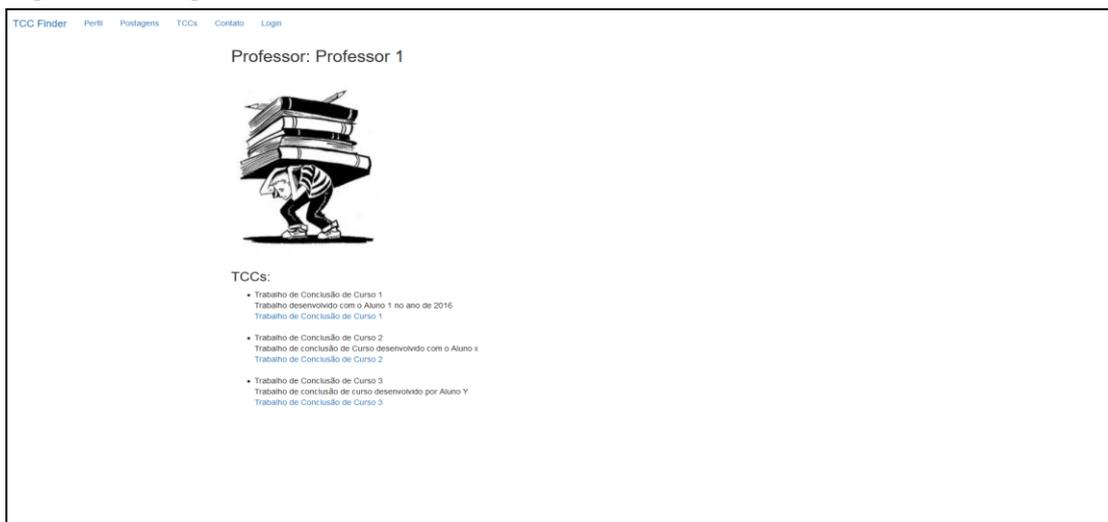


The screenshot displays the 'TCCs' page for an authenticated professor. At the top, a navigation menu includes 'TCC Finder', 'Home', 'Perfil', 'Postagens', 'TCCs', 'Contato', and 'Logout'. The main content area is titled 'Upload arquivo' and contains a form with the following fields: 'Name' (a text input), 'Desc' (a larger text area), and 'File' (a file selection button labeled 'Selecionar arquivo...' with the text 'Nenhum arquivo selecionado' below it). A blue 'Submit' button is positioned below the form. Below the form, the text 'Professor: Professor 1' is displayed. At the bottom of the page, there is a cartoon illustration of a person carrying a large stack of books on their back.

Fonte: Autoria Própria.

A Figura 49 ilustra a página de TCCs para um usuário não autenticado ou que não possua as permissões necessárias para publicar nesta página.

Figura 49 – Página de TCCs do Professor Não Autenticado



Fonte: Autoria Própria.

4.12 PÁGINA DE CONTATO

A página de contato possibilita que um usuário não autenticado, como por exemplo, um aluno que está à procura de informações, entre em contato com o professor. Nesta página há um formulário em que o discente preenche os campos: assunto, nome, curso, mensagem e seu e-mail para contato. Ao pressionar o botão *Submit*, as informações são enviadas ao e-mail cadastrado para aquele professor. A Figura 50 exibe a página de contato.

