

Universidade Federal de Uberlândia
Faculdade de Computação
Bacharelado em Sistemas de Informação

Heitor Henrique Nunes

Desenvolvimento de um Portal de Anúncios
Direcionado a Imóveis

Heitor Henrique Nunes

**Desenvolvimento de um Portal de Anúncios
Direcionado a Imóveis**

Trabalho de Conclusão de Curso apresentado
à Faculdade de Computação da Universidade
Federal de Uberlândia como requisito exigido
parcial à obtenção do grau de Bacharel em
Sistemas de Informação.

Área de concentração: Sistemas de Informação

Orientador: Daniel Antônio Furtado

Uberlândia
2019

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE COMPUTAÇÃO
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

Os abaixo assinados, por meio deste, certificam que leram e recomendam para a Faculdade de Computação da Universidade Federal de Uberlândia a aceitação do Trabalho de Conclusão de Curso intitulado “**Desenvolvimento de um Portal de Anúncios Direcionado a Imóveis**” por **Heitor Henrique Nunes** como requisito exigido parcial à obtenção do grau de **Bacharel em Sistemas de Informação**.

Uberlândia, 9 de Julho de 2019

Orientador: _____

Prof. Dr. Daniel Antônio Furtado
Universidade Federal de Uberlândia

Banca Examinadora:

Prof. Dr. André Ricardo Backes
Universidade Federal de Uberlândia

Prof^a. Dr^a. Maria Adriana Vidigal de Lima
Universidade Federal de Uberlândia

Gostaria de dedicar este trabalho de conclusão de curso a todos que estiveram comigo nesta caminhada. Àqueles que me ensinaram, apoiaram, incentivaram e me guiaram.

Agradecimentos

Agradeço a todos meus familiares e namorada pela compreensão nas ausências em finais de semanas, feriados, datas importantes e ao apoio incondicional. Em especial ao meu primo Rodolfo, quem me apresentou o curso e não mediu esforços para que eu me entusiasmasse e iniciasse minha carreira na área.

Aos meus amigos que fiz na graduação – especialmente ao Steffan - agradeço pelo incentivo, motivação, finais de semana estudando, pelas piadas, pelos jogos, corujões e até algumas saídas para bares ou rodízios. Sei que muitos seguirão caminhos diferentes do meu, mas tenho certeza que alguns serão para a vida toda.

Sou grato também ao meu orientador, que me auxiliou com dicas e sugestões ao longo do trabalho, assim como em algumas disciplinas da graduação. A todos os professores que de alguma maneira me proporcionaram o conhecimento, tanto de maneira acadêmica quanto lições para o resto da minha vida, muito obrigado!

*“A confiança não vem do ato de estar sempre certo,
mas de não ter medo de estar errado.”*

Peter T. McIntire

Resumo

Este Trabalho de Conclusão de Curso mostra o processo de criação e desenvolvimento de um portal de anúncios direcionado ao ramo imobiliário a fim de facilitar a divulgação de imóveis e a prospecção de novos clientes, com o diferencial de trazer uma interface moderna e responsiva. O sistema foi desenvolvido em *PHP* utilizando o grande potencial dos *frameworks* Laravel e Materialize. Ainda há possibilidades de melhoria na aplicação, mas este portal já é capaz de suprir as necessidades de um portal de imóveis de uma imobiliária.

Palavras-chave: Sistema para Imobiliárias, Anúncio de Imóveis, Aluguel e Venda de Imóveis, PHP, Laravel.

Lista de ilustrações

Figura 1 – Página inicial - Udimóveis Imobiliária	16
Figura 2 – Página de detalhes do imóvel - Udimóveis Imobiliária	17
Figura 3 – Página de detalhes do imóvel - Airbnb	18
Figura 4 – Localização do imóvel no mapa - Airbnb	18
Figura 5 – Página inicial do portal	31
Figura 6 – Página de detalhes do imóvel	32
Figura 7 – Formulário de interesse do imóvel	33
Figura 8 – Painel de gestão do portal	34
Figura 9 – Listagem de imóveis	35
Figura 10 – Cadastro de cliente	36
Figura 11 – Telas do portal visualizadas em um <i>smartphone</i>	37

Lista de tabelas

Tabela 1 – Comparativo SQL vs. Eloquent	27
---	----

Lista de siglas

API *Application Programming Interface*

CRUD *Create-Read-Update-Delete*

CSS *Cascading Style Sheets*

DBaaS *Database as a Service*

ER *Entidade-Relacionamento*

HTML *Hypertext Markup Language*

HTTP *Hypertext Transfer Protocol*

JS *JavaScript*

MVC *Model-View-Controller*

ORM *Object Relational Mapper*

PaaS *Platform as a Service*

PHP *PHP Hypertext Preprocessor*

SGBD *Sistema Gerenciador de Banco de Dados*

SQL *Structured Query Language*

TI *Tecnologia da Informação*

UI *User Interface*

URL *Uniform Resource Locator*

UX *User Experience*

WWW *World Wide Web*

Sumário

1	INTRODUÇÃO	11
1.1	Lacuna	11
1.2	Propósito	12
1.3	Contribuições	12
1.4	Divisão do Trabalho	13
2	FUNDAMENTAÇÃO TEÓRICA	14
2.1	Conceitos Adotados	14
2.2	Trabalhos Correlatos	16
3	DESENVOLVIMENTO	20
3.1	Planejamento do Sistema	20
3.1.1	Requisitos	20
3.1.2	Modelagem de Dados	23
3.2	Tecnologias Utilizadas	23
3.2.1	Composer	24
3.2.2	Framework Laravel	24
3.2.3	Framework Materialize	28
3.2.4	Heroku	28
4	RESULTADOS	30
4.1	Apresentação	30
5	CONCLUSÃO	38
	REFERÊNCIAS	39

CAPÍTULO 1

Introdução

Na nossa sociedade a conquista de um imóvel nos remete a uma segurança financeira e realização pessoal que faz com que as pessoas poupem – ou se endividem – por anos a fim de adquirir sua tão sonhada casa própria. Há também aqueles que veem tal aumento patrimonial como uma boa oportunidade de investimento para obter lucro ou garantir uma certa estabilidade financeira no futuro.

Não são apenas as transferências patrimoniais que fomentam o mercado imobiliário, uma grande fatia deste negócio é constituída pelo setor de locação, negócio que geralmente interessa àqueles proprietários que pretendem esperar o melhor momento para vender seu bem, ou àqueles que buscam complementar renda obtendo um pouco de lucro enquanto não utilizam o ativo.

Independente do nicho, este mercado vem crescendo mundialmente a passos largos (INGAIA, 2018), e isto pode ser para os profissionais da área de Tecnologia da Informação (TI) uma oportunidade única de desenvolver novas soluções e ferramentas úteis ao setor, contribuindo ainda mais para o desenvolvimento do ramo.

No Brasil já existem inúmeros sistemas que procuram facilitar o comércio de imóveis (compra e venda ou aluguel). Há sistemas específicos para imobiliárias, *websites* de anúncios, aplicativos e também sistemas mais genéricos que, em comum, buscam agilizar a comunicação entre os interessados no negócio. Mas o perfil destas pessoas vem se tornando mais exigente, os usuários esperam encontrar sistemas cada vez mais interativos e simples, porém eficazes.

1.1 Lacuna

O processo de comercialização de um imóvel pode ser muito complicado sem a ajuda de um profissional do ramo ou de alguma tecnologia bem desenvolvida que facilite a comunicação entre o anunciante e o interessado, pois como há riscos para ambas as partes, a falta de informações essenciais, dificuldade de comunicação ou complexidade do

processo pode inviabilizar a negociação ou, pior, fazer com que alguém se engane e venha a se arrepender da operação.

Contar com a ajuda de um corretor tem seu lado positivo, como não se preocupar muito com a burocracia envolvida, além da confiança passada pelo contato pessoal com um intermediário. Contudo, é adicionada uma quantia extra ao valor de comercialização do imóvel devido aos custos a serem pagos pelos serviços deste profissional. Com a competitividade acirrada, quanto mais acréscimos forem feitos ao preço final, mais difícil é fechar um negócio.

Optando por substituir – integral ou parcialmente – a figura do corretor por um portal de anúncios, em nossa região ainda há barreiras tecnológicas a serem superadas. Dos portais que abrangem nossa área, muitos possuem irregularidades ou funcionalidades insuficientes. Há, por exemplo, telas não responsivas, tecnologias antigas sendo empregadas, má usabilidade e principalmente imagens de baixa qualidade ou mal expostas para apresentar o astro do negócio, o imóvel. Este cenário, que oferece uma experiência sem qualidade ao usuário, gera desinteresse, desconfiança ou até irritação, podendo causar a desistência do acordo.

1.2 Propósito

Assim, a proposta deste projeto é desenvolver uma aplicação *web* que seja simples, objetiva e principalmente intuitiva, de forma a facilitar a comercialização de imóveis pelas imobiliárias de nossa região. O objetivo é criar um portal de anúncios para atender as necessidades específicas deste setor e melhorar significativamente a experiência de seus usuários.

Pretende-se utilizar funcionalidades que agregam usabilidade, como páginas limpas que visam facilitar a experiência de usuários iniciantes, capacidade de exibir imagens de alta resolução e bem expostas para melhorar o foco nos detalhes dos imóveis oferecidos, filtros de busca mais completos e com boa personalização, além de formulários de contato mais simples e diretos.

1.3 Contribuições

Com esta aplicação as imobiliárias terão uma alternativa para maximizar os lucros de sua atividade, pois contarão com um portal de anúncios moderno, podendo reduzir os custos de corretagem e agilizando o fechamento do negócio através da comunicação mais rápida com o seu cliente.

