
Curso de Ciência da computação
Universidade Estadual de Mato Grosso do Sul

PORTAL UNIVERSITÁRIO *WEB*: MÓDULO MORADAS

Guilherme Fonseca Corrêa Pires de Araújo

MSc. André Chastel Lima(Orientador)

Dourados - MS

2018

PORTAL UNIVERSITÁRIO *WEB*: MÓDULO MORADAS

Guilherme Fonseca Corrêa Pires de Araújo

Este exemplar corresponde à redação final da monografia da disciplina Projeto Final de Curso devidamente corrigida e defendida por Guilherme Fonseca Corrêa Pires de Araújo e aprovada pela Banca Examinadora, como parte dos requisitos para a obtenção do título de Bacharel em Ciência da Computação.

Dourados, 25 de novembro de 2018

MSc. André Chastel Lima(Orientador)

Curso de Ciência da Computação
Universidade Estadual de Mato Grosso do Sul

PORTAL UNIVERSITÁRIO *WEB*: MÓDULO MORADAS

Guilherme Fonseca Corrêa Pires de Araújo

Novembro de 2018

Banca Examinadora:

Prof. Msc. André Chastel Lima(Orientador)
Área de Computação - UEMS

Profa. Dra. Glaucia Gabriel Sass
Área de Computação - UEMS

Prof. Dr. Cleber Valgas Gomes Mira
Área de Computação - UEMS

AGRADECIMENTOS

Agradeço primeiramente a Deus que permitiu que tudo isso acontecesse ao longo de minha vida, e não somente nestes anos como universitário, mas em todos os momentos, pois é meu maior mestre.

Agradeço a Universidade Estadual do Mato Grosso do Sul, seu corpo docente, direção e administração que oportunizaram o conhecimento na área escolhida. Ao meu orientador MSc. Andre Chastel Lima, pelo empenho dedicado à elaboração deste trabalho. E também a todos os professores que se dedicaram no meu processo de formação profissional

Obrigado a todos da família, principalmente meu pai Hamilton, minha mãe Milma, minhas irmãs Gabriele e Milena, minha tia Neide e meu tio Toninho, e minha prima Aniele, e por último e não menos importante, aos meus avós Nadir e Sebastião, que me acolheram em sua casa nesse período de estudo, estes, que entenderam minha ausência, principalmente nos dias que estive os três períodos na universidade.

À Amanda minha namorada, pessoa com quem amo partilhar a vida. Com você tenho me sentido mais vivo de verdade. Obrigado pelo carinho, a paciência e por sua capacidade de me trazer paz na correria de cada semestre. Agradeço meus amigos Álvaro, Calebe e Gustavo, que tanto me ajudaram, me deram força e apoio nesse período, aos demais colegas de turma, pelo companheirismo e amizade.

Agradeço também ao Ministério Universidades Renovadas, na qual, pude exercer a missão de evangelização, e também de ser evangelizado, e das pessoas que conheci através desse ministério. Agradeço a todos que direta ou indiretamente fizeram parte da minha formação

Sê humilde para evitar o orgulho, mas voa alto para alcançar a sabedoria. Santo Agostinho

RESUMO

Dourados é considerada um polo universitário dentro do estado de Mato Grosso do Sul. Ainda dentro desse contexto, a cidade conta com o Campus Universitário situado fora da cidade onde está instalada a UEMS e UFGD. Assim, o estudo consiste na elaboração de uma página web para facilitar a escolha de onde os acadêmicos, professores, servidores e técnicos da Cidade Universitária de Dourados ou de outros polos universitários podem possivelmente morar. Hoje diversos estudantes de outras cidades quando vêm para Dourados se deparam com uma dificuldade em encontrar moradia. O Portal Universitário consiste no desenvolvimento da página web para informar aqueles que buscam uma: kitnet, casa, apartamento, república ou apenas um quarto para dormir.

Palavras-chave: Campus Universitário, Moradia, Python, Django.

ABSTRACT

Dourados is considered a university center within the state of Mato Grosso do Sul. Also within this context, the city counts on the University Campus located outside the city where UEMS and UFGD are installed. Thus, the study consists of the elaboration of a web page to facilitate the choice of where the academics, teachers, servers and technicians of the University City of Dourados can possibly live. Today several students from other cities when they come to Dorados are faced with a difficulty in finding housing. The University Portal consists of the development of the web page to inform those who are looking for a kitnet, house, apartment, republic or just a bedroom to sleep.

Key-words: University Campus, Home, WEB, Python, Django.

SUMÁRIO

	Lista de ilustrações	xvii
1	INTRODUÇÃO	1
1.1	Objetivos Gerais	2
1.1.1	Objetivos	2
1.2	Organizações dos capítulos	2
2	METODOLOGIA E DESENVOLVIMENTO	3
2.1	Introdução	3
2.2	Metodologia	3
2.3	Definição dos Requisitos do Usuário	3
2.4	Especificação dos Requisitos do Sistema	3
2.4.1	Requisitos Funcionais	4
2.4.2	Requisitos Não Funcionais	4
2.4.3	Descrição de Usuário	5
2.5	Visão de Caso de Uso	5
2.5.1	Modelo de Casos de Uso	5
2.5.1.1	Descrição dos atores	6
2.5.1.2	Descrição dos casos de uso	6
2.6	Visão lógica	10
2.6.1	Modelo de análise	10
2.6.1.1	Visão de Classes Participantes	10
2.6.1.1.1	Caso de uso: Acessar Site	10
2.6.1.1.2	Caso de uso: Fazer Login	10
2.6.1.1.3	Caso de uso: Cadastrar Imóvel	11
2.6.1.1.4	Caso de uso: Aprovar Imóvel	11
2.6.1.1.5	Caso de uso: Atualizar Imóvel	12
2.6.1.1.6	Caso de uso: Remover Imóvel	12
2.6.1.1.7	Caso de uso: Filtrar Imóvel	13
2.6.1.2	Diagramas de Sequência	13
2.6.1.2.1	Caso de uso: Acessar Site	13
2.6.1.2.2	Caso de uso: Fazer Login	14
2.6.1.2.3	Caso de uso: Cadastrar Imóvel	14
2.6.1.2.4	Caso de uso: Aprovar Imóvel	15
2.6.1.2.5	Caso de uso: Atualizar Imóvel	15

2.6.1.2.6	Caso de uso: Remover Imóvel	16
2.6.1.2.7	Caso de uso: Filtrar Imóvel	16
3	REFERENCIAL TEÓRICO	18
3.1	Campus universitário	18
3.2	Tecnologias utilizadas	18
3.2.1	Python	18
3.2.2	Django	19
3.2.3	Material Design	19
3.2.4	Materialize	19
3.2.5	HTML5 e CSS3	20
3.2.6	PostgreSQL	20
3.2.7	[API]Google Maps	21
3.2.8	Ubuntu	22
3.2.9	Visual Studio Code	22
4	IMPLEMENTAÇÃO	24
4.1	Codificação	24
4.1.1	Conexão - API Google Maps	24
4.1.2	Trecho do código - Filtro	24
4.1.3	Script - Marcar imóveis	25
4.1.4	Classe - Tabela para objeto	25
4.1.5	Marcador - Google Maps	26
5	RESULTADOS	28
5.1	Explicação das telas	28
5.1.1	<i>Django Administration</i>	29
5.2	Teste de aceitação	30
6	CONCLUSÃO	34
	REFERÊNCIAS	35
	APÊNDICE A – FORMUÁRIO DE ACEITAÇÃO	36

LISTA DE ABREVIATURAS E SIGLAS

UEMS	Universidade Estadual do Mato Grosso do Sul
UFGD	Universidade Federal da Grande Dourados
RU	Requisito de Usuário
RF	Requisito Funcional
RNF	Requisito Não Funcional
IDE	<i>Integrated Development Environment</i>
SDK	<i>Software Development Kit</i>
UML	Linguagem de Modelagem Unificada
WEB	<i>World Wide Web</i>
API	<i>Application Programming Interface</i>
XML	<i>eXtensible Markup Language</i>
HTML	Linguagem de Marcação de Hipertexto
CSS	<i>Cascading Style Sheets</i>
CDN	<i>Content Delivery Network</i>
SQL	<i>Structured Query Language</i>
SGBD	Sistema Gerenciador de Banco de Dados
JSON	<i>JavaScript Object Notation</i>
XML	<i>Extensible Markup Language</i>

LISTA DE ILUSTRAÇÕES

Figura 1 – Diagrama modelo casos de uso	5
Figura 2 – Visão de Classe Participante: Acessar Site	10
Figura 3 – Visão de Classe Participante: Fazer Login	11
Figura 4 – Visão de Classe Participante: Cadastrar Imóvel	11
Figura 5 – Visão de Classe Participante: Aprovar Imóvel	11
Figura 6 – Visão de Classe Participante: Atualizar Imóvel	12
Figura 7 – Visão de Classe Participante: Remover Imóvel	12
Figura 8 – Visão de Classe Participante: Filtrar Imóvel	13
Figura 9 – Diagrama de Sequência: Acessar Site	13
Figura 10 – Diagrama de Sequência: Fazer Login	14
Figura 11 – Diagrama de Sequência: Cadastrar Imóvel	15
Figura 12 – Diagrama de Sequência: Aprovar Imóvel	15
Figura 13 – Diagrama de Sequência: Atualizar Imóvel	16
Figura 14 – Diagrama de Sequência: Remover Imóvel	16
Figura 15 – Diagrama de Sequência: Filtrar Imóvel	17
Figura 16 – Conexão API	24
Figura 17 – Filtro de imóveis	24
Figura 18 – Marcar imóveis	25
Figura 19 – Classe Model	26
Figura 20 – Marcador	26
Figura 21 – Visualização de imóvel	28
Figura 22 – Cadastrar Imóvel	29
Figura 23 – Administração do site	29
Figura 24 – Aprovação de Imóveis	30
Figura 25 – Permissão de usuários	30
Figura 26 – Pergunta 1	31
Figura 27 – Pergunta 2	31
Figura 28 – Pergunta 3	32
Figura 29 – Pergunta 4	32

1 INTRODUÇÃO

A cidade de Dourados é considerada um polo universitário dentro do estado de Mato Grosso do Sul. Atualmente a cidade possui 5 faculdades sendo duas públicas o que atrai um grande número de estudantes, tanto da cidade quanto da região. Ainda dentro desse contexto, Dourados possui uma cidade universitária situada distante da região central onde estão instaladas a Universidade Estadual Mato Grosso do Sul - UEMS e Universidade Federal da Grande Dourados - UFGD. Existem cursos matutinos, vespertinos, noturnos e integrais movimentando o mercado imobiliário da cidade por conta daqueles estudantes que vêm de fora.

