



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE AUTOMAÇÃO E SISTEMAS
CURSO DE GRADUAÇÃO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Marina Silva Tavares

Impacto do uso do *framework* AMP na taxa de conversão do agendamento de visitas de imóveis na página de descrição de um imóvel em um SaaS de imobiliária

Florianópolis
2020

Marina Silva Tavares

Impacto do uso do *framework* AMP na taxa de conversão do agendamento de visitas de imóveis na página de descrição de um imóvel em um SaaS de imobiliária

Relatório final da disciplina DAS5511 (Projeto de Fim de Curso) como Trabalho de Conclusão do Curso de Graduação em Engenharia de Controle e Automação da Universidade Federal de Santa Catarina em Florianópolis.

Orientador: Prof. Rômulo Silva de Oliveira, Dr.
Supervisor: Andrio Renan Gonzatti Frizon, Eng.

Florianópolis
2020

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Tavares, Marina Silva

Impacto do uso do framework AMP na taxa de conversão do agendamento de visitas de imóveis na página de descrição de um imóvel em um SaaS de imobiliária / Marina Silva Tavares ; orientador, Rômulo Silva de Oliveira, 2020.

77 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Centro Tecnológico, Graduação em Engenharia de Controle e Automação, Florianópolis, 2020.

Inclui referências.

1. Engenharia de Controle e Automação. 2. Javascript. 3. Accelerated Mobile Page. 4. front-end. 5. performance em dispositivos móveis. I. Silva de Oliveira, Rômulo. II. Universidade Federal de Santa Catarina. Graduação em Engenharia de Controle e Automação. III. Título.

Este trabalho é dedicado a minha família, aos colegas
de curso e de trabalho.

AGRADECIMENTOS

Agradeço aos meus pais, meus avós e ao meu irmão pelo apoio incondicional e paciência durante todos esses anos de graduação, não conseguiria passar por tudo isso sem a incrível base familiar que vocês me propuseram. Gostaria de também agradecer ao meu namorado, Vinicius, por todo o incentivo, compreensão e suporte emocional durante esse período.

A todos os meus colegas durante a graduação e principalmente à turma de 15.2 e 16.2 que me acolheram após a transferência à UFSC Florianópolis, deixo o meu agradecimento por se tornarem a minha família dentro da universidade. Agradeço também à todo corpo docente do DAS e ao do curso de ECA em Blumenau, por todos os anos de aprendizado. Em agradecimento e memória também ao professor Cláudio Melo, que me ajudou em todo o processo de escolha de curso e me motivou a ser uma Engenheira.

Gostaria de agradecer também à Jungle Devs por toda a confiança durante o período de estágio e pela oportunidade de fazer parte do projeto Woliver. Agradeço pelo Marco e Igor por todo apoio durante a realização do projeto, e também à Michelle e Augusto, por serem os meus mentores e por contribuírem ao meu crescimento pessoal e profissional durante tudo esse processo.

RESUMO

Este documento é referente ao desenvolvimento do Projeto de Fim de Curso realizados entre o mês de Agosto e Novembro de 2020 na empresa Jungle Devs. O projeto consiste na aplicação da tecnologia Accelerated Mobile Pages (AMP) em um SaaS de imobiliária, o Canal Digital, na página de descrição de um imóvel disponível para a locação, objetivando uma melhora na performance na página e a análise do impacto dessa na taxa de conversão do número de agendamentos realizados na plataforma, na qual foram adquiridos parâmetros a partir de um teste A/B. Será apresentada brevemente a empresa Jungle Devs, o seu atual funcionamento e a metodologia utilizada para o gerenciamento do projeto (como as metodologias *Agile* e *Scrum*). Para a compreensão do uso da tecnologia serão explicados conceitos básicos utilizados para o desenvolvimento das tarefas, o atual escopo do projeto, as imobiliárias já inscritas e as atividades desenvolvidas para a aplicação do AMP. Por fim, será analisado o impacto do uso da tecnologia, a comparação da performance na página em ReactJS e AMP e a análise do desenvolvimento.

Palavras-chave: AMP, Front-end, Javascript, Desenvolvimento Web, ReactJS, Performance em Mobile.

ABSTRACT

This document refers to the development of the End of Course Project carried out between August and November 2020 at Jungle Devs. The project consists of the application of AMP technology in a real estate SaaS, Canal Digital, on the description page of a property available for rental, aiming at improving the performance on the page and analyzing the impact of this on the rate of conversion of the number of appointments made on the platform, in which parameters were acquired from an A/B test. The company Jungle Devs, its current operation and the methodology used for project management (such as Agile and Scrum) will be briefly presented. To understand the use of technology, basic concepts used for the development of tasks will be explained, the current scope of the project, the real estate agencies already registered and the activities developed for the application of AMP. Finally, the impact of the use of technology, the comparison of performance on the page in ReactJS and AMP and the analysis of development will be analyzed.

Keywords: AMP, Front-end, Javascript, Web Development, ReactJS, Mobile Performance.

LISTA DE FIGURAS

Figura 1 – Tipos de dispositivos usados por usuários ativos na internet	13
Figura 2 – Logo da empresa Jungle Devs	16
Figura 3 – Funcionamento de uma <i>sprint</i>	18
Figura 4 – Funcionamento da metodologia Kanban	19
Figura 5 – Classificação dos <i>Story Points</i>	20
Figura 6 – Informação de uma atividade de uma <i>sprint</i>	21
Figura 7 – Exemplo de estruturação em HTML	23
Figura 8 – Exemplo de elementos no DOM	23
Figura 9 – <i>Website</i> com e sem estilização	24
Figura 10 – Representações das diferentes implementações do Cascading Style Sheets (CSS) em um HTML	24
Figura 11 – Funcionamento de um <i>website</i> CSR	26
Figura 12 – Funcionamento de um <i>website</i> SSR	27
Figura 13 – Exemplo de componentização do <i>ReactJS</i>	28
Figura 14 – Representação do funcionamento de componentes, propriedades e estados no <i>ReactJS</i>	29
Figura 15 – Representação site AMP em busca no <i>Google</i>	31
Figura 16 – Fluxograma do agendamento de visita em <i>desktop</i>	38
Figura 17 – Gráfico de linhas com dados das imobiliárias com mais visualizações únicas	39
Figura 18 – Gráfico de linhas com dados das imobiliárias com menos visualiza- ções únicas	39
Figura 19 – Gráfico de linhas com dados das imobiliárias com menos visualiza- ções únicas	40
Figura 20 – Gráfico de áreas com tipos de dispositivo utilizado por usuário	40
Figura 21 – Gráfico de áreas com tipos de dispositivo utilizado por novos usuários	41
Figura 22 – Página da descrição do imóvel em <i>mobile</i>	42
Figura 23 – Página da descrição do imóvel em <i>desktop</i>	43
Figura 24 – Notas da página de descrição de um imóvel no <i>lighthouse</i>	44
Figura 25 – Nova User Interface (UI) no <i>mobile</i>	45
Figura 26 – Nova UI no <i>desktop</i>	46
Figura 27 – Código obrigatório para a renderização do AMP	48
Figura 28 – Estado de fotos selecionados na seção de seleção de mídias em dispositivos móveis	49
Figura 29 – Estado de fotos selecionados na seção de seleção de mídias em <i>desktop</i>	50
Figura 30 – Modal com todas as fotos do imóvel	50

Figura 31 – Estado de vídeo selecionados na seção de seleção de mídias em <i>mobile</i>	51
Figura 32 – Estado de vídeo selecionados na seção de seleção de mídias em <i>desktop</i>	51
Figura 33 – Estado de Mapa e Rua selecionados na seção de seleção de mídias em <i>mobile</i>	52
Figura 34 – Estado de Mapa e Rua selecionados na seção de seleção de mídias em <i>desktop</i>	52
Figura 35 – Informações gerais do imóvel em <i>mobile</i>	53
Figura 36 – Informações gerais do imóvel em <i>desktop</i>	54
Figura 37 – Cartões informativos em <i>mobile</i>	55
Figura 38 – Cartões informativos em <i>desktop</i>	56
Figura 39 – Imóveis semelhantes em <i>mobile</i>	56
Figura 40 – Imóveis semelhantes em <i>desktop</i>	57
Figura 41 – Rodapé em <i>mobile</i>	57
Figura 42 – Cabeçalho em <i>desktop</i> com o botão de opções selecionados	58
Figura 43 – Rodapé em <i>desktop</i>	58
Figura 44 – Fluxograma do teste A/B	59
Figura 45 – Notas da página de descrição de um imóvel em AMP no <i>Lighthouse</i>	61
Figura 46 – Comparação entre tempo de carregamento entre a página em <i>ReactJS</i> e AMP	63
Figura 47 – Comparação entre o progresso visual entre a página em <i>ReactJS</i> e AMP	63
Figura 48 – Tipos de requisições para o carregamento para as páginas em AMP e <i>ReactJS</i>	64
Figura 49 – Gráfico de pizza com dados do tipo de página de descrição do imóvel visitada	65
Figura 50 – Gráfico de pizza com o percentual de agendamento realizados com base na página de descrição do imóvel de origem	67

LISTA DE TABELAS

Tabela 1 – Lista de imobiliárias registradas no Woliver	36
Tabela 2 – Comparação entre resultados obtidos no <i>Lighthouse</i> entre página em <i>ReactJS</i> e AMP	62
Tabela 3 – Número de visitas na página de descrição de imóvel nas imobiliárias	65
Tabela 4 – Taxa de rejeição na página de descrição do imóvel nas imobiliárias	66
Tabela 5 – Número de visitas agendadas nas imobiliárias por tipo de página acessada	66
Tabela 6 – Taxa de conversão de agendamento de visitas	67

LISTA DE ABREVIATURAS E SIGLAS

AMP	Accelerated Mobile Pages
API	Application Programming Interface
CSS	Cascading Style Sheets
DOM	Document Object Model
GA	Google Analytics
HTML	Hypertext Markup Language
HTTP	Hyper Text Transfer Protocol
HTTPS	Hyper Text Transfer Protocol Secure
JSON	JavaScript Object Notation
PO	Product Owner
QA	Quality Assurance
SaaS	Software as a Service
SEO	Search Engine Optimization
SSR	Server Side Rendering
UI	User Interface
URL	Uniform Resource Locator

SUMÁRIO

1	INTRODUÇÃO	13
1.1	MOTIVAÇÃO	13
1.2	OBJETIVOS	14
2	EMPRESA	16
2.1	SOBRE A JUNGLE DEVS	16
2.1.1	Programa Academy	17
2.2	GERENCIAMENTO DE PROJETO	17
2.2.1	Story Points	19
2.2.2	Jungle Rocks	20
3	CONCEITOS BÁSICOS	22
3.1	HTML	22
3.1.1	Document Object Model	22
3.2	CSS	22
3.3	JAVASCRIPT	25
3.3.1	Client Side Rendering	25
3.3.2	Server Side Rendering	26
3.4	REACTJS	27
3.5	ACCELERATED MOBILE PAGES	29
3.6	GIT	31
3.7	SOFTWARE AS A SERVICE	31
3.8	WHITE LABEL	32
3.9	GOOGLE ANALYTICS	32
3.10	GOOGLE DATA STUDIO	32
3.11	GOOGLE LIGHTHOUSE	32
3.11.1	Performance	33
3.11.2	Acessibilidade	33
3.11.3	Boas práticas	33
3.11.4	SEO	34
4	CANAL DIGITAL	35
4.1	O PROJETO	35
4.1.1	Agendamento de visita	36
4.2	COMPORTAMENTO DOS USUÁRIOS NA PLATAFORMA	37
4.3	A PÁGINA DE DESCRIÇÃO DO IMÓVEL	40
4.3.1	Performance da página	42
5	ATIVIDADES DESENVOLVIDAS	45
5.1	NOVA INTERFACE DA PÁGINA DE DESCRIÇÃO DO IMÓVEL	45
5.2	DESENVOLVIMENTO DA PÁGINA EM AMP	46

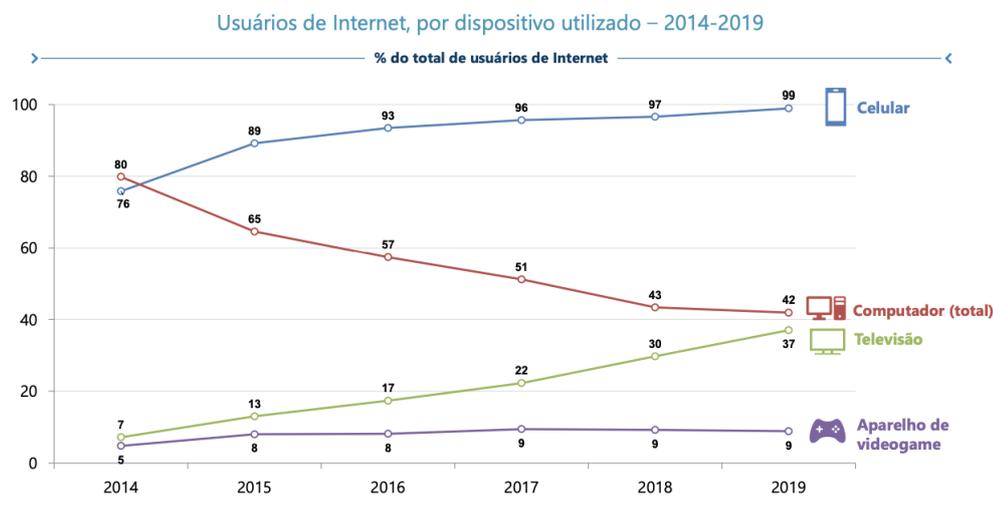
5.2.1	Seleção de Mídias	47
5.2.1.1	Fotos	49
5.2.1.2	Vídeo	50
5.2.1.3	Mapa e Rua	51
5.2.2	Informações gerais do imóvel	52
5.2.3	Valor de aluguel e ação de agendar visita	53
5.2.4	Deslocamento	54
5.2.5	Cartões informativos	54
5.2.6	Imóveis parecidos	55
5.2.7	Cabeçalho e Rodapé	56
5.2.8	Google Analytics	58
5.3	TESTE A/B	58
6	ANÁLISE DOS RESULTADOS	61
6.1	PERFORMANCE DA PÁGINA	61
6.1.1	Google Lighthouse	61
6.1.2	WebPageTest.org	62
6.2	IMPACTO DO USO AMP NA PLATAFORMA	64
6.2.1	Página de descrição do imóvel	64
6.2.2	Taxa de conversão no número de agendamento de visitas	66
6.3	RESUMO	67
6.4	PERSPECTIVAS FUTURAS	68
7	CONCLUSÃO	69
	REFERÊNCIAS	71

1 INTRODUÇÃO

1.1 MOTIVAÇÃO

De acordo com a pesquisa TIC Domicílio 2019, realizada pelo Centro Regional de Estudos para Desenvolvimento da Sociedade da Informação, 71% da população brasileira possui acesso à internet (CETIC, 2019). Como é possível analisar no gráfico representado na figura 1, a grande maioria dos usuários ativos (99%) preferem utilizar o celular como o principal meio de navegação à internet, e 58% dos usuários ativos só possuem o dispositivo móvel como meio de acesso.

Figura 1 – Tipos de dispositivos usados por usuários ativos na internet



Fonte – Pesquisa TIC Domicílio 2019

Levando esses dados em consideração, deve-se garantir que a navegação na *web* seja acessível principalmente a todos os dispositivos móveis, garantindo que o usuário tenha uma boa experiência em um *website* tanto no computador quanto no celular. Porém, não é o que acontece atualmente. Em uma pesquisa realizada pelo *Google Research* em 2017, o tempo médio para uma página carregar totalmente em um celular é de 22 segundos (SPERO, 2017) o que remete ao grande problema: dos usuários que acessam uma página num celular cerca de 53% desistem de acessar *websites* caso esse demore mais do que 3 segundos para carregar. Ou seja, quanto mais demorado ou menos performático um *website* for, maior será a taxa de evasão.

Observando essa diminuição da performance dos *websites* na internet, o *Google*, como principal mecanismo de busca na internet, desenvolveu uma tecnologia chamada AMP (*Accelerated Mobile Pages*) que promete uma renderização até 4x mais rápida do que uma página normal de Hypertext Markup Language (HTML). O que se torna muito atrativo para a comunidade *web*, considerando que cada segundo pode

impactar na taxa de conversão (percentual utilizado para mensurar resultados (MORAED, 2019)) do site em 20%, ou seja, para uma melhor experiência do usuário final o site deve ser o mais rápido possível.

Tendo em vista essa ascensão do uso dos celulares e a popularização da implementação do AMP em projetos de *front-end*, a Jungle Devs, empresa onde foi realizado este trabalho de PFC, gostaria de estudar a eficiência e observar o impacto do uso do AMP na taxa de conversão em um dos maiores projetos da empresa: o Canal Digital, uma plataforma web de imobiliárias.

1.2 OBJETIVOS

O objetivo deste PFC foi o estudo do *framework* criado pelo Google, o *Accelerated Mobile Pages* e a implementação desse num *SaaS* de imobiliárias chamado Canal Digital, o qual possui como principal funcionalidade o agendamento de visitas. Logo será medindo o impacto, através de um teste A/B, do uso dessa tecnologia através do número de agendamentos de visitas realizadas, com a análise final da taxa de conversão de visitas marcadas.

ESTRUTURA DO DOCUMENTO

Este documento será formado por mais 6 capítulos, sendo estruturados da seguinte maneira:

No capítulo 2 será apresentada a empresa na qual foi realizada o projeto, a Jungle Devs, como também a sua atual estruturação, as metodologias utilizadas para gerenciamento de projeto bem como o programa *Academy* oferecido pela empresa.

O capítulo 3 contém o conteúdo referente aos conceitos básicos para a compreensão das atividades posteriormente desenvolvidas. Aqui será descrito as linguagens de programação utilizadas, o funcionamento de bibliotecas ou *frameworks* essenciais e também serviços essenciais para a aquisição de indicadores para a análise do impacto do projeto em questão.

No capítulo 4 apresenta-se o produto no qual será aplicado o trabalho realizado nesse documento, o Canal Digital, indicando o funcionamento do principal fluxo e a problemática a ser tratada.

No capítulo 5 serão apresentadas todas as atividades realizadas no trabalho, no qual é explicado o processo de desenvolvimento de cada seção da página de informações de um imóvel através do uso da tecnologia AMP e também a aplicação do teste A/B no Server Side Rendering (SSR) do Canal Digital.

Já no capítulo 6, serão apresentadas as análises do que foi aplicado no Canal Digital e o impacto do uso do AMP na plataforma através da taxa de conversão no

número de visitas realizadas, e a diferença na performance entre uma página em AMP e em *ReactJS*.

Por fim, no capítulo 7 apresenta-se as conclusões da tecnologia aplicada visando os itens analisados e serão levantadas perspectivas futuras da aplicação da tecnologia no projeto.

2 EMPRESA

Neste capítulo será retratado a empresa na qual foi desenvolvido o projeto: a Jungle Devs. Tratando um pouco da sua história, seu posicionamento atual, o programa de estágio oferecido e também a metodologia adotada para o gerenciamento de projeto.

2.1 SOBRE A JUNGLE DEVS

A Jungle Devs é uma empresa do setor de TI sediada em Florianópolis, atuando na área de desenvolvimento de software com o foco em produtos *Web* e *Mobile*. São oferecidos serviços na área de *front-end* (utilizando a tecnologia *ReactJS* para *web* e *Swift*, *Kotlin*, *React-Native* e *Flutter* em *mobile*), *back-end* sendo *Django* seu principal *framework*, *DevOps* trabalhando com *AWS*, Design de UI/UX, prestação de consultoria e também atuando na parte de análise de dados.

Figura 2 – Logo da empresa Jungle Devs



Fonte – Jungle Devs

A empresa foi fundada em 2018 e atualmente (novembro de 2020) possui sete sócios, todos com formação no curso de Engenharia de Controle e Automação na UFSC, e outros 48 empregados sendo eles Engenheiros de Software, Gerente de Projetos, Designers e estagiários. A missão da empresa é repensar a maneira nas quais as pessoas aprendem e criam tecnologias, o ponto de partida para isso foi a criação do programa Academy que está diretamente ligada a sua visão dos primeiros 2 anos da empresa: a consolidação do programa e também aumentar o número de produtos desenvolvidos. O principal objetivo da empresa é o desenvolvimento de pessoas.

A divisão da empresa é realizada em 4 Tribos na qual cada possui um gerente de projeto, um ou dois designers e de 4 a 5 desenvolvedores das diferentes *stacks* trabalhadas na empresa. Cada tribo trabalha com diferentes projetos, podendo ser de clientes do Brasil, Estados Unidos, Austrália e Espanha. Atualmente a empresa possui parceria com empresas brasileiras e também dos Estados Unidos, África do Sul, Polônia e Austrália. Tendo em vista os diversos clientes e cenários foi estabelecido

que a primeira língua da empresa é o inglês, sendo reforçado a necessidade dessa habilidade para se inserir no mercado de trabalho.