1.4 Divisão do Trabalho

Este documento está organizado da seguinte maneira:

- ❑ **Capítulo 2 — Fundamentação Teórica:** descreve sucintamente os principais conceitos necessários ao desenvolvimento do portal de anúncios para imobiliárias e cita alguns *websites* existentes que serviram como inspiração para uma melhor construção do trabalho.
- ❑ **Capítulo 3 — Desenvolvimento:** descreve os requisitos que guiaram o desenvolvimento do sistema, além de apresentar as linguagens de programação e os *frameworks* adotados, o modo de armazenamento dos dados e o modelo de distribuição do sistema.
- ❑ **Capítulo 4 — Resultados:** apresenta o portal desenvolvido neste trabalho, exibindo suas principais características e funcionalidades.
- ❑ **Capítulo 5 — Conclusão:** traz as últimas considerações e os desafios enfrentados na etapa de elaboração deste sistema, deixando espaço para trabalhos futuros.

CAPÍTULO 2

Fundamentação Teórica

Este capítulo está organizado em duas seções. A seção 2.1 é relativa ao referencial teórico e tem como objetivo expor os principais conceitos e tecnologias utilizados no trabalho. A seção 2.2 apresenta brevemente alguns sistemas correlatos que serviram de apoio na elaboração deste.

2.1 Conceitos Adotados

Com o surgimento das redes móveis unido à evolução e popularização da *Internet*, possuir sistemas instalados localmente em servidores dedicados e utilizá-los unicamente em computadores locais ou em uma pequena rede empresarial não é mais uma necessidade crucial como já fora outrora. Atualmente, desenvolvedores podem manter quaisquer sistemas funcionando diretamente na *Internet* e fazer com que seus usuários possam acessá-los de diferentes dispositivos conectados à *World Wide Web* (WWW), onde quer que eles estejam.

A *Hypertext Markup Language* (HTML) é a linguagem base para a programação na *Internet* (DEV MEDIA, 2012), que é interpretada pelos navegadores e possibilita a publicação e o compartilhamento na WWW de conteúdo contendo textos, imagens, áudios e vídeos, dentre outros formatos de mídia, formando uma grande rede de informações e entretenimento.

Assim como outras linguagens de programação, o *HTML* melhorou com o tempo e está na quinta versão, o *HTML5*, que tornou mais fácil a manipulação de seus elementos e melhorou sua integração com outras linguagens. Contudo, ainda é de extrema importância complementá-la com o *Cascading Style Sheets* (CSS) e o JavaScript (JS) para obter uma página melhor estilizada, intuitiva e interativa.

O CSS, que está na terceira versão, foi criado com o objetivo de aplicar estilos e comportamentos visuais aos elementos de uma página *web* (TECMUNDO, 2009), deixando com que as linguagens de marcação, como o HTML, se encarreguem apenas das tarefas de marcar e estruturar o documento. Essa divisão de responsabilidades só traz ganhos

aos usuários, já que as linguagens de marcação podem ser otimizadas para fornecer uma maior quantidade de conteúdo enquanto a linguagem de estilização os deixa muito mais intuitivos e atraentes.

O JS é uma linguagem de programação que permite manipular os elementos HTML e seus estilos CSS de forma dinâmica, interativa e em tempo real (CANALTECH, 2015), deixando as páginas *web* mais completas e robustas do ponto de vista do desenvolvedor, e mais agradáveis e inteligentes na visão do usuário.

A combinação destas três tecnologias dá ao desenvolvedor a possibilidade de criar páginas excepcionais, mas alguns recursos ainda são essenciais para que uma aplicação seja capaz de colher informações e exiba-as futuramente em outro dispositivo ou a outro usuário sem abrir mão da segurança destas informações, senão a página será somente um documento estático e imutável incapaz de fornecer muita utilidade a diversos usuários. Dentre esses recursos vale mencionar o servidor de aplicação em conjunto com uma linguagem *server-side* para receber e enviar informações ao usuário, e um banco de dados para guardar com segurança todo o conteúdo.

Dentre diversas alternativas, foi adotado o *PHP Hypertext Preprocessor* (PHP) – originalmente *Personal Home Page* –, uma linguagem de programação leve e rápida, voltada para o lado do servidor (BECODE, 2017). É através dela que serão recebidas, processadas e respondidas todas requisições dos clientes. O PHP foi adotado por ter uma curva de aprendizagem mais baixa quando comparado a outras linguagens de programação tais como Java e C#, e possui uma extensa documentação na *Internet*, com vários casos de uso, fóruns, artigos, guias e uma comunidade extremamente ativa.

E para atender a necessidade de persistir todas as informações processadas de maneira segura e organizada, é utilizado um Sistema Gerenciador de Banco de Dados (SGBD) relacional (DEV MEDIA, 2006).

Todos estes recursos serão utilizados na nuvem através de uma *Plataform as a Service* (PaaS) que irá hospedar as páginas e disponibilizá-las aos usuários através de um servidor PHP, e de um *Database as a Service* (DBaaS) para armazenar e disponibilizar os dados processados pela aplicação.

A PaaS é um tipo de solução que disponibiliza um espaço *online* para instalação de aplicações completas para disponibilizá-las aos usuários através da rede mundial. A adoção desta abordagem livra o desenvolvedor de grande parte da montagem e manutenção de um servidor de aplicação, pois toda configuração é prontamente fornecida ao contratar o serviço. O DBaaS funciona de forma similar, porém fornecendo na nuvem uma instância que gerencia um banco de dados. Juntos, os serviços criam um ambiente pronto para executar o sistema.

2.2 Trabalhos Correlatos

Analizando o mercado *online* de imóveis, pode-se encontrar diversos sistemas que utilizam os conceitos abordados neste projeto. Para mostrar o conflito de usabilidade e navegação que toma conta deste nicho, tomamos como exemplo o portal regional da **Udimóveis Imobiliária** (UDIMÓVEIS, 2018), que possui um sistema desenvolvido pela empresa de TI SmartWEB (SMARTWEB, 2000), e o *Airbnb* (AIRBNB, 2018), uma referência mundial de locação de espaços privados, fundado por Brian Chesky, Joe Gebbia e Nathan Blecharczyk. Os dois sistemas mostram o contraste das atuais tecnologias de desenvolvimento web quando aplicadas sem um devido uso de seu potencial e quando aliadas a conhecimentos em *design*, usabilidade e experiência do usuário.