Com o crescimento do mercado corporativo, muitas empresas estão em busca de mais lucro. Tendo em vista que, toda empresa moderna tem se atualizado em relação as novas exigências tecnológicas, muitas tem adotado páginas *web* para oferecer uma forma prática e rápida de usuários e clientes para terem acesso ao seus produtos.

Aplicações executadas diretamente de um celular podem estar conectadas diretamente a internet e obtendo informações atualizadas direto de um servidor confiável. Muitas empresas já oferecem seus serviços para seus clientes como, por exemplo, muitos bancos na qual é possível fazer pagamento, transferências, tirar extrato entre muitas outras opções direto do celular.

O objetivo deste estudo consiste na elaboração de uma página *web* para facilitar a escolha de imóveis onde os acadêmicos, professores, servidores e técnicos da cidade universitária de Dourados ou de outros polos universitários possam ir a morar.

O presente trabalho apresenta o desenvolvimento de uma página *web* para disponibilizar os pontos de moradias universitária na cidade de Dourados.

A página *web* consistirá de duas partes, a primeira é onde os locadores irão colocar as informações dos imóveis(localização, preço e contato) e a outra é onde o locatário irá visualizar essas informações.

O página consiste em duas partes. A primeira parte do módulo web permitirá que o locador faça o cadastro das informações gerais de seu imóvel para locação, como: endereço, tipo do imóvel, preço, etc. A segunda parte é a do locatário onde ele irá consultar as melhores opções de morada que cabem no seu orçamento e gosto.

Com esta página *web* a necessidade dos usuários, poderão ser atendidas, então os mesmos irão beneficiar se na praticidade quanto ao deslocamento até o destino que mais lhe convém, de acordo com a escolha feita.

1.1 Objetivos Gerais

O objetivo geral deste estudo é construir uma página *web* para a locação de imóveis universitários, que estará disponível gratuitamente para acesso de qualquer dispositivo com acesso a internet.

1.1.1 Objetivos

- Elaborar uma página *WEB* em *Python* usando o *Framework* Django voltada para o âmbito universitário.
- Mostrar opções de moradias ofertadas usando a API Google Maps.
- Otimizar o tempo de escolha para a melhor moradia.
- Trazer informações sobre todos os anúncios de moradias em Dourados exceto de imobiliárias .

1.2 Organizações dos capítulos

O trabalho é composto por 6 capítulos. O Capítulo 1 apresenta a introdução e objetivos do trabalho. O Capítulo 2, são apresentadas metodologia e desenvolvimento do projeto. Já o Capítulo 3 e apresenta o referencial teórico utilizado neste projeto. O Capítulo 4 apresenta a implementação do projeto. No Capítulo 5, são apresentadas os resultados obtidos no desenvolvimento do projeto. E, finalmente, no Capítulo 6, são apresentadas as conclusões finais do projeto.

2 METODOLOGIA E DESENVOLVIMENTO

2.1 Introdução

Neste capítulo estaremos apresentando a metodologia aplicada para o desenvolvimento da página WEB.

2.2 Metodologia

O estudo consistiu na criação da página web, Portal Universitário, para atender o público universitário da cidade de Dourados.

O objetivo maior é a disponibilização de uma plataforma onde todos possam acessar aluguéis de: repúblicas, pensionatos, apartamentos, kitnets e casas, facilitando assim a escolha de onde morar.

Para isso, utilizou-se a linguagem de modelagem UML para a documentação de requisitos e arquitetura da página. Após a construção da modelagem, iniciou-se o processo da programação na linguagem *Python* com o framework *Django*. Em seguida serão executados testes de programação e aceitação que serão exibidos ao final no projeto.

2.3 Definição dos Requisitos do Usuário

Nesta seção estão descritos os Requisitos do Usuário (RU), ou sejam, as necessidades levantadas pelo usuário. Foram realizadas entrevistas com os usuários e estudos da documentação utilizada por outros desenvolvedores de páginas similares para formulação dos requisitos aqui descritos.

- RU1: Inserção de imóvel para locação.
- RU2: Remoção do imóvel.
- RU3: Filtrar informações de imóveis.
- RU4: Pesquisa de imóvel.

2.4 Especificação dos Requisitos do Sistema

Os requisitos do sistema descrevem o comportamento externo do sistema e suas restrições operacionais. Os requisitos definidos nas próximas seções delimitam o sistema baseado nos requisitos de usuário.

2.4.1 Requisitos Funcionais

Baseado nos requisitos do usuário esta seção descreve de forma mais detalhada como os requisitos serão desenvolvidos no sistema.

- RF1.** Acessar a página web com o endereço: <http://127.0.0.1:8000/> .Usuário: Classe A, B, C (Requisitos Relacionados: - - -).
- RF2.** Login de locador com os seguintes dados: usuário e senha. Usuário: Classe B (Requisitos Relacionados: RF1).
- RF3.** O locador insere o imóvel com os seguintes dados: titulo, sexo, tipo de imóvel, endereço, valor, contato e detalhes. Usuário: Classe B (Requisitos Relacionados: RF1 e RF2).
- RF4.** Aprovação de imóvel com os seguintes dados: informações do imóvel e campo de aprovação. Usuário: Classe A (Requisitos Relacionados: RF3).
- RF5.** Atualização de informações do imóvel com os seguintes dados: login, senha, imóvel e campo(s) a ser(em) alterado(s). Usuário: Classe A e B (Requisitos Relacionados: RF3).
- RF6.** Remoção do imóvel com os seguintes dados: login, senha e imóvel. Usuário: Classe A e B (Requisitos Relacionados: RF3).
- RF7.** RF7. Filtrando o imóvel para visualização com os seguintes dados: gênero, tipo e preço. Usuário: Classe A B C (Requisitos Relacionados: RF1, RF3 e RF4).

2.4.2 Requisitos Não Funcionais

São aqueles não diretamente relacionados às funções específicas, como as propriedades emergentes, confiabilidade, tempo de resposta, espaço de armazenamento, restrições de um dispositivo de E/S e representações de dados.

- RNF1.** Para ter o imóvel retirado ou alterado é necessário solicitação via e-mail.(Requisitos Relacionados: RF5 e RF6).
- RNF2.** Não é obrigatório filtrar os três campos para efetuar uma pesquisa. (Requisitos Relacionados: RF7).
- RNF3.** Pode selecionar mais de uma opção no mesmo filtro. (Requisitos Relacionados: RF7).
- RNF4.** Não há aprovação de cadastro de locador. (Requisitos Relacionados: RF2).

RNF5. Qualquer usuário tem acesso à página de pesquisa. (Requisitos Relacionados: RF1, RF2 RF7).

RNF6. Imóveis com mensagens inapropriadas ou informações indevidas não serão aprovados. (Requisitos Relacionados: RF4).

2.4.3 Descrição de Usuário

Esses são os usuários classificados pelo sistema nessa etapa de trabalho:

- Usuário A: Administrador do sistema e responsável pela aprovação, atualização e remoção de imóveis.
- Usuário B: Encarregado de inserir imóveis no por portal universitário.
- Usuário C: Todo aquele que acessa a página do portal universitário.

2.5 Visão de Caso de Uso

2.5.1 Modelo de Casos de Uso

Nesta seção será apresentado na Figura 1 o Modelo de Casos de Uso: Diagrama de Casos de Uso, Descrição dos Trabalhadores e as Descrições desses Casos de Uso. O objetivo é demonstrar a estrutura e dinâmica do sistema. Além de definir uma visão comum da organização por clientes, usuários e desenvolvedores

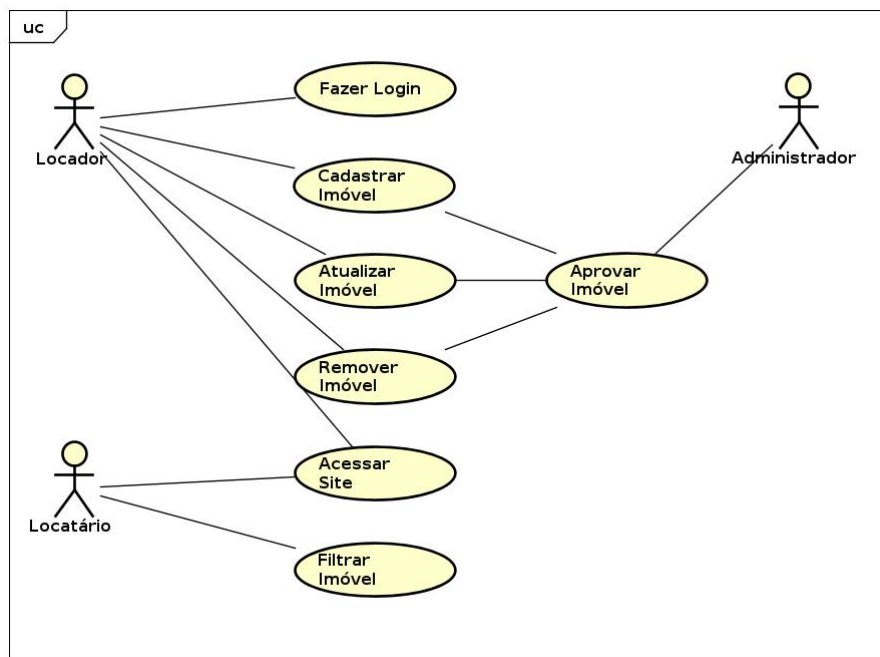


Figura 1 – Diagrama modelo casos de uso

2.5.1.1 Descrição dos atores

Os seguintes atores foram identificados nessa etapa de trabalho:

- **Locador:** Proprietário que está cedendo o imóvel para a locação e será responsável por cadastrar as informações dos imóveis.
- **Locatário:** Pessoa que está tomando o imóvel a título de locação apenas irá consultar as informações da página.
- **Administrador:** Desenvolvedor do portal universitário e responsável pela aprovação/atualização/remoção de imóvel.