2.1.1 Programa *Academy*

Um dos grandes diferenciais da Jungle Devs é o seu programa *Academy* que é um estágio focado principalmente em preparar e dar condições a graduandos que estão no final do curso a adquirirem experiência e conhecimento para se inserirem no mercado de trabalho brasileiro ou exterior. O principal pré-requisito para a entrada nesse programa é a vontade de aprender, não é necessário ter o domínio de uma específica linguagem de programação.

O programa tem a duração mínima de 1 ano no qual nesse período tem contato direto com as principais tecnologias utilizadas no mercado, aprendendo conceitos, linguagens de programação e utilização das ferramentas usadas pela equipe além de ter acesso a um ambiente de desenvolvimento de software profissional. Os participantes são parte de um programa mentor-mentorado, no qual os mentores auxiliam os mentorados não só no desenvolvimento profissional, mas também como pessoal.

Antes de ter contato com projetos reais da empresa, são passados desafios para consolidar os conceitos aprendidos. São também realizados testes de Quality Assurance (QA) em projetos durante o seu desenvolvimento, os que consistem na utilização do produto para certificar o seu funcionamento, ou ainda encontrar falhas indesejados ocasionando comportamento inesperado e reportá-las, antes da entrega final ao cliente.

Conforme o estagiário aprimora o seus conhecimentos em determinada área este fará parte no desenvolvimento de projetos, realizando novas funções, participando de reuniões de planejamento junto a gerentes de projetos, designers e engenheiros de software, arrumando erros de código, criando e também fazendo lançamento de projetos.

Atualmente a Jungle Devs oferece 7 tipos diferentes de programas *academy*: *front-end* com desenvolvimento em *ReactJS* e *React-Native*, *back-end*, *mobile* com desenvolvimento em *iOS* ou *Android*, design e em gerenciamento de projetos. No momento da escrita desse documento, 20 pessoas já passaram pelo programa *Academy* e 13 delas foram efetivadas na empresa.

2.2 GERENCIAMENTO DE PROJETO

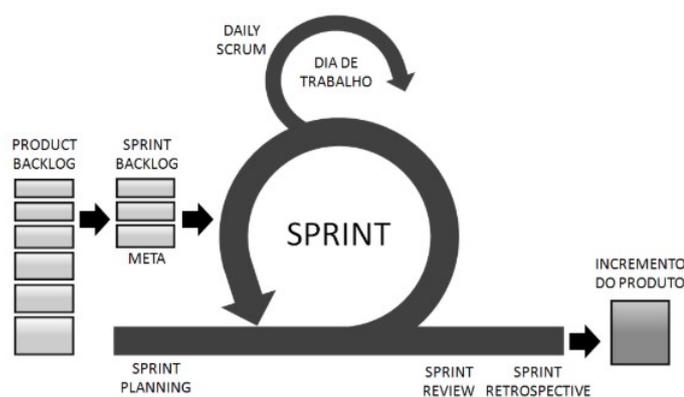
Para atender às necessidades, garantir a execução e a entrega dos produtos de *software* da empresa são utilizadas prática de metodologia ágeis (JUSTO, s.d.) na gestão dos projetos, mais especificamente o *framework* Scrum (PAULA, 2020) e o *Kanban*.

O *framework* Scrum impõe que um projeto seja dividido em ciclos de atividades com uma duração delimitada entre 1 semana até 1 mês, denominados *sprints*. O projeto final é separado em uma determinada quantidade de *sprints*, em que cada membro da equipe é alocado a trabalhar em uma quantidade delimitada de horas.

O início de um projeto, e propriamente do Scrum, é oficializado quando o cliente e o PO da empresa elaboram a visão e o objetivo na entrega final de um projeto, traçando elementos como o de um *Roadmap* do projeto, que seriam as etapas da sua realização ou uma linha do tempo das etapas a serem entregues no projeto e o *Product Backlog*, que é uma lista de requisitos do projeto ordenada de acordo com a prioridade de cliente. A cada *sprint*, são separadas algumas atividades do *Product Backlog* para serem realizadas durante esse período, sendo listadas como *Sprint Backlog* (OLIVEIRA, s.d.) e designadas para cada membro da equipe.

Na Jungle Devs cada *sprint* tem duração de duas semanas e o seu acompanhamento é realizado através de 2 principais reuniões: a *Sprint Planning Meeting* em que é uma reunião em que o PO apresenta aos membros da equipe do projeto as suas atividades esperadas para o *sprint*, e o *Sprint Review Meeting*, na qual a equipe apresenta as atividades realizadas. Além dessas reuniões, todos os dias é realizado o *Stand Up Meeting*, que seria um encontro de no máximo 20 minutos em que cada membro da equipe descreve brevemente as atividades realizadas nas últimas 24h. Esse funcionamento é representado graficamente na figura 3. Cada equipe de *scrum* na Jungle Devs seriam representadas pelas Tribos.

Figura 3 – Funcionamento de uma *sprint*

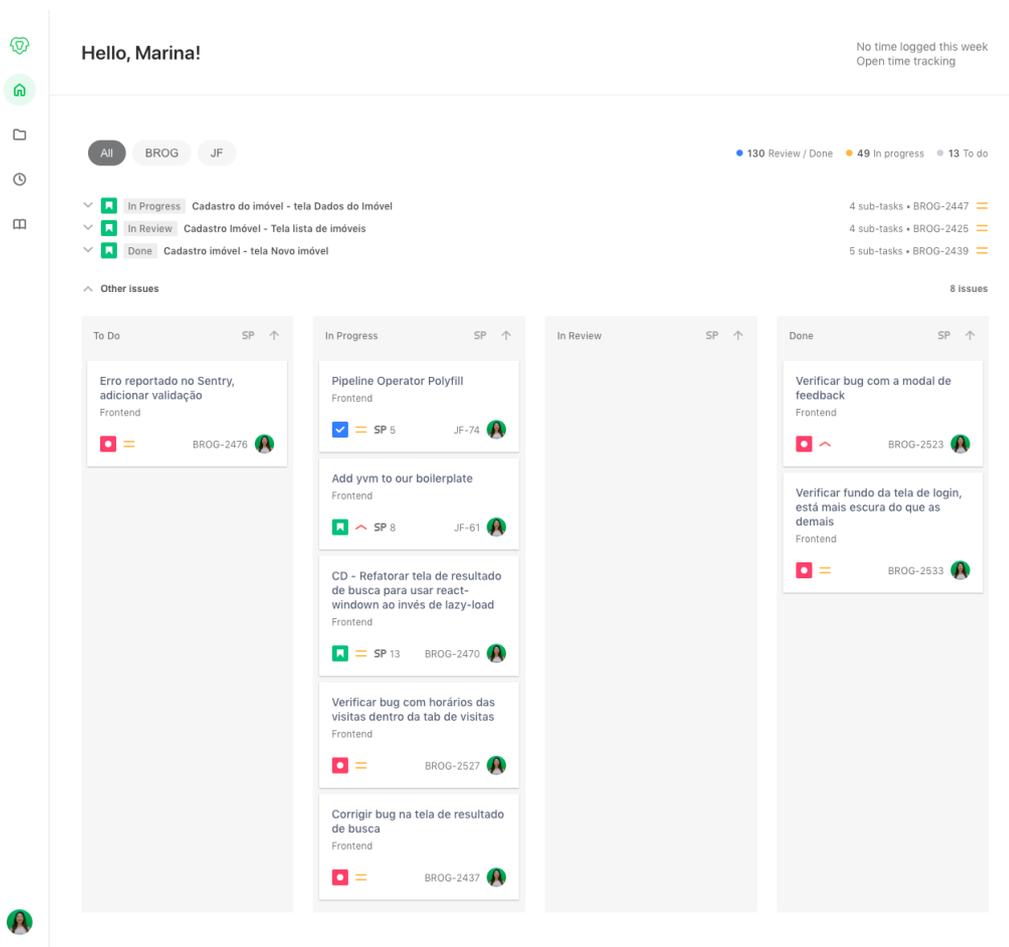


Fonte – Site evolve

Também é utilizada a metodologia Kanban (ESPINHA, s.d.), uma maneira visual de administrar o fluxo de tarefas que devem ser produzidas na empresa (*To-do*), que estão sendo feitas (*in progress*) e que estão sendo revisadas (*review*). A quarta coluna

do kanban utilizado pela empresa, demonstrado na figura 4 é utilizada pelo *Project Owner* para checar a funcionalidades das atividades realizadas.

Figura 4 – Funcionamento da metodologia Kanban



Fonte – Site Jungle Rocks

2.2.1 Story Points

Uma outra maneira de medir a dificuldade das atividades alocadas em um projeto é pelo *story points*, um artifício para medir o esforço que essa tarefa requer (LEVISON, 2010). É importante enfatizar que o tempo para realizar a atividade não é um ponto de referência para o caso. Quem define a quantidade de *story points* em cada tarefa é a pessoa que é alocada a essa.

O conjunto de *story points* serve como indicador de desempenho ao Product Owner (PO) para saber o quanto pode ser entregue ou o quanto esforço a equipe irá passar durante uma *sprint*.

Na Jungle Devs, os *story points* são determinados a partir da sequência Fibonacci, variando os números de 1 à 21, sendo 1 uma tarefa que não requer muito

esforço e 21 uma extremamente trabalhosa. Uma descrição da sequência utilizada pelos membros da empresa é representada na figura 6.

Figura 5 – Classificação dos *Story Points*

Story point	Definition
1	Extremely simple Tasks, that take almost no thinking, meaning you already know what and where in the code base the task will be accomplished
2	Intermediate between 1 and 3
3	A Task that you know what to do and requires little work
5	A Task that you know what to do and requires work
8	A Task that you know what to do and requires a lot of work
13	A Task that you don't exactly know what to do, that you are a little insecure about, but you have ideas on how to start
21	A Task that you have no idea on how to start, usually tasks that require testing, security, results validation or breaking changes, which are difficult to measure. It's important to evaluate if the task could be broken down into smaller tasks

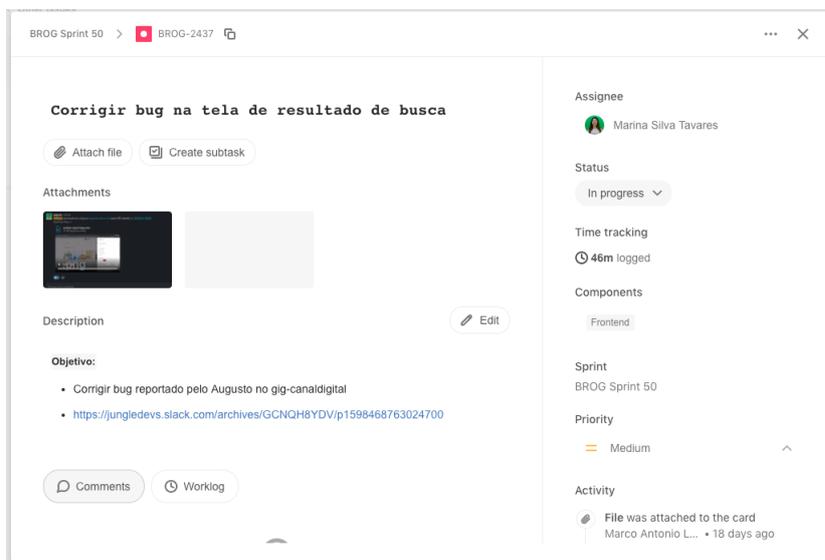
Fonte: Jungle Devs

2.2.2 *Jungle Rocks*

O *Jungle Rocks* é um software realizado pela própria empresa para automatizar diversas atividades internas da empresa, dentre elas foi realizada um desenvolvimento de uma área para o gerenciamento de projetos. Nessa o PO tem a possibilidade de popular o *sprint backlog*, atribuir atividades aos diversos membros da equipe, fazer análise da *sprint* de acordo com os *story points* preenchidos e também visualizar o *kanban* geral do projeto.

Além das seções específicas para o PO, há também a área voltada à equipe do projeto na qual cada membro tem acesso ao seu *kanban* pessoal, em que filtra as próprias tarefas da *sprint*, como demonstrado na figura 4. Em cada *card* presente no *kanban* em que ao serem clicados mostram uma *modal* com todas as informações necessárias para a elaboração da atividades.

Figura 6 – Informação de uma atividade de uma *sprint*



Fonte: Jungle Rocks

3 CONCEITOS BÁSICOS

Neste capítulo serão descritas conceitualmente as tecnologias atualmente utilizadas no projeto e as necessárias para melhorar a performance na página. Os conceitos serão fundamentais para a compreensão das soluções propostas nos próximos capítulos. As tecnologias descritas em sequência são focadas principalmente na *stack* de *front-end* do *software* em questão.

3.1 HTML

Um dos elementos essenciais para a criação e estruturação básica de uma página web é o HTML, sigla para Hypertext Markup Language ou Linguagem de Marcação de Hipertexto, o qual permite a inserção de conteúdos em um website tais como textos, imagens, vídeos, documentos, etc. O HTML não é considerado uma linguagem de programação por não possuir a habilidade de criar funções dinâmicas, mas é através dele que um navegador consegue realizar a leitura e renderização de um *website* (G., s.d.).

O funcionamento do HTML é através do conjunto de *tags*, uma *tag* é a maneira pela qual um conteúdo é marcado ou classificado em um *website*. Existem diversos tipos de *tags* como cabeçalhos, parágrafos, listas, tabelas, imagens ou ancoras e esses são utilizados de acordo com a necessidade de uma página (MARQUES, s.d.). As *tags* não são mostradas na sua renderização, mas são essenciais para o navegador renderizar o conteúdo presente em uma página através de arquivos de terminação “.html” ou “.htm”. Um exemplo da estruturação de um HTML é representada na figura 7, na qual possui cabeçalhos e parágrafos separados no conteúdo da página.

3.1.1 Document Object Model

O Document Object Model (DOM) é a maneira pela qual os navegadores leem o HTML de uma página (MALDONADO, 2019). Após a leitura é realizada uma representação estruturada, como uma árvore, onde o DOM define métodos em que a estrutura é acessada. É possível alterar os seus elementos através do *Javascript*.

Um exemplo de representação de uma página é mostrada na figura 8. O *Document* é a “raiz” do documento, os *elements* são as *tags* instanciadas no HTML, os *texts* são os textos que são colocados entre as *tags* HTML e o *attribute* é um atributo presente a um elemento HTML.

3.2 CSS

A estilização e customização das tags HTML é possível através do uso de *Cascading Style Sheets* (Folha de Estilo em Cascatas), mais conhecido como CSS

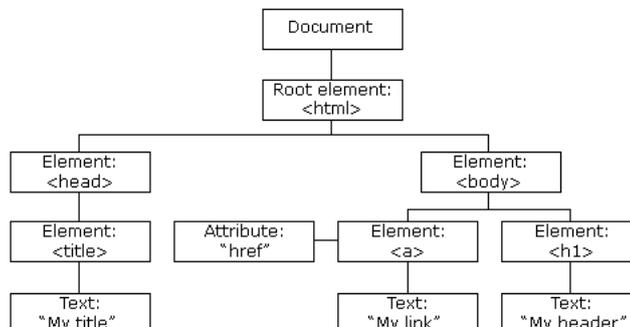
Figura 7 – Exemplo de estruturação em HTML

```

<html>
  <body>
    <h1>This is the Main Heading</h1>
    <p>This text might be an introduction to the rest of
    the page. And if the page is a long one it might
    be split up into several sub-headings.</p>
    <h2>This is a Sub-Heading</h2>
    <p>Many long articles have sub-headings to help you
    follow the structure of what is being written.
    There may even be sub-sub-headings (or lower-level
    headings).</p>
    <h2>Another Sub-Heading</h2>
    <p>Here you can see another sub-heading.</p>
  </body>
</html>
  
```

Fonte – Blog Kathep

Figura 8 – Exemplo de elementos no DOM

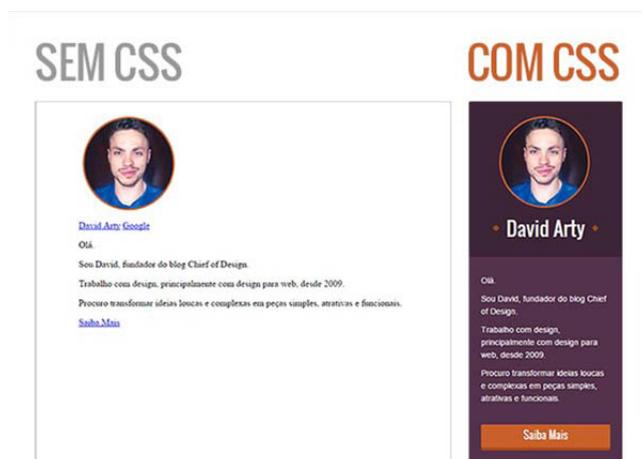


Fonte – Página w3schools

(GONÇALVES, 2019). O seu desenvolvimento foi essencial para a separação e a formatação das páginas web dando a possibilidade de incluir elementos com diferentes cores, formatos e tamanhos de fontes ou layouts entre as páginas (PEREIRA, 2020). Na figura 9 é representado um arquivo HTML com e sem CSS, no qual é possível notar as diferentes possibilidades de customização de uma seção.

É importante salientar que a existência do CSS é inteiramente ligada a um arquivo HTML, o conjunto dos arquivos dão infinitas possibilidades para a estilização de uma página web, além de também dar a possibilidade da elaboração de diferentes layouts para diferentes tipos de dispositivos, tornando uma página responsiva (CELKE, s.d.). Existem três maneiras de inserir o CSS em uma página HTML:

Figura 9 – Website com e sem estilização



Fonte – Página *Chief od Design*

1. CSS *inline*: a estilização é possível através do atributo *style* de uma *tag*;
2. CSS incorporado: Uma *tag* `<style>` com o CSS de uma página é incorporada dentro da *tag head* do arquivo HTML;
3. CSS externo: Um arquivo com a extensão “.css“ é incorporado na seção *head* do HTML;

As maneiras descritas acima são exemplificadas na figura 10.

Figura 10 – Representações das diferentes implementações do CSS em um HTML

Inline CSS

```
<p style="color: blue;">This is a paragraph.</p>
```

Internal CSS

```
<head>
  <style type = text/css>
    body {background-color: blue;}
    p { color: yellow;}
  </style>
</head>
```

External CSS

```
<head>
  <link rel="stylesheet" type="text/css" href="style.css">
</head>
```

Fonte – Página *BitDegree*

3.3 JAVASCRIPT

O *Javascript* é uma linguagem de programação utilizada amplamente pela comunidade de desenvolvimento *web*. A linguagem foi criada na década de 90 por Brendan Eich com o apoio da empresa *Netscape* para revolucionar sites estáticos (criado apenas com HTML e CSS) trazendo funcionalidades mais complexas para o desenvolvedor, como formulários interativos, animações gráficas, mapas e alertas ao usuário, funcionando em conjunto com o HTML e CSS de maneira flexível no *front-end* deixando *websites* mais dinâmicos fornecendo uma melhor experiência ao usuário final, sendo esse o principal objetivo (LONGEN, 2019b). Atualmente ela é presente em variados navegadores, dispositivos móveis e sistemas operacionais, arquivos *Javascript* são diferenciados pela extensão “.js” ou presente dentro de *tags* `<script>` nos arquivos HTML (LONGEN, 2019a).

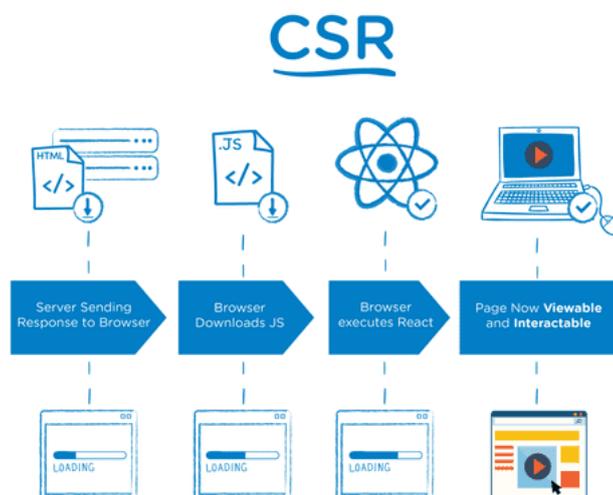
Por ser uma linguagem de programação interpretada (ou de *scripting*) não é necessário a compilação completa para a execução, sendo interpretada a cada linha pelo motor de renderização do navegador utilizado, logo após a renderização dos arquivos HTML e CSS (MORRIS, s.d.). Além de ser uma linguagem de programação de *scripting* é também estruturada, suporta estilos de programação funcionais, imperativas e orientados a eventos.