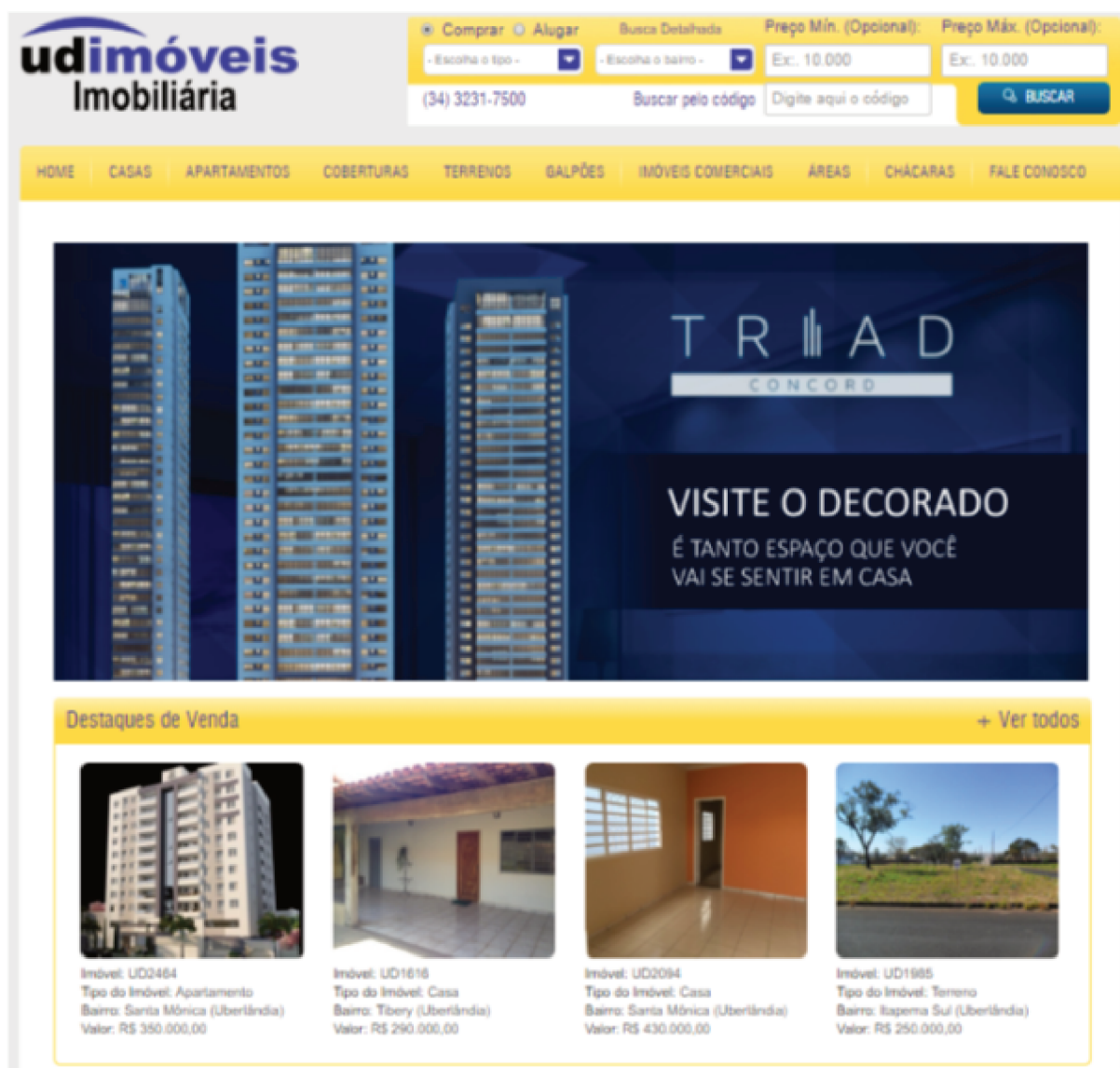


Figura 1 – Página inicial - Udimóveis Imobiliária

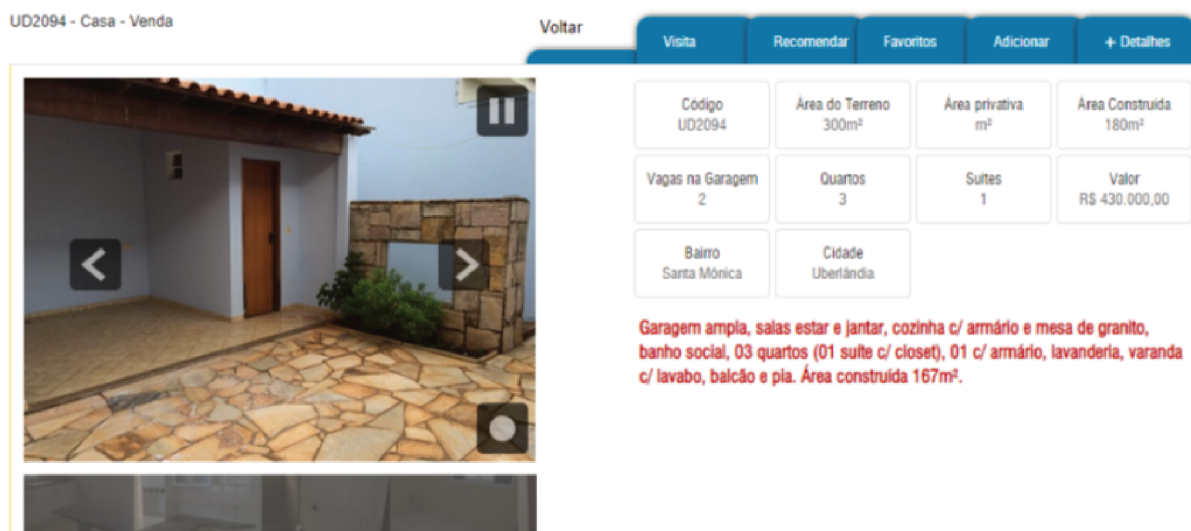


Figura 2 – Página de detalhes do imóvel - Udimóveis Imobiliária

A Udimóveis Imobiliária fornece aos seus usuários um sistema de compra e aluguel de imóveis de todo tipo, incluindo chácaras e lotes ao seu variado rol de bens comercializados.

O sistema possui um simples filtro que ajuda o usuário a localizar com mais facilidade o imóvel desejado. A página inicial (Figura 1) conta também com um *banner* rotativo destacando algum novo empreendimento e com uma reduzida lista que exibe as principais ofertas do sistema referentes a venda e aluguéis em destaque. Cada imóvel também possui uma página própria (Figura 2) com detalhes adicionais.

Mesmo parecendo ser um sistema completo, percebe-se algumas falhas que podem ser melhorados, principalmente em relação à aplicação da CSS no portal, fontes fora do padrão, elementos mal posicionados, imagens de baixa qualidade e até algumas ofertas sem fotos, podendo gerar um desconforto para o usuário finalizar a negociação.

Além disso, as páginas não são bem apresentadas em dispositivos móveis, pois em vez do conteúdo se adaptar ao tamanho da tela um zoom reduz tudo para que a página se encaixe no espaço menor, deixando os textos pequenos e ilegíveis.

Não obstante a tais problemas, uma possível funcionalidade que ajudaria o sistema a atrair uma maior quantidade de clientes seria disponibilizar a localização do imóvel ofertado por meio de um mapa, mas nem mesmo um endereço aproximado é divulgado, e para qualquer detalhe adicional sobre o anúncio é preciso entrar em contato com a empresa.

Já o Airbnb (Figura 3) apresenta o oposto disto. Este é um *site* onde anfitriões podem anunciar espaços para alugar e viajantes podem reservá-las de qualquer lugar do mundo por uma temporada. Os desenvolvedores utilizam de maneira correta a estrutura do HTML, as disposições da folha de estilo CSS e a dinâmica do JS juntamente com um design arrojado e moderno, deixando o sistema elegante, intuitivo e mais atrativo.

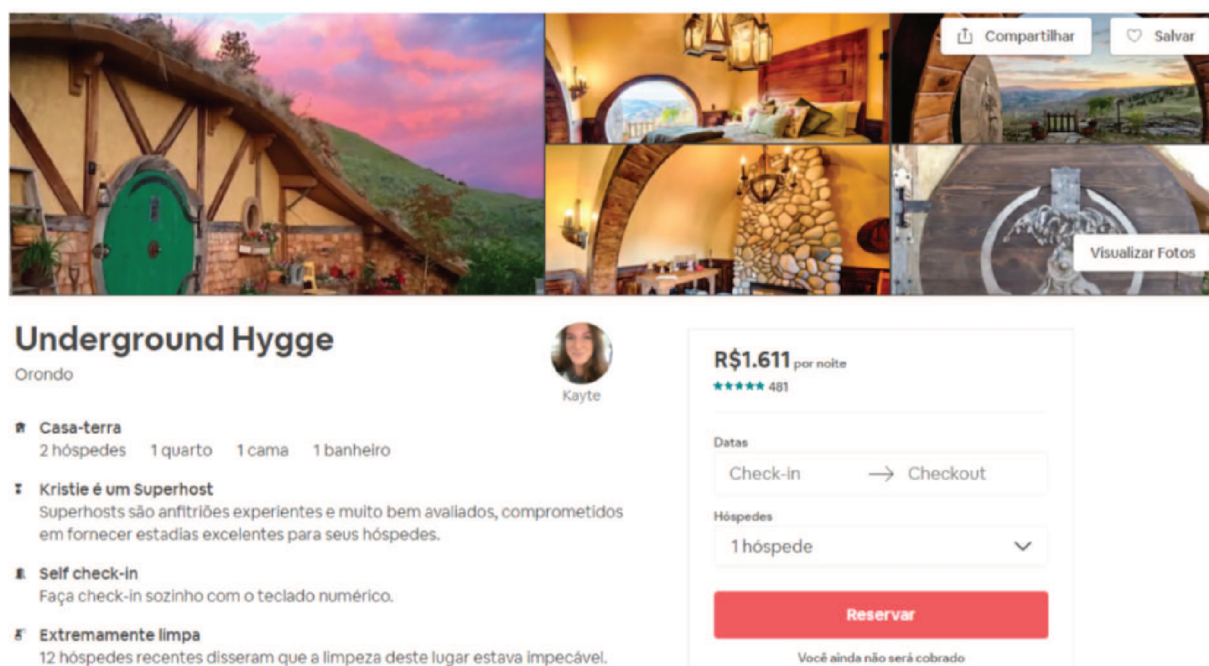


Figura 3 – Página de detalhes do imóvel - Airbnb

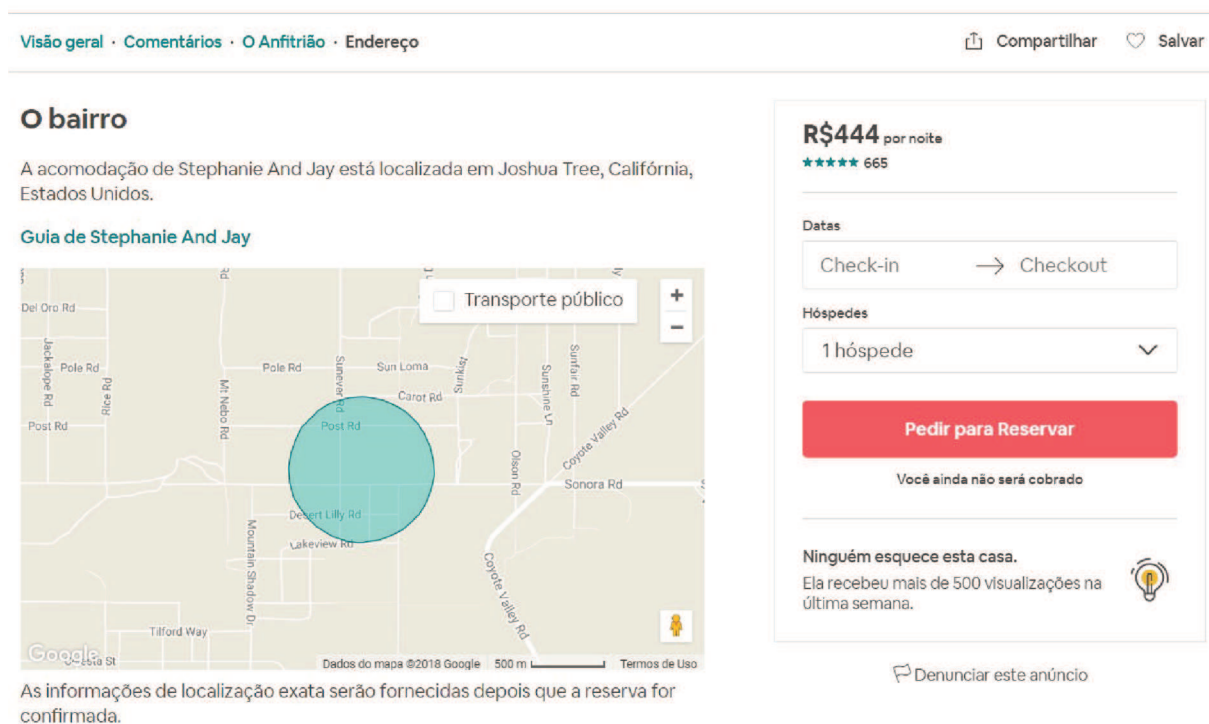


Figura 4 – Localização do imóvel no mapa - Airbnb

Como destaque, possui um sistema inteligente de troca de mensagens onde os interessados podem se comunicar de forma segura, além de uma plataforma confiável para transferência de valores. Além disso, esbanja filtros completos e inteligentes, mapas (Figura 4) para dar uma maior visibilidade da área em que o imóvel se localiza e imagens de alta qualidade, proporcionando ao usuário uma experiência muito próxima à de estar presente no ambiente.