2.5.1.2 Descrição dos casos de uso

Caso de uso:	Acessar site
Ator:	Locatário ou Locador
Sumário:	Este caso de uso permite aos atores locatário ou locador acessarem a página web portal universitário.
Pré condição:	Ter acesso à internet.
RF: RF1	RNF: RNF5
Fluxo principal:	
<ol style="list-style-type: none"> 1. O usuário irá acessar a página com o seguinte endereço: http://127.0.0.1:8000/ 2. O sistema abrirá a página inicial. 3. O sistema encerra o caso de uso. 	
Pós condição:	O usuário poderá visualizar todos os imóveis cadastrados.
Relacionamentos:	Não existem relacionamentos associados a este caso de uso.

Caso de uso:	Fazer login
Ator:	Locador
Sumário:	Este caso de uso permite o ator locador logar na página web portal universitário.
Pré condição:	Estar no site portal universitário.
RF: RF2	RNF: RNF4 e RNF5
Fluxo principal:	
<ol style="list-style-type: none"> 1. O locador faz o login com os seguintes dados: login e senha. 2. É liberado o campo “Cadastrar Imóvel”. 3. O sistema encerra o caso de uso. 	
Fluxo Alternativo (1): Login/Senha não encontrado.	
<ol style="list-style-type: none"> 1. O locador irá se cadastrar com as seguintes informações: login e senha. 2. O sistema salvará as informações do novo cadastro. 3. O sistema emite uma mensagem informando usuário cadastrado. 4. O sistema retorna para o passo 2 do cenário principal. 	
Pós condição:	O locador pode cadastrar um imóvel
Relacionamentos:	Não existem relacionamentos associados a este caso de uso.

Caso de uso:	Cadastrar imóvel
Ator:	Locador
Sumário:	Este caso de uso permite o ator locador cadastrar imóveis na página web portal universitário.
Pré condição:	Ter um login válido.
RF: RF3	RNF:
Fluxo principal:	
<ol style="list-style-type: none"> 1. O locador acessa o local de cadastro de imóveis. 2. O locador seleciona o marcador e arrasta até o local desejado. 3. O locador preenche os dados do imóvel com os seguintes dados: título, sexo, tipo de imóvel, endereço, valor, contato e detalhes. 4. O locador confirma o cadastro do imóvel. 5 O sistema encerra o caso de uso. 	
Pós condição:	O imóvel esta sob aprovação do administrador.
Relacionamentos:	Não existem relacionamentos associados a este caso de uso.

Caso de uso:	Aprovar Imóvel
Ator:	Administrador
Sumário:	Este caso de uso permite ao ator Administrador aprovar imóveis na página web portal universitário.
Pré condição:	Ter imóveis para serem aprovados
RF: RF4	RNF:
Fluxo principal:	
<ol style="list-style-type: none"> 1. O administrador acessa a área de administração do site. 2. O administrador acessa a listagem de imóveis. 3. O administrador aprova os imóveis pendentes. 4. O sistema encerra o caso de uso. 	
Pós condição:	O imóvel pode ser visualizado no portal universitário.
Relacionamentos:	Não existem relacionamentos associados a este caso de uso.

Caso de uso:	Atualizar imóvel
Ator:	Locador
Sumário:	Este caso de uso permite ao ator locador requerer a atualização de um imóvel da página web portal universitário.
Pré condição:	Ter um login válido.
RF: RF5	RNF: RNF1
Fluxo principal:	
<ol style="list-style-type: none"> 1. O locador acessa o local de alteração de imóvel. 2. O sistema emite uma mensagem informando o e-mail que ele deve enviar a solicitação de alteração de informação do imóvel. 3. O sistema encerra o caso de uso. 	
Pós condição:	Enviar o e-mail de alteração de informação do imóvel.
Relacionamentos:	Não existem relacionamentos associados a este caso de uso.

Caso de uso:	Remover imóvel
Ator:	Locador
Sumário:	Este caso de uso permite o ator locador requerer a remoção do imóvel da página web portal universitário.
Pré condição:	Ter um login válido.
RF: RF6	RNF: RNF1
Fluxo principal:	
<ol style="list-style-type: none"> 1. O locador acessa o local de alteração de imóvel. 2. O sistema emite uma mensagem informando o e-mail que ele deve enviar a solicitação de exclusão do imóvel. 3. O sistema encerra o caso de uso. 	
Pós condição:	Enviar o e-mail de exclusão do imóvel.
Relacionamentos:	Não existem relacionamentos associados a este caso de uso.

Caso de uso:	Filtro imóvel
Ator:	Locatário e Locador
Sumário:	Este caso de uso permite os usuários locatário e locador fazer um filtro de busca de imóvel da página web portal universitário.
Pré condição:	Estar acessando o portal universitário.
RF: RF7	RNF: RNF2, RNF3 e RNF5
Fluxo principal:	
<ol style="list-style-type: none"> 1. O usuário seleciona o filtro que deseja. 2. O usuário confirma os filtros aplicados. 3. O sistema exibe os imóveis compatíveis com o filtro aplicado. 4. O sistema encerra o caso de uso. 	
Pós condição:	Visualização dos imóveis filtrados
Relacionamentos:	Não existem relacionamentos associados a este caso de uso.

2.6 Visão lógica

2.6.1 Modelo de análise

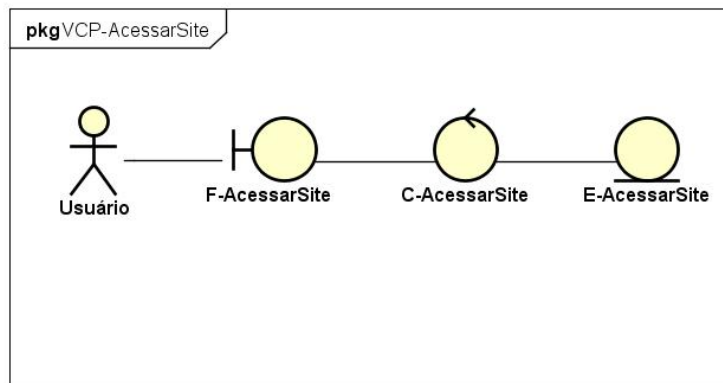
O modelo de análise contém as classes de análise. Ele pode ser um artefato temporário, como no caso em que evolui para um modelo de design, ou pode existir através de parte ou de todo o projeto e, talvez, servindo como uma visão geral conceitual do sistema. Para o modelo de análise será elaborado a Visão de Classes Participantes (VCP) e o Diagrama de Sequência para cada caso de uso identificado na Figura 8.

2.6.1.1 Visão de Classes Participantes

2.6.1.1.1 Caso de uso: Acessar Site

A Figura 2 apresenta a visão de classe participante do caso de uso acessar site. Com as classes de fronteira, de controle de entidade.

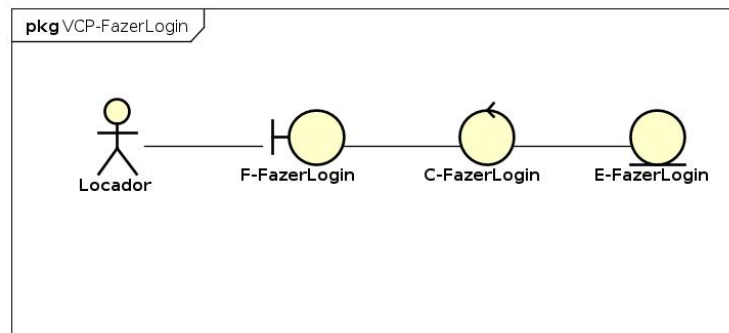
Figura 2 – Visão de Classe Participante: Acessar Site



2.6.1.1.2 Caso de uso: Fazer Login

A Figura 3 apresenta a visão de classe participante do caso de uso cadastrar cliente. Com as classes de fronteira, de controle e entidade.

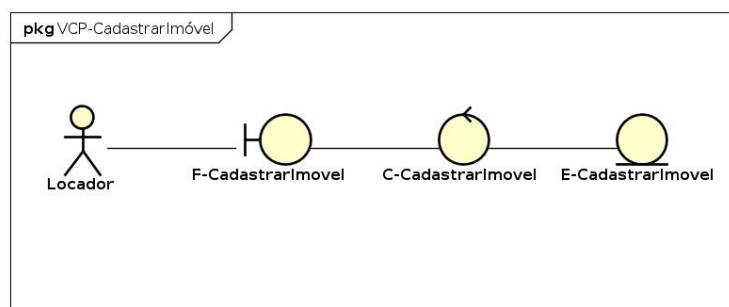
Figura 3 – Visão de Classe Participante: Fazer Login



2.6.1.1.3 Caso de uso: Cadastrar Imóvel

A Figura 4 apresenta a visão de classe participante do caso de uso cadastrar cliente. Com as classes de fronteira, de controle e entidade.

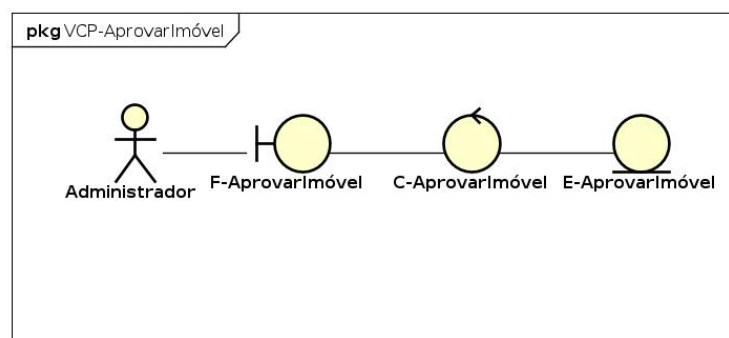
Figura 4 – Visão de Classe Participante: Cadastrar Imóvel



2.6.1.1.4 Caso de uso: Aprovar Imóvel

A Figura 5 apresenta a visão de classe participante do caso de uso cadastrar cliente. Com as classes de fronteira, de controle e entidade.