Figura 50 – Página de Contato

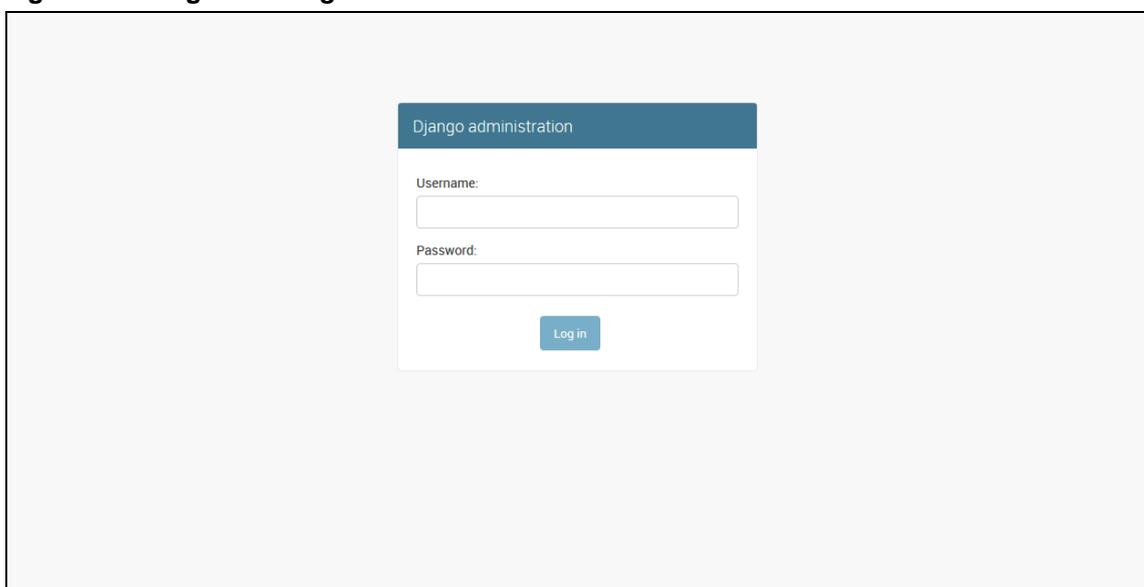
Fonte: Autoria Própria.

4.13 LOGIN AMBIENTE ADMINISTRATIVO

A Figura 51 ilustra a página de *login* no ambiente administrativo da aplicação. Como já citado anteriormente, o ambiente administrativo foi adaptado do *Framework Django* (HOLOVATY; KAPLAN-MOSS, 2009).

Esta tela é responsável por fazer a autenticação do usuário no sistema administrativo.

Figura 51 – Página de *Login* Ambiente Administrativo.

The image shows a screenshot of the Django administration login page. It features a central white box with a blue header that reads "Django administration". Below the header, there are two input fields: "Username:" and "Password:". A blue "Log in" button is positioned below the password field. The entire form is centered on a light gray background.

Fonte: Autoria Própria.

4.14 PÁGINA INICIAL DO AMBIENTE ADMINISTRATIVO

A Figura 52 ilustra a página inicial do ambiente administrativo. Nesta são exibidas as tabelas que foram criadas no banco de dados e, por meio destas, o professor responsável pode realizar qualquer alteração na aplicação.

Figura 52 – Página Inicial do Ambiente Administrativo

The screenshot displays the Django administration interface. At the top, there is a header with the text "Django administration". Below this, the main content area is titled "Site administration".

The "Site administration" section is divided into three main categories, each with a blue header bar:

- AUTHENTICATION AND AUTHORIZATION**: This section contains two items: "Groups" and "Users". Each item has a green "+ Add" button and a yellow pencil "Change" button.
- BLOG**: This section contains four items: "Areas", "Materias", "Professors", and "Relacao_area_professors". Each item has a green "+ Add" button and a yellow pencil "Change" button.
- PERFIL**: This section contains three items: "Arquivos", "Categorias", and "Posts". Each item has a green "+ Add" button and a yellow pencil "Change" button.

On the right side of the interface, there is a "Recent actions" panel. It is titled "Recent actions" and contains a sub-section "My actions". This panel lists several actions, each with a yellow pencil icon (indicating a successful action) or a red 'x' icon (indicating a failed action):

- arquivo object (Arquivo) - successful
- Teste de Post (Post_perfil) - failed
- segundo teste (Post_perfil) - failed
- teste 3 (Post_perfil) - failed
- teste 3 (Post_perfil) - failed
- i (Post_perfil) - failed
- i (Post_perfil) - failed

Fonte: Autoria Própria.

5 CONCLUSÃO

Neste trabalho foi desenvolvida uma aplicação *web* com o intuito de auxiliar os alunos da Universidade Tecnológica Federal do Paraná – Campus Ponta Grossa, a encontrar um orientador para o desenvolvimento do trabalho de conclusão de curso. Por meio desta aplicação os alunos podem realizar consultas a informações de um determinado professor, áreas e disciplinas que lhe interessam.

A aplicação desenvolvida também permite que os alunos possam enviar mensagens ao docente, facilitando assim a forma de entrar em contato com o mesmo para sanar dúvidas sobre pesquisas ou até mesmo, para propor um tema a ser desenvolvido como trabalho de conclusão de curso.