O objetivo deste projeto é desenvolver um sistema para imobiliárias inspirado nos conceitos vistos em sites modernos como o Airbnb e afastar os erros cometidos nos pequenos portais de nossa região.

CAPÍTULO 3

Desenvolvimento

Este capítulo está dividido em duas seções. A seção 3.1 é relativa ao planejamento do sistema e traz os requisitos atendidos e de que forma estão relacionados, e a seção 3.2 apresenta com mais detalhes as tecnologias utilizadas na construção do portal imobiliário.

3.1 Planejamento do Sistema

A modelagem do sistema é uma das primeiras e principais fases de desenvolvimento de um projeto. É considerada uma etapa de análise onde definirá a forma que o projeto será construído. Nela desenvolve-se os modelos abstratos do sistema, onde cada um representa uma visão ou perspectiva diferenciada do mesmo.

Priorizando as necessidades do usuário final foram definidos os requisitos do sistema e criado o modelo entidade-relacionamento do portal apresentados nos tópicos seguintes.

3.1.1 Requisitos

Um requisito é determinado pelo que se espera de uma funcionalidade, ou seja, as habilidades que o sistema deverá ter para suprir as necessidades do usuário final. Um conjunto de requisitos bem estruturados e fundamentados é a base para o sucesso da aplicação. Mapeá-los adequadamente nesta etapa ajuda a reduzir a descoberta de novas necessidades após a entrega do produto e, assim, reduz a manutenção necessária após a implantação do sistema.

Os requisitos do portal estão documentados na lista a seguir:

- R. 1 **Cadastro de Cliente:** o sistema deve contemplar um *Create-Read-Update-Delete* (CRUD) para clientes. Estes irão fornecer seus dados a um funcionário que irá identificá-lo no sistema como proprietário (anunciante), comprador ou inquilino. Cada cliente poderá anunciar, comprar ou alugar quantos imóveis desejar. O cadastro deve ser único, mesmo que o cliente efetue operações diferentes como, por

exemplo, um cliente pode estar vendendo um imóvel enquanto compra outro e é inquilino de uma terceira propriedade.

R. 2 **Cadastro de Tipos:** o sistema deverá suportar imóveis de dois tipos: casa ou apartamento. Entretanto, deverá contemplar um CRUD para possibilitar que a imobiliária estenda tais tipos e crie quantos subtipos forem necessários ao seu negócio.

R. 3 **Cadastro de Imóvel:** o sistema deve contemplar um CRUD para imóveis, no qual o funcionário irá definir seu tipo e se ele será ofertado para venda ou aluguel. Cada imóvel será relacionado a apenas um proprietário responsável. Ademais, os itens abaixo devem ser atendidos:

Fotos: os imóveis devem ter uma galeria de fotos, possibilitando o envio de várias imagens de uma só vez;

Mapa: a localização do imóvel deve ser registrada para exibição em um mapa;

Características: registrar dados como quantidade de dormitórios, vagas, banheiros, suítes e valor do condomínio, se houver;

Infraestruturas: registrar as infraestruturas que o imóvel possui;

R. 4 **Registro de Interesses:** o sistema deve contemplar um CRUD para serem registrados todos os interesses de um imóvel. O interesse identifica o provável cliente e pode ser incluído tanto por um funcionário quanto pelo sistema, caso haja manifestação de um interessado na área pública do portal.

R. 5 **Agendamento de Visitas:** o sistema deve contemplar um CRUD para registrar as visitas aos imóveis. Um funcionário da imobiliária ficará encarregado de combinar estas visitas junto aos interessados.

R. 6 **Cadastro de Propostas:** o sistema deve contemplar um CRUD para registrar as propostas de compra ou locação de um imóvel. A proposta é o passo anterior à concretização do negócio, onde o valor proposto não precisa ser exatamente o valor anunciado e, em caso de finalização da venda, este valor deve ser considerado. Uma proposta pode ser aprovada ou recusada.

R. 7 **Registro de Vendas:** o sistema deve contemplar um CRUD para registrar as vendas, dando a possibilidade de anexar o documento do contrato ao registro. Quando a venda é concretizada o imóvel deve ser automaticamente removido da página de anúncios do portal, mas deve ficar disponível no sistema.

R. 8 **Registro de Aluguéis:** o sistema deve contemplar um CRUD para registrar as locações, dando a possibilidade de anexar o documento do contrato ao registro.

Quando a locação é concretizada o imóvel deve ser automaticamente removido da página de anúncios do portal, mas deve ficar disponível no sistema.

- R. 9 **Relatórios:** o sistema deve fornecer uma página para rápida visualização das vendas e locações concretizadas em determinado período.
- R. 10 **Cadastro de Funcionário:** o sistema deve contemplar um CRUD para funcionários. O funcionário será o usuário do sistema que utilizará as funcionalidades previstas nos requisitos anteriores. Poderá haver mais de um usuário e eles devem ser capazes de acessar simultaneamente o sistema.
- R. 11 **Área restrita:** o sistema deve possuir uma área cujo o acesso é restrito aos colaboradores da imobiliária através de um formulário de autenticação com usuário e senha. É nesta área que eles poderão cadastrar os clientes da imobiliária, gerir os imóveis negociados e administrar os interesses, visitas e propostas para cada um, além de mapear as vendas e alugueis e visualizar seus relatórios, ou seja, os requisitos anteriores devem ser protegidos por esta autenticação.
- R. 12 **Área pública:** o sistema deve ter uma área acessível publicamente, sem requerer qualquer autenticação. Nela o usuário visualiza todos os imóveis disponíveis na imobiliária, tanto para compra quanto para locação. Esta área contemplará as seguintes necessidades:
 - Página Inicial:** esta página deve concentrar todos os anúncios, anúncios em destaque, acesso rápido ao formulário de busca e às demais páginas do portal. A paginação é uma necessidade, visto que haverá muitos anúncios ativos simultaneamente;
 - Página de Detalhes:** esta página deve detalhar cada imóvel, trazendo todas as suas características, mapa, fotos, etc., além de prover um meio de comunicação do interessado com a imobiliária que automaticamente registra um interesse no sistema;
 - Formulário de Busca:** permitir buscas de imóveis utilizando vários filtros, como tipo de oferta e de imóvel, localização, valor, características e infraestruturas;
 - Página de Contato:** formulário de contato com a imobiliária;
 - Página Institucional:** uma página com informações sobre a imobiliária, personalizável pelos próprios funcionários.
- R. 13 **Responsividade:** o sistema deve ser capaz de se adequar automaticamente para cada visor em que ele está sendo exibido.
- R. 14 **Criptografia:** somente armazenar as senhas após cifrá-las usando o algoritmo **Bcrypt** (PROVOS; MAZIERES, 1999), para prevenir vazamento de dados.

- R. 15 **Remoção em Cascata:** sempre que uma entidade for excluída, após a confirmação do usuário, excluí-la juntamente com todas suas entidades filhas.
- R. 16 **Controle de versão:** usar **Git** para gerenciamento das diferentes versões do projeto.

3.1.2 Modelagem de Dados

Um modelo Entidade-Relacionamento (ER) é um modelo de dados conceitual de alto nível bastante popular (ELMASRI; NAVATHE, 2009). Este modelo é muito utilizado para projetar bancos de dados e sua montagem visa melhorar a persistência de dados de forma a garantir a recuperação das informações.

O modelo ER deste sistema foi criado abstraindo o comportamento do ramo imobiliário em algumas entidades principais: “Funcionário, Cliente, Imóvel e Contrato”. Além de especializá-las ou generalizá-las, para abranger as demais funcionalidades almejadas pelo sistema foram levantadas mais algumas entidades: “Visita e Anúncio”.

Um dos requisitos do sistema é possibilitar o cadastro de casas ou apartamentos (Requisito R. 2). Entretanto, ambos são imóveis, então as entidades “Casa e Apartamento” são mera especialização da entidade “Imóvel”. De forma semelhante, como um imóvel só pode ser vendido ou alugado (Requisito R. 3), outra vez é utilizado o conceito de especialização, criando as entidades “Imóvel para alugar e Imóvel para vender”. Assim, há quatro possíveis combinações de características para um imóvel.

Outro requisito estipula que um cliente pode ser “Proprietário, Comprador ou Inquilino” (Requisito R. 1). Como estes dois últimos tem, na visão do negócio, interesse em um imóvel anunciado, suas entidades foram generalizadas em “Interessado”. É interessante trabalhar com esta generalização pois, como ambos podem visitar um imóvel, fica mais claro a relação desta entidade com a de “Visita”.

Por fim, um último conceito interessante que foi modelado é o da operação de comercialização, dividida em “Venda ou Aluguel”. Tanto a venda como o aluguel fazem parte de um processo de comercialização intermediado por um “Funcionário” da imobiliária, porém, da compra somente participam as entidades “Comprador e Imóvel para vender”, enquanto do aluguel somente participam o “Inquilino e o Imóvel para alugar”. Estas singularidades são possíveis graças às especializações do “Imóvel, Interessado e Contrato”.