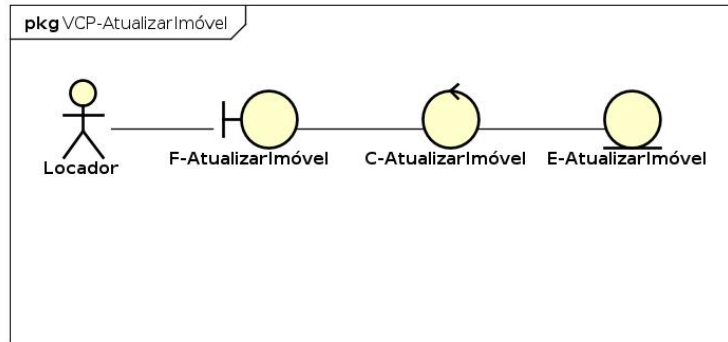
Figura 5 – Visão de Classe Participante: Aprovar Imóvel



2.6.1.1.5 Caso de uso: Atualizar Imóvel

A Figura 6 apresenta a visão de classe participante do caso de uso cadastrar cliente. Com as classes de fronteira, de controle e entidade.

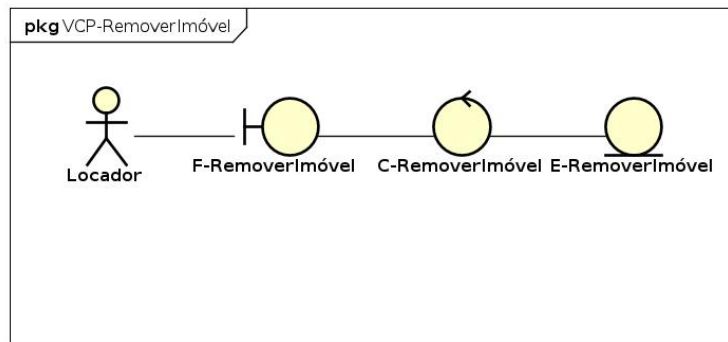
Figura 6 – Visão de Classe Participante: Atualizar Imóvel



2.6.1.1.6 Caso de uso: Remover Imóvel

A Figura 7 apresenta a visão de classe participante do caso de uso cadastrar cliente. Com as classes de fronteira, de controle e entidade.

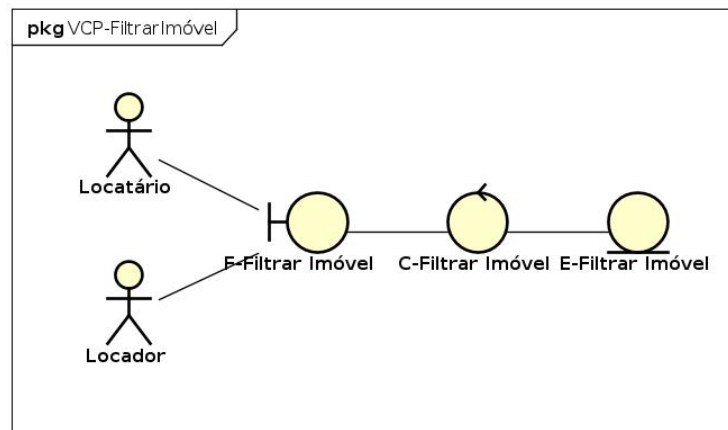
Figura 7 – Visão de Classe Participante: Remover Imóvel



2.6.1.1.7 Caso de uso: Filtrar Imóvel

A Figura 8 apresenta a visão de classe participante do caso de uso cadastrar cliente. Com as classes de fronteira, de controle e entidade.

Figura 8 – Visão de Classe Participante: Filtrar Imóvel

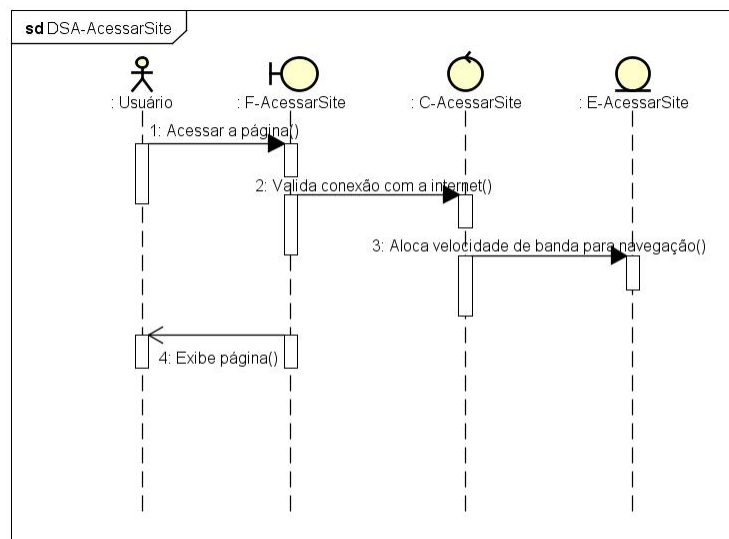


2.6.1.2 Diagramas de Sequência

2.6.1.2.1 Caso de uso: Acessar Site

A Figura 9 apresenta a visão do diagrama de sequência do caso de uso acessar site. Com as classes de fronteira, de controle e entidade.

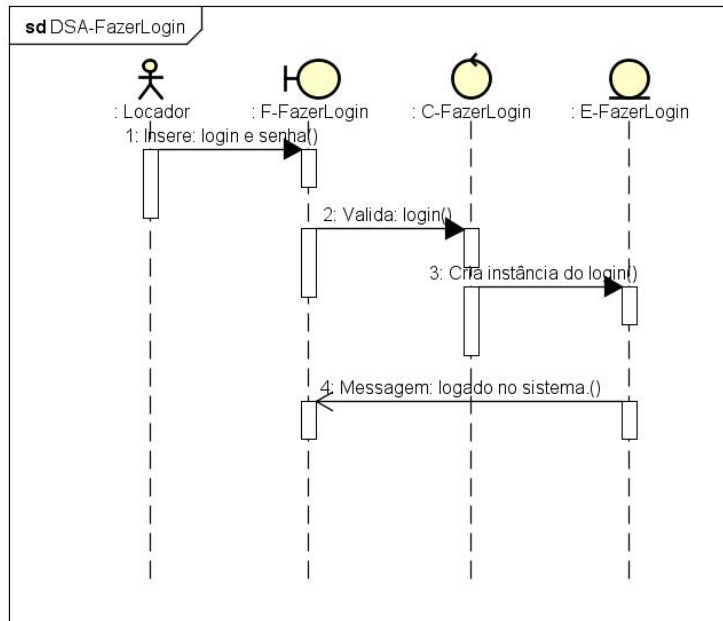
Figura 9 – Diagrama de Sequência: Acessar Site



2.6.1.2.2 Caso de uso: Fazer Login

A Figura 10 apresenta a visão do diagrama de sequência do caso de uso fazer login. Com as classes de fronteira, de controle de entidade.

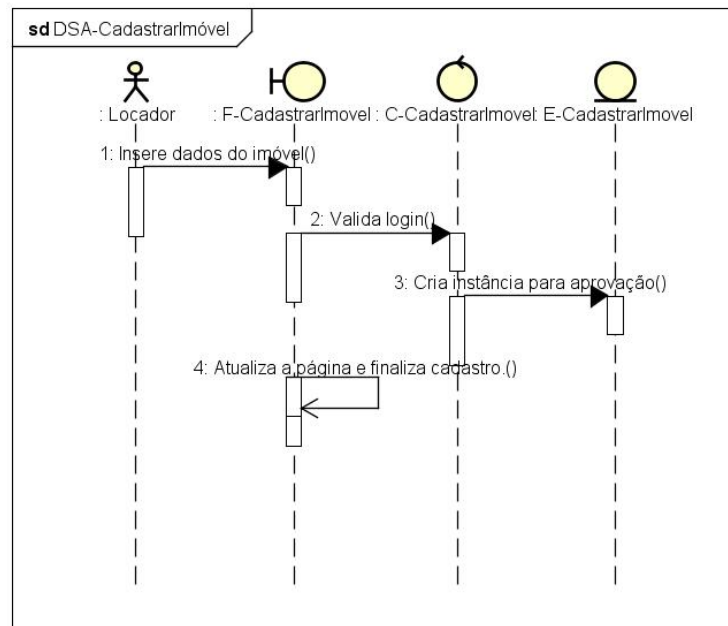
Figura 10 – Diagrama de Sequência: Fazer Login



2.6.1.2.3 Caso de uso: Cadastrar Imóvel

A Figura 11 apresenta a visão do diagrama de sequência do caso de uso cadastrar imóvel. Com as classes de fronteira, de controle de entidade.

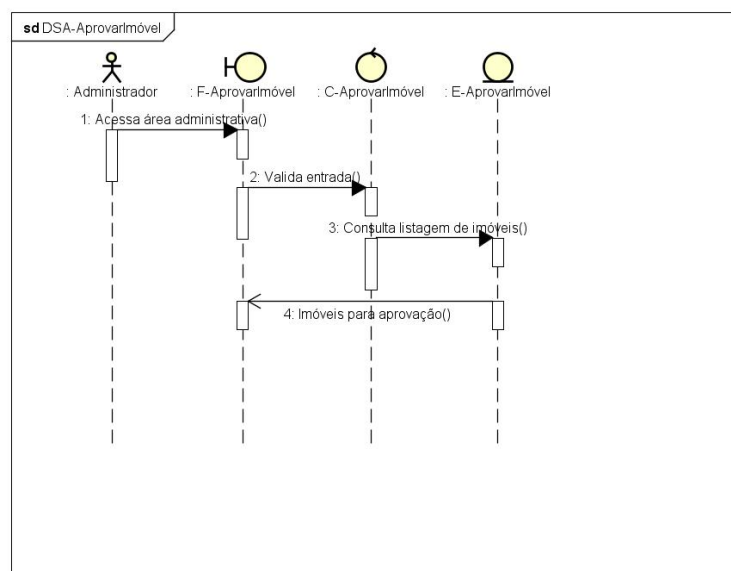
Figura 11 – Diagrama de Sequência: Cadastrar Imóvel



2.6.1.2.4 Caso de uso: Aprovar Imóvel

A Figura 12 apresenta a visão do diagrama de sequência do caso de uso aprovar imóvel. Com as classes de fronteira, de controle de entidade.