A comunidade *web* disponibiliza diversas bibliotecas que constistem na reutilização de códigos paara uma função necessário ao desenvolvimento de outros projetos, e *frameworks*: que são um conjunto de bibliotecas (junção de funções). Os principais *frameworks* utilizados sao: Angular.js e Vue.js, e as bibliotecas existentes são JQuery.js, e o ReactJS, que foi utilizado para a realização desse projeto. O uso do *Javascript* não se restringe ao *front-end*, existe uma plataforma runtime chamada NodeJS que permite o desenvolvimento de *back-end* utilizando o lado do servidor.

3.3.1 Client Side Rendering

Inicialmente o *Javascript* foi criado para ser processado sem um servidor, chamado de lado do cliente ou *client-side* (BREUX, 2018). A execução desse se dá pela requisição do HTML e do CSS de um site para um servidor específico, que é executado e após o recebimento do arquivo público o navegador irá fazer o processamento do arquivo *Javascript* e e assim que o *script* for processado a página ficará visível, o fluxo descrito é na figura 11.

O carregamento pelo motor de renderização do navegador traz como consequências a demora do carregamento da página inicial porém as páginas subsequentes apresentam uma melhor performance, uma vez que o *Javascript* já foi carregado pelo navegador.

Figura 11 – Funcionamento de um *website* CSR

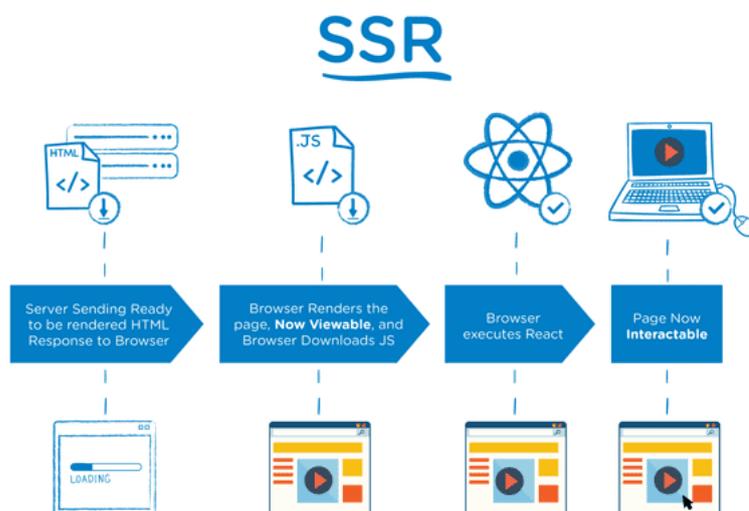
Fonte – Medium

3.3.2 Server Side Rendering

Como citado anteriormente o *Javascript* foi originalmente desenvolvido para ser processado em *client-side*. Porém, com o avanço das tecnologias e o seu uso notou-se que o tempo de carregamento pode ser maior com esse tipo de processamento, já que o CSR faz com que os códigos fonte sejam uma página vazia em que o *Javascript* renderiza todas as funcionalidades, está diretamente ligado a fatores como o tamanho da página a ser carregada ou ainda em qual dispositivo o site está sendo visualizado (ROCHA, 2018), como mostrado na figura 12. Uma das alternativas adotadas para minimizar os problemas citados foi a criação do *server-side*, o processamento do lado do servidor através da utilização do *NodeJS* (SMITH, 2020).

Para o funcionamento do SSR é necessário que o conteúdo seja convertido inteiramente para HTML (junto com o CSS) no servidor e renderizado para ser mandado novamente para o navegador. Uma das grandes vantagens do SSR é o tempo de carregamento, que tende ser menor do que o CSR, uma vez que não depende inteiramente da performance de um navegador (BURKHOLDER, 2019).

Como o SSR permite que os sites sejam retornados como um página HTML indexável, ela se torna uma boa alternativa para o uso de Search Engine Optimization (SEO), que consiste no processo de otimizar o conteúdo de um site tornando acessível para mecanismos de buscas como o Google e Bing mostrarem como resultado de uma busca de usuário, aumentando a quantidade e qualidade de tráfego para um site (WHAT... , s.d.).

Figura 12 – Funcionamento de um *website* SSR

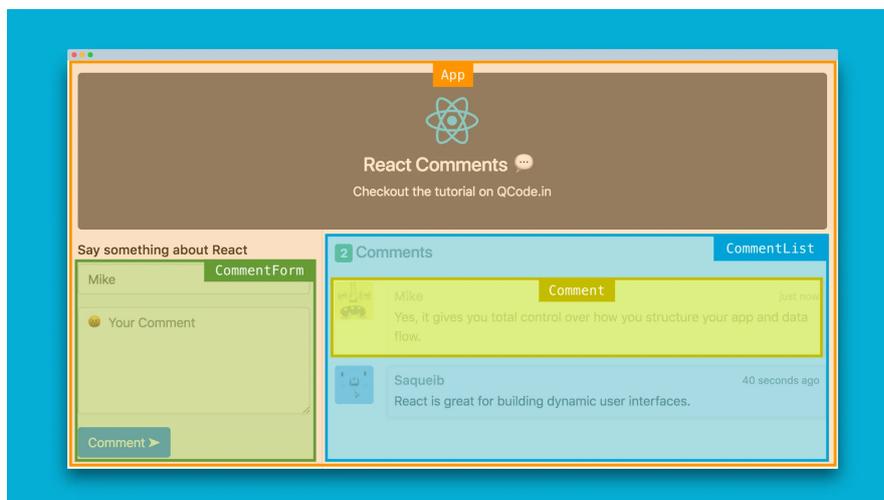
Fonte – Medium

3.4 REACTJS

Para o desenvolvimento *front-end web* do Canal Digital foi utilizada uma biblioteca de *Javascript* declarativa e focada principalmente no desenvolvimento de UI, o *ReactJS*. Foi criada em 2011 pelo *Facebook* e em 2013 foi lançada oficialmente como *open-source* e atualmente é uma das bibliotecas mais utilizadas no desenvolvimento *web* e está presente no repositório de grandes empresas de software como *Netflix*, *Airbnb*, *Instagram* e diversos outros softwares (LONGEN, 2019c).

O funcionamento do *ReactJS* é baseado principalmente em componentes que podem ser reutilizados ao longo da aplicação em que os seus dados podem variar de acordo com o tempo, uma página em *ReactJS* pode ser formada por um ou múltiplos componentes (TUTORIAL: . . . , s.d.). Um componente pode ter dois diferentes tipos de classificação: componente funcional, que são funções provenientes do *Javascript* puro podem receber propriedade de outros componentes, e componentes de classe, que além de receber propriedade de outro componente tem a habilidade de controlar o estados (JUNIOR, p. B.; JUNIOR, B., 2018). Um exemplo de componentização do *ReactJS* é representado na figura 13, na qual o componente *App* representa a página acessada, em que essa possui outros dois componentes: o *CommentForm* e o *CommentList*, no qual esse último pode ser formado por múltiplos componentes de *Comment*.

O estado de um componente contém dados ou informações necessárias para a renderização deste, podendo ser alterado por alguma ação do usuário ou por eventos

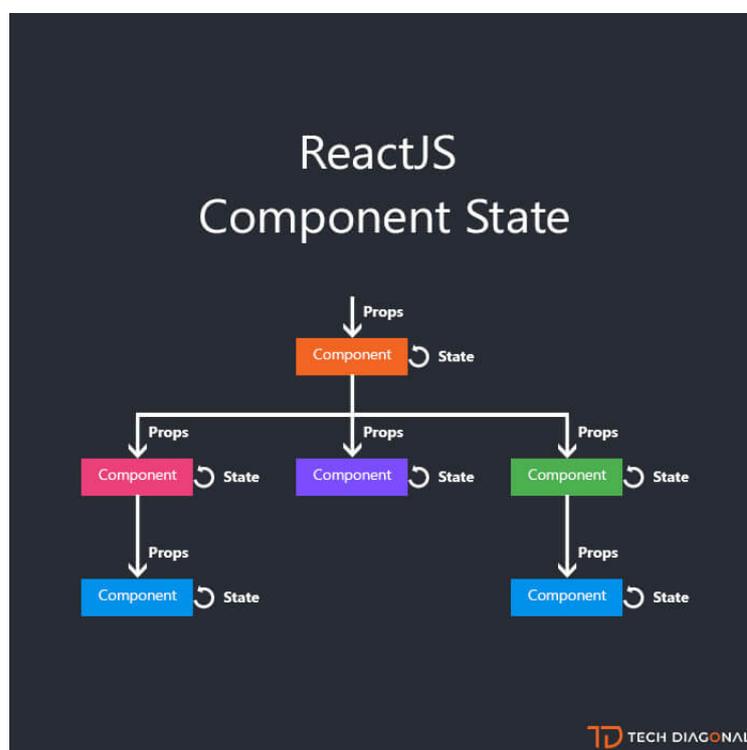
Figura 13 – Exemplo de componentização do *ReactJS*

Fonte – Página QCode

ocasionados pela aplicação ou de alguma condição gerada pelas propriedades recebidas (HOLLA, s.d.). As *props* ou propriedades de um componente funcionam como entradas de uma função, que podem alterar o funcionamento de um componente. Uma representação do funcionamento de propriedades e estados dentro de um ou vários componentes é representado na figura 14.

Uma das grandes vantagens do *ReactJS* é a *Virtual DOM*, quando um componente é atualizado em uma página *web* a DOM deveria ser atualizada de acordo com os novos elementos a serem compartilhados, porém como a manipulação na DOM é muito lenta é utilizado a *Virtual DOM*. Através de um processo chamado de reconciliação, a VDOM analisa os componentes afetados e sincroniza possíveis alterações, sem afetar muito a DOM, trazendo uma melhora significativa as aplicações *web* (HANASHIRO, 2020).

Como o *ReactJS* é uma biblioteca amplamente utilizada pela comunidade *web* há diversas *features open-source* adicionando funcionalidades extras ao projeto, tais como: *Reach-Router*, responsável pela rota dos links do projeto possibilitando o redirecionamento de um link para outro *Reach-Router*, *React Hooks*, que possibilita o desenvolvedor a manipular estados em componentes funcionais (FERNANDES, 2019), e também o uso de *Redux*, que o funcionamento irá ser explicado a seguir. A utilização dessas bibliotecas externas é possível graças a utilização do *yarn*, um gerenciador de pacotes para o *NodeJS*.

Figura 14 – Representação do funcionamento de componentes, propriedades e estados no *ReactJS*

Fonte – Página *Tech Diagonal*

3.5 ACCELERATED MOBILE PAGES

O AMP, sigla para *Accelerated Mobile Pages* (páginas aceleradas para dispositivos móveis), é uma tecnologia *open-source* criada pelo *Google* em 2015 com apoio de empresas como *Twitter*, *Linkedin* e *Pinterest*, que visa otimizar e padronizar as páginas web a serem carregadas rapidamente em dispositivos móveis sem a necessidade de possuir a mais veloz conexão de internet. O principal objetivo do uso desta tecnologia é diminuir as taxas de rejeição e abandono durante o fluxo de *websites*, melhorando o desempenho e as taxa de conversões (COMO... , s.d.) potencializando o uso do SEO nas páginas web. A melhora da performance é uma consequência dos seguintes sete pontos trabalhados pelo framework AMP (HOW... , s.d.):

1. Execução assíncrona do *Javascript* AMP: Um aspecto para a demora da renderização de uma página pode ser ocasionado pelo uso de *Javascript*, já que esse pode bloquear o DOM a realizar a estruturação da página. Sendo assim, o uso do *Javascript* dentro de web que usam AMP só devem ser assíncronos.
2. Recursos dimensionados estaticamente: Imagens, vídeos, *iframes* ou *ads* devem possuir seus tamanhos instanciados como atributos na suas tags HTML, assim o

AMP consegue posicionar o elemento de acordo com o seu HTML antes de ser feito o download da mídia instanciada.

3. Mecanismos de extensão não podem bloquear a renderização: *Scripts* personalizado devem informar ao sistema AMP que eventualmente terá uma *tag* personalizada, fazendo que o AMP não permita o bloqueio de renderização da página.
4. Retirar *Javascript* provenientes de bibliotecas externas da rota crítica: É necessário evitar o uso de bibliotecas externas que utilizam carregamento síncrono, separando o código em *sandboxed iframes*. Realizando essa separação a execução da página principal não é bloqueada.
5. CSS deve ser *inline* e *size-bound*: Em páginas AMP HTML é possível utilizar CSS *inline* com o tamanho máximo de 50 kilobytes.
6. Acionamento de Fontes customizadas devem ser eficiente: não há nenhum *request* HTTP antes do carregamento da fonte do código
7. Minimização do recálculo do estilo/layout: O cálculo para o layout da página pode ser custoso para o navegador, sendo assim no AMP o DOM lê todo o conteúdo antes de estruturar uma página para garantir no máximo um cálculo dos estilos por quadro.
8. Carregar apenas animações *GPU-accelerated*: A animação e a transição na transformação e opacidade do CSS que são os únicos permitidos no AMP.
9. Priorizar o carregamento de recursos: Todos os downloads de recursos são controlados pelo AMP, priorizando o carregamento de recursos essenciais e busca recursos com carregamento mais lento, sendo os últimos a serem de fato carregados.
10. Carregar páginas instantaneamente: É utilizada uma Application Programming Interface (API) de pré-conexão garantindo que as solicitações em HTTP sejam imediatas, logo uma página pode ser renderizada antes que o usuário final declare explicitamente o direcionamento até ela.

O funcionamento da arquitetura do *framework* é separado em três partes:

1. AMP HTML: Para garantir a performance e o desenvolvimento da página estática é necessário que algumas tags HTML sejam alteradas por web-components fornecidos pelo AMP, como a tag de imagem e video.
2. AMP JS: Como descrito anteriormente, um dos principais objetivos do AMP é garantir a renderização rápida das páginas, tornando tudo que é externo assíncrono evitando possíveis bloqueios na interpretação da página.

3. AMP CDN: Quando são utilizados o Google Cache para armazenar o AMP HTML das páginas.

Como a tecnologia é desenvolvida pelo *Google* esta é facilmente detectada no mecanismo de busca, para diferenciar uma página AMP entre as demais, basta procurar o símbolo do AMP (um círculo com um relâmpago) entre os resultados da busca como é possível ver na figura 15. As páginas que possuem a tecnologia AMP são preferenciadas nas buscas.

Figura 15 – Representação site AMP em busca no *Google*



Fonte – Página *Google*

3.6 GIT

O Git é um sistema de controle de versão *open-source*, que é uma forma de registrar as mudanças realizadas em um arquivo ou um conjunto de arquivos ao longo do seu desenvolvimento, permitindo a recuperação de versões já realizadas. Sendo possível não somente a visualização, mas também reverter arquivos, analisar quem submeteu as mudanças e facilitando a procura de erros na lógica.

A grande vantagem de utilizar um sistema de controle de versão de arquivos é a possibilidade da contribuição simultânea dos integrantes de um projeto. A plataforma mais utilizada para o uso online é o *GitHub*: um serviço web que serve de repositório online dos programas de um projeto (TRÊS. . . , 2019), possibilitando o compartilhamento dos códigos entre outros desenvolvedores, e também a requisição de *Pull Requests*, no qual é pedido uma “avaliação” entre outros desenvolvedores da área.

3.7 SOFTWARE AS A SERVICE

O Canal Digital é um Software as a Service (SaaS), o projeto não é vendido como um produto mas sim como um serviço como uma aplicação de um serviço *web*, sem a necessidade de um programa em específico (PATEL, 2020). Sua principal característica é o pagamento de uma assinatura para utilizar do serviço. A sua utilização

pode ser realizada desde *desktops* até *mobiles* e o seu funcionamento requer uma conexão estável com a internet.

3.8 WHITE LABEL

O Canal Digital também utiliza o modelo de negócio de White Label, no qual o serviço ou produto de software original tem a sua logo ou marca retirada, em que a empresa que requer o serviço aplica a sua marca em cima. Ou seja, o serviço prestado tem as mesmas funcionalidades em todas as aplicações, o que pode mudar de uma a outra são as cores principais utilizadas entre um e outro como também a divulgação da marca (JÚNIOR, 2019).

3.9 GOOGLE ANALYTICS

O *Google Analytics* é um sistema que monitora dados de tráfego em um *website*, é uma plataforma gratuita que permite fornecer informações do comportamento dos usuários fornecendo muito além do que a contagem de visitas diárias. Utilizando o monitoramento de tráfego é possível também obter informações referentes a taxa de conversão ou o comportamento dos usuários dentro do *website* captando indicadores como: quanto tempo permanecem em uma página, qual é o fluxo do site mais utilizado e também a origem do usuário, sendo possível analisar também quais campanhas de marketing estão dando certo, ou caso o redirecionamento de páginas de busca estão sendo relevantes. A interpretação do GA é essencial para um bom entendimento do comportamento do usuário no *website* em questão (VALLE, 2020).

3.10 GOOGLE DATA STUDIO

O *Data Studio* também é um serviço do *Google* voltado principalmente para construção de relatório ou *dashboards* customizáveis (PATEL, 2019). Possibilitando a aquisição de dados do *Google Analytics* (GA) e a criação de tabelas, dados, infográficos ou até mapas.

3.11 GOOGLE LIGHTHOUSE

Para medir o desempenho de um *website* é utilizada uma ferramenta *open-source* do *Google* chamado *Lighthouse*, que faz testes na página selecionada e gera relatórios referentes a performance, acessibilidade, boas práticas e SEO, gerando uma nota de 0 a 100 a cada um desses indicadores (PHILLIPS, 2019). Os valores entre 0 e 49 são considerados como baixos, de 50-89 medianos e 90-100 bons.

A bateria de testes realizados são denominadas *audit*, que envolvem submeter a página selecionada em um ambiente de um dispositivo móvel antigo e lento e com

uma conexão instável de *internet*. Essa simulação é realizada através da limitação do pacote de dados e diminuindo a capacidade da CPU. Caso a página seja rápida nessas condições de testes de *stress*, a performance será ainda melhor em telefones mais novos. A seguir serão detalhados os principais pontos analisados durante os testes de cada indicador.

3.11.1 Performance

Esse indicador é responsável pela medição da *performance* e indica métricas que podem aumentar a velocidade de um *website*, as principais métricas serão listadas a seguir:

- **First Contentful Paint:** A sua tradução seria o primeiro conteúdo renderizado, medindo o período de tempo necessário para um navegador renderizar o primeiro elemento DOM (FIRST... , 2019).
- **First Meaningful Paint:** Incrementando o item anterior, esse mede o tempo para a renderização de um conteúdo significativo ao usuário.
- **First CPU Idle:** É o tempo em que a página demora para de ser renderizada completamente.
- **Speed Index:** Baseado num parâmetro de 1,25 segundos, esse mostra a velocidade no qual um conteúdo é visível.
- **Time to Interactive:** Determina o período em que a página passa a ser interativa.
- **Estimated Input Latency:** O período necessário que a página leva para responder a um evento realizado pelo usuário.

3.11.2 Acessibilidade

O indicador de acessibilidade é responsável pela verificação das *tags* de cabeçalho, determinando se existe coerência ou não (LIGHTHOUSE... , s.d.). Além das *tags* também verifica possíveis contrastes incorretos e a existência da descrição de imagens ou links. Diferentemente do indicador de performance os itens checados não apresentam notas parciais e sim de ordem binária, caso passe no teste a nota é 100 e caso não é 0. O *audit* da acessibilidade é formado por uma lista com mais de 30 itens e a nota é determinada através de um peso entre esses.

3.11.3 Boas práticas

As boas práticas estão relacionadas ao uso de protocolos Hyper Text Transfer Protocol Secure (HTTPS) ao uso de *cross-origin links* e do *cache* da aplicação (PHIL-

LIPS, 2019). Também é relacionado a aspectos da UI da aplicação, caso há alguma imagem com proporção de tela diferente da original.

3.11.4 SEO

Esse indicador está mais relacionado a boas práticas do sentido de SEO, informando se a estruturação das informações estão corretas ou não (PHILLIPS, 2019).

4 CANAL DIGITAL

Esta seção tem como principal objetivo a explicação da plataforma Canal Digital, descrevendo o desenvolvimento da plataforma realizada pela Jungle Devs, a contextualização atual do projeto e por fim será mostrado a atual página do imóvel, seus parâmetros atuais e a justificativa da implementação do AMP.