Adicionalmente, esta aplicação possibilita aos professores disponibilizar trabalhos já orientados ou outros arquivos que julgarem importantes para auxiliar aos alunos na escolha de um tema para seu próprio trabalho, ou simplesmente para utilizá-los como referências.

Para o desenvolvimento deste trabalho foi utilizada a metodologia de gerenciamento de projeto *Kanban* (IKONEN et al., 2011) e a documentação foi modelada utilizando a linguagem unificada de modelagem (UML) (BOOCH; RUMBAUGH; JACOBSON, 2006). A partir do uso destas, pode-se criar um ciclo de trabalho que possibilitasse o desenvolvimento da aplicação sem o desperdício de esforços e de tempo, a fim de buscar uma melhor qualidade no produto final.

5.1 TRABALHOS FUTUROS

Nesta subseção são apresentadas algumas sugestões para trabalhos futuros. São eles:

- Refatorar a interface da aplicação *web* seguindo diretrizes de autores da área de IHC;
- Integrar a aplicação com o repositório ROCA (ROCA, 2017) que é utilizado pela Universidade Tecnológica Federal do Paraná para disponibilizar trabalhos de conclusão de curso desenvolvidos pelos alunos;

- Criar um repositório com temas de trabalhos disponíveis para serem desenvolvidos;
- Permitir que os alunos possam se cadastrar para receberem avisos sobre as postagens feitas pelo professor;
- Refatorar as seções de postagens para permitir que imagens e vídeos possam ser anexados às mensagens;
- Estender a utilização da aplicação para outros departamentos e instituições;

REFERÊNCIAS

AHMAD, M. O.; et al. **Kanban in software development**: A systematic literature review. Santander, Spain: *Software Engineering and Advanced Applications (SEAA)*, 2013 39th EUROMICRO Conference, 2013.

ANDERSON, M. **Technology Device Ownership**.2015. Disponível em: <<http://www.pewinternet.org/2015/10/29/technology-device-ownership-2015/>> Acesso em: 18 mai. 2016.

AUTOMATTIC. **Gravatar**. 2017. Disponível em: < <http://pt.gravatar.com/>> Acesso em: 10 abr. 2017.

BELL, D. **UML basics**: The component diagram. Disponível em: <<https://www.ibm.com/developerworks/rational/library/dec04/bell/index.html>> Acesso em: 17 nov. 2016.

BENITO-OSORIO, D; et al. *Web 5.0: the future of emotional competences in higher education*. **Global Business Perspectives**, Manhattan, NY, EUA v.1, n. 3, p. 274-287, dez. 2013.

BERARDI, D.; CALVANESE, D.; DE GIACOMO, G. Reasoning on UML class diagrams. **Artificial Intelligence**, v. 168, n. 1, p. 70-118, out. 2005.

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I; **UML**: guia do usuário. Rio de Janeiro, RJ: Elsevier Brasil, 2006.

BORGES, R. M.; MOTA, A. C. Integrating UML and formal methods. **Electronic Notes in Theoretical Computer Science**, v. 184, p. 97-112, 2007.

DOS SANTOS PIRES, A. **O kernel do Linux**: A definição, importância e funcionalidades. Disponível em: < <https://www.vivaolinux.com.br/artigo/O-kernel-do-Linux-A-definicao-importancia-e-funcionalidades?pagina=2>>. Acesso em: 14 jun. 2016.

FLANAGAN, D. **JavaScript**: the definitive guide. O'Reilly Media, Inc. Sebastopol, CA, Estados Unidos 2011.

HOLOVATY, A.; KAPLAN-MOSS, J. **The definitive guide to Django: Web development done right**. Apress, Nova York, NY, Estados Unidos, 2009.

IKONEN, M.; et al. **On the impact of Kanban on software project work: An empirical case study investigation**. Engineering of Complex Computer Systems (ICECCS), 2011 16th IEEE International Conference on. IEEE, 2011.

KAMBIL, A. What is your *Web 5.0* strategy? **Journal of Business Strategy**, Vol. 29, p. 56–58, West Yorkshire, WA, Estados Unidos, 2008.

KANBAN FLOW. **Kanbanflow**. 2017. Disponível em: <<https://Kanbanflow.com/>> Acesso em: 15 fev. 2017.