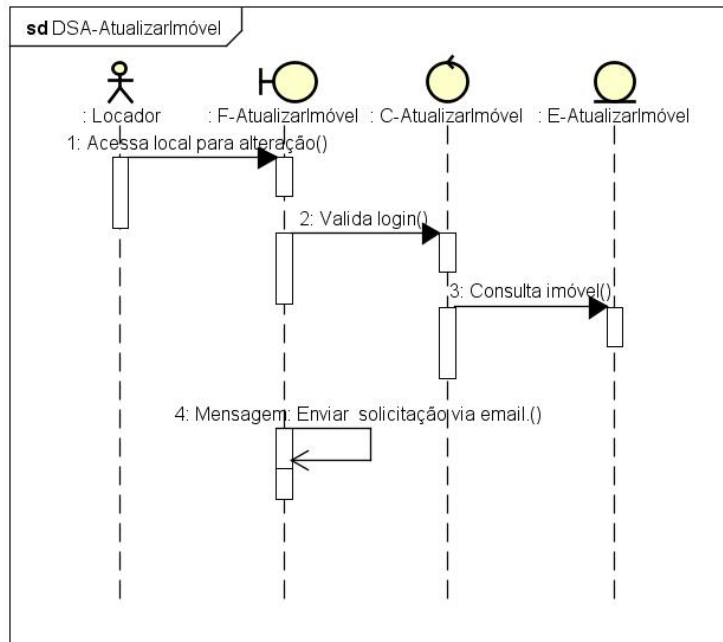
Figura 12 – Diagrama de Sequência: Aprovar Imóvel



2.6.1.2.5 Caso de uso: Atualizar Imóvel

A Figura 13 apresenta a visão do diagrama de sequência do caso de uso atualizar imóvel. Com as classes de fronteira, de controle de entidade.

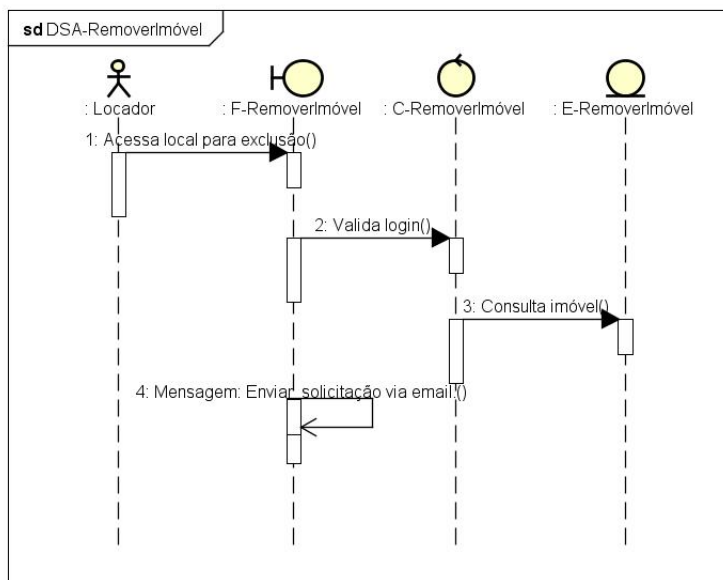
Figura 13 – Diagrama de Sequência: Atualizar Imóvel



2.6.1.2.6 Caso de uso: Remover Imóvel

A Figura 14 apresenta a visão do diagrama de sequência do caso de uso remover imóvel. Com as classes de fronteira, de controle de entidade.

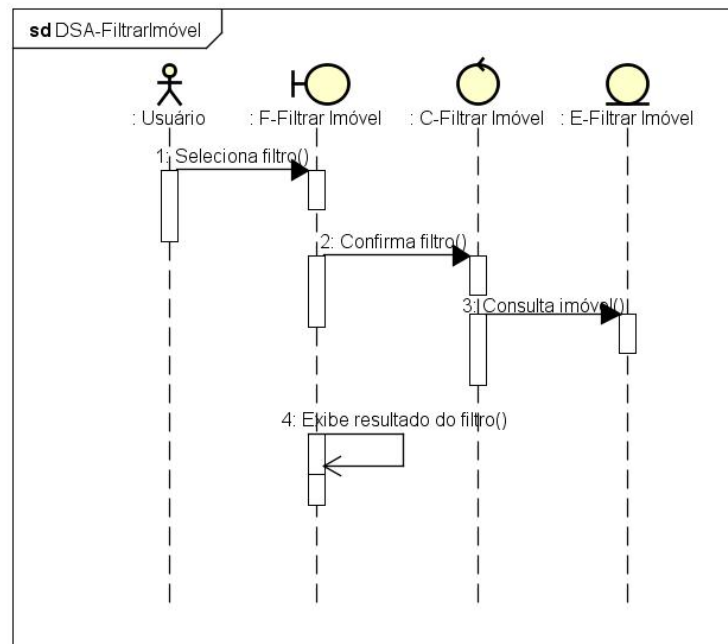
Figura 14 – Diagrama de Sequência: Remover Imóvel



2.6.1.2.7 Caso de uso: Filtrar Imóvel

A Figura 15 apresenta a visão do diagrama de sequência do caso de uso filtrar imóvel. Com as classes de fronteira, de controle de entidade.

Figura 15 – Diagrama de Sequência: Filtrar Imóvel



Neste capítulo de metodologia e desenvolvimento, apresentou-se os casos de uso, diagrama de classe, diagramas de sequencia que compõe o estudo.

3 REFERENCIAL TEÓRICO

Neste capítulo apresentamos o referencial teórico que serviu como base de estudos para o desenvolvimento do trabalho de conclusão de curso. Assuntos como Campus Universitário e opções de moradia foram abordados. Finalmente, foram apresentadas as tecnologias aplicadas para a construção da página WEB Portal Universitário.

3.1 Campus universitário

O Campus universitário é uma área onde se encontra as instalações de uma faculdade podendo ser em um grande centro ou fora dele. Todo o terreno e prédios que compõem a universidade formam o campus universitário. Apesar do nome campus remeter a ideia de uma área verde, não é necessariamente obrigatório possuir essa característica (BUFFA; PINTO, 2009)

3.2 Tecnologias utilizadas

O sistema funciona em uma página web, podendo ser acessado por qualquer dispositivo com acesso a Internet. O processo de desenvolvimento baseou-se nas fases do Processo Unificado e na UML 2.0. Os diagramas serão desenvolvidos na ferramenta Case Astha Community. A implementação ocorreu na linguagem python em usando o Framework Django e com o gerenciador de banco de dados PostgreSQL, e o editor de texto Visual Studio Code, ambos instalados e uma maquina com o sistema operacinal Linux Ubuntu.

3.2.1 Python

Python é uma linguagem de altíssimo nível orientada a objeto, de tipagem dinâmica e forte, interpretada.

Em suas características o *Python* tem uma sintaxe clara e concisa que favorece a legibilidade do código-fonte, tornando a linguagem mais produtiva.

A linguagem inclui diversas estruturas de alto nível (listas, dicionários, data/hora, complexos e outras) e uma vasta coleção de módulos prontos para uso, além de *frameworks* de terceiros que podem ser adicionados. Também inclui recursos encontrados em outras linguagens modernas, tais como geradores, introspecção, persistência, metaclasses e unidades de teste. Multiparadigma, a linguagem suporta programação modular e funcional, além da orientação a objetos. Mesmo os tipos básicos no *Python* são objetos. A linguagem é interpretada através de bytecode pela máquina virtual *Python*, tornando o código portátil.

Com isso é possível compilar aplicações em uma plataforma e rodar em outros sistemas ou executar direto do código-fonte.

Python é um software de código aberto (com licença compatível com *General Public License[CPL]*, porém menos restritiva, permitindo que o *Python* seja inclusive incorporado em produtos proprietários). A especificação da linguagem é mantida pela *Python Software Foundation(PSF)*(BORGES, 2014)

3.2.2 Django

Django é um *framework* para aplicações *web* gratuito e de código aberto, escrito em *Python*. Um *web framework* é um conjunto de componentes que ajuda você a desenvolver sites de forma mais rápida e fácil.

Quando você está construindo um site, sempre precisa de um conjunto similar de componentes: uma maneira de lidar com a autenticação do usuário (inscrever-se, fazer login, fazer logout), um painel de gerenciamento para o seu site, formulários, uma forma de upload de arquivos, etc.

Felizmente, outras pessoas perceberam muito tempo atrás que desenvolvedores *web* enfrentam problemas similares quando estão construindo um novo site, então elas se juntaram e criaram *frameworks* (*Django* é um deles) que te dão componentes já prontos para usar.

Frameworks existem para evitar que você tenha que reinventar a roda e para ajudar a aliviar parte do trabalho extra quando você está construindo um novo site. (BORGES, 2014)

3.2.3 Material Design

O Material Design é um guia abrangente para o projeto visual, de movimento e de interação para diversas plataformas e dispositivos”(DEVELOPERS, 2017) O material fornece estilo atualizado para seus aplicativos, possibilitando definir a paleta de cores e as animações para as transições de atividades, além de fornecer novos widgets(componente de uma interface gráfica) para exibir cartões e listas com estilos e animação, também permitem criar animações personalizadas.(DEVELOPERS, 2017).

3.2.4 Materialize

O Materialize é um Framework desenvolvido para facilitar e agilizar o desenvolvimento quando o assunto é Material Design. Além de ágil, possui uma infinidade de vantagens que permitem que na maioria das vezes dispensam o uso de outros frameworks para complementá-lo.

Para baixar os arquivos do Materialize foi utilizado o seguinte site <https://materializecss.com/>, nele você verá uma documentação explicando como fazer o download dos arquivos e todas

as maneiras que você tem para fazer isso, ou também pode usar a CDN dele como geralmente muitos fazem com o JQUERY(biblioteca de funções JavaScript) Os arquivos são bem distribuídos e não diferem dos demais Frameworks que existem.

Uma das principais vantagens é o fato de ser muito similar ao Material Design desenvolvido pela google. Outra vantagem que podemos observar são os amplos recursos de JQuery que ele possui como: Slider, Lightbox, Captions, Modais, Transitions, Waves entre muitos outros.