4.1 O PROJETO

No final do ano de 2018 a Brognoli, uma agência imobiliária localizada em Florianópolis contratou os serviços de *back-end* e *front-end* da Jungle Devs para o desenvolvimento de uma plataforma *web*, objetivando a simplificação da burocracia na locação de um imóvel tornando a experiência de aluguel de uma maneira 100% digital. Porém, a plataforma não ficou restrita apenas à Brognoli, essa voltou-se ao desenvolvimento de um SaaS se transformando em um serviço de aluguel de imóveis em *White Label*, ou seja, oferecendo a possibilidade as imobiliárias que assinassem o serviço terem a liberdade de alterar o nome, cores, logo, imagens, lista de imóveis ou até selecionar serviços oferecidos da plataforma Canal Digital.

Atualmente o Canal Digital pertence à empresa *Woliver* que possui 18 imobiliárias como assinantes, listadas na tabela 1. A plataforma tem diversos fluxos, que não só contemplam o agendamento de visita, mas sim a experiência de alugar um imóvel de maneira digital com a possibilidade de realizar propostas no imóvel, a iniciação da negociações, a análise de crédito, caso necessário há a requisição de garantia, e também a assinatura do contrato digitalmente, completando o ciclo do aluguel. Além de também oferecer o cadastro do imóvel na plataforma pelo usuário.

Há cerca de 83 mil imóveis cadastrados entre todas as imobiliárias disponíveis, com mais de 28 mil usuários cadastrados, 1137 propostas realizadas e 5 contratos assinados na plataforma. Cabe ressaltar que nem todas imobiliárias finalizam o contrato do imóvel no Canal Digital, justificando a baixa quantidade desse. Esses números foram coletados pelo *Metabase* existente do projeto.

Além do Canal Digital, a *Woliver* também requisitou a realização de outro projeto, o *Backoffice*, responsável pela parte administrativa do Canal Digital voltada ao uso da imobiliária e de seus corretores. Para todos os fluxos realizados no Canal Digital há visualização por parte do corretor responsável pela visita, negociação, análise de crédito ou contrato de um determinado cliente. O *Backoffice* também é um *White Label*.

Os dois projetos são desenvolvidos ambos *front-end* e *back-end* pela Jungle Devs, trabalhado na tribo *Woliver* formado por um time de 6 pessoas, sendo 2 desenvolvedores *back-end*, 3 *front-ends* e 1 gerente de projeto. A autora deste documento atuou como líder da *stack* de *front-end* do projeto. O *design* das páginas foram oferecidos pelo cliente *Woliver*.

Tabela 1 – Lista de imobiliárias registradas no Woliver

Imobiliária	URL	Localização
Yes ABC	https://yesabc.woliver.net/	Santo André, SP
Santa Maria	https://santamaria.woliver.net	Chapecó, SC
Casarao	https://casarao.woliver.net/	Santa Maria, RS
Ayla	https://ayla.woliver.net	Florianópolis, SC
Auxiliadora	https://auxiliadora.woliver.net	Porto Alegre, RS
Simao	https://simao.woliver.net	Erechim, RS
Rosa	https://rosa.woliver.net	Cuiabá, MT
Cardinali	https://cardinali.woliver.net	São Carlos, SP
Forte Guimaraes	https://forteguimaraes.woliver.net	Ribeirão Preto, SP
Hunters	https://hunters.woliver.net	Campinas, SP
Arnel	https://arnel.woliver.net	Passo Fundo, RS
Cohab	https://cohab.woliver.net	Aracaju, SE
Guaira	https://guaira.woliver.net	São Bernardo do Campo, SP
Anage	https://anage.woliver.net	Joinville, SC
Prolar	https://prolar.woliver.net	Caxias do Sul, RS
Brognoli	https://brognoli.woliver.net	Florianópolis, SC
Brognoli Digial	http://brognolidigital.woliver.net/	Florianópolis, SC
Nova Freitas	https://novafreitas.woliver.net	São José dos Campos, SP
Terraz	https://terraz.woliver.net	Florianópolis, SC

Fonte – Jungle Devs

Como mencionado anteriormente, há a presença de diversos fluxos na plataforma em que todos tem o objetivo final da alocação de um imóvel. Porém, para a compreensão do tópico do documento realizado será explicado somente um fluxo que será o de agendamento de uma visita.

4.1.1 Agendamento de visita

O fluxo de agendamento de visita é o principal do Canal Digital, depois da realização desse o cliente final consegue realizar a proposta, uma análise de crédito e a assinatura do contrato final. O seu funcionamento é composto de 7 passos principais:

1. **Acesso à página inicial:** Ao acessar a página inicial de uma imobiliária o usuário terá a possibilidade de selecionar valores como: cidade de interesse, bairro, valor máximo de aluguel e número de quartos no imóvel.
2. **Página resultado de busca:** Após a filtragem selecionada, será possível visualizar os imóveis disponíveis com resultados exatos e similares a busca realizada. Para visualizar mais informações sobre um imóvel é necessário clicar no cartão do imóvel em questão.

3. **Página detalhamento do imóvel:** Todas as informações disponíveis do imóvel estarão dispostas nessa página, caso o usuário deseje realizar uma visita basta clicar no botão “Agendar visita”.
4. **Escolha de horário disponível:** A marcação de um horário de visita é definido nessa página em que mostra todos os momentos disponíveis.
5. **Confirmação do horário:** Informações básicas serão dispostas nessa página para realizar a confirmação do agendamento da visita.
6. **Login/Cadastro:** Composto de 3 passos de um aviso de login/cadastro fornecimento do número do telefone celular, confirmação do número disponibilizado através do recebimento de um SMS. Esse passo só é realizado caso o usuário não esteja logado na plataforma.
7. **Inserção ou confirmação de informações básicas:** Neste momento o usuário insere ou apenas confirma suas informações básicas, como o nome completo, *email* e CPF. Após a confirmação a ação do agendamento é realizada.

Após os passos descritos anteriormente, o usuário é direcionado a uma página em que esse preenche como foi a sua experiência de agendamento na plataforma e depois é mostrado todas as informações sobre a sua visita: horário e local. As páginas assim como o seu funcionamento é representado em um fluxograma, encontrado na figura 16.

O fluxo descrito anteriormente é a maneira pela qual o usuário consegue realizar o agendamento dependendo somente do Canal Digital. No entanto há outras formas em que o usuário pode acessar diretamente as rotas de agendamento, pulando a etapa de buscar um imóvel e visitar a página referente a descrição do imóvel. Existe a possibilidade de que a imobiliária em questão utilize outro site, com outra interface da página de descrição de imóvel, e direcione o usuário apenas para a etapa de agendamento no Canal Digital.

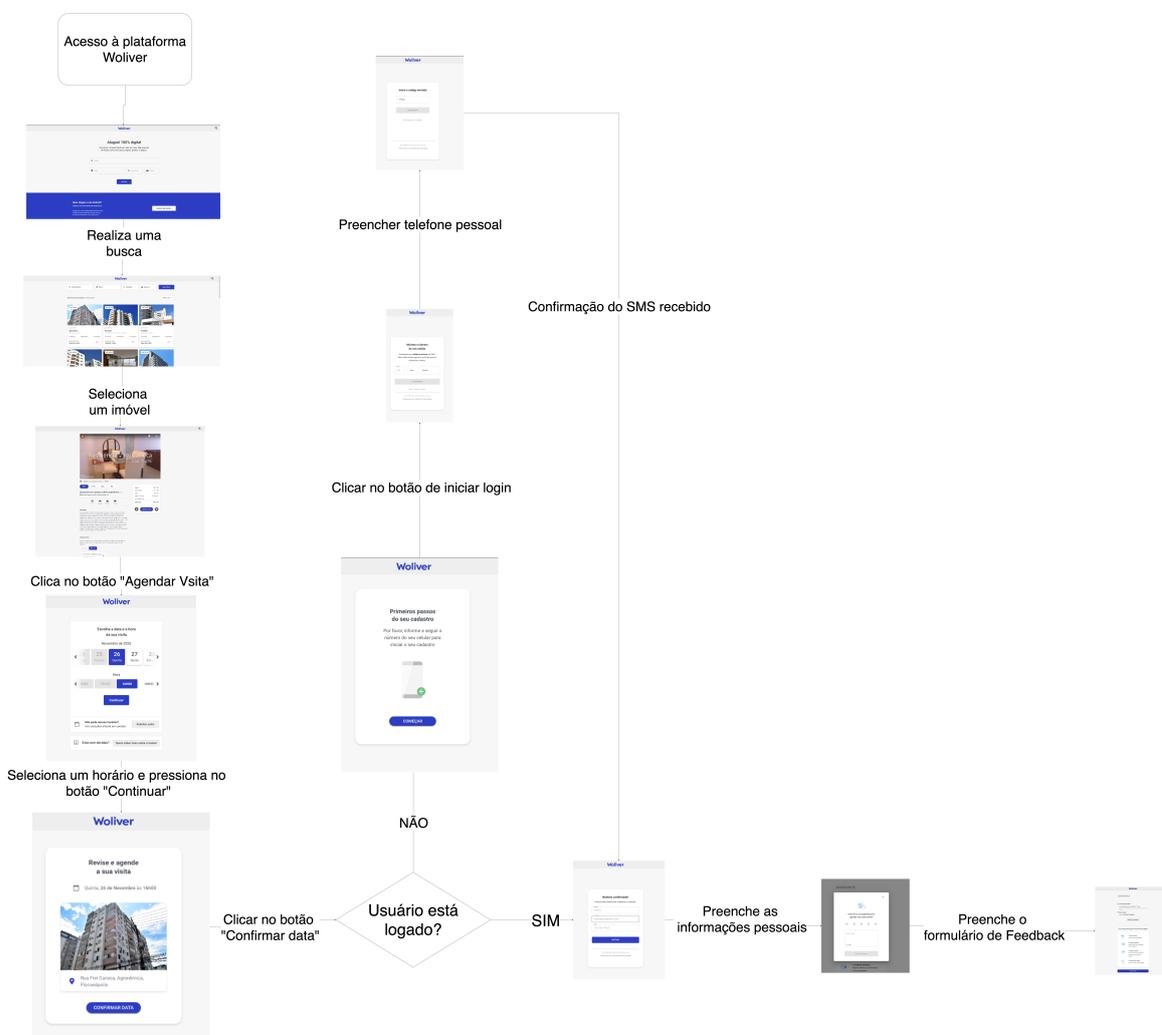
4.2 COMPORTAMENTO DOS USUÁRIOS NA PLATAFORMA

Para ser possível analisar o atual comportamento dos usuários no Canal Digital foram manipulados os dados adquiridos do *Google Analytics* através do uso do *Data Studio* pela autora do documento. As análises a seguir foram realizadas nas imobiliárias em que o acesso aos dados foi fornecido, sendo elas a Santa Maria, Ayla, Brognoli, Brognoli Digital, Terraz, Arnel, Nova Freitas, Hunters, Prolar e Guaira.

Primeiramente foi realizada uma análise das visualizações únicas por página nos meses de Agosto a Outubro de 2020, devido a limitações do *Data Studio* foi possível agregar dados de 5 imobiliárias por gráfico. Sendo assim, as imobiliárias foram

Figura 16 – Fluxograma do agendamento de visita em *desktop*

FLUXO AGENDAMENTO DE VISITA



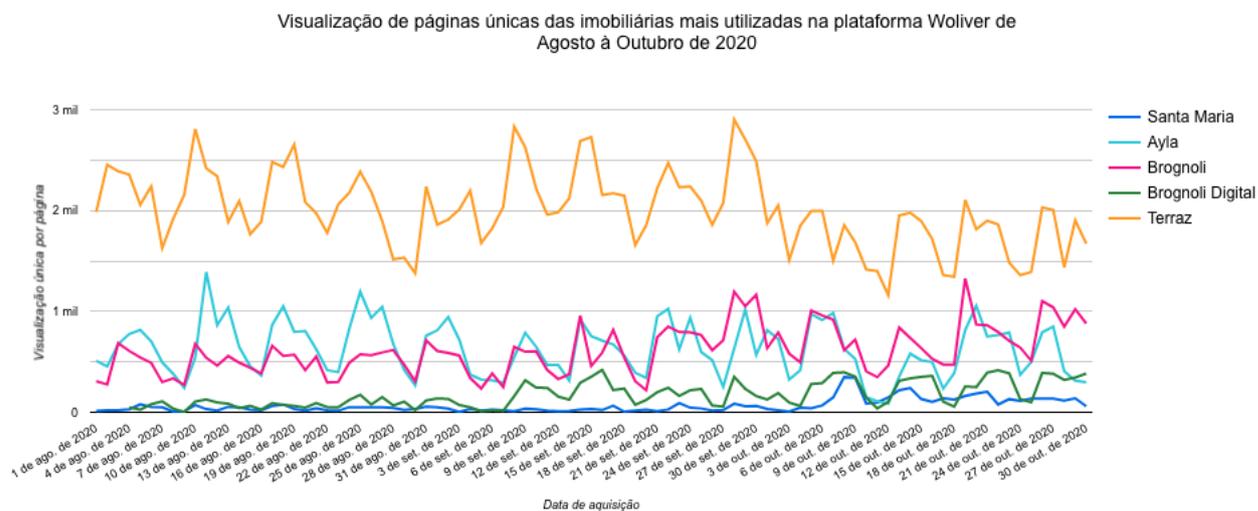
Fonte – Autora

separadas naquelas que tem mais e menos visualizações por páginas. O gráfico 17 apresentam as imobiliárias mais visitadas sendo elas ordenadas em: Terraz, Brognoli, Ayla, Brognoli Digital e Santa Maria. O gráfico da figura 18 apresenta os dados referentes às imobiliárias menos visualizadas no mês de outubro, sendo elas a Nova Freitas, Prolar, Arnel, Guaira e Hunters.

Sendo assim, devido a limitação do *Data Studio* e pela melhor distribuição e consistência de dados, as futuras análises deste relatório serão baseada nas 5 imobiliárias com mais visualizações entre os meses de Agosto e Outubro de 2020.

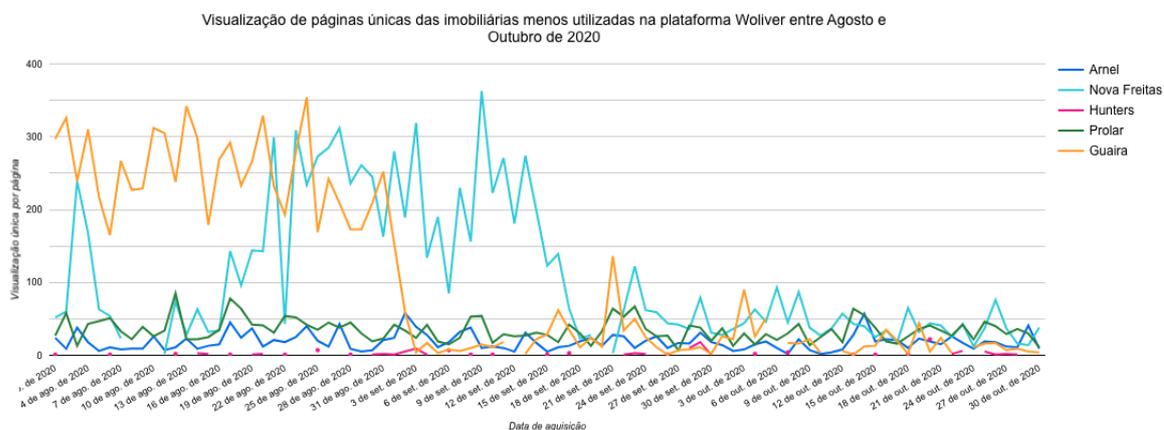
A próxima análise será referente ao fluxo de usuários na plataforma, com a separação entre usuários e quais desses são novos usuários na plataforma, e também

Figura 17 – Gráfico de linhas com dados das imobiliárias com mais visualizações únicas



Fonte – Autora

Figura 18 – Gráfico de linhas com dados das imobiliárias com menos visualizações únicas



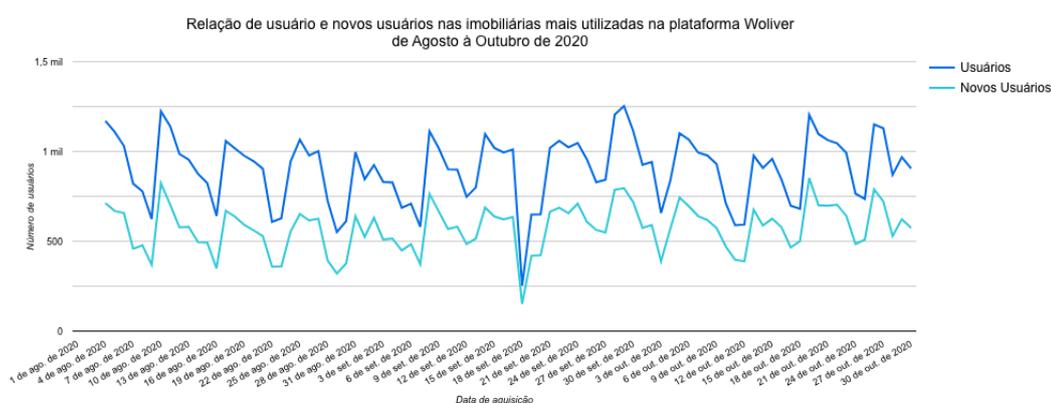
Fonte – Autora

o tipo de dispositivo utilizado para o acesso a plataforma.

No gráfico da figura 19 é representado o número de usuário ativos da plataforma durante os meses de Agosto a Outubro de 2020, na qual percebe-se que a maioria dos usuários ativos são os novos e a rotação entre os usuários é constante.

Validando a hipótese levantada pela TIC Domicílios 2019, representada na figura 1, foi analisado o tipo de dispositivo em que o Canal Digital é acessado por todos os usuários e também pelos novos. Ambos os gráficos, presentes nas figuras 20 e

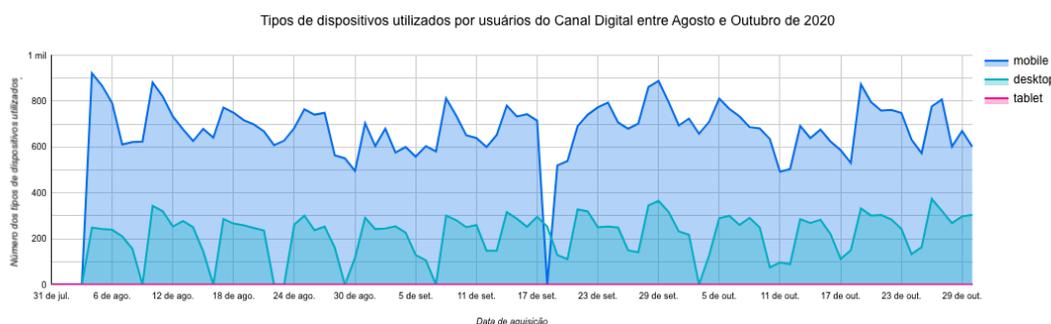
Figura 19 – Gráfico de linhas com dados das imobiliárias com menos visualizações únicas



Fonte – Autora

21 evidenciam que a grande maioria dos usuários acessam a plataforma através de dispositivos móveis. Sendo assim o desenvolvimento da plataforma deve ser projetada para ser mais performática principalmente para celulares, mantendo o usuário por mais tempo do *website* ou na página em questão. Na próxima seção será verificado a performance da principal página de plataforma, a de descrição do imóvel, e se caso ela realmente atenda os requisitos para atrair os usuários no Canal Digital.