3.2 Tecnologias Utilizadas

Para o desenvolvimento do portal imobiliário foram utilizadas diversas ferramentas com o intuito de enriquecer as linguagens de programação adotadas, visando agilizar a fase de criação das aplicações e melhorar seu desempenho. O componente que proporciona tal resultado é conhecido como *framework*, que adiciona recursos à uma linguagem

de programação, tal como a PHP, estendendo suas funcionalidades. Neste sentido, foi adotado o *framework* Laravel para o desenvolvimento no *back-end*, ampliando a flexibilidade da linguagem de servidor adotada, e o *framework* Materialize CSS para facilitar a programação do *front-end*, trazendo novos componentes em HTML e CSS prontos para utilização.

A distribuição do sistema também conta com *facilitadores*, serviços na nuvem especializados em guardar, executar e disponibilizar a aplicação aos usuários finais. A hospedagem das páginas da *Application Programming Interface* (API) e dos dados do sistema utilizam a PaaS Heroku, um serviço na nuvem capaz de oferecer a disponibilidade e escalabilidade que a aplicação necessitar.

Essas tecnologias são apresentadas brevemente a seguir.

3.2.1 Composer

O **Composer** (ADERMANN; BOGGIANO, 2012) é uma ferramenta que permite gerenciar, de forma fácil, as dependências do PHP. Com poucos comandos é possível definir todas as bibliotecas, próprias ou de terceiros, que serão utilizadas em um projeto. O gerenciador fica, então, encarregado de obter a versão mais atual de cada uma delas e de criar um *autoloader* deixando-as prontas para uso pelo programador.

O gerenciador não é necessário, mas é uma facilidade. Um exemplo de utilização do Composer é na criação do projeto em Laravel, executando a seguinte linha de código:

```
composer create-project --prefer-dist laravel/laravel imobiliaria
```

Desta maneira, será baixado, instalado e configurado um projeto para seu desenvolvimento ser realizado utilizando o *framework* Laravel. Toda a estrutura de pastas e todas as bibliotecas do Laravel com as configurações básicas já serão criadas. O programador pode iniciar o desenvolvimento imediatamente.

3.2.2 Framework Laravel

O **Laravel** (OTWELL, 2011) é um *framework* PHP para desenvolvimento rápido, livre e *open source*. Seu principal objetivo é proporcionar que o desenvolvedor tenha um código bem estruturado em mãos e consiga implementá-lo de forma rápida e elegante, assim como diz no seu *slogan*: “O *framework* PHP para artesões da *web*”.

Foi desenvolvido para acabar com a rotina entediante e repetitiva dos desenvolvedores *web*, pois foi criado com intuito de ser simples e fácil de aprender. Possui uma documentação intuitiva e bem estruturada, facilitando as pesquisas e estudos para a utilização de forma aprofundada dos recursos providos pelo *framework*.

Além disso, o Laravel já conta com bastante aceitação do mercado em geral, possui uma equipe de desenvolvedores ativa e extremamente competente, além de uma grande comunidade, o que torna o suporte e o crescimento do *framework* muito ágeis. Com estas características ele passou a ser um dos mais populares *frameworks* para PHP da atualidade (TRENDS, 2019).

3.2.2.1 Rotas

Uma das funções básicas do Laravel é o controle de requisições HTTP, denominado unicamente de **Rotas**. As rotas fazem o mapeamento da *Uniform Resource Locator* (URL) digitada no navegador para alguma ação implementada no sistema. Pode-se resumir a uma URL amigável. O mapeamento é simples, feito da seguinte maneira:

```
Route::get('/', ['as'=>'site.home', 'uses'=>'Site\HomeController@index']);
Route::get('/contato', ['as'=>'site.contato',
    'uses'=>'Site\PaginaController@contato']);
```

Então, quando o usuário acessa “www.imobiliaria.com” estará acessando a rota “/”, ou melhor dizendo, a página principal do site. Ao acessar “www.imobiliaria.com/contato” estará acessando a rota “/contato”, e assim por diante. Quando o servidor recebe a requisição, o sistema de Rotas do Laravel procura no mapeamento qual *Controller* e qual função ele deve executar.

Nos casos acima está sendo utilizado o protocolo HTTP juntamente com o método mais comum, o *GET*, onde parâmetros são passados no cabeçalho da requisição (na URL). Cada um dos casos está chamando um controlador que, por sua vez, possui um método que lista e redireciona para as correspondentes páginas. No segundo exemplo, que acessa a rota “/contato”, o servidor deverá executar a função “contato()” do *Controller* “PaginaController” que está no diretório “Site” do servidor, e retornar ao usuário a página “Contato”.

Continuando este exemplo, dentro da página “Contato” o usuário preencherá um formulário para se comunicar com a imobiliária, e então clicará no botão “Enviar”. O botão executará uma chamada à seguinte rota:

```
Route::post('/contato.enviar', ['as'=>'site.contato.enviar',
    'uses'=>'Site\PaginaController@enviarContato']);
```

Neste caso também está sendo utilizando o protocolo HTTP, porém desta vez com o método *POST*, cujo parâmetros são passados no corpo da requisição (escondendo da URL), garantindo uma maior segurança quanto à informação preenchida pelo usuário.

Também é chamado um controlador que possui um método que irá captar as informações preenchidas e posteriormente realizar o envio do contato.

Este é basicamente o conceito de rotas no Laravel: ao ser requisitado em uma determinada URL o sistema de rotas define o que fazer, como redirecionar para uma página ou enviar para o controlador decidir qual ação realizar. Parâmetros podem ser passados na URL ou no corpo da requisição, e podem ser capturadas pelo *framework* e utilizadas pelo desenvolvedor para executar ações específicas.

3.2.2.2 Blade

O **Blade** é o *template engine* (processador de modelo) do Laravel. A sua diferença em relação aos outros está na flexibilidade que tem, pois ele não restringe o uso de PHP nativo misturado à sua própria sintaxe. Os arquivos do Blade devem ter a extensão “.blade.php”.

O principal objetivo do Blade é reduzir a quantidade de código PHP inserido no HTML e aumentar o reuso, e para isto ele utiliza de uma série de diretivas que podem ser inseridas junto ao código de acordo com a necessidade do programador e da página em questão. Exemplo:

```
<ul>
    @if($imovel->galeria()->count())
        @foreach($galeria as $foto)
            <li>
                
            </li>
        @endforeach
    @else
        <li>
            
        </li>
    @endif
</ul>
```

Aqui temos um trecho de código da página que exibe um imóvel, especificamente a(s) foto(s) do mesmo. Este trecho está misturado à sintaxe do Blade para fazer um laço que monta a galeria de imagens. São usadas algumas diretivas proporcionadas pelo Blade neste código, tais como as diretivas condicionais “@if / @else / @endif” e as diretivas de repetição “@foreach / @endforeach”. Com a diretiva condicional verifica se existe alguma imagem cadastrada na galeria do respectivo imóvel e, caso positivo, utiliza-se a diretiva de repetição para mostrar na página todas as imagens do imóvel, mas caso não haja uma galeria para aquele imóvel, será exibida somente a imagem padrão do imóvel.

Outros dois grandes recursos do Blade são as heranças e seções, elas permitem trabalhar facilmente com o conceito de *master page* (página mestre), que é uma página sem conteúdo mas com um *layout* reutilizável por outras páginas, evitando repetição de código.

3.2.2.3 Eloquent

O **Eloquent** é o *Object Relational Mapper* (ORM) padrão do Laravel. É uma técnica de desenvolvimento de *software* usada para reproduzir tabelas de banco de dados em formato de objetos relacionais, deixando a manutenção dos dados mais simples e fácil. Com esta técnica o programador não precisa escrever códigos na linguagem *Structured Query Language* (SQL). Com a interface ORM é possível realizar operações de CRUD, além de criar as tabelas e montar relacionamentos 1:1 (um para um), 1:N (um para muitos) e N:M (muitos para muitos).

Utilizando um padrão de registro ativo, o Eloquent transforma a tabela do banco de dados em uma estrutura *Model-View-Controller* (MVC). Por exemplo, para listar todos os usuários da tabela de usuários ou encontrar um usuário específico basta usar funções simples e intuitivas que equivalem à *query* SQL desejada (Tabela 1).

SQL	Eloquent ORM
<code>select * from usuario</code>	<code>Usuario::all()</code>
<code>selec * from usuario where id = 1</code>	<code>Usuario::find(1)</code>

Tabela 1 – Comparativo SQL vs. Eloquent

3.2.2.4 Artisan Console

Artisan é o nome da interface de linha de comando incluída no Laravel. Ela fornece vários comandos que auxilia o desenvolvedor no momento da implementação da aplicação. O Artisan é impulsionado pelo poderoso componente de console do Symfony Framework (SAS, 2005). Com este console o trabalho inicial do desenvolvedor é poupado, vejamos:

```
php artisan make:model Cliente -m
php artisan make:controller ClienteController
```

Com um comando como o primeiro no exemplo acima, a ferramenta já cria diversos arquivos e configurações no projeto para implementar a entidade “Cliente”, bastando agora somente preencher seus atributos, relacionamentos e regras de negócios. Do mesmo modo, o segundo comando já monta o arquivo padrão de um controlador.

3.2.3 Framework Materialize

Antes de começar a explicar o **Materialize** é preciso entender um pouco sobre o *Material Design*, pois é deste conceito que nasceu o Materialize. O Material Design (GOOGLE, 2014) é um conceito que foi criado pela equipe de *User Experience* (UX) e *User Interface* (UI) da Google com intuito de unificar a experiência de seus usuários, tanto nos computadores quanto nos dispositivos móveis, com premissas modernas e simples.

O Materialize (MATERIALIZE, 2014) é um conjunto de bibliotecas que tem como objetivo facilitar a estilização de sites utilizando a linguagem CSS e JS, e traz consigo diversos componentes prontos para uso e personalizáveis, baseado nas premissas do Material Design.