KENNEDY, B.; MUSCIANO, C. **HTML & XHTML: The Definitive Guide**. O'Reilly, 2006.

KIM, H; et al. **Test cases generation from UML activity diagrams**. *Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, 2007. SNPD 2007. Eighth ACIS International Conference on. Vol. 3. IEEE, 2007.

KLEIN, E. **A UML no desenvolvimento ágil de sistemas**. Disponível em: <<http://www.mobiltec.com.br/blog/index.php/a-uml-no-desenvolvimento-agil-de-sistemas/>>. Acesso em: 01 jun. 2016.

MARCONDES, C. A. **Programando em HTML 4.0**. São Paulo, SP: Editora Érica, 2000.

MELTON, J. Sql language summary. **ACM Computing Surveys (CSUR)**, v. 28, n. 1, p. 141-143, New York, NY, EUA, 1996.

MEYER, E. A. **CSS: The Definitive Guide: The Definitive Guide**. O'Reilly Media, Inc., 2006.

MICROSOFT. **Visual Studio Code**. 2017. Disponível em: <<https://code.visualstudio.com/>> Acesso em: 15 mar. 2017.

MICROSOFT. **Bash on ubuntu on Windows**. 2017. Disponível em: <<https://msdn.microsoft.com/en-us/commandline/wsl/about/>> Acesso em: 9 jan. 2017.

NEGRINO, T.; SMITH, D. **JavaScript para world wide web**. Pearson Education, 2001.

NIC.br. **NIC.br anuncia resultados da pesquisa sobre o uso da Internet no Brasil**, 2014. Disponível em: <<http://nic.br/noticia/nic-br-anuncia-resultados-da-pesquisa-sobre-o-uso-da-internet-nas-empresas/111>> Acesso em: 20 jun. 2016.

POUSHTER, J. **Smartphone Ownership and Internet Usage Continues to Climb in Emerging Economies**: But advanced economies still have higher rates of technology use. Pew Research Center, fev. 2016.

PYTHON.ORG. **Creation of virtual environments**. 2017. Disponível em: <<https://docs.python.org/3/library/venv.html>> Acesso em: 20 fev. 2017

PYTHON.ORG. **Should I use Python 2 or Python 3 for my development activity?** 2017. Disponível em: <<https://wiki.python.org/moin/Python2orPython3>> Acesso em: 27 abr. 2017.

RIBEIRO, A.M. **UML**. Disponível em: <http://homepages.dcc.ufmg.br/~amendes/GlossarioUML/glossario/conteudo/distribuicao/diagrama_de_distribuicao.htm>. Acesso em: 07 jun. 2016.

ROCA. **Repositório de Outras Coleções Abertas**. 2017. Disponível em: <<http://repositorio.roca.utfpr.edu.br>> Acesso em: 20 abr. 2017

RUMBAUGH, J; et al. **Modelagem e projetos baseados em objetos**. Vol. 8., Rio de Janeiro, RJ: Editora Campus, 1994.

SANNER, M. F. **Python**: a programming language for *software* integration and development. J Mol Graph Model, La Jolla, CA, EUA, v. 17, n. 1, p. 57-61, 1999.

SENGUPTA, S.; BHATTACHARYA, S. **Formalization of UML use case diagram-a Z notation based approach**. In Computing & Informatics, 2006. ICOCI'06. International Conference on p. 1-6. IEEE, jun, 2006.

StarUML 2.8.0. **Download**. 2016. Disponível em: <<http://staruml.io/download>>. Acesso em: 10 dez. 2016.

UTFPR. **Regulamento do Trabalho de Conclusão de Curso (TCC) para os Cursos de Graduação da UTFPR.** Disponível em: <http://www.utfpr.edu.br/estrutura-universitaria/pro-reitorias/prograd/legislacao/utfpr-1/regulamento_tcc_utfpr.pdf> Acesso em: 01 abr. 2016.

W3C. **HTML & CSS.** Disponível em: <<https://www.w3.org/standards/webdesign/htmlcss>>. Acesso em: 06 jun. 2016.

W3C. **HTML.** Disponível em: <<https://www.w3.org/html/>>. Acesso em: 06 jun. 2016.

WSGI.ORG. **What is WSGI?** 2017. Disponível em: <<https://wsgi.readthedocs.io/en/latest/what.html>> Acesso em: 27 mai. 2017.