Figura 20 – Gráfico de áreas com tipos de dispositivo utilizado por usuário

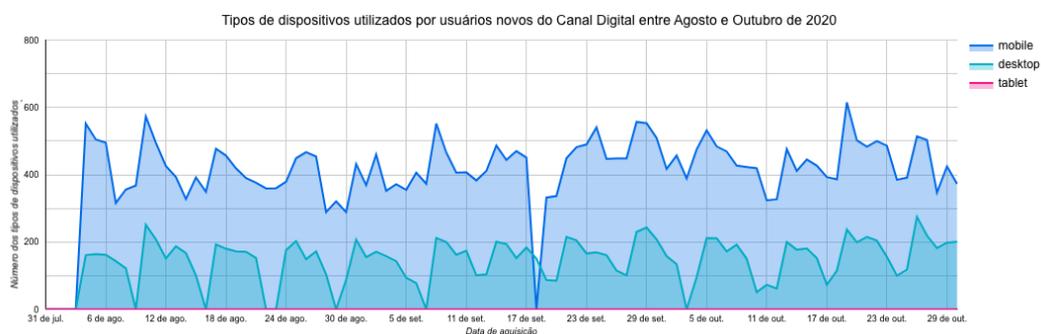


Fonte – Autora

4.3 A PÁGINA DE DESCRIÇÃO DO IMÓVEL

Qualquer tipo de ação realizada no Canal Digital depende da associação de um imóvel, logo considera-se que a principal página de todas na plataforma é a de descrição desse. A rota dessa página é composta pela URL inicial de cada imobiliária, que pode ser encontrados na tabela 1 na coluna de URL, adicionando '/imovel' junto a

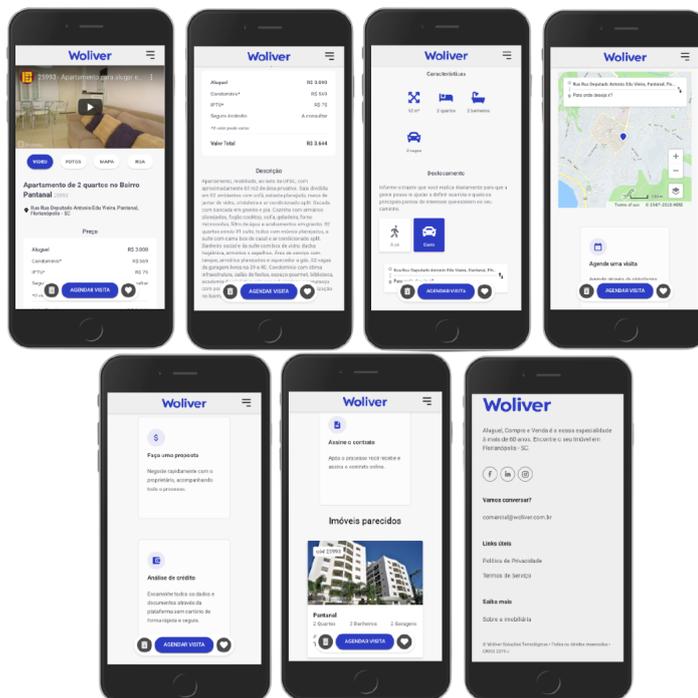
Figura 21 – Gráfico de áreas com tipos de dispositivo utilizado por novos usuários



Fonte – Autora

um parâmetro único de cada imóvel, chamado na *codebase* de *external id*. A página é mostrada na figura 23 na versão *desktop* e na 22 em *mobile*. Atualmente a página é dividida nas seguintes seções:

- **Seleção das mídias:** A primeira seção da página é composta da apresentação das mídias presentes do imóvel, sendo elas: fotos, vídeos, posicionamento de um mapa estático e um mapa com visualização da rua. A seleção do tipo de mídia a ser mostrada é possível através de um botão de seleção.
- **Informações gerais do imóvel:** Seguindo para a seção com informações textuais do imóvel, com um título indicando o tipo de imóvel, a quantidade de quartos e o bairro de localização com o seu respectivo endereço. Há a disposição sobre as informações de área, número de quartos, banheiros, vagas disponíveis e uma breve descrição sobre o imóvel.
- **Valor de aluguel e ação de agendar visita:** Cartão informativo sobre o valor do aluguel e a soma desse com possíveis taxas. Nessa seção existe o botão essencial para o redirecionamento da página de escolha de horário para a visita. Aqui também há a opção do usuário favoritar ou excluir o imóveis das opções atuais.
- **Deslocamento:** Apresentação de um mapa dinâmico com a presença de um campo de texto no qual é possível adicionar possíveis rotas até o endereço do imóvel.
- **Cartões informativos:** Seção com informações dos próximos passos do aluguel do imóvel, esse texto varia de acordo com a imobiliária.
- **Imóveis parecidos:** De acordo com as características desse imóvel são disponibilizados alguns imóveis parecidos com o selecionado.

Figura 22 – Página da descrição do imóvel em *mobile*

Fonte – Woliver

Ambas as páginas das figuras 21 e 23 foram desenvolvidas em *ReactJS* em 2019 por outra desenvolvedora, foram utilizadas bibliotecas externas como a *HERE* para a demonstração de mapas interativos ao usuário. A UI da página será reestruturada e a autora do documento foi encarregada de algumas seções, que serão explicadas no capítulo 5.

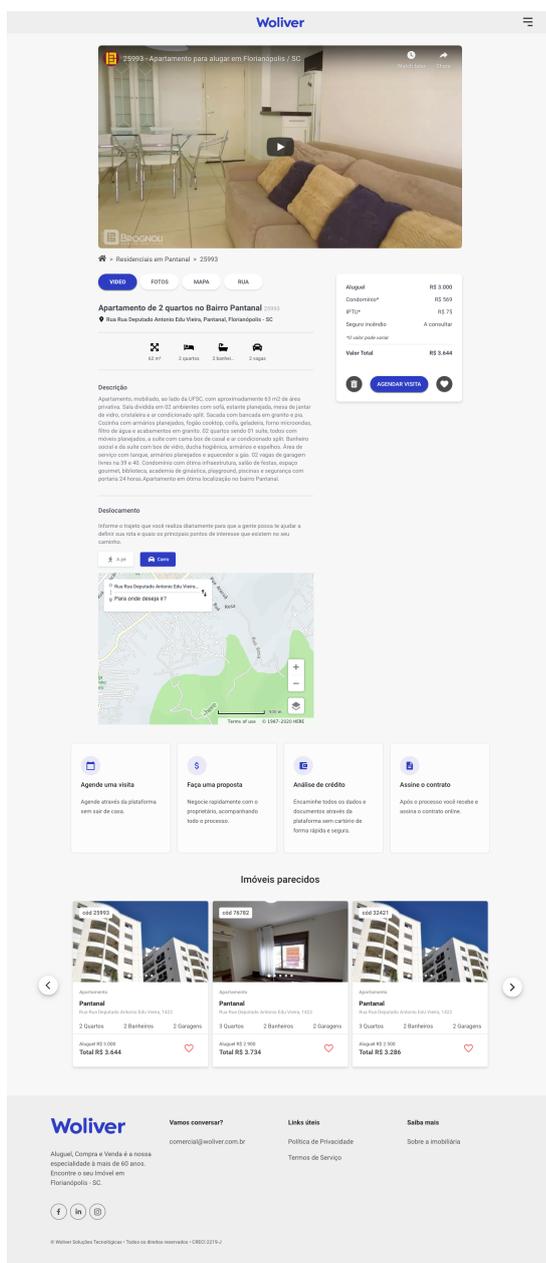
4.3.1 Performance da página

Como já destacado na última seção, a grande maioria dos usuários que acessam a plataforma o fazem através de dispositivos móveis, sendo a página do imóvel a principal página da plataforma, essa deve ser mais performática possível para o usuário final.

Ao realizar o teste na página de um imóvel na imobiliária Demo, um ambiente no qual são demonstrados aos possíveis clientes uma demonstração do produto, foram obtidos os resultados demonstrados na figura 24. A nota 3 de 100 é bem crítica à performance da página, não atendendo os requisitos da grande maioria dos usuários frequentes e novos.

Com as métricas com resultados bem abaixo do esperado foram estudadas alternativas para a implementação de técnicas ou tecnologias que possam aumentar a nota e respectivamente melhorar a experiência do usuário final na plataforma. Foi

Figura 23 – Página da descrição do imóvel em *desktop*

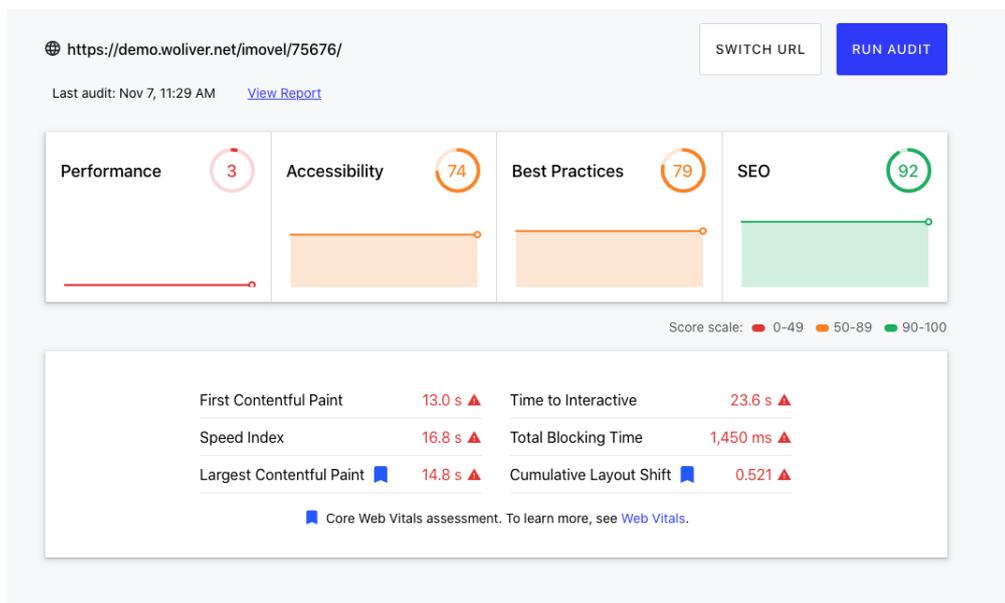


Fonte – Woliver

recomendado o uso da tecnologia criada pelo *Google*, o AMP, que promete uma melhor performance na página. Sendo assim foi realizado um estudo sobre essa nova tecnologia e se o uso dessa trará impacto real ao usuário final na plataforma, comprovando que o parâmetro medido no *Lighthouse* é realmente eficaz.

Para entender o impacto dessa tecnologia na plataforma, foi optado fazer a realização de um teste A/B entre a página atual em *ReactJS* e em AMP e observando o impacto dessas no número de agendamento realizados das imobiliárias através da

Figura 24 – Notas da página de descrição de um imóvel no *lighthouse*



Fonte – Site Web Dev Measure

análise dessa taxa de conversão. O documento focará do desenvolvimento por parte da autora na implementação do AMP na plataforma e o estudo do impacto do uso dessa.

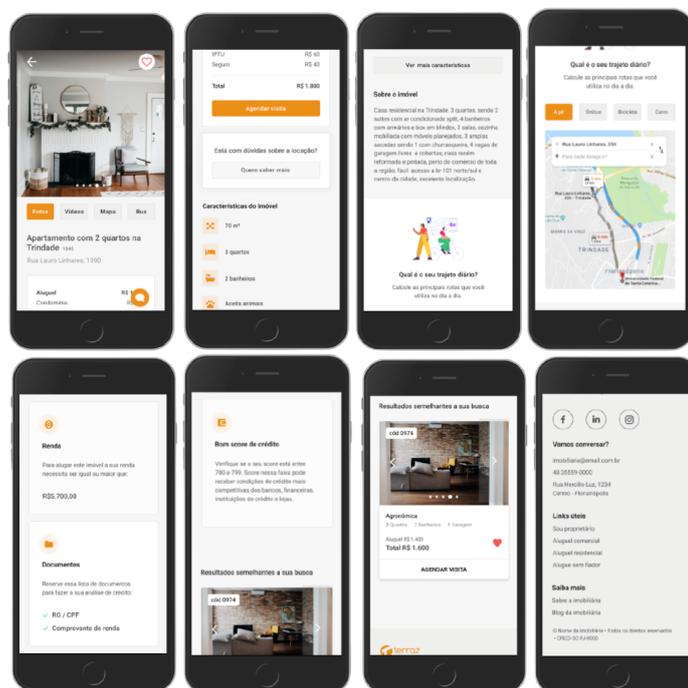
5 ATIVIDADES DESENVOLVIDAS

Neste capítulo será tratada a parte de implementação da tecnologia AMP no Canal Digital, realizado na página de descrição de imóvel. Será destacando o desenvolvimento com a tecnologia AMP replicado utilizando a nova UI disponibilizada pelo cliente Woliver, e como foi adicionado no código base do Canal Digital. Também será explicando a implementação do teste A/B. A autora do documento ficou responsável pelo desenvolvimento total da página em AMP, da implementação do teste A/B e pelas análises subsequentes.

5.1 NOVA INTERFACE DA PÁGINA DE DESCRIÇÃO DO IMÓVEL

A nova UI da página de usuário conta com a customização dos elementos já existentes, como botões com bordas diferentes e a atualização da disposição de algumas seções. Porém, as divisões da páginas descritas na seção 4.3 mantiveram-se as mesmas. A UI do *mobile* é encontrada na figura 25 e a 26 é a UI em *desktop*.

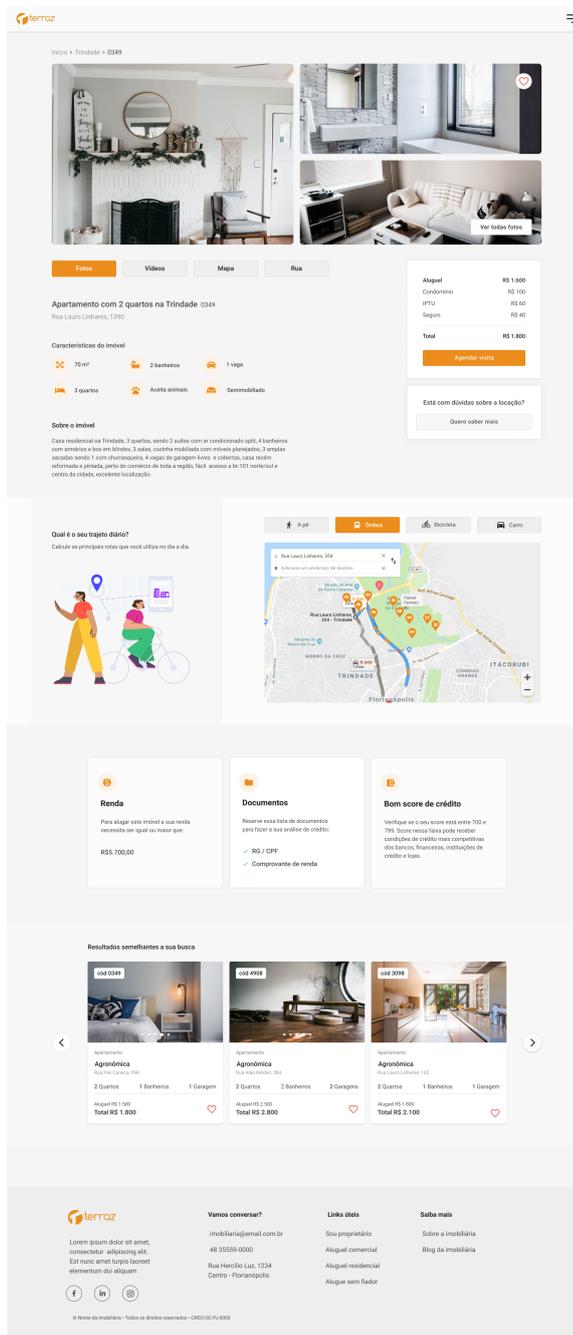
Figura 25 – Nova UI no *mobile*



Fonte – Figma Woliver

No *design* foi utilizado como base a imobiliária Terraz que apresenta como cor primária o laranja, para a demonstração do que foi desenvolvido e para melhor diferenciação será mostrado figuras da página de uma imobiliária teste, chamada *Demo Woliver*, que possui a cor primária azul escuro.

Figura 26 – Nova UI no desktop



Fonte – Figma Woliver

5.2 DESENVOLVIMENTO DA PÁGINA EM AMP

As diversas limitações do AMP, como a não utilização do uso de bibliotecas externas com *Javascript*, impedem qualquer tipo de reaproveitamento do código existente até o momento. Logo, toda a nova *view* foi realizada seguindo as normas estabelecidas por esse *framework*.

Para a criação de uma página em AMP é necessário estruturar o HTML da página com *tags* obrigatórias, fornecidas através de *boilerplate* (figura 27) presente na documentação do AMP, para que o navegador e o *Google* diferenciem uma página normal e uma com o *framework*. Cabe ressaltar que os próximos passos serão desenvolvidos dentro da *tag body*, e todo o CSS utilizado foi do tipo *inline* com um atributo *amp-custom* especificado pela documentação do AMP (COMO... , s.d.).

A única biblioteca externa em *Javascript* disponível de uso é a do próprio AMP e como mencionado na seção 3.5 são fornecidos diferentes componentes em HTML que facilitam o desenvolvimento da página, caso não ocorresse a existência desses a página seria realizada somente com puro HTML e CSS.

Antes da especificação de cada divisão da página de descrição de imóveis, é necessário o entendimento de componentes fundamentais para o funcionamento geral da página fornecidos pelos componentes HTML AMP, sendo o primeiro o *amp-list* no qual a sua principal funcionalidade é fazer a requisição, utilizando protocolos Hyper Text Transfer Protocol (HTTP) do tipo GET à API da imobiliária de acordo com o Uniform Resource Locator (URL) fornecido, sendo possível adquirir a informações no formato JavaScript Object Notation (JSON) do imóvel da página acessada. Na página em questão foram utilizados 2 diferentes URLs, uma que retorna todas as informações de um imóvel e outra que retorna uma lista de imóveis semelhantes. Junto ao *amp-list* é necessária a utilização do *amp-mustache* no qual é possível fazer a atribuição de onde cada valor recebido pelo JSON será devidamente populado. Ao lidar com estados ou a interatividade é necessário utilizar o *amp-bind*, que possibilita a criação de variáveis dinâmicas, ou seja, que mudam de valor conforme a interatividade do usuário. Posto que não é possível adicionar *Javascript* proveniente de bibliotecas externas, há a opção de adicionar funções limitadas como a de formatação de uma *string*, com o componente *amp-script*.

Apesar do AMP ser focado para o desenvolvimento de páginas em dispositivos móveis foi optado também a realização da *view* em *desktop*, já que a estruturação é relativamente parecida, e também pela possibilidade de realizar páginas responsivas através do uso o *boilerplate* do *framework* em questão. O desenvolvimento da *view* será descrito conforme a divisão realizada na seção 4.3, fazendo a diferenciação em *mobile* e *desktop* dentro de cada divisão da página citada.

5.2.1 Seleção de Mídias

A primeira seção tem 4 diferentes estados sendo eles o de “Fotos”, “Vídeos”, “Mapa” e “Rua”. Os estados de ativo e desativo é possibilitado através da utilização do *amp-selector*, que representa um controle de uma lista de opções. Será necessário analisar cada estado já que em cada um deles foi utilizado um componente AMP HTML. Na UI dos diferentes dispositivos existem botões de seleção, logo será explicado a

Figura 27 – Código obrigatório para a renderização do AMP

```
<!doctype html>
<html amp lang="en">

<head>
  <meta charset="utf-8">
  <script async src="https://cdn.ampproject.org/v0.js"></script>
  <title>Hello, AMPs</title>
  <link rel="canonical" href=
  "https://amp.dev/documentation/guides-and-tutorials/start/create/basic_markup/">
  <meta name="viewport" content="width=device-width,minimum-scale=1,initial-scale=1">
  <script type="application/ld+json">
    {
      "@context": "http://schema.org",
      "@type": "NewsArticle",
      "headline": "Open-source framework for publishing content",
      "datePublished": "2015-10-07T12:02:41Z",
      "image": [
        "logo.jpg"
      ]
    }
  </script>
  <style amp-boilerplate>
  body {
    -webkit-animation: -amp-start 8s steps(1, end) 0s 1 normal both;
    -moz-animation: -amp-start 8s steps(1, end) 0s 1 normal both;
    -ms-animation: -amp-start 8s steps(1, end) 0s 1 normal both;
    animation: -amp-start 8s steps(1, end) 0s 1 normal both;
  }

  @-webkit-keyframes -amp-start {
    from {
      visibility: hidden;
    }

    to {
      visibility: visible;
    }
  }

  @-moz-keyframes -amp-start {
    from {
      visibility: hidden;
    }

    to {
      visibility: visible;
    }
  }

  @-ms-keyframes -amp-start {
    from {
      visibility: hidden;
    }

    to {
      visibility: visible;
    }
  }

  @-o-keyframes -amp-start {
    from {
      visibility: hidden;
    }

    to {
      visibility: visible;
    }
  }

  @keyframes -amp-start {
    from {
      visibility: hidden;
    }

    to {
      visibility: visible;
    }
  }
</style><noscript>
<style amp-boilerplate>
  body {
    -webkit-animation: none;
    -moz-animation: none;
    -ms-animation: none;
    animation: none;
  }
</style>
</noscript>
</head>

<body>
  <h1>Welcome to the mobile web</h1>
</body>
</html>
```

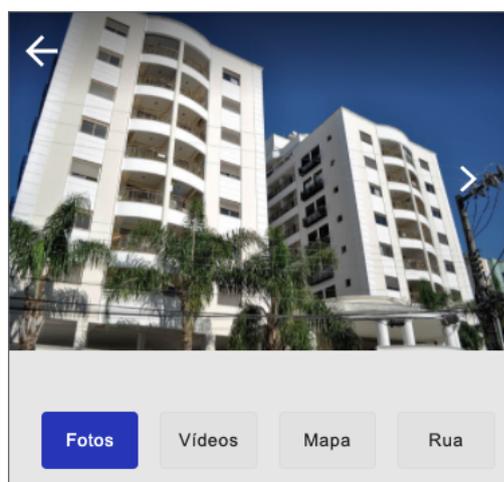
Fonte – Página amp dev

lógica implementada em cada item com a especificação realizada para os diferentes dispositivos.