Com o Materialize, basta inserir uma classe para transformar um simples campo de hora em um *modal* que exibe um relógio para selecionar a hora desejada, da mesma forma que vemos nos aplicativos de despertador dos *smartphones* atuais, por exemplo.

Nos módulos e ferramentas relacionados à linguagem CSS do Materialize, alguns itens merecem destaque, como o sistema de cores que é extremamente amplo e simples de implementar, e o sistema de grades que tem foco na responsividade, deixando o sistema pronto para ser executado em *smartphones* ou *tablets*, além dos computadores.

Além do exemplo já mencionado anteriormente, o *framework* possui outros componentes que facilitam a implementação para o programador e melhora muito a experiência do usuário, tais como, botões, cartões, conjunto de ícones, barras de navegação, rodapés e a paginação.

Os módulos relacionados à linguagem de programação JS possuem as ferramentas mais dinâmicas do framework. A principal é o menu de navegação próprio para *tablets* e *smartphones*, que tem um funcionamento diferente do visto nos computadores. Outras ferramentas importantes são os menus *dropdown*, *sliders* e carrosséis para apresentar imagens, *modals* e *tooltips*, apenas para citar os mais comuns.

3.2.4 Heroku

Com o crescimento da computação em nuvem e o aumento significativo dos serviços que são proporcionados por estas plataformas, os programadores ganharam agilidade e simplicidade no processo de implementação e distribuição de seus sistemas. É possível escolher dentre contratar um simples *Web Service* ou até uma solução mais completa, dependendo da necessidade do projeto.

Diversas empresas competem neste mercado, como a Amazon e a Microsoft. Ultimamente um fornecedor que vem ganhando mercado é o Heroku (SALESFORCE, 2007), pois oferece uma PaaS, que é um tipo de serviço que abstrai o desenvolvedor de detalhes de infraestrutura, facilita a manutenção, extensão e escalabilidade. Ele também fornece o

DBaaS, um banco de dados na nuvem que pode ser integrado a uma aplicação, hospedada ou não em seu domínio.

O Heroku foi escolhido por fornecer os dois serviços essenciais a esta aplicação em um só local, por ter um plano inicial gratuito, e também por ser de fácil utilização.

CAPÍTULO 4

Resultados

Este capítulo apresenta o portal de anúncios para imobiliárias desenvolvido neste trabalho e traz as principais telas criadas. Para demonstrar os recursos implementados segue-se o fluxo de uma transação, da prospecção à venda, passando pelos seus principais cadastros para uma melhor abstração das funcionalidades.

4.1 Apresentação

A estrutura deste sistema foi criada com base no conceito de aplicação *Web* e foi dividida em duas áreas: uma de acesso público onde usuários podem navegar livremente, e outra de acesso restrito desenvolvida para que somente os colaboradores da imobiliária possam acessar.

Na área pública (Figura 5), os usuários encontram no topo da tela um painel publicitário que tem como principal funcionalidade ampliar a visibilidade e aumentar as chances de comercialização de determinados imóveis. Além de ser um impulsionador de vendas, este painel cria uma nova forma de monetização para a imobiliária, pois ela pode oferecê-lo de forma adicional ao anúncio padrão, onde cada proprietário pode optar por este investimento a fim de concretizar seu negócio de maneira mais rápida.

Abaixo do painel publicitário é exibida uma lista com todos os imóveis anunciados, apresentados em forma de cartões que trazem as principais características da propriedade, uma imagem principal e seu valor comercial. Como esta lista pode ser grande foi desenvolvida com um sistema de paginação, limitando cada tela a vinte anúncios e trazendo *links* de navegação entre as ofertas. Acima desta lista há um formulário de pesquisa que possibilita filtrar os resultados de acordo com a necessidade particular de cada visitante. É possível filtrar por venda ou aluguel, tipo de imóvel, localização (cidade, bairro ou rua), quantidade de dormitórios, banheiros, suítes e vagas na garagem. Caso o internauta tenha em mente algumas infraestruturas que deseja ter em seu futuro imóvel é possível especificá-las e, além disso, é possível determinar uma faixa de valores para apresentação.

Alliance

Home Sobre Contato



Apartamento no Saraiva

Apartamento próximo a UFU, com muita segurança e uma ótima infraestrutura.

[MAIS →](#)

Imóveis

Oferta

Qualquer oferta ▾

Tipo de Imóvel

Qualquer Tipo ▾

Cidade

Qualquer Cidade ▾

Endereço/Bairro

Dormitórios

Qualquer Qtd. ▾

Banheiros

Qualquer Qtd. ▾

Suítes

Qualquer Qtd. ▾

Vagas de Garagem

Qualquer Qtd. ▾

Infraestruturas

Marque suas preferências ▾

Faixa de Valor (R\$)

0

1500000

FILTRAR



À venda
Apartamento no Centro

R. Cel. Antônio Alves Pereira, 731
Uberlândia/MG

R\$ 890.000,00

VER MAIS...



À venda
Casa no Jardim Karaíba

R. Marcolino Ribeiro, 481
Uberlândia/MG

R\$ 1.500.000,00

VER MAIS...



Para alugar
Casa no Santa Mônica

R. José Carrijo, 1279
Uberlândia/MG

R\$ 4.000,00

VER MAIS...



Para alugar
Apartamento no Saraiva

Av. Nicodemos Alves dos Santos, 100
Uberlândia/MG

R\$ 3.500,00

VER MAIS...

Alliance

Sistema de Imobiliária.

Links

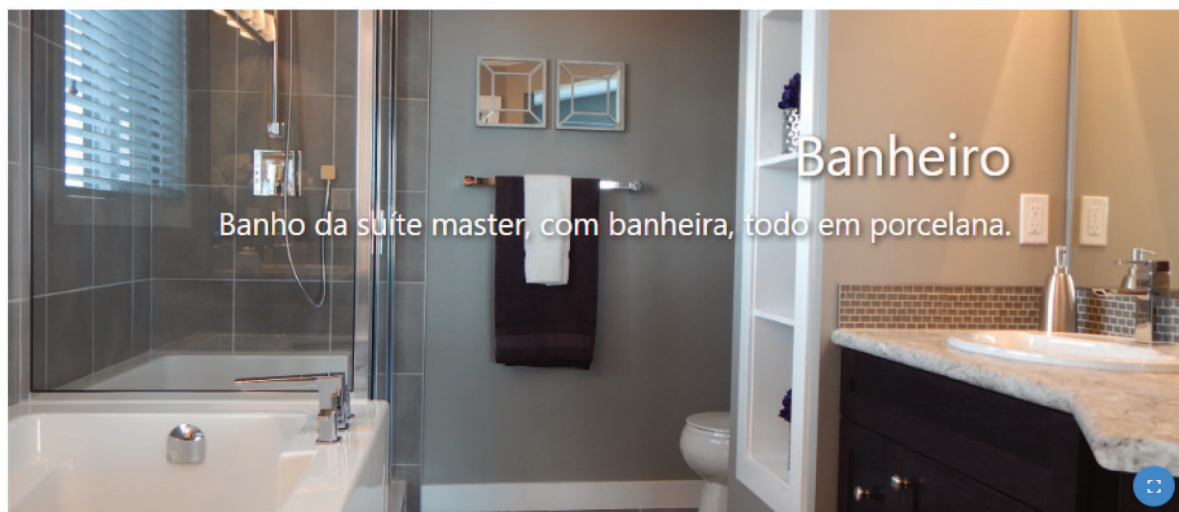
Home
Sobre
Contato

© 2018 Copyright Alliance

Heitor Nunes

Figura 5 – Página inicial do portal

Apartamento no Saraiva



Apartamento localizado próximo a UFU, situado em uma via de rápido acesso, facilitando a locomoção para os principais pontos da cidade. Possui um supermercado em seu quarteirão, facilitando a compra de mantimentos. Portaria 24 horas, com câmeras espalhadas por todo condomínio. Parquinho, quadra, piscina e um salão para festas com churrasqueira e muito mais, para deixar seu lazer com ainda mais qualidade.

Oferta: Para alugar **Tipo:** Apartamento **Vaga(s):** 2 **Dormitório(s):** 3 **Banheiro(s):** 2 **Suíte(s):** 1

Endereço: Av. Nicodemos Alves dos Santos, 100 **Cidade:** Uberlândia



Detalhes:

Sacada, Piscina, Elevador, Churrasqueira, Gás encanado, Parquinho, Quadra, Salão de Festas, Ar condicionado, Portaria 24h.

Condomínio: R\$ 950,00

R\$ 4.500,00

TENHO INTERESSE

Figura 6 – Página de detalhes do imóvel

Tenho Interesse



Apartamento no Saraiva

Preencha o formulário.

Nome Completo

Telefone

E-mail

Seu endereço de e-mail

Mensagem

ENVIAR

Figura 7 – Formulário de interesse do imóvel

Os filtros são cumulativos, ou seja, marcando mais de um critério, serão exibidos somente imóveis que atenderem a todos.

Ao clicar em um dos cartões ou anúncio em destaque da página inicial abre uma nova tela que exibe detalhes completos daquele imóvel. Cabe destacar que apenas uma página foi desenvolvida, porém a mesma se adapta dinamicamente ao conteúdo particular de cada imóvel que exibe.

A tela de detalhes do imóvel (Figura 6) possui uma galeria de fotos com uma breve descrição onde o usuário conseguirá visualizar com maior precisão todos os detalhes de cada área da propriedade. Estas imagens podem ser de alta qualidade e resolução e é possível ampliá-las para uma melhor visualização, ocupando toda área disponível no visor. A página traz uma descrição muito mais detalhada do imóvel e com as características mais interessantes em destaque, entregando ao usuário informações com mais qualidade e facilidade de visualização.

