

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA**

Alexandre Garcez Nunes e Luís Eduardo Vieira Tarouco Bueno

**APLICAÇÃO WEB PROGRESSIVA PARA O
GERENCIAMENTO UNIFICADO DE PONTOS DE
FIDELIDADE ENTRE BARES E RESTAURANTES**

Florianópolis

2020

Alexandre Garcez Nunes e Luís Eduardo Vieira Tarouco Bueno

**APLICAÇÃO WEB PROGRESSIVA PARA O
GERENCIAMENTO UNIFICADO DE PONTOS DE
FIDELIDADE ENTRE BARES E RESTAURANTES**

Trabalho de conclusão de curso apresentado como parte dos requisitos para a obtenção do grau de “Bacharel em Sistemas de Informação”.

Florianópolis, 06 de novembro 2020.

Banca Examinadora:

Prof. Leandro José Komosinski
Orientador

Prof. Ricardo Pereira e Silva

Prof. Mateus Grellert da Silva

SUMÁRIO

Resumo	
Abstract	
Lista de Figuras	
Lista de Abreviaturas e Siglas	
1 INTRODUÇÃO	13
1.1 OBJETIVO GERAL	14
1.2 OBJETIVOS ESPECÍFICOS	14
2 FUNDAMENTAÇÃO TEÓRICA E TECNOLÓGICA	17
2.1 MARKETING RELACIONAL	17
2.1.1 Perspectiva do Marketing Relacional	18
2.2 <i>JAVASCRIPT</i>	19
2.2.1 Conceitos JavaScript	20
2.3 APLICAÇÃO WEB PROGRESSIVA	22
2.3.1 Vantagens versus Desvantagens	22
2.3.1.1 Vantagens	23
2.3.1.2 Desvantagens	23
2.3.2 Visão Geral da Arquitetura	24
2.3.2.1 <i>Service Worker</i>	24
2.3.2.2 <i>Application Shell</i>	24
2.4 VUE	25
2.4.1 Visão Geral da Arquitetura	26
2.5 FIREBASE	27
2.5.1 Firebase User Authentication	28
2.5.2 Firebase Realtime Database	29
2.5.2.1 <i>Setup</i>	30
2.5.2.2 Leitura dos Dados	30
2.5.2.3 Escrever Dados	31
2.6 ENCERRAMENTO	31
3 SOLUÇÕES EXISTENTES	33
3.1 ZYGO FERRAMENTA DE FIDELIZAÇÃO	33
3.2 DOTZ	34
3.3 ENCERRAMENTO	35
4 SOLUÇÃO PROPOSTA	37
4.1 ESPECIFICAÇÕES	37
4.1.1 Regras de Negócio	37
4.1.1.1 Usuários	38
4.1.1.2 Estabelecimento	38

4.1.1.3 Estabelecimentos - Programas de Fidelidade	39
4.1.1.4 Estabelecimentos - Produtos	39
4.1.1.5 Estabelecimentos - Cupons	39
4.1.1.6 Estabelecimentos - Pontuar Clientes	39
4.1.2 Requisitos Funcionais	40
4.1.3 Diagrama de Classes	44
4.1.4 Casos de Uso	45
4.1.5 Persistência de Dados e Integração	47
4.2 DESENVOLVIMENTO	50
4.2.1 Tornando a aplicação uma PWA	51
5 RESULTADOS OBTIDOS	55
5.1 INTERFACE ESTABELECIMENTO ADMINISTRADOR ..	56
5.2 INTERFACE CLIENTE USUÁRIO	61
6 CONCLUSÃO	65
6.1 TRABALHOS FUTUROS	66
APÊNDICE A – Código-Fonte	71
APÊNDICE B – Artigo	75
REFERÊNCIAS	87

RESUMO

Com a disseminação da tecnologia nos mais diversos setores, como é o caso do setor gastronômico, notamos uma marcante presença no uso de aplicações de software que visam fomentar o lucro e a qualidade dos produtos e serviços prestados por estabelecimentos comerciais, bem como fornecer e facilitar informações a seus clientes. Muitos desses estabelecimentos não possuem uma certa clareza sobre suas fraquezas, não conseguindo, desta forma, alcançar todo o potencial possível. Nesse sentido, um dos pontos importantes a ser levado em consideração, é o que esse estabelecimento pode e deve fazer para fidelizar e conquistar novos clientes.

No mercado gastronômico atual, podemos encontrar uma diversidade de programas de fidelidade que prometem mitigar esse problema, todavia, por serem muitos e dos mais diversos, acabam pecando na usabilidade e simplicidade, criando uma espécie de resistência ao seu uso pelos próprios clientes.