5.2.1.1 Fotos

O estado de “Fotos” é o inicial do componente e para o dispositivo móvel a sua representação será através de *carousel*, um slideshow em que com o deslize horizontal mostra os elementos presentes no componente (MARK OTTO, s.d.), que neste caso são as fotos disponíveis do imóvel. O uso dessa funcionalidade foi possível através da utilização do *amp-carousel*, já que o conteúdo desse são fotos, como pode ser analisado na figura 28, será necessária a utilização do *amp-img*. Apesar de já existir uma *tag* em HTML para a imagem, optou-se por utilizar a versão em AMP uma vez que essa determina a priorização ou o atraso do carregamento dos componentes de acordo com a posição da janela de visualização, largura de banda da internet, recursos do sistema ou outros.

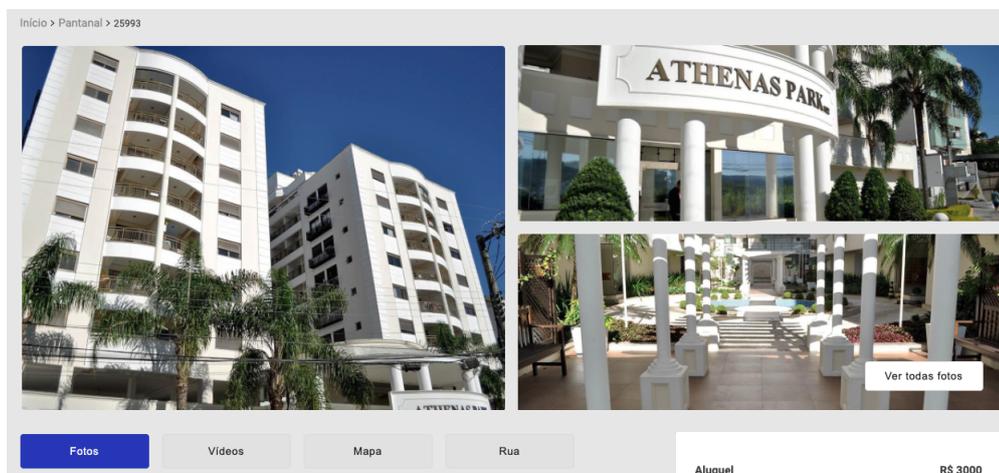
Figura 28 – Estado de fotos selecionados na seção de seleção de mídias em dispositivos móveis



Fonte – Woliver

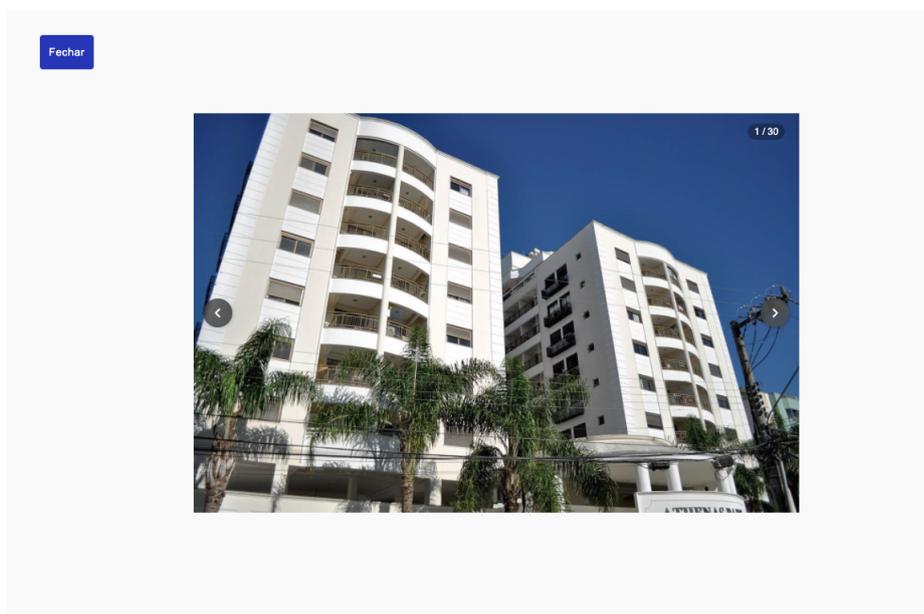
Já para a versão de “Fotos” em computadores é separada em 3 imagens como retratado na figura 29. Para o usuário visualizar as outras fotos do imóvel há a possibilidade de clicar no botão “Ver todas fotos” acionando o uso do componente *amp-lightbox*, que funciona como uma janela secundária, chamada também de modal (MÁRCIOD’ÁVILA, 2006), em o conteúdo é um *carousel* que contém todas as imagens do imóvel conforme a figura 30. Nesta opção não foi utilizado o *amp-carousel* mas sim o *amp-base-carousel*, que é uma versão mais customizável do *carousel* em que é possível adicionar o componente de *amp-inline-gallery-pagination*, o qual indica a paginação atual da mídia selecionada.

Figura 29 – Estado de fotos selecionados na seção de seleção de mídias em desktop



Fonte – Woliver

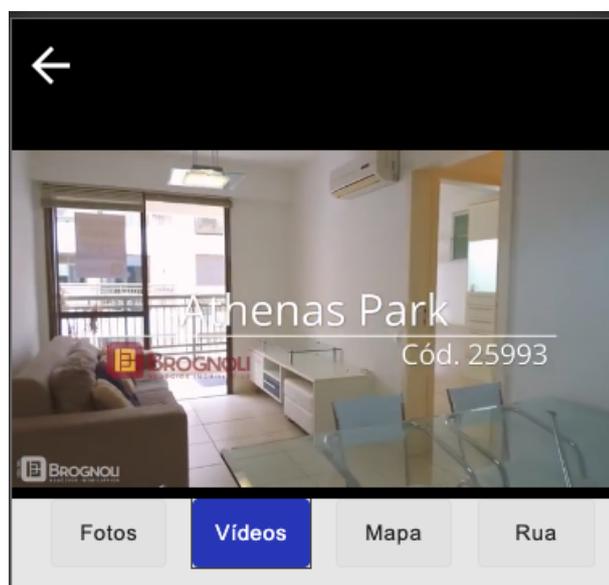
Figura 30 – Modal com todas as fotos do imóvel



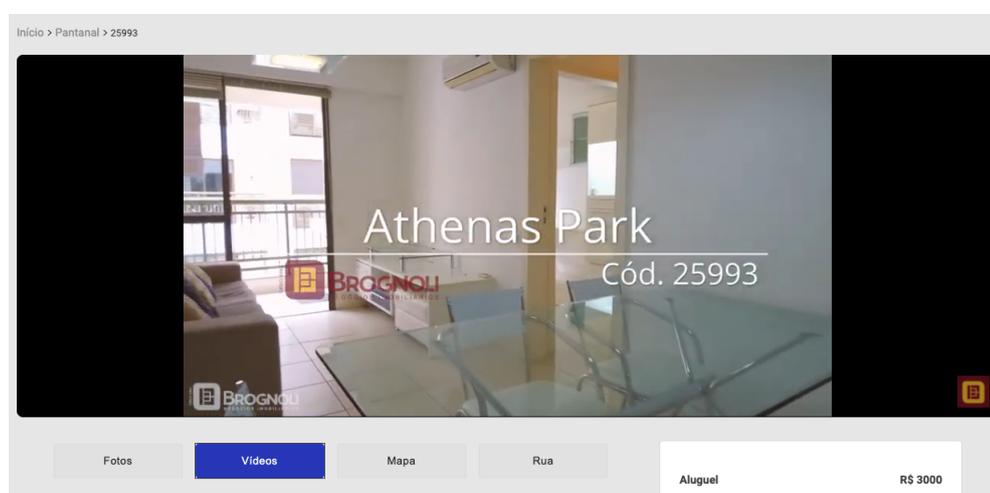
Fonte – Woliver

5.2.1.2 Vídeo

Os vídeos são links do youtube, para isso foi utilizado o *amp-youtube*, recebendo o *dataId* do vídeo em questão, que é a id fornecida pelo URL do vídeo. A disposição do componente ficou igual em ambos os dispositivos como é representado nas figuras 31 e 32.

Figura 31 – Estado de vídeo selecionados na seção de seleção de mídias em *mobile*

Fonte – Woliver

Figura 32 – Estado de vídeo selecionados na seção de seleção de mídias em *desktop*

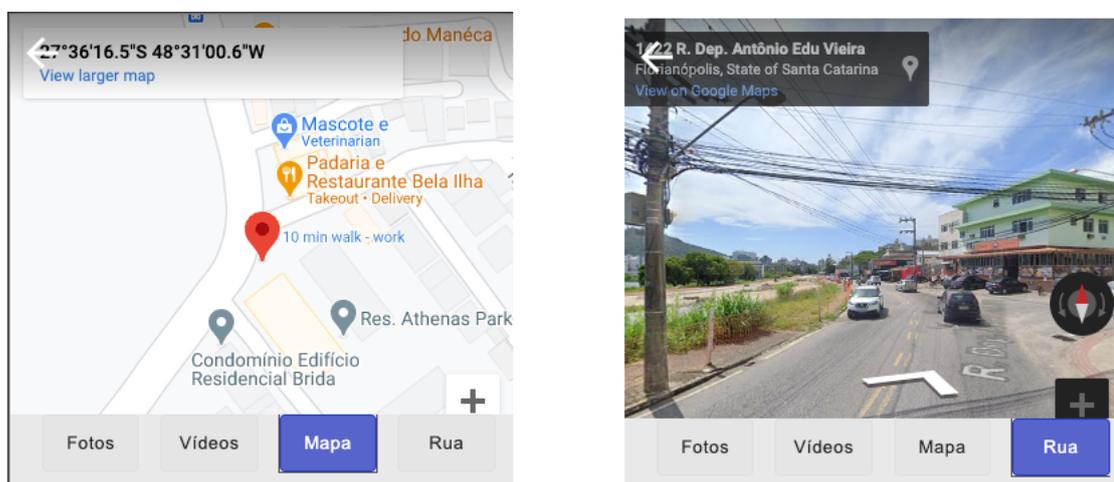
Fonte – Woliver

5.2.1.3 Mapa e Rua

Apesar de serem representações diferentes, ambas utilizam o componente *amp-iframe*, para mostrar o mapa em diferentes versões é necessário a utilização do *Maps Embed API* do *Google Maps*. Essa API fornece um mapa interativo com uma requisição em HTTP (GET... , 2020) de até 5 tipos diferentes. Para seguir os modelos requisitados no design foi utilizado na requisição o mapa do tipo *place* para a opção "Mapa" e o do

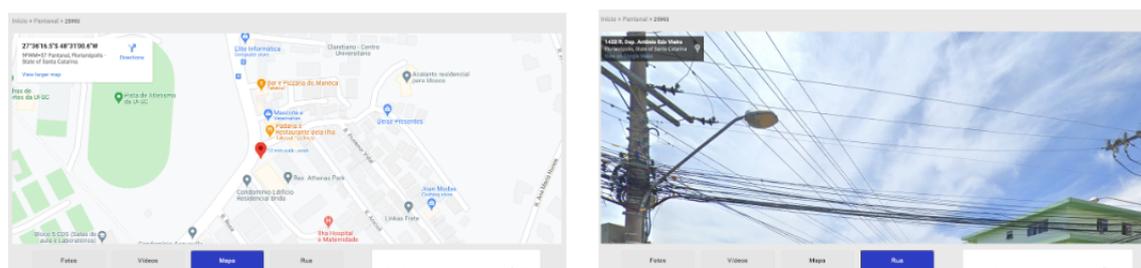
tipo *streetview* para a opção “Rua”. A visualização desse estado pode ser encontrada nas figuras 33 e 34.

Figura 33 – Estado de Mapa e Rua selecionados na seção de seleção de mídias em *mobile*



Fonte – Woliver

Figura 34 – Estado de Mapa e Rua selecionados na seção de seleção de mídias em *desktop*



Fonte – Woliver

5.2.2 Informações gerais do imóvel

As informações gerais aqui são consideradas como: o título do imóvel, endereço, as características e a sua descrição. A disposição do título varia conforme o dispositivo utilizado, para o *mobile* ele vem logo em sequência da divisão de seleção de mídia e é separado das outras informações do imóvel pela seção de valor de aluguel e ação de agendar visita como mostrado na figura 35, no *desktop* todas as informações ficam unidas no lado esquerdo da página como na figura 36.

Figura 35 – Informações gerais do imóvel em *mobile*

Fonte – Woliver

Para o título do imóvel foi necessário criar uma função para estabelecer qual é o tipo de imóvel, quantos imóveis e o bairro. Para as características as informações são mais demonstrativas e os itens são mostrados de acordo com a disponibilidade do dado ou não no JSON do imóvel, como por exemplo, caso o valor da permissão de animais seja falso o item desaparece desta lista. Foi necessário incluir a função para adicionar a palavra referente ao dado como plural ou não, como no caso de quarto e quartos.

5.2.3 Valor de aluguel e ação de agendar visita

Os cartões são encontrados nas figuras 35 e 36, foi necessário criar uma função para formatar a *string* em formato monetário, para os valores relacionados ao aluguel e também realizar a somatória do aluguel total. O botão de "Agendar visita" é alterado

Figura 36 – Informações gerais do imóvel em *desktop*

Item	Valor
Aluguel	R\$ 3000
Condomínio	R\$ 569
IPTU	R\$ 75
Total	R\$ 3.644

Fonte – Woliver

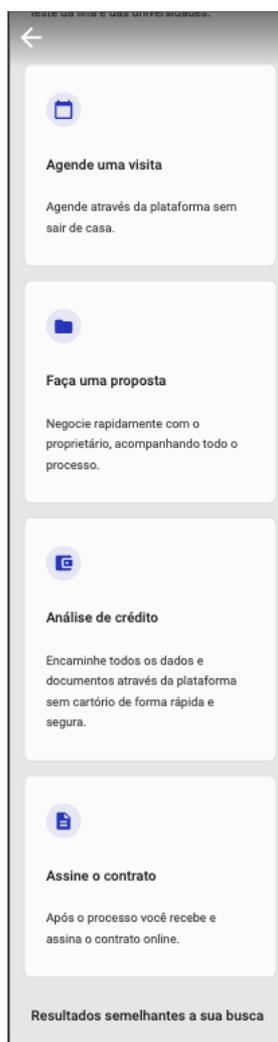
conforme o status do usuário com o imóvel: se esse esteja logado na plataforma é realizado uma requisição para o API checando caso esse já tenha realizado uma visita no mesmo, caso sim o botão se altera para “Fazer proposta” como também o direcionamento desse. O botão “Quero saber mais” direciona o usuário para o fluxo de mais informações do imóvel.

5.2.4 Deslocamento

Devido às limitações definidas pelas regras gerais do AMP e a impossibilidade de realizar ou utilizar um componente *amp-iframe* que mudasse de rota dinamicamente requerindo diversos requisitos à API do Google Maps, foi optado por descartar essa seção da página em sua versão AMP.

5.2.5 Cartões informativos

O uso dos componentes do AMP não foi necessário nessa seção uma vez que o servidor que fornece as informações necessárias sobre as imobiliárias já que as informações dos títulos e as descrições são alteradas conforme as informações iniciais de cada imobiliária, fornecidas por uma requisição inicial por parte do servidor ao banco de dados do projeto. Houve alterações em relação ao design que foi a adição de mais um cartão e a alteração dos ícones para que haja informações sobre a visita, proposta, análise de créditos e contrato. O que muda na visualização dos dispositivos dessa divisão é a disposição vertical na figura 37 e horizontal na 38.

Figura 37 – Cartões informativos em *mobile*

Fonte – Woliver

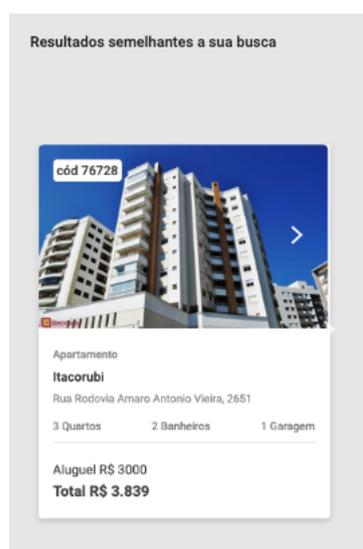
5.2.6 Imóveis parecidos

Nessa divisão também foi utilizado um *carousel* mas ao invés de mostrar somente imagens esse irá mostrar um elemento com informações básicas do imóvel demonstrados. Esse elemento em questão também terá um *carousel* com as algumas imagens do imóvel e para a demonstração das informações básicas do imóvel. Foram realizadas funções de formatação de *string*, semelhantes as utilizadas anteriormente, como representado na figura 39.

Para a visualização em *desktop* foi necessário alterar o *carousel* das fotos por uma seção de três imagens e a parte dos resultados similares. Ao invés de utilizar o *amp-carousel* foi necessário alterar para o *amp-base-carousel* para ser possível a adição de 3 elementos simultaneamente, como demonstrado na figura 40.

Figura 38 – Cartões informativos em *desktop*

Fonte – Woliver

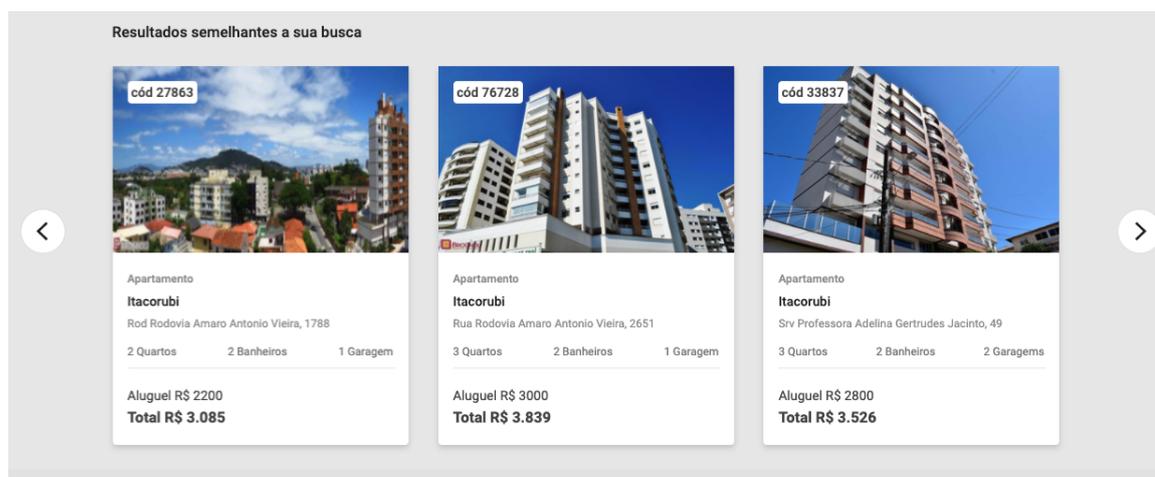
Figura 39 – Imóveis semelhantes em *mobile*

Fonte – Woliver

5.2.7 Cabeçalho e Rodapé

Para a realização do cabeçalho e do rodapé na versão *mobile* não foi necessário a utilização de componentes provenientes do AMP. O cabeçalho, que pode ser visto na figura 33 é fixado conforme a página é explorada, e o rodapé, mostrado na figura 41, encontra-se no final da página. Para obter a logo de cada imobiliária foi necessário realizar uma requisição para formatá-lo com a cor primária da imobiliária em questão.

Já na versão em *desktop* foi necessário replicar o design do cabeçalho dispostos nas outras páginas da aplicação, sendo semelhante ao da figura 23 ao invés do da figura 26. Houve também a necessidade da adição de um botão que abrisse uma modal com botões que possibilitasse o usuário ser redirecionado à página que lista os imóveis

Figura 40 – Imóveis semelhantes em *desktop*

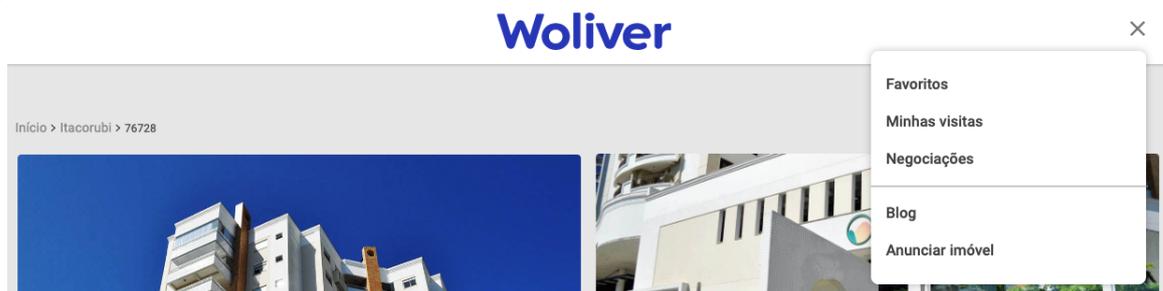
Fonte – Woliver

Figura 41 – Rodapé em *mobile*

Fonte – Woliver

favoritas, outra referente às visitas marcadas ou para as negociações pendentes, além de ter o redirecionamento ao blog da imobiliária ou a página que dá acesso ao fluxo de anunciar imóveis, como demonstrado na figura 43. A realização dessa modal foi feita com o componente *amp-lightbox*.