3.2.5 HTML5 e CSS3

“HTML (HyperText Markup Language - Linguagem de Marcação de HiperTexto) é o mais básico construtor de blocos da *web*”(NETWORK, 2017b). HTML define o conteúdo que a página *web* irá exibir . Existem também outras linguagens que podem ser usadas para implementar a interface gráfica e outras funcionalidade . “ "HyperText"(HiperTexto) refere-se a links que conectam uma página a outra, seja dentro do *website* ou entre *websites*”(NETWORK, 2017b). Links estão ligando um conteúdo a outro na internet o que a torna de extrema importância. A marcação feita pelo HTML engloba texto, imagens e muitos conteúdos. (SILVA, 2008)

“CSS (Cascading Style Sheets) é uma linguagem de estilo usada para descrever a apresentação de um documento escrito em HTML ou em XML”(NETWORK, 2017a). O objetivo do CSS é descrever como serão mostrados os elementos na tela. É também uma das linguagens mais conhecidas na *open web*.(SILVA, 2008)

3.2.6 PostgreSQL

O PostgreSQL é um sistema de gerenciamento de banco de dados objeto-relacional (SGBDOR) baseado no POSTGRES Versão 4.2 desenvolvido pelo Departamento de Ciência da Computação da Universidade da Califórnia em Berkeley. O POSTGRES foi pioneiro em vários conceitos que somente se tornaram disponíveis muito mais tarde em alguns sistemas de banco de dados comerciais.

O PostgreSQL é um descendente de código fonte aberto deste código original de Berkeley, que suporta grande parte do padrão SQL e oferece muitas funcionalidades modernas, como:

- comandos complexos
- chaves estrangeiras
- gatilhos
- visões
- integridade transacional

- controle de simultaneidade multiversão

Além disso, o PostgreSQL pode ser ampliado pelo usuário de muitas maneiras como, por exemplo, adicionando novos

- tipos de dado
- funções
- operadores
- funções de agregação
- métodos de índice
- linguagens procedurais

Devido à sua licença liberal, o PostgreSQL pode ser utilizado, modificado e distribuído por qualquer pessoa para qualquer finalidade, seja particular, comercial ou acadêmica, livre de encargos.(MOMJIAN, 2000)

3.2.7 [API]Google Maps

É um serviço público e gratuito que qualquer pessoa pode usar em seus sites e aplicações. Desde que o usuário final não seja cobrado, pode usar este serviço, no caso de uso comercial existe a versão paga da API, mas não é o caso para este trabalho que visa apenas o estudo e compreensão da mesma.(DEVMEDIA, 2018)

A API do Google Maps oferece mapas como imagens ou elementos interativos no qual você define um estilo para eles com marcadores, linhas, cores, polígonos e imagens de acordo com sua marca.

Na verdade o Google Maps possui várias API's que podem ser incorporadas ao site/aplicação dependendo de cada caso. Vejamos na tabela abaixo essas API's e suas descrições retiradas do site oficial da API.

API	Descrição
Google Maps JavaScript API	Incorpore um mapa do Google em sua página da <i>web</i> usando JavaScript. Manipule o mapa e adicione conteúdo com a ajuda de vários serviços.
Google Maps API for Flash	Use essa API <i>ActionScript</i> para incorporar um mapa do Google na sua página da <i>web</i> ou aplicativo baseado em Flash. Manipule o mapa em três dimensões e adicione conteúdo com a ajuda de vários serviços.
Google Earth API.	Incorpore um verdadeiro globo digital em 3D à sua página da <i>web</i> . Leve os seus visitantes a qualquer lugar da Terra (até mesmo nas profundezas dos oceanos) sem tirá-los de sua página da <i>web</i> .
Google Static Maps API	Incorpore uma imagem simples e rápida do Google Maps em sua página da <i>web</i> ou site para celular sem precisar de códigos <i>JavaScript</i> ou qualquer carregamento dinâmico de página.
Serviços da <i>web</i>	Use solicitações de URL para acessar informações de geocodificação, rotas, elevação e lugares dos aplicativos cliente e manipule os resultados em <i>JSON</i> ou <i>XML</i> .
Google Maps Data API	Visualize, armazene e atualize dados de mapa por meio de feeds da Google Data API, usado um modelo de elementos (marcadores, linhas e formas) e coleções de elementos.

3.2.8 Ubuntu

Ubuntu é um sistema operacional de código aberto, construído a partir do núcleo Linux, baseado no Debian e utiliza GNOME como ambiente de desktop de sua mais recente versão com suporte de longo prazo(LTS). É desenvolvido pela Canonical Ltd.

Geralmente é executado em computadores pessoais e também é popular em servidores de rede, geralmente executando a versão Ubuntu Server, com recursos de classe empresarial. O Ubuntu também está disponível para tablets e smartphones, com a edição Ubuntu Touch.

A proposta do Ubuntu é oferecer um sistema que qualquer pessoa possa utilizar sem dificuldades, independentemente de nacionalidade, nível de conhecimento ou limitações físicas. O sistema deve ser constituído principalmente por software livre e deve também ser isento de qualquer taxa.(CANONICAL, 2004)

3.2.9 Visual Studio Code

O Visual Studio Code é um editor de código-fonte desenvolvido pela Microsoft para Windows, Linux e macOS. Ele inclui suporte para depuração, controle Git incorporado,

realce de sintaxe, complementação inteligente de código, snippets e refatoração de código. Ele também é customizável, fazendo com que os usuários possam mudar o tema do editor, teclas de atalho e preferências. Ele é um software livre e de código aberto, apesar do download oficial estar sob uma licença proprietária.(MICROSOFT, 2018)

Neste capítulo vimos um contexto sobre Campus Universitário. Apresentou-se também todas as linguagens e ferramentas que serão utilizadas no desenvolvimento do projeto.

4 IMPLEMENTAÇÃO

Neste capítulo estaremos mostrando um pouco do desenvolvimento do portal web, sua codificação, scripts e classes, e também mostrarei algumas telas e trechos do código.

4.1 Codificação

4.1.1 Conexão - API Google Mapas

Na Figura 16 temos a conexão com API do Google Mapas, na linha 95 é referenciado o link do mapa, na linha 96 e pego os pontos de latitude e longitude para setar o quadrante da cidade no caso Dourados e na linha 97 e definido o zoom em relação ao mapa.

Figura 16 – Conexão API

```
90 <script src="https://maps.googleapis.com/maps/api/js?key=AIzaSyDfTJcdAY1POyEs1-Wpwmklj9nE44i5THA&libraries=geometry"></script>
91 <script>
92   var map;
93   var imobiles = {{ imobiles| safe}};
94   function initMap() {
95     map = new google.maps.Map(document.getElementById('map'), {
96       center: { lat: -22.231291451905935, lng: -54.82970655663214 },
97       zoom: 13
98     });
99   }
```

4.1.2 Trecho do código - Filtro

Na Figura 17 temos um trecho das condições que trata os filtros de imóveis.

Figura 17 – Filtro de imóveis

```
53     if initial_value:
54         object_list = object_list.filter(value__gte=initial_value)
55     if final_value:
56         object_list = object_list.filter(value__lte=final_value)
57     if genders:
58         object_list = object_list.filter(gender__in=genders)
59     if kinds:
60         object_list = object_list.filter(kind__in=kinds)
61     return object_list
```

4.1.3 Script - Marcar imóveis

Na Figura 18 temos o Script do Django que permite o usuário marcar o imóvel no mapa na hora de cadastrar, ele captura os pontos de latitude e longitude, podemos ver a sua representação no capítulo seguinte na Figura 22.

Figura 18 – Marcar imóveis

```
161 <script>
162   var map;
163   var mapCenter = { lat: -22.231291451905935, lng: -54.82970655663214 };
164   {% if form.longitude.value and form.latitude.value %}
165     var markerPosition = { lat: {{ form.latitude.value }}, lng: {{ form.longitude.value }} };
166   {% else %}
167     var markerPosition = { lat: -22.231291451905935, lng: -54.82970655663214 };
168   {% endif %}
169   function initMap() {
170     map = new google.maps.Map(document.getElementById('map'), {
171       center: mapCenter,
172       zoom: 13
173     });
174   }
175   initMap();
176   var marker = new google.maps.Marker({
177     position: markerPosition,
178     map: map,
179     draggable: true,
180   });
181   marker.addListener('dragend', function (a, b, c) {
182     document.getElementById('latitude').value = marker.position.lat();
183     document.getElementById('longitude').value = marker.position.lng();
184   })
185 </script>
```

4.1.4 Classe - Tabela para objeto

Na Figura 19 temos o modelo de dados onde é feita a conversão da tabela do banco de dados para um objeto python/django.

Figura 19 – Classe Model

```

19  class Immobile(models.Model):
20      user = models.ForeignKey(User, on_delete=models.CASCADE, verbose_name='Usuário')
21      title = models.CharField(max_length=200, verbose_name='Título')
22      gender = models.CharField(max_length=2, choices=GENDERS, verbose_name='Sexo')
23      kind = models.CharField(max_length=3, choices=KINDS, verbose_name='Tipo de imóvel')
24      address = models.CharField(max_length=200, verbose_name='Endereço')
25      value = models.FloatField(verbose_name='Valor')
26      contact = models.TextField(verbose_name='Contato')
27      details = models.TextField(verbose_name='Detalhes')
28      latitude = models.CharField(max_length=200)
29      longitude = models.CharField(max_length=200)
30      approved = models.BooleanField(default=False, verbose_name='Aprovado')
31

```

4.1.5 Marcador - Google Maps

Na Figura 20 temos o trecho de código que marca os pontos no mapa e desenha o marcador, podemos ver a sua representação no capítulo seguinte na Figura 22.