Pensando nisso, o presente projeto apresenta como solução o desenvolvimento de uma Aplicação *Web* Progressiva responsável por gerenciar e disponibilizar programas de fidelidade a clientes, assim como permitir aos usuários de bares e restaurantes o controle de todos os seus pontos de fidelidade obtidos. A aplicação visa, em primeiro lugar, a simplicidade, tanto com relação aos usuários clientes de estabelecimentos gastronômicos quanto aos gestores destes estabelecimentos, ambos com acesso a plataforma por diferentes interfaces.

Palavras-chave: *PWA*. *Vue*. *Firebase*. Restaurantes. Marketing Relacional. Fidelidade. Cupom.

ABSTRACT

With the dissemination of technology in the most diverse sectors, as is the case of the gastronomic sector, we note a marked presence in the use of software applications that aim to promote profit and the quality of products and services provided by commercial establishments, as well as providing and facilitating information to your customers. Many of these establishments are not clear about their domains and weaknesses, thus failing to reach their full potential. In this sense, one of the important points to be taken into account is what this establishment can and must do to retain and win new customers.

In the current gastronomic market, we can find a variety of loyalty programs that promise to mitigate this problem, however, because they are many and the most diverse, they end up sinning in usability and simplicity, creating a kind of resistance to their use by the customers themselves.

With this in mind, this work presents as a solution a progressive web application of loyalty points in an integrated way. The application aims, in the first place, for simplicity and usability, both in relation to the part destined to customers and the part destined to restaurants. In addition, but not less important, the application has a kind of wallet points for its customers, making it not necessary to have a specific website or application for each restaurant that the customer has points, which means that, in a single application, it is possible to manage the points of several restaurants.

Keywords: *PWA. Vue. Firebase* Restaurants. Relationship marketing. Loyalty program.

LISTA DE FIGURAS

Figura 1	Evolução do Marketing Relacional	17
Figura 2	Tecnologias populares <i>Stack Overflow</i>	20
Figura 3	Exemplo do funcionamento do <i>Application Shell</i>	25
Figura 4	Vue - Visão geral	26
Figura 5	Exemplo de estrutura JSON.	29
Figura 6	<i>Setup Firebase</i>	30
Figura 7	Código HTML.	30
Figura 8	Escrevendo no Banco Firebase.	31
Figura 9	Portal Fidelidade	33
Figura 10	Site oficial Clube Dotz.	34
Figura 11	Diagrama de Classes.	45
Figura 12	Casos de Uso	47
Figura 13	Base de Dados na interface Firebase.	48
Figura 14	Representação estrutura base de dados.	49
Figura 15	Preços Firebase.	49
Figura 16	Configuração Inicial Firebase	50
Figura 17	Trecho de código do método login	51
Figura 18	Trecho de código do arquivo manifest.json	52
Figura 19	Trecho de código do arquivo index.html.	53
Figura 20	Métricas de Qualidade.	56
Figura 21	Tela de login.	57
Figura 22	Tela inicial Administrador Restaurante.	58
Figura 23	Tela Gerenciamento Administrador Restaurante.	59
Figura 24	Gerenciamento de cupons e planos de fidelidade	60
Figura 25	Simulador de Cupons.	61
Figura 26	Tela Inicial Usuário Cliente.	62
Figura 27	Tela Dados Cadastrais usuário cliente.	63

LISTA DE ABREVIATURAS E SIGLAS

TIC	Tecnologia da Informação e Comunicação	13
HTML	<i>Hyper Text Markup Language</i>	20
SQL	<i>Structured Query Language</i>	20
PWA	<i>Progressive Web App</i>	22
DOM	<i>Document Object Model</i>	24
CSS	<i>Cascading Style Sheets</i>	24
JSON	JavaScript Object Notation	27
SDK	<i>Software Development Kit</i>	28
ID	<i>Identity</i>	29
URL	<i>Uniform Resource Locator</i>	30
CPF	Cadastro de Pessoa Física	34
UML	<i>Unified Modeling Language</i>	37
MVP	<i>Minimum Viable Product</i>	37
API	<i>Application Programming Interface</i>	50
SEO	<i>Search Engine Optimization</i>	55
CRM	<i>Customer Relationship Management</i>	66

1 INTRODUÇÃO

Com o surgimento de novas tecnologias e o aprimoramento das já existentes, projetos e aplicações disseminaram em diversos modelos de negócios afetando não só a maneira como os produtos e serviços são aplicados na sociedade mas também na forma com que as pessoas realizam negócios.