O rodapé da página em *desktop* foi similar ao realizado em *mobile*, houve somente a adaptação para a disposição dos itens de maneira horizontal como retratado na figura 43.

Figura 42 – Cabeçalho em *desktop* com o botão de opções selecionados

Fonte – Woliver

Figura 43 – Rodapé em *desktop*

Fonte – Woliver

5.2.8 Google Analytics

O armazenamento de dados realizados na plataforma é feita pelo GA. Para ser possível ter a aquisição dos dados na página em AMP é necessária a adição do componente *amp-analytics* na *tag head* do HTML, para diferenciar cada página no GA foi determinado que a rota a ser armazenada será */amp/imovel/externalId*, no qual o *external id* representa um id único do imóvel.

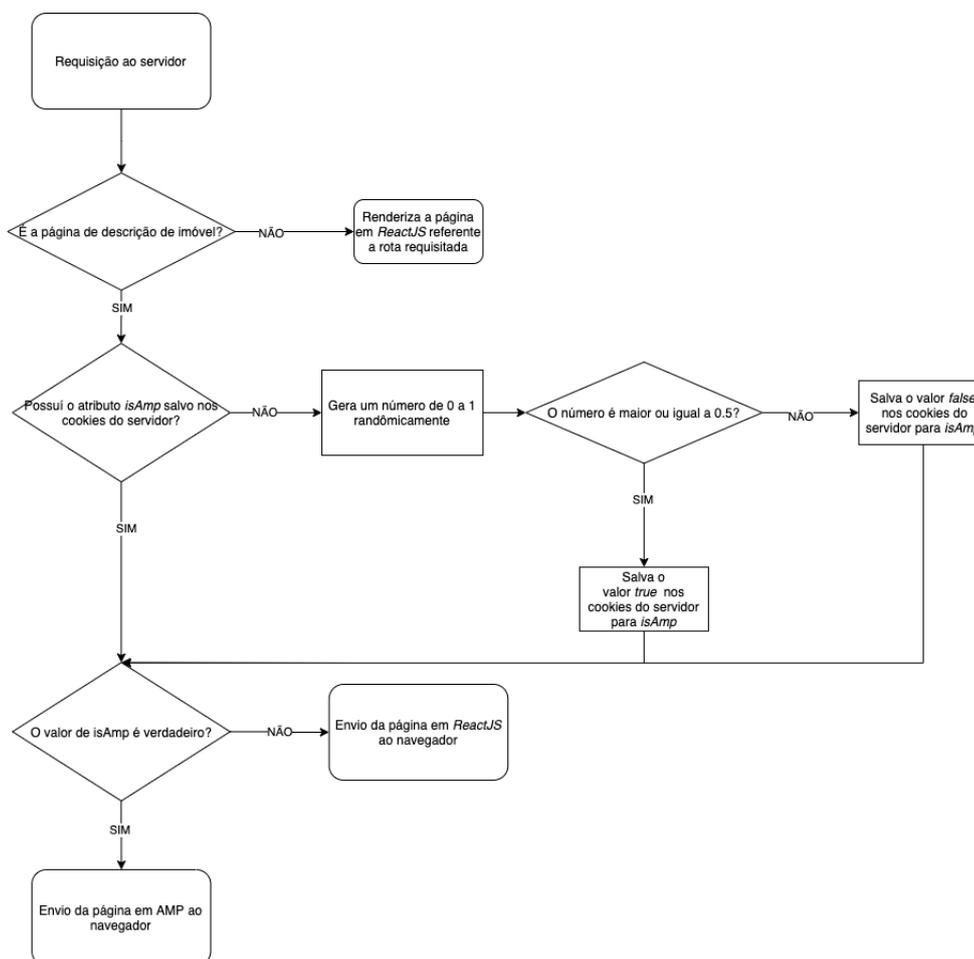
5.3 TESTE A/B

O *front-end* Canal Digital é desenvolvido atualmente com o *framework ReactJS* em que funciona com SSR, ou seja, todas as *views* são processadas pelo lado do servidor, e foi por isso que foi possível a implementação de uma página em AMP dentro do projeto.

O teste A/B foi feito no servidor do projeto e é esse que determina o que será enviado ao navegador e renderizado na tela. A realização do teste é esquematizado

em um fluxograma da figura 44, esse utiliza o artifício do navegador *cookies*, que são a maneira pela qual o site consegue armazenar e acessar dados salvos temporariamente referentes ao usuário no navegador (BATISTA, 2019).

Figura 44 – Fluxograma do teste A/B



Fonte – Autora

O funcionamento do teste A/B é baseado na rota de requisição feita ao servidor e o valor da variável *isAmp* presente nos *cookies* do navegador. Primeiramente é verificado se a rota é a referente a página de descrição de imóvel, caso sim há a verificação da existência da variável *isAmp* nos *cookies* do navegador do usuário, que apresenta um valor booleano, que direcionam a resposta *true* ao servidor para retornar a página em AMP ou a página em *ReactJS* quando *isAmp* é falso. Caso o usuário não apresente a variável nos *cookies* é sorteado um valor entre 0 e 1, sendo 0 primeiro inclusivo e 1 exclusivo, através da função *Math.random()* presente no *Javascript* (MATH. . . , 2019), e verificado se o valor é maior ou igual à 0.5, se o valor seja maior o atributo *isAmp* recebe *true* e caso menos *false*, determinando a resposta

do servidor. Sendo assim há a chance de 50% de um usuário ser direcionado para uma página AMP.

Utilizar *cookies* impede que o usuário caia de maneira aleatória em uma página em AMP ou de *ReactJS*, mantendo uma consistência nos dados capturados pelo GA.

6 ANÁLISE DOS RESULTADOS

A página dos imóveis desenvolvida e descrita no capítulo 5 foi enviada para a produção no dia 30 de outubro de 2020. Os resultados analisador a seguir serão referentes a um período de 15 dias a partir da captura dos dados através do GA, de 31 de outubro até 14 de novembro de 2020. Primeiramente, será analisado o impacto do uso no AMP na performance da página, e depois, o impacto da utilização desse na taxa de conversão para a realização do agendamento de visita.

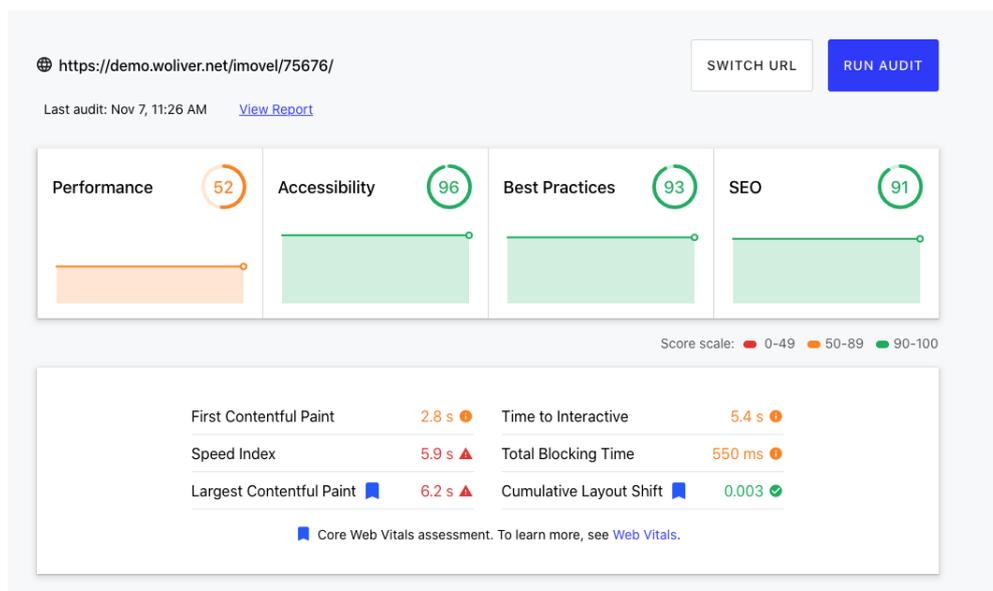
6.1 PERFORMANCE DA PÁGINA

Para a comparação entre as performances das páginas em *ReactJS* e AMP, foram realizados testes através de duas plataformas: o *Google Lighthouse*, já mencionado anteriormente, e também no site *WebPageTest.org*, uma ferramenta *open-source* que também mede métricas para a análise de performance (ISHAM, 2015).

6.1.1 *Google Lighthouse*

Foi realizado o teste na mesma página do imóvel que o mostrado na figura 24 e obteve-se resultados com o aumento da nota de performance em mais de 10 vezes do que a anterior, retratado na figura 45. Além de também aumentar as notas no quesito de acessibilidade, boas práticas e SEO.

Figura 45 – Notas da página de descrição de um imóvel em AMP no *Lighthouse*



Fonte – Site Web Dev Measure

O aumento considerável na nota foi ocasionado pelo principal ponto do AMP: não utilização de *Javascript* provenientes de bibliotecas externas. Houve também a diminuição dos valores das métricas relevantes ao tempo de renderização e de interação ao usuário, destacadas na tabela 2.

Tabela 2 – Comparação entre resultados obtidos no *Lighthouse* entre página em *ReactJS* e AMP

	Página ReactJS	Página AMP
Performance	3	52
Acessibilidade	74	96
Boas práticas	79	93
<i>First Content Paint</i>	13,0s	2,8s
<i>Speed Index</i>	16,8s	5,9s
<i>Largest Contentful Paint</i>	14,8s	6,2s
<i>Time to Interactive</i>	23,6s	5,4s
<i>Total Blocking Time</i>	1,450ms	550ms
<i>Culmutative Layout Shift</i>	0,521	0,003

Fonte – Autora

A nota da acessibilidade aumentou pelo uso correto de atributos como *aria-label*, fornecendo um título ou uma descrição a componentes nos quais um texto não é visível, em imagens, *links* ou nos mapas. O quesito de boas práticas foram relacionados ao uso de AMP e também pela nova disposição da UI da nova página. As métricas relacionadas a Performance listadas na tabela (*First Contentful Paint*, *Speed Index*, *Largest Contentful Paint*, *Time to Interactive*, *Total Blocking Time*) são relacionadas ao tempo de carregamento total da página, e a partir dessas é possível comparar o tempo de carregamento entre as diferentes páginas, em todos os pontos o AMP superou a página em *ReactJS*. Outra métrica a ser destacada é o *Culmutative Layout Shift*, ou Deslocamento Culmutativo de Layout, que mede a estabilidade visual da página, ou seja, o quanto a UI pode mudar variando o tamanho da janela do usuário, para essa métrica uma nota é considerada boa se for menor do que 0,1 e uma nota maior do que 0,25 é dita como ruim, a página em AMP traz uma qualificação muito melhor do que a em *ReactJS*, que ultrapassa a métrica estipulada como nota ruim.

Pelas notas geradas no *Google Lighthouse*, o uso do AMP traz uma melhora significativa a experiência do usuário, principalmente nos dispositivos móveis.

6.1.2 *WebPageTest.org*

Conforme realizado no *Google Lighthouse*, foi simulada a visita na página em um dispositivo Moto G4 com uma conexão 3G lenta, simulando um teste em um celular com baixa capacidade de CPU e com conexão instável da internet. O *site*

WebPageTest.org disponibiliza a opção de realizar a comparação entre dois testes (no caso entre *ReactJS* e AMP), que serão expostos a seguir.

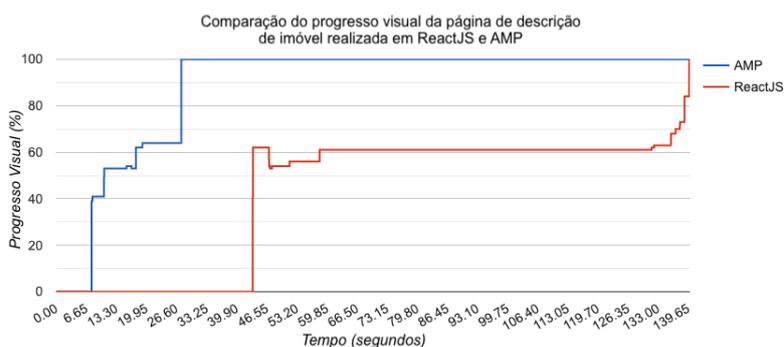
Inicialmente é efetuada a comparação entre o tempo de carregamento total da página e a visualização do carregamento de cada item da página. O resultado obtido é demonstrado na figura 46 e a diferença entre os períodos é visivelmente notada. O tempo de carregamento da página em AMP é de cerca de 28 segundos enquanto a de *ReactJS* ultrapassa de 2 minutos. Outra maneira de visualizar a diferença entre carregamento é através do gráfico presente na figura 47, que indica o percentual do conteúdo a ser visualizado em relação ao tempo, no qual a página em AMP termina o seu carregamento antes da em *ReactJS* começar a renderização de seus componentes.

Figura 46 – Comparação entre tempo de carregamento entre a página em *ReactJS* e AMP



Fonte – Site *WebPageTest.org*

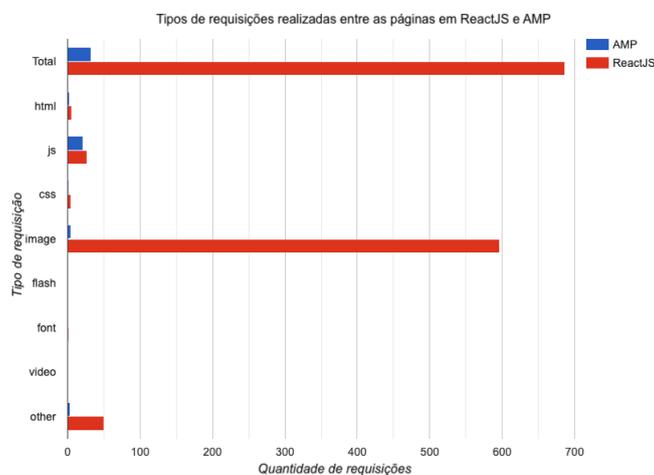
Figura 47 – Comparação entre o progresso visual entre a página em *ReactJS* e AMP



Fonte – Site *WebPageTest.org*

Um dos indicadores fornecidos pelo *WebPageTest.org* são as requisições realizadas para o carregamento da página, indicado no gráfico da figura 48. Pelos resultados obtidos é possível compreender o período relativamente maior de carregamento da página em *ReactJS*, ocasionado pelo alto número de requisições de imagens.

A página em AMP é destacada em ambos os testes realizados, o tempo de carregamento se mostrou ao menos 15 vezes mais rápido do que o previamente

Figura 48 – Tipos de requisições para o carregamento para as páginas em AMP e *ReactJS*

Fonte – Site WebPageTest.org

desenvolvido. Ambos os gráficos das figuras 47 e 48 destacam a melhora significativa no quesito de velocidade de carregamento da página, comprovando a eficiência do uso do AMP na página de descrição de imóvel.

6.2 IMPACTO DO USO AMP NA PLATAFORMA

O entendimento do comportamento do usuário na plataforma é possível através da captura dos dados realizados pelo GA. Para compreender o impacto do uso do AMP foi analisado primeiramente parâmetros referentes à página da descrição do imóvel, comparando dados da página em AMP e em *ReactJS* e depois o impacto de cada página na taxa de conversão de agendamento de visitas na plataforma.

6.2.1 Página de descrição do imóvel

Para ser possível adquirir dados referentes à página de descrição do imóvel, foi adicionado um filtro no conteúdo total de cada imobiliária para mostrar apenas aqueles referentes as páginas que contenham na rota “/imovel/” ou “/amp/imovel/”, filtrando respectivamente as páginas em *ReactJS* e AMP.

Primeiramente foram adquiridos os dados referentes ao número de visualizações de cada tipo de página, os dados obtidos das cinco imobiliárias mais visualizadas estão dispostos na tabela 3. Através desses dados é possível realizar a proporção de cada tipo de página visitada, avaliando a funcionalidade do teste A/B, disposto no gráfico da figura 49. Apesar de a proporção não ser de exatamente de 50%, conforme estabelecido no servidor do Canal Digital, a diferença ainda é pequena (2,1%) entre

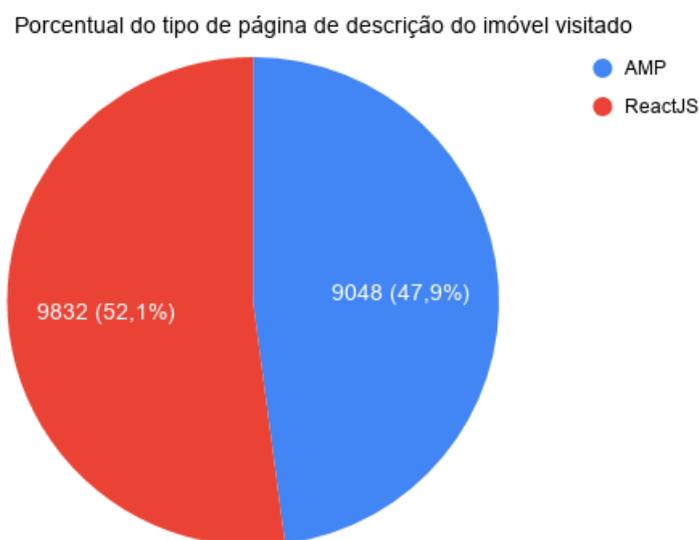
os tipos de páginas, não interferindo as análises seguintes.

Tabela 3 – Número de visitas na página de descrição de imóvel nas imobiliárias

	Terraz	Ayla	Brognoli	Brognoli Digital	Santa Maria
AMP	4299	1268	2662	635	184
ReactJS	4924	1820	2207	685	196

Fonte – Google Analytics Woliver

Figura 49 – Gráfico de pizza com dados do tipo de página de descrição do imóvel visitada



Fonte – Google Analytics Woliver

Outra métrica importante a se considerar na página em questão é a taxa de rejeição. No GA a taxa de rejeição é o percentual referente aos usuários que acessaram a página e não interagiram com o conteúdo (DIAS, 2019). No cenário da página de descrição do imóvel, quanto menor a taxa, maior é o número de usuários que seguem interagindo com a plataforma. Os dados referentes a cada imobiliária encontram-se na tabela 4, em média a taxa de rejeição das páginas em AMP são de 47,5%, sendo menores do que a em *ReactJS* que tem o valor de 56,3%, indicando que o impacto da performance da página também trouxe benefícios da diminuição de cerca de 9% na taxa de rejeição.

Tabela 4 – Taxa de rejeição na página de descrição do imóvel nas imobiliárias

	Terraz	Ayla	Brognoli	Brognoli Digital	Santa Maria
AMP	55,22%	44,81%	45,90%	41,35%	50,00%
ReactJS	59,09%	53,16%	56,69%	62,70%	50,00%

Fonte – Google Analytics Woliver

6.2.2 Taxa de conversão no número de agendamento de visitas

Para obter o número referente à taxa de conversão obtido com o uso ou não de AMP é adquirido os números de visitas realizadas pela plataforma. Nesse item é necessário utilizar o conceito de sessão do GA, que medem as ações que o usuário realiza em um *website*. Atualmente a ação de realizar a visita é contabilizada quando o usuário acessa a rota “/agendamento/passo-5” ou “agendamento/passo-5/”. Através dessa classificação é possível diferenciar os agendamentos provenientes da página de descrição do imóvel pelo filtro do conteúdo das sessões, pela funcionalidade de *Segments* do GA.

O número de visitas agendadas durante o período de 15 dias da aplicação estão expostos na tabela 5. Como já mencionado no capítulo 4, existem outras maneiras de acessar o fluxo de agendamento de visita diferenciados por “Outros Fluxos”. Como este documento é focado apenas no fluxo completo no Canal Digital será analisado a origem dos fluxos em *ReactJS* e AMP na figura 50. A melhora na performance ocasionada pelo AMP trouxe impactos significativos no número de visitas realizados, sendo mais da metade (61,2%) dos agendamentos visitas provenientes da página desenvolvida neste documento.