Através da API do **Google Maps** (GOOGLE, 2005) foi incorporado nesta tela um mapa com a exata localização do imóvel. Neste mapa o usuário terá acesso a todas as informações relacionadas a endereço e também conseguirá visualizar pontos de interesse nas proximidades da propriedade, tais como supermercados, hospitais, escolas, padarias, restaurantes e farmácias, dentre outros.

Por fim, caso o usuário desperte interesse em conhecer a propriedade e queira marcar uma vista, ele poderá entrar em contato com a imobiliária através do botão “TENHO INTERESSE”, onde preencherá um formulário (Figura 7) para preenchimento de seus dados pessoais e uma mensagem. Este formulário irá registrar um interesse no sistema

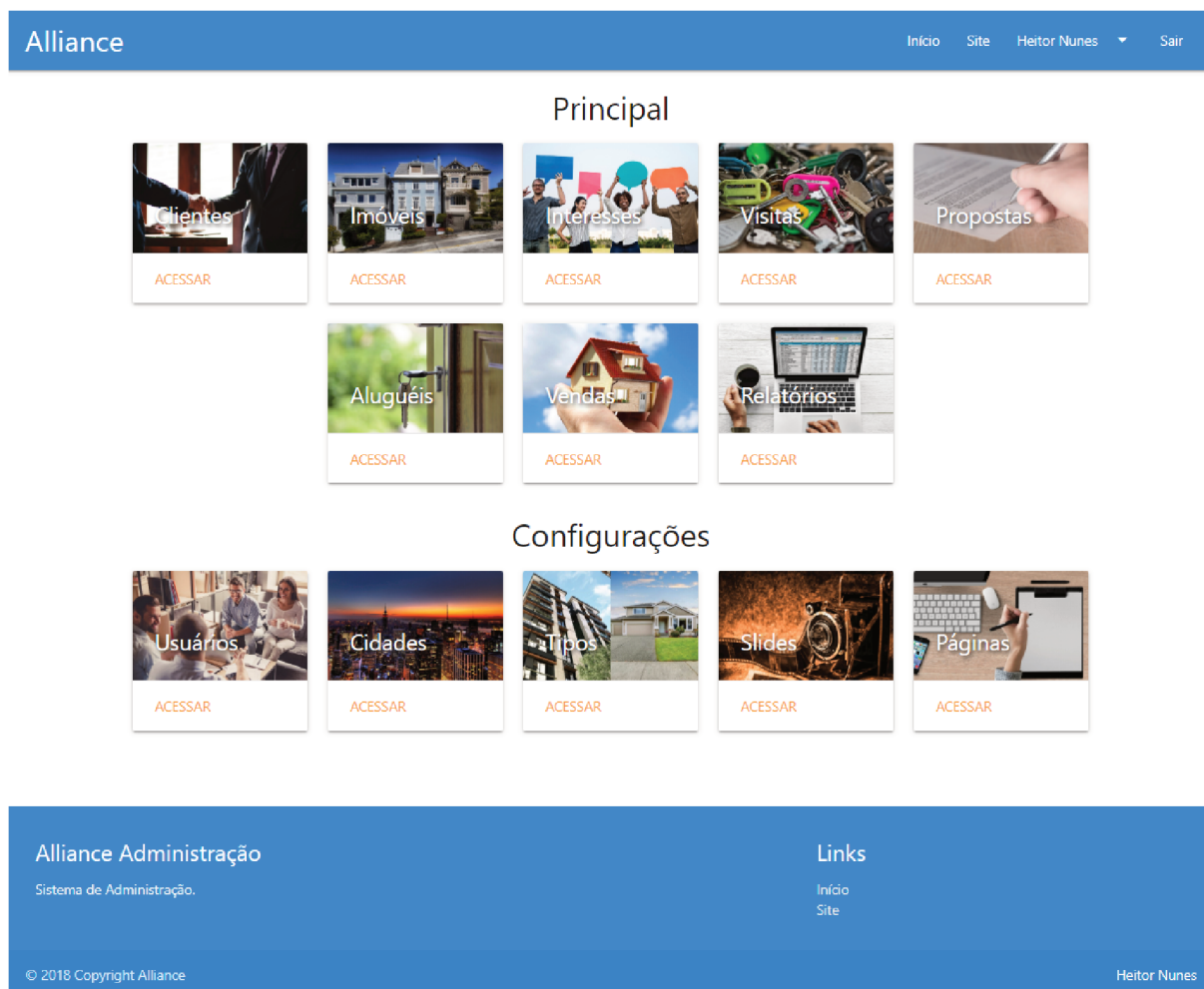


Figura 8 – Painel de gestão do portal

da imobiliária para o imóvel em questão, onde um funcionário tem a oportunidade de visualizar afim de retornar o contato com o cliente para, por exemplo, agendar uma visita ou passar mais informações sobre o imóvel.

A área pública do portal possui ainda uma página institucional personalizada, onde a imobiliária pode colocar informações sobre a empresa, endereço e telefone de contato.

Na área restrita, acessível apenas após efetuar *login*, encontram-se as demais funcionalidades do sistema. Esta área possui um painel de gestão do portal (Figura 8) onde se encontram inúmeros cartões que dão acesso a todas os cadastros do sistema, separados em regras de negócio e configurações do portal. Para evitar que haja necessidade de retornar a este painel sempre que quiser trocar de uma rotina para outra, o menu superior também traz um *dropdown* com acesso direto a todos os recursos.

Em seu primeiro acesso o funcionário deverá cadastrar todas as cidades que a imobiliária atua e os tipos de imóveis que comercializa. Estes tipos estão restritos a casas ou apartamentos, mas não há limite para subtipos, por exemplo, pode-se registrar mansões que pertencem ao tipo casa ou duplex e triplex que pertencem ao tipo apartamento. Esta

Lista de Imóveis

Início > Lista de Imóveis

Titulo	Proprietário	Valor	Oferta	Imagem	Publicado	Ação
Apartamento no Centro	Maria Joaquina Antunes	R\$ 890.000,00	Vende		Sim	EDITAR INTERESSES DELETAR
Casa no Jardim Karaiba	Heitor Henrique Nunes	R\$ 1.500.000,00	Vende		Nao	EDITAR INTERESSES DELETAR
Casa no Santa Mônica	Claudio Gonçalves Brito	R\$ 4.000,00	Aluga		Sim	EDITAR INTERESSES DELETAR
Apartamento no Saraiva	Gabriela Afonso Dias	R\$ 4.500,00	Aluga		Sim	EDITAR INTERESSES DELETAR

ADICIONAR

Figura 9 – Listagem de imóveis

possibilidade deixa o portal mais flexível e customizado. As cidades e tipos cadastrados serão exibidos como opções na ferramenta de busca da página inicial do portal, além de estarem presentes como campos no cadastro dos imóveis.

Estas são as premissas necessárias para utilizar o restante do sistema. A partir desse ponto já é permitido inserir clientes, imóveis, interesses, visitas, propostas, aluguéis e vendas. Entretanto, na parte de configurações o funcionário pode, ainda, criar usuários para mais colaboradores, inserir anúncios em destaque e até editar a página institucional ou personalizar a página de contato com a imobiliária.

Todos os CRUDs do sistema possuem o mesmo funcionamento. Ao acessar um cadastro abre-se uma tela com a lista de todos os registros existentes com seus detalhes mais relevantes (Figura 9), e para cada item alguns botões de ação que variam de acordo com a entidade (por exemplo, na lista de imóveis há um botão de ação exclusivo que mostra uma lista com todos os interesses registrados nele, enquanto que na lista de clientes há um botão de ação adicional que mostra os imóveis pertencentes àquele proprietário).

Quando um colaborador desejar excluir algum registro é necessária uma confirmação final, pois todas entidades foram configuradas para realizar uma exclusão em cascata dos registros filhos conforme requisito R. 15, evitando assim falhas referenciais. Já incluir ou editar um registro direciona o usuário para a mesma tela, porém o sistema trata a forma que ela será apresentada de acordo com a ação que originou o acesso à página.

A maioria dos campos nos formulários é do tipo textual. Logo, esses campos estão sujeitos a serem preenchidos de forma equivocada. Para tratar isto o sistema conta