Segundo Centro Regional de Estudos para o Desenvolvimento da Sociedade da Informação em sua Pesquisa sobre o uso das tecnologias de informação e comunicação nas empresas brasileiras - TIC Empresas 2019, 98% das empresas utilizaram internet nos últimos 12 meses, sendo o mercado de alimentação e alojamento o menos expressivo com 91% de utilização (BRASIL, 2020).

Utilizar a internet como plataforma de negócio ou como apoio para resolver problemas e atrair clientes tem sido cada vez mais crucial para a competitividade e sustentabilidade das empresas. Aprimorar a relação de lealdade e oferecer experiências bem sucedidas ao invés de apenas bons produtos e serviços tem feito a diferença na escolhas dos clientes diante a tantas opções de mercado.

Estratégia muito utilizada no relacionamento empresa-cliente e que possui como principal objetivo estabelecer relações mutuamente satisfatórias de longa duração, o Marketing Relacional é definido por Berry Parasuraman (1991) como um conceito que tem por objetivo atrair clientes e manter relações duradouras, é o processo que utiliza os recursos da empresa para criar a melhor experiência possível e o máximo valor para o cliente, englobando um conjunto de atividades de Marketing orientadas para o estabelecimento de relações a longo prazo, com base nas promessas mútuas de confiança entre a empresa e o cliente.

Considerando isso, este projeto busca explorar o Marketing Relacional junto às empresas do setor de gastronomia, como bares e restaurantes, oferecendo uma solução tecnológica que incentive e auxilie as empresas na construção de boas estratégias de Marketing Relacional e forneça aos clientes aproximação com a rede de benefícios além da facilidade em pontuar e gerenciar em um único local seus pontos obtidos nos diferentes estabelecimentos frequentados.

Para isto, é proposto o desenvolvimento de uma Aplicação Web Progressiva, disponível para navegação multiplataforma. A aplicação visa uma experiência confiável e semelhante a de um aplicativo nativo, permitindo o acesso em qualquer dispositivo através de um navegador

de internet. De acordo com Richard (2020) as PWAs são caracterizadas como rápidas, respondendo instantaneamente as interações dos usuários; confiáveis, com carregamento rápido independente da qualidade da rede; e envolvente, com interações e navegação semelhante aos aplicativos nativos, possibilitando notificações push e inserção de um atalho da aplicação na tela inicial do dispositivo móvel.

Neste projeto a utilização do PWA se constitui principalmente pela disponibilidade a todos os públicos, independente do dispositivo, pela possibilidade de utilização sem baixar a aplicação, pela alternativa em progressivamente instalar a aplicação após a aprovação do seu conteúdo, e por fim, pelo engajamento promovido após a aproximação e fácil acessibilidade fornecida aos usuários. Todas essas características apresentadas vão ao encontro da aplicação desenvolvida.

No decorrer do trabalho serão apresentadas as principais características e vantagens do Marketing de Relacionamento e a forma como a aplicação incorporará essas estratégias em sua solução.

Na sequência serão apresentadas as tecnologias que em conjunto com as estratégias de mercado trarão benefícios aos usuários da aplicação, sejam eles estabelecimentos ou clientes.

E por fim, o projeto da solução desenvolvida contemplado por sua análise, recursos utilizados no desenvolvimento da aplicação e seus resultados obtidos.

1.1 OBJETIVO GERAL

Desenvolver uma Aplicação Web Progressiva responsável por armazenar em um único local todos os pontos de fidelidade obtidos pelo usuário cliente com os estabelecimentos parceiros da aplicação, assim como proporcionar aos estabelecimentos uma solução inteligente que auxilie na divulgação e criação de campanhas bem sucedidas para retenção de clientes.

1.2 OBJETIVOS ESPECÍFICOS

- Desenvolver uma aplicação com visual simples e fácil uso para todos os públicos de usuário;
- Desenvolver uma aplicação dinâmica e funcional através de tecnologias novas;

- Desenvolver uma aplicação que amplie a divulgação de estabelecimentos que possuem benefícios aos clientes;
- Desenvolver uma aplicação que permita aos clientes de bares e restaurantes gerenciar em um único local todos os pontos de fidelidade obtidos em diferentes estabelecimentos;
- Auxiliar através da aplicação desenvolvida os estabelecimentos na criação de programas de fidelidade e disponibilização de cupons;

2 FUNDAMENTAÇÃO TEÓRICA E TECNOLÓGICA