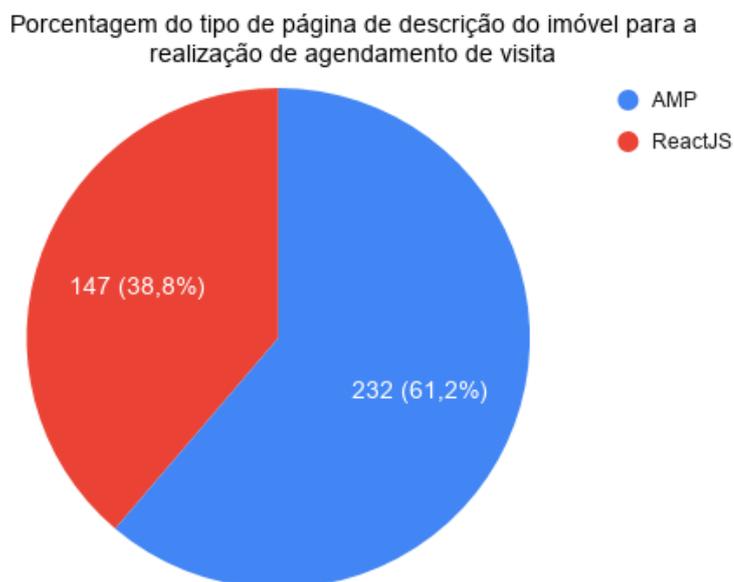
Tabela 5 – Número de visitas agendadas nas imobiliárias por tipo de página acessada

	Terraz	Ayla	Brognoli	Brognoli Digital	Santa Maria
AMP	77	48	90	16	1
ReactJS	51	24	61	10	1
Outros fluxos	76	84	138	105	11

Fonte – Google Analytics Woliver

Com os dados obtidos nas tabelas 3 e 5 é possível calcular o impacto do uso do AMP na taxa de conversão do número de agendamento de visita utilizando a fórmula da equação 1. A taxa de conversão em cada imobiliária estão dispostos na tabela 4. Realizando a média das cinco imobiliárias listadas, a taxa de conversão das páginas em AMP obteve um percentual de 2,40% contra 1,42% da em *ReactJS*. Apesar da taxa de conversão ser um percentual baixo, a diferença de quase o dobro em algumas

Figura 50 – Gráfico de pizza com o percentual de agendamento realizados com base na página de descrição do imóvel de origem



Fonte – Google Analytics Woliver

imobiliárias, como no Ayla, comprovam o impacto positivo do uso do AMP na principal página do Canal Digital, quase dobrando o número já existente.

$$TaxaConversao = \frac{VisitasMarcadas}{VisitasPaginaImovel} \quad (1)$$

Tabela 6 – Taxa de conversão de agendamento de visitas

	Terraz	Ayla	Brognoli	Brognoli Digital	Santa Maria
AMP	1,79%	3,79%	3,38%	2,52%	0,54%
ReactJS	1,04%	1,32%	2,76%	1,46%	0,51%

Fonte – Google Analytics Woliver

6.3 RESUMO

O estudo e desenvolvimento da página em AMP com a implementação do teste A/B foi importante para a obtenção das métricas relacionadas a performance e o real impacto no usuário final. Os valores obtidos pelo *Google Lighthouse* já se mostraram satisfatórios pelo aumento significativo na nota relacionada à performance. Os resultados dos testes provenientes do *website WebPageTest.org* confirmaram a

diminuição do tempo de carregamento total, trazendo gráficos que ressaltam a eficácia do AMP, que termina o seu carregamento total antes mesmo do que a página *ReactJS* comece esse processo.

O impacto do uso do *framework* comprova que a performance, considerando principalmente o tempo de carregamento, é de suma importância a experiência do usuário final. Apesar de trazer algumas perdas em alguns componentes do design proposto como ocorreu com o mapa de deslocamento podem ser descartadas, já que houve o aumento de mais de 1% da taxa de conversão do agendamento de visita e também pela diminuição da taxa de rejeição. O aumento do número de visitas agendadas em páginas em AMP trazem um impacto considerável no produto, que após esse período de testes, deve ser retirado o teste A/B no projeto Canal Digital e implementado uma lógica no servidor sempre retornar a página desenvolvida junto a remoção da página em *ReactJS* na *codebase* do projeto.

6.4 PERSPECTIVAS FUTURAS

O uso do *framework* AMP impactou positivamente o Canal Digital, oferecendo ao usuário final uma experiência ainda mais objetificada aos dispositivos móveis. Isso pode ser comprovado pelo número de visitas aumentarem de uma maneira significativa ocasionado por cerca de 50% dos usuários que acessaram a plataforma entre os dias 31 de outubro e 14 de novembro de 2020.

Como mencionado no capítulo 5, não foi possível adicionar o mapa dinâmico de geração de rotas em relação ao endereço da página do imóvel. A opção do uso da página em AMP implicaria na exclusão dessa funcionalidade. Caso seja optado por manter a página desenvolvida em *ReactJS* é sugerido encontrar alguma maneira de controlar as múltiplas requisições ocasionadas pelas imagens da página, usando um componente de *Lazy Loading*, por exemplo.

Para um impacto ainda maior no número final de agendamento de visitas a serem realizadas pelo Canal Digital, seria sugerido a atualização das páginas relacionadas à escolha do imóvel para que sejam utilizadas também em AMP. Porém, para que isso seja possível deve ser proposto outro UI para a página de busca, que consiga ser desenvolvido em AMP. Como os outros fluxos não são tão expostos aos usuários, não há necessidade da aplicação desse tipo de tecnologia.

O trabalho realizado não só foi importante para a aplicação de AMP exclusivamente no Canal Digital, mas serve também como parâmetro para a empresa Jungle Devs do que abrange a tecnologia em questão. Apesar de ser uma tecnologia inovadora, não se vê a necessidade da aplicação dessa em todos os projetos em *front-end* na empresa. O AMP é ideal para projetos em que precisam de usuários provenientes de fluxos de pesquisa, como o *Google* por exemplo, em que a permanência do usuário num fluxo inicial é essencial para o mantimento do produto.

7 CONCLUSÃO

O objetivo deste projeto foi a realização do estudo do AMP, um *framework* de componentes de *Web*, a aplicação desse na principal página de um SaaS de imobiliária, a de descrição de um imóvel. Finalizando com uma análise geral do seu impacto na taxa de conversão no número de visitas, com os dados obtidos através de um teste A/B.

Mesmo sendo uma tecnologia relativamente nova, a documentação referente aos componentes provenientes do AMP foram suficientes para a compreensão e uso desses. O que foi uma surpresa em comparação a outras documentações de tecnologias criadas pelo *Google*, como ocorre com o *GA* e o *Data Studio*, que são mais restritas. Das regras estipuladas pelo AMP alguns itens que foram mais difíceis de adaptar como os “Recursos dimensionados estaticamente” e a “Minimização do recálculo do estilo/layout”, já que o design visa ser mais responsivo e não com tamanho fixos.

Em relação ao desenvolvimento da página o maior desafio foi a não utilização de bibliotecas externas, o que geralmente visa a facilitar a criação de sites. O impedimento do uso da linguagem da programação fez com que a autora utilizasse apenas HTML e CSS, diferente do que estava sendo utilizado no resto do projeto (*ReactJS*), alguns elementos que já estavam prontos, como o cabeçalho e rodapé, tiveram que ser refeitos.

Dependendo do conteúdo da página a ser desenvolvida em AMP pode haver certos elementos que não poderão ser implementados pois não cumprem as regras implementadas pelo *framework*. Caso haja a necessidade de fazer múltiplos requisitos dinamicamente a uma API, como ocorreu na seção de deslocamento da página do imóvel por exemplo, impossibilitou a adição de um mapa que apresentasse a uma rota do imóvel atual a uma localização selecionada pelo usuário. O ponto de ter a página definitiva sem o mapa dinâmico mas com alta performance será levado em consideração pelo PO do projeto para a decisão da permanência ou não da página em AMP no Canal Digital.

Por relatos encontrados em artigos referentes à tecnologia em questão geraram expectativas em relação ao seu impacto no uso da plataforma. Os resultados obtidos nos quesitos da performance foram comprovados serem relevantes ao usuário final nos quesitos da taxa de rejeição e também pelo número total de agendamentos realizados entre as páginas.

A contribuição da autora foi de suma importância não somente ao número de visitas agendadas no Canal Digital, mas também serve como parâmetro na empresa Jungle Devs para a avaliação da utilização desse *framework* de componentes *web*.

Para ter um impacto ainda maior no número de agendamento de visitas é suge-

rido que os próximos passos sejam a refatoração das páginas anteriores à página de descrição do imóvel (a página inicial e a de busca) para a utilização em AMP, já que a grande maioria dos usuários tendem a usar dispositivos móveis para o acesso da plataforma. Caso isso seja possível dentro do escopo, sugere-se a implementação de um *design* que seja favorável ao uso do AMP, sem funcionalidades com a necessidade de muito lógica em uma página. Já que o desenvolvimento em AMP naturalmente leva mais tempo de execução do que em *ReactJS*.

Outro ponto que pode trazer uma melhora ainda maior na performance é a utilização de um otimizador sugerido pela documentação do AMP. No escopo da realização desse projeto não foi possível devido a versão do *NodeJS* utilizada.

Além desses aspectos referentes ao AMP, esse período foi importante para a autora se inteirar ainda mais sobre o projeto, fazendo parte também da tomada de decisão referentes a outros fluxos que estavam sendo realizados paralelamente pela tribo.

A realização da graduação do curso de Engenharia de Controle e Automação e os conhecimentos adquiridos foram essenciais para a formação da profissional que a autora visa ser. Apesar do documento não enquadrar fatos estudados teoricamente em controle, a maneira a qual é aprendida de lidar com problemas e a análise através de métricas foram realizadas inúmeras vezes ao decorrer do curso, foram replicados durante o período de estágio. Conteúdos aprendidos em matérias referentes à desenvolvimento do código foram fundamentais para a escolha profissional de Engenharia de Software.

O ambiente de aprendizado fornecido pela empresa Jungle Devs foi essencial para consolidar e reforçar conceitos previamente adquiridos. O projeto realizado nesse documento consolida fielmente os princípios do programa *Academy* oferecido pela empresa, mostrando a importância desse para o desenvolvimento pessoal e profissional.

REFERÊNCIAS

BATISTA, Adonis. **Você sabe o que são cookies na internet? Conheça os 3 tipos.**

[S.l.]: Hariken, jan. 2019. Disponível em: <https://blog.hariken.co/voce-sabe-o-que-sao-cookies-na-internet-conheca-os-3-tipos/>.

BREUX, Guillaume. **Client-side vs. Server-side vs. Pre-rendering for Web Apps.**

[S.l.]: Toptal, set. 2018. Disponível em: <https://www.toptal.com/front-end/client-side-vs-server-side-pre-rendering>.

BURKHOLDER, Benjamin. **JavaScript SEO: Server Side Rendering vs Client Side Rendering.** [S.l.: s.n.], fev. 2019. Disponível em:

<https://medium.com/@benjburkholder/javascript-seo-server-side-rendering-vs-client-side-rendering-bc06b8ca2383>.

CELKE. **Inserir CSS inline, CSS incorporado e CSS externo.** [S.l.: s.n.]. Disponível em: <https://celke.com.br/artigo/inserir-css-inline-css-incorporado-e-css-externo>.

COMITÊ GESTOR DA INTERNET NO BRASIL. **TIC Domicílios 2019: A pesquisa TIC Domicílios tem o objetivo de mapear o acesso às TIC nos domicílios urbanos e rurais do país.** Rio de Janeiro, dez. 2019.

COMO usar AMP com o Google Ads - Ajuda do Google Ads. [S.l.]: Google. Disponível em: https://support.google.com/google-ads/answer/7495018?hl=pt-BR&ref_topic=3121935.

DIAS, Tassia. **Entenda o que é bounce rate e saiba se a taxa de rejeição afeta ou não o seus rankings no Google.** [S.l.]: Rock Content, out. 2019. Disponível em:

<https://rockcontent.com/br/blog/bounce-rate/>.

ESPINHA, Roberto Gil. **Kanban O que é e TUDO sobre como gerenciar fluxos de trabalho.** [S.l.: s.n.]. <https://artia.com/kanban/>.

FERNANDES, Diego. **React Hooks: Como utilizar, motivações e exemplos práticos.** [S.l.]: Blog da Rocketseat, mai. 2019. Disponível em:

<https://blog.rocketseat.com.br/react-hooks/>.

FIRST Contentful Paint. [S.l.: s.n.], mai. 2019. Disponível em:

<https://web.dev/first-contentful-paint/>.

G., Domantas. **Web What is HTML? The Basics of Hypertext Markup Language Explained.** [S.l.: s.n.]. [https://www-hostinger-](https://www-hostinger-com.cdn.ampproject.org/c/s/www.hostinger.com/tutorials/what-is-html/amp/)

[com.cdn.ampproject.org/c/s/www.hostinger.com/tutorials/what-is-html/amp/](https://www-hostinger-com.cdn.ampproject.org/c/s/www.hostinger.com/tutorials/what-is-html/amp/).

GET Started Maps Embed API. [S.l.]: Google, set. 2020. Disponível em:

<https://developers.google.com/maps/documentation/embed/get-started>.

GONÇALVES, Ariane. **O que é CSS? Guia Básico para Iniciantes.** [S.l.: s.n.], ago. 2019. Disponível em:

<https://www.hostinger.com.br/tutoriais/o-que-e-css-guia-basico-de-css/>.

HANASHIRO, Akira. **O que é DOM, Virtual DOM e Shadow DOM?** [S.l.]: Akira Hanashiro, abr. 2020. Disponível em:

<https://www.treinaweb.com.br/blog/o-que-e-dom-virtual-dom-e-shadow-dom/>.

HOLLA, Suhas. **Chapter 5 – ReactJS Component State.** [S.l.]: Suhas Holla.

Disponível em: https://www.techdiagonal.com/reactjs_courses/beginner/reactjs-component-state/.

https://www.techdiagonal.com/reactjs_courses/beginner/reactjs-component-state/.

HOW AMP works. [S.l.]: AMP Project. Disponível em:

<https://amp.dev/about/how-amp-works/>.

ISHAM, Mark. **The SEO Expert's Guide to Web Performance Using WebPageTest.**

[S.l.]: Moz, jul. 2015. Disponível em: <https://moz.com/blog/the-seo-experts-guide-to-web-performance-using-webpagetest>.

JÚNIOR, Ademilson. **O que é White Label e quais suas principais vantagens.**

[S.l.: s.n.], Set 2019. Disponível em: <https://blog.configr.com/o-que-e-white-label-e-suas-principais-vantagens/>.

JUNIOR, por Bogнар; JUNIOR, Bogнар. **React – Componentes de Classe e Funcionais.** [S.l.]: Bogнар Junior, ago. 2018. Disponível em:

<https://bognarjunior.wordpress.com/2018/08/13/react-componentes-de-classe-e-funcionais/>.

JUSTO, Andreia Silva. **Conheça 3 metodologias ágeis que vão transformar o seu jeito de conduzir projetos.** [S.l.: s.n.].

<https://www.euax.com.br/2019/04/metodologias-ageis/>.

LEVISON, Mark. **O que são os "Story Points"? Eles são necessários?** [S.l.]: InfoQ, mar. 2010. Disponível em:

[%5Curl%7Bhttps://www.infoq.com/br/news/2010/03/story-points/%7D](https://www.infoq.com/br/news/2010/03/story-points/).

LIGHTHOUSE accessibility scoring. [S.l.: s.n.]. Disponível em:

<https://web.dev/accessibility-scoring/>.

LONGEN, Andrei. **O que é JavaScript e Como Funciona.** [S.l.: s.n.], dez. 2019a. Disponível em:

<https://www.weblink.com.br/blog/programacao/o-que-e-javascript/>.

LONGEN, Andrei. **O que é JavaScript: comece a aprender sobre JavaScript agora.** [S.l.]: Gabriel Zampieri, jan. 2019b. Disponível em:

<https://www.hostinger.com.br/tutoriais/o-que-e-javascript/>.

LONGEN, Andrei. **O Que é React e Como Funciona? - Guia para Iniciantes.** [S.l.: s.n.], mai. 2019c. Disponível em:

<https://www.hostinger.com.br/tutoriais/o-que-e-react-javascript>.

MALDONADO, Leonardo. **Entendendo o DOM (Document Object Model) - Tableless - Website com artigos e textos sobre Padrões Web, Design, Back-end e Front-end tudo em um só lugar.** [S.l.: s.n.], fev. 2019.

<https://tableless.com.br/entendendo-o-dom-document-object-model/>.

MÁRCIOD'ÁVILA. **Janela modal na web.** [S.l.: s.n.], dez. 2006. Disponível em:

http://www.mhavila.com.br/topicos/web/window_modal.html.

MARK OTTO, Jacob Thornton. **Carousel.** [S.l.: s.n.]. Disponível em:

<https://getbootstrap.com.br/docs/4.1/components/carousel/#:~:>

[text=0%20carousel%20%C3%A9%20um%20slideshow,controles%20anterior,%20pr%C3%B3ximo%20e%20indicadores..](https://getbootstrap.com.br/docs/4.1/components/carousel/#:~:)

MARQUES, Rafael. **O que é HTML? Entenda de forma descomplicada | Homehost.** [S.l.: s.n.].

<https://www.homehost.com.br/blog/tutoriais/o-que-e-html/>.

MATH.RANDOM(). [S.l.: s.n.], mai. 2019. Disponível em:

[https://developer.mozilla.org/pt-](https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Math/random)

[BR/docs/Web/JavaScript/Reference/Global_Objects/Math/random](https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Math/random).

MORAED, Daniel. **Taxa de conversão: descubra agora como calcular!** [S.l.]: Rock Content, nov. 2019. Disponível em:

<https://rockcontent.com/br/blog/taxa-de-conversao/>.

MORRIS, Scott. **What Is JavaScript? A Guide for Total Beginners.** [S.l.]: Scott Morris. Disponível em: <https://skillcrush.com/blog/javascript/>.

OLIVEIRA, Welliton. **O que é scrum? Conceito, definições e etapas.** [S.l.: s.n.].

<https://evolvemp.com/o-que-e-scrum-conceito-definicoes-e-etapas/>.

PATEL, Neil. **Google Data Studio: O Que É e Como Criar Relatórios (13 Dicas).**

[S.l.]: Neil Patel, dez. 2019. Disponível em:

<https://neilpatel.com/br/blog/google-data-studio/>.

PATEL, Neil. **SaaS: O Que É, Como Funciona, Exemplos e Benefícios.** [S.l.: s.n.],

nov. 2020. Disponível em: <https://neilpatel.com/br/blog/saas/>.

PAULA, GILLES B. DE. **Tudo sobre Metodologia Scrum: o que é e como essa ferramenta pode te ajudar a poupar tempo e gerir melhor seus projetos.**

[S.l.: s.n.], set. 2020. <https://www.treasy.com.br/blog/scrum/>.

PEREIRA, Ana Paula. **O que é CSS?** [S.l.: s.n.], out. 2020. Disponível em:

<https://www.tecmundo.com.br/programacao/2705-o-que-e-css-.htm>.

PHILLIPS, Michael. **What is Google Lighthouse and How Do You Use It?** [S.l.]:

Michael Phillips, ago. 2019. Disponível em:

<https://www.greengeeks.com/blog/google-lighthouse-how-you-use-it/>.

ROCHA, Fernando. **O que é Server Side Rendering e como usar na prática.**

[S.l.: s.n.], dez. 2018. Disponível em: <https://medium.com/techbloghotmart/o-que-%C3%A9-server-side-rendering-e-como-usar-na-pr%C3%A1tica-a840d76a6dca>.

SMITH, Stephanie. **What Is Server-Side Rendering?** [S.l.]: Stephanie Smith, jul.

2020. Disponível em:

<https://dev.to/seal125/what-is-server-side-rendering-22ik>.

SPERO, Jason. **The Digital Consumer Behaviors Shaping Mobile - Think with Google**. [S.l.: s.n.], mai. 2017. <https://www.thinkwithgoogle.com/consumer-insights/consumer-trends/consumer-behavior-mobile-digital-experiences/>.

TRÊS Primeiros passos - Noções Básicas de Git. [S.l.: s.n.], nov. 2019. Disponível em: <https://git-scm.com/book/pt-br/v1/Primeiros-passos-No%C3%A7%C3%B5es-B%C3%A1sicas-de-Git>.

TUTORIAL: Intro to React. [S.l.: s.n.]. Disponível em: <https://reactjs.org/tutorial/tutorial.html>.

VALLE, Alberto. **O que é Google Analytics**. [S.l.: s.n.], mai. 2020. Disponível em: <https://www.academiadomarketing.com.br/o-que-e-google-analytics/>.

WHAT is SEO? Search Engine Optimization 2020. [S.l.: s.n.]. Disponível em: <https://moz.com/learn/seo/what-is-seo>.