Figura 20 – Marcador

```

101  var infowindow = new google.maps.InfoWindow({
102  });
103  var marker
104  immovebles.map(function (immobile) {
105      marker = new google.maps.Marker({
106          position: { lat: parseFloat(immobile.latitude), lng: parseFloat(immobile.longitude) },
107          map: map,
108          title: immobile.title,
109          ...immobile
110      });
111      marker.addListener('click', function () {
112          infowindow.setContent(`
113              <h6>${this.title}</h6>
114              <p>
115                  Genero: ${this.gender}<br>
116                  Tipo: ${this.kind}<br>
117                  Valor: R$ ${this.value}<br>
118              </p>
119              <p>
120                  Detalhes: ${this.details}<br>
121              </p>
122              <p>
123                  Contato: ${this.contact}<br>
124              </p>
125              `)
126          infowindow.open(map, this)
127      })
128  })

```

Foi mostrado neste capítulo pequenos trechos de código, dentre eles, scripts, funções, condições e classes que compõe uma certa parcela no desenvolvimento do trabalho.

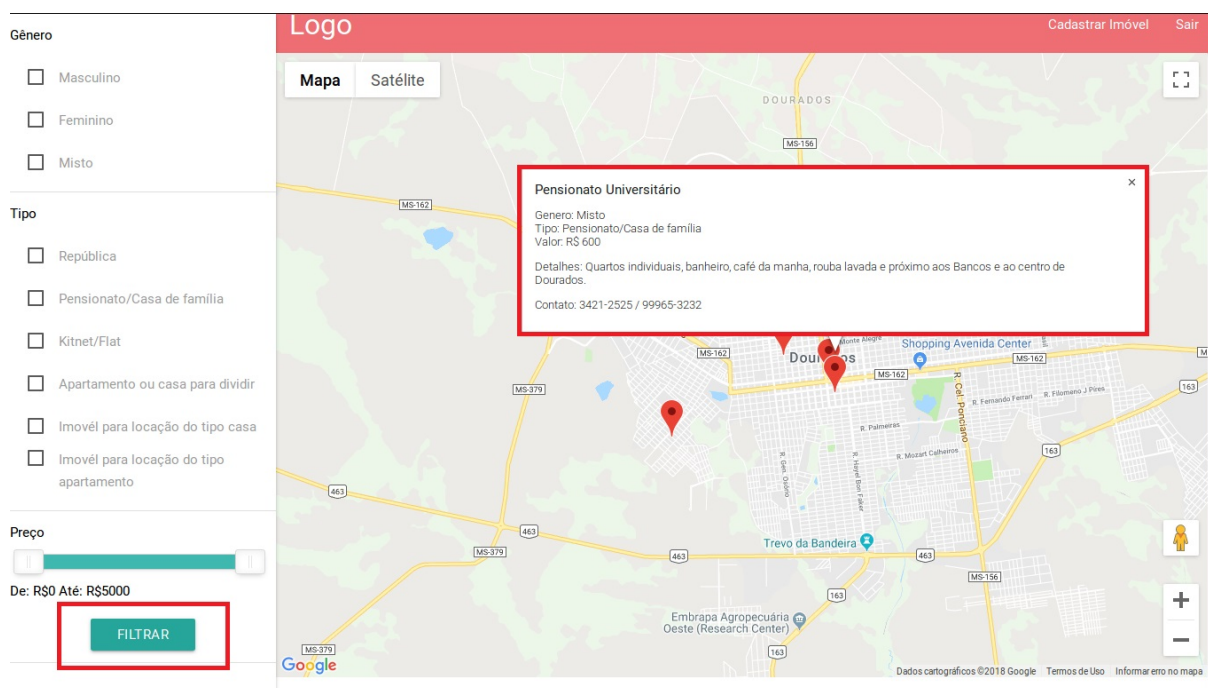
5 RESULTADOS

Nesse capítulo são apresentados os resultados referentes ao trabalho e também serão demonstrado as telas do sistema com suas descrições de funcionalidades. Resultados de teste de aceitação serão exibidos em formato compilado baseado nos relatórios avaliativos aplicados ao publico conforme descrito em anexo.

5.1 Explicação das telas

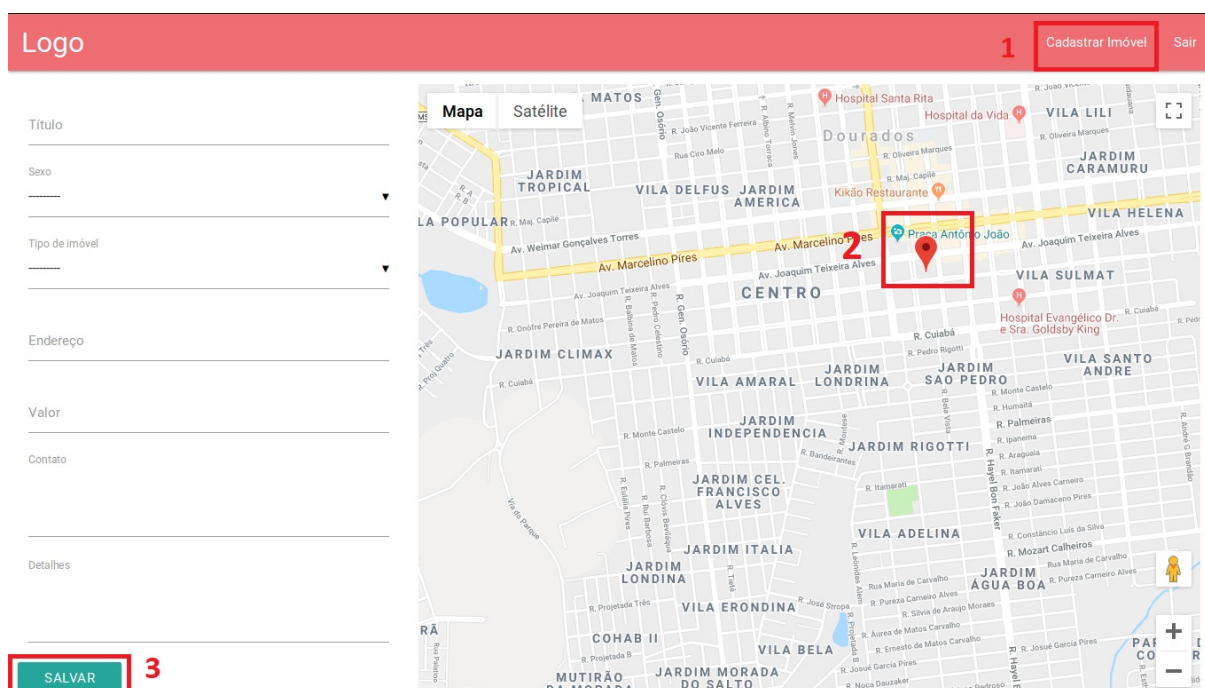
Na Figura 21 podemos observar os marcadores no Google mapas e quando selecionado gera uma caixa flutuante com as informações do imóvel(titulo, gênero, tipo, valor, detalhes e contato). No lado esquerdo da tela estão os filtros de buscar.

Figura 21 – Visualização de imóvel



Na Figura 22 temos o passo a passo para a inserção de um novo imóvel para locação, com o usuário já logado no sistema ele irá clicar no botão cadastrar imóvel, logo posicionar o marcador no local desejável, preencher obrigatoriamente todos os campos(titulo, sexo, tipo do imóvel, endereço, valor, contato e detalhes), e por fim clicar no botão salvar.

Figura 22 – Cadastrar Imóvel



5.1.1 Django Administration

Painel de administração do site: Administração do site, Figura 23, somente o administrador tem acesso a essa área.

Figura 23 – Administração do site



Painel de administração do site: Imóveis, Figura 24

Vemos no quadro o usuário que inseriu o imóvel e se está aprovado ou não.

Figura 24 – Aprovação de Imóveis

Django administration

Home · Immobiles · Immobiles

Select immobile to change

Search

Action: [-----] Go 0 of 11 selected

TÍTULO	SEXO	TIPO DE IMÓVEL	USUÁRIO	APROVADO
<input type="checkbox"/> teste casa 5	Misto	Apartamento ou casa para dividir	admin	✘
<input type="checkbox"/> Apartamento Só vem	Masculino	Imóvel para locação do tipo apartamento	guilherme	✔

No primeiro quadro vemos os usuários já cadastrados no sistema, e no segundo a permissão que é dividida em quem é administrador e quem não é.

Figura 25 – Permissão de usuários

Django administration

Home · Authentication and Authorization · Users

Select user to change

Search

Action: [-----] Go 0 of 4 selected

USERNAME	EMAIL ADDRESS	FIRST NAME	LAST NAME	STAFF STATUS
<input type="checkbox"/> admin				✔
<input type="checkbox"/> clovis				✘
<input type="checkbox"/> guilherme				✘
<input type="checkbox"/> joao				✘

4 users

5.2 Teste de aceitação

No período de 01/06/2018 a 30/09/2018 foi elaborado uma pesquisa utilizando redes sociais, com um formulário constituído por quatro perguntas, que esteve disponível na plataforma do google.

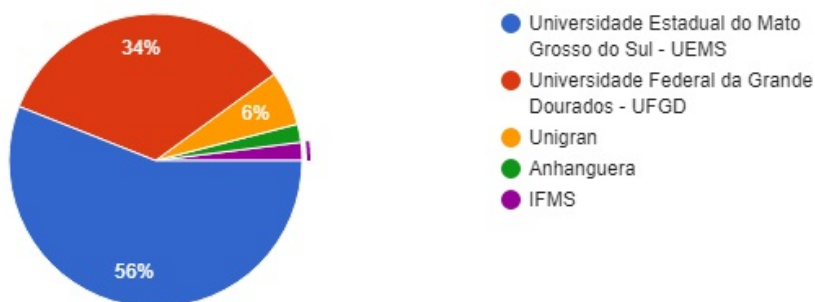
O intuito foi facilitar, através de uma página web, a escolha de moradia dos acadêmicos, professores, servidores e técnicos da região de Dourados, cidade Universitária.

Pergunta, Figura 26

Figura 26 – Pergunta 1

Qual Universidade estuda/trabalha ?

50 respostas



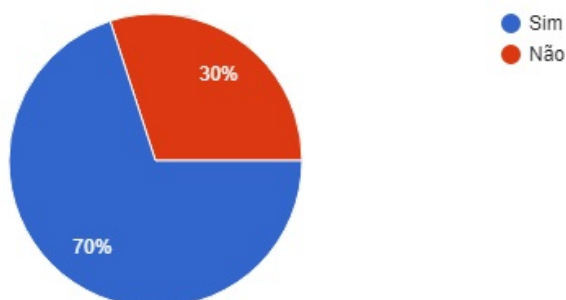
- Universidade Estadual do Mato Grosso do Sul - UEMS: 56%
- Universidade Federal da Grande Dourados - UFGD: 34%
- Universidade da Grande Dourados: Unigran: 6%
- Anhanguera: 2%
- Instituto Federal do Mato Grosso do Sul - IFMS: 2%

Pergunta, Figura 27

Figura 27 – Pergunta 2

Reside em Dourados?

50 respostas



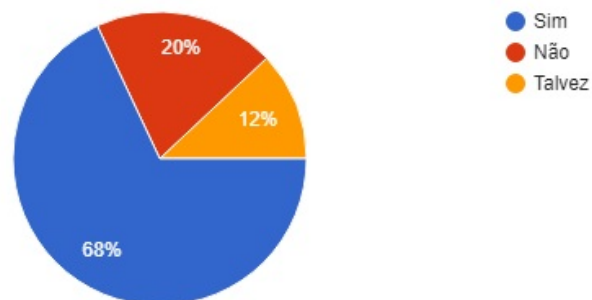
- SIM: 70%
- NÃO: 30%

Pergunta, Figura 28

Figura 28 – Pergunta 3

Teve dificuldades para encontrar moradia ?

50 respostas



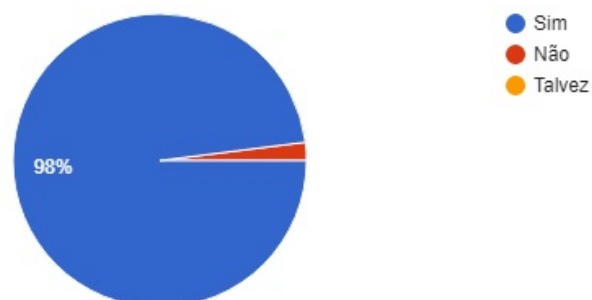
- SIM: 68%
- NÃO: 20%
- TALVEZ: 12%

Pergunta, Figura 29

Figura 29 – Pergunta 4

Se tivesse uma pagina web(site) em que pudesse encontrar opções de moradia universitária em Dourados você aprovaria ?

50 respostas



- SIM: 98%
- NÃO: 2%
- TALVEZ: 0%

Neste capítulo foi mostrado as telas de uso do site que compõe o estudo e as respostas do formulário de aceitação que foi aplicado e que esta nos apêndices.

6 CONCLUSÃO

Este trabalho, de modo geral, se propôs a abrir um novo horizonte a ser explorado pelo portal universitário *web*. O objetivo de desenvolver funcionalidades geográficas ao projeto foi alcançado. Atingir este objetivo não foi uma tarefa simples, pois, exigiu muito trabalho e diversos cuidados para que, o resultado gerado possa ser muito bem aproveitado no futuro.

Com o uso da API do Google maps foram gerados pontos no mapa que possibilitam uma visão alternativa das informações de moradias na página web portal universitário. Para que pudesse ser possível alcançar este objetivo, gastou-se uma parcela do tempo considerável para compreender a dimensão e necessidades da área acadêmica. Os resultados obtidos foram apresentados e visualizados como uma forma de aplicação da teoria na página portal universitário. No processo de conquistar os objetivos definidos neste trabalho, houveram diversas descobertas intelectuais, até o momento, em que foi possível transpor para a prática a ideia inicial. As descobertas e novos aprendizados gerados por esta atividade, agregam valor ao profissional.

No decorrer do desenvolvimento deste trabalho, diversos desafios apareceram, como, compreender uma nova linguagem de programação *Python* e um *framework Django*, sendo que este é um tema que não está no curriculum do curso de Ciência da Computação, dificultando ainda mais o trabalho. Os outros desafios estavam relacionados a utilização das tecnologias para a execução do trabalho. Em relação as tecnologias empregadas, a facilidade em fazer com que estas operassem juntas, foi um dos grandes marcos, fazendo este trabalho seguir um fluxo mais tranquilo no desenvolvimento.

Os testes de aceitação apresentados no trabalho somaram para as ciências envolvidas, já as áreas de desenvolvimento *web*, georreferenciamento e análise de necessidade, foram mais um exemplo de aplicação prática, se tornando, um ponto positivo para a comunidade acadêmica.

Para projetos futuros, o grande objetivo a ser atingido é conseguir utilizar o que foi proposto e construído neste trabalho, para compor os outros dois módulos do portal universitário *web*, o restaurantes e eventos, ambos voltados para o público universitário, essa é uma proposta minha, o autor.

Por fim, este trabalho proporcionou diversas aprendizagens para o meu curriculum, enquanto acadêmico, gerando também, experiência para meu futuro profissional. Com essa pesquisa pude aliar teoria e prática que é base do nosso curso e universidade, que anseia pela pesquisa e extensão.

REFERÊNCIAS

- BORGES, L. E. *PYTHON para desenvolvedores*. Novatec. [S.l.]: Novatec, 2014.
- BUFFA, E.; PINTO, G. A. *Arquitetura e educação: campus universitários Brasileiros*. [S.l.]: EdUFSCar, 2009.
- CANONICAL. *The story of Ubuntu*. 2004. Disponível em: <<https://pt.wikipedia.org/wiki/Ubuntu>,<https://www.ubuntu.com/about>>. Acesso em: 03-09-2018.
- DEVELOPERS, A. *Conheça o Android Studio*. 2017. Disponível em: <<https://developer.android.com/studio/intro/index.html?hl=pt-br>>. Acesso em: 07-06-2018.
- DEVMEDIA. *O que é a Google Maps API?* 2018. Disponível em: <<https://www.devmedia.com.br/introducao-a-google-maps-api/26967>>. Acesso em: 05-04-2018.
- MICROSOFT. *MICROSOFT SOFTWARE LICENSE TERMS*. 2018. Disponível em: <<https://code.visualstudio.com/license>>. Acesso em: 03-09-2018.
- MOMJIAN, B. *PostgreSQL: Introduction and Concepts*. 2000. Disponível em: <pgdocptbr.sourceforge.net/pg80/index.html>. Acesso em: 03-09-2018.
- NETWORK, M. D. *CSS*. 2017. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/CSS>>. Acesso em: 13-06-2018.
- NETWORK, M. D. *HTML*. 2017. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/HTML>>. Acesso em: 08-06-2018.
- SILVA, M. S. *Criando Sites com HTML: Sites de Alta Qualidade com HTML e CSS*. 0. ed. [S.l.]: NOVATEC, 2008.

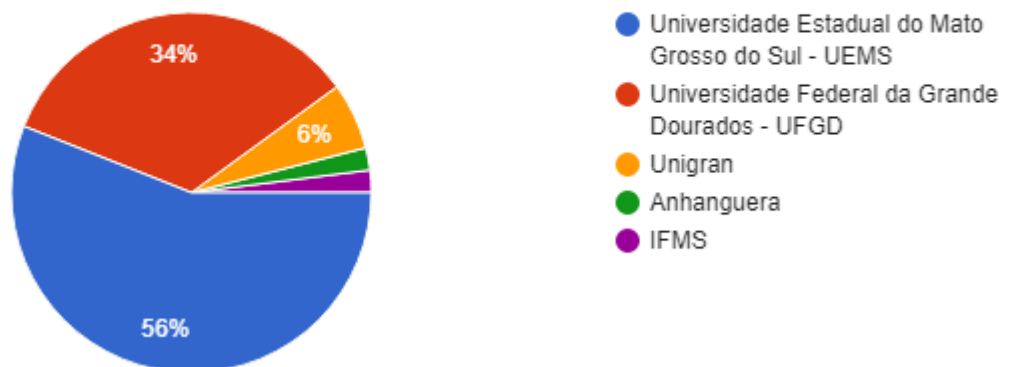
APÊNDICE A — FORMUÁRIO DE ACEITAÇÃO

Portal Universitário WEB

A pesquisa de campo a seguir, tem a finalidade de estudar a demanda de universitários que aprovam ou desaprovam o projeto de conclusão de curso, que dessa forma, visa desenvolver uma página web (site) para facilitar a escolha de onde os acadêmicos, professores, servidores e técnicos de Dourados possam morar.

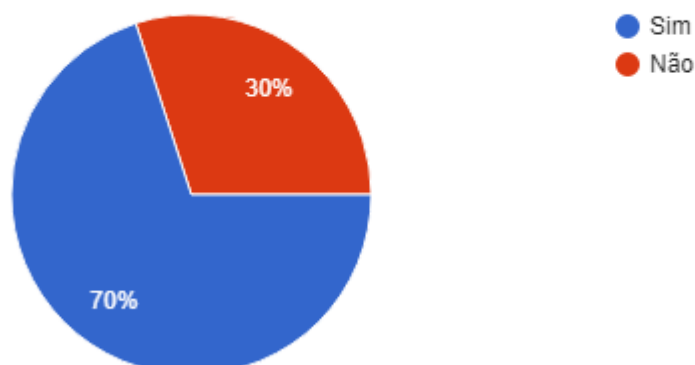
Qual Universidade estuda/trabalha ?

50 respostas



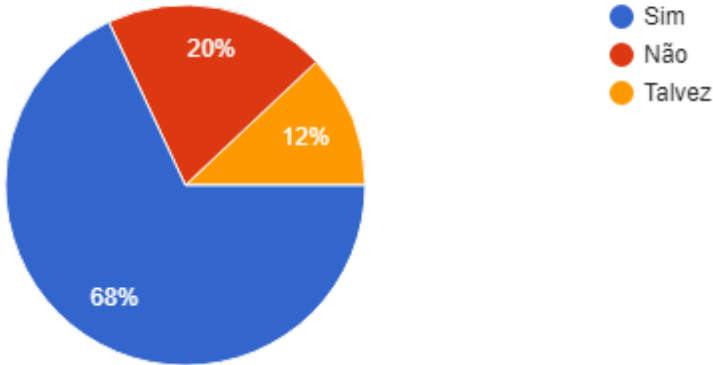
Reside em Dourados?

50 respostas



Teve dificuldades para encontrar moradia ?

50 respostas



Se tivesse uma página web(site) em que pudesse encontrar opções de moradia universitária em Dourados você aprovaria ?

50 respostas