Editar Cliente

Início > Lista de Clientes > Editar Cliente

Tipo

☒ Proprietário

☐ Comprador

☐ Inquilino

Nome Completo

Heitor Henrique Nunes

CPF

499.174.740-64

Telefone

(34) 98888-5457

E-mail

heitor@mail.com

Endereço de e-mail

CEP

38400-239

Cidade

Uberlândia

Endereço

Rua Professor João Alves Castilho, 1879

ATUALIZAR

ADICIONAR IMÓVEL

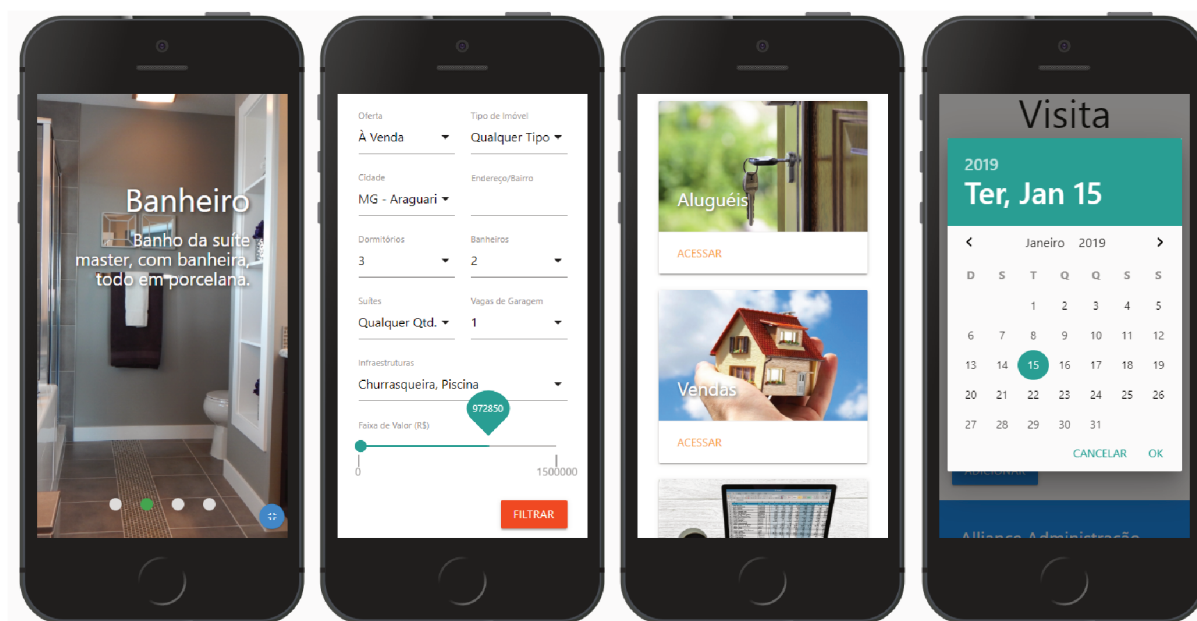
Figura 10 – Cadastro de cliente

com validações pontuais que evitam alguns erros de preenchimento como, por exemplo, verificações de CPF, telefone, e-mail, CEP, valores e taxas.

É necessário sempre cadastrar o cliente proprietário antes de registrar seu imóvel. Vale destacar que no cadastro de clientes (Figura 10) é necessário definir se ele é proprietário, comprador ou inquilino, podendo acumular papéis. No cadastro de um imóvel o sistema deixará relacionar a ele apenas os clientes que tiverem o papel de proprietário. Da mesma forma, vendas só são permitidas aos compradores e aluguéis somente aos inquilinos.

Após o sistema possuir propriedades cadastradas a imobiliária poderá registrar interesses a determinado imóvel. Um colaborador pode incluir interesses através do sistema nos casos em que o cliente procura a imobiliária, ou o próprio sistema pode inseri-los quando um cliente preenche o formulário de contato na página do anúncio. Um interesse contém uma mensagem, que pode ser uma dúvida do cliente em relação ao imóvel, uma intenção de compra, ou uma solicitação de visita. A partir das informações registradas o funcionário entrará em contato com o cliente e pode agendar uma visita à propriedade, quando for o caso.

Um imóvel pode receber vários interesses, visitas e propostas. Uma proposta é uma posição de valor feita pelo interessado ao proprietário, por intermédio da imobiliária. O valor da proposta não é, necessariamente, o valor do anúncio original, permitindo assim uma negociação pelas partes, e tais propostas podem ser recusadas ou aprovadas.

Figura 11 – Telas do portal visualizadas em um *smartphone*

Finalmente, quando uma proposta for aprovada, um contrato poderá ser registrado no sistema.

O contrato contempla a concretização da comercialização, seja de compra ou locação, e seu registro automaticamente oculta o imóvel do portal de anúncios. Caso este contrato seja aluguel, um prazo deverá ser firmado e, ao fim deste, a imobiliária deverá disponibilizar o imóvel novamente. O registro do contrato suporta o carregamento de um arquivo para, em geral, servir como um repositório de documentos contratuais assinados, facilitando a consulta em situações futuras.

Como diversas informações são registradas no sistema, foi desenvolvido uma funcionalidade para gerar relatórios, onde a imobiliária poderá extrair relatórios de venda, aluguel ou os dois em um determinado período. Com este relatório é possível analisar todas as vendas e alugueis daquele período, incluindo os valores referente a comissões, dando uma ferramenta para tomada de decisões ao gerente da imobiliária.

O sistema foi totalmente projetado pensando em facilitar a experiência do usuário. Tanto a área pública quanto a restrita foram desenhadas para que o acesso e a compreensão de cada funcionalidade se torne mais agradável. Foram utilizados rótulos, ícones, imagens e até prefixos ou sufixos para indicar com maior precisão o que está sendo requerido em determinado campo. O site foi totalmente criado pensando no conceito de responsividade (Figura 11). É possível acessar qualquer uma das funcionalidades do sistema de um celular, tablete ou computador, porém em dispositivos menores algumas funcionalidades estarão recolhidas enquanto outras precisarão de dar uma rolagem na tela, porém sempre com a ideia de melhorar a experiência do usuário de acordo com o dispositivo em uso.

CAPÍTULO 5

Conclusão

Ao final do projeto foi implementada uma aplicação *web* voltada ao comércio de imóveis. Com o objetivo de facilitar a usabilidade dos funcionários, o sistema pode ser acessado tanto em seus computadores quanto em *smartphones* ou outros dispositivos menores. Além disso, esta portabilidade também atende as necessidades de clientes e deixa sua experiência no negócio mais agradável.

O sistema apresentado foi de extrema importância para ampliar o conhecimento acadêmico e ajudar a construir o lado profissional. Este objetivo só foi alcançado graças aos conhecimentos adquiridos durante a formação acadêmica em Sistemas de Informação, principalmente aos relacionados às disciplinas de Banco de Dados, Programação para Internet, Modelagem de Software, Redes de Computadores e Programação Orientada a Objetos, além de estudos adicionais com foco nos *frameworks* utilizados.

Durante o desenvolvimento foi necessário superar alguns desafios. A modelagem do banco de dados foi refeita inúmeras vezes até se obter um modelo adequado, pois haviam diversas formas de representar o mesmo cenário. Além disso, adaptar todas as telas do sistema para que fossem reproduzidas fielmente em diferentes dispositivos exigiu inúmeros testes e modificações de *layout*.

Almejo continuar o desenvolvimento do sistema implementando novas funcionalidades afim de torná-lo mais atrativo comercialmente, tais como melhorar o filtro de imóveis, adicionando uma busca por mapa, no qual o usuário poderá escolher seu imóvel com base em uma localização geográfica, ampliar os tipos de imóveis, possibilitando comercializar lotes, fazendas e pontos comerciais, dentre outros, além de também adicionar restrições de acesso às áreas do sistema de acordo com o cargo dos funcionários, como restringindo relatórios aos gerentes e cadastros aos atendentes.

Referências

- ADERMANN, N.; BOGGIANO, J. **Composer Dependency Manager for PHP**. 2012. <<https://getcomposer.org/>>. (Accessado em 11/05/2018).
- AIRBNB. **Descubra Lugares Incríveis**. 2018. <<https://www.airbnb.com.br/>>. (Accessado em 24/06/2018).
- BECODE. **O que é PHP e porque você precisa aprender HOJE!** 2017. <<https://becode.com.br/o-que-e-php/>>. (Accessado em 24/06/2018).
- CANALTECH. **O que é e como funciona a linguagem JavaScript?** 2015. <<https://canaltech.com.br/internet/O-que-e-e-como-funciona-a-linguagem-JavaScript/>>. (Accessado em 24/06/2018).
- DEVMEDIA. **Conceitos Fundamentais de Banco de Dados**. 2006. <<https://www.devmedia.com.br/conceitos-fundamentais-de-banco-de-dados/1649>>. (Accessado em 24/06/2018).
- _____. **O que é o HTML5**. 2012. <<https://www.devmedia.com.br/o-que-e-o-html5/25820>>. (Accessado em 24/06/2018).
- ELMASRI, R.; NAVATHE, S. **Sistemas de banco de dados**. [S.l.]: ADDISON WESLEY BRA, 2009. ISBN 9788588639171.
- GOOGLE. **Mapas personalizados - Google Maps Platform - Google Cloud**. 2005. <<https://cloud.google.com/maps-platform/maps/?hl=pt>>. (Accessado em 24/06/2018).
- _____. **Material Design**. 2014. <<https://material.io/>>. (Accessado em 11/05/2018).
- INGAIA. **Entenda o crescimento do mercado imobiliário pelo mundo**. 2018. <<http://www.ingaia.com.br/entenda-o-crescimento-do-mercado-imobiliario-pelo-mundo/>>. (Accessado em 11/05/2018).
- MATERIALIZE. **Materialize CSS**. 2014. <<https://materializecss.com/>>. (Accessado em 11/05/2018).
- OTWELL, T. **Laravel - The PHP Framework For Web Artisans**. 2011. <<https://laravel.com/>>. (Accessado em 11/05/2018).

PROVOS, N.; MAZIERES, D. **USENIX Annual Technical Conference**. 1999. <https://www.usenix.org/legacy/event/usenix99/provos/provos_html/index.html>. (Accessado em 17/11/2018).

SALESFORCE. **Cloud Application Platform - Heroku**. 2007. <<https://www.heroku.com/>>. (Accessado em 11/05/2018).

SAS, S. **Symfony, High Performance PHP Framework for Web Development**. 2005. <<https://symfony.com/>>. (Accessado em 11/05/2018).

SMARTWEB. **Plataforma de E-commerce e Desenvolvimento de Websites**. 2000. <<http://www.smartweb.com.br/>>. (Accessado em 11/05/2018).

TECMUNDO. **O que é CSS?** 2009. <<https://www.tecmundo.com.br/programacao/2705-o-que-e-css-.htm>>. (Accessado em 24/06/2018).

TRENDS, G. **Laravel, CodeIgniter, Symfony, Cakephp, Zend - Comparação do Google Trends**. 2019. <<https://trends.google.com.br/trends/explore?q=laravel,CodeIgniter,symfony,cakephp,Zend>>. (Accessado em 28/05/2019).

UDIMÓVEIS. **Udimóveis Imobiliária**. 2018. <<http://www.udimoveis.com.br/>>. (Accessado em 24/06/2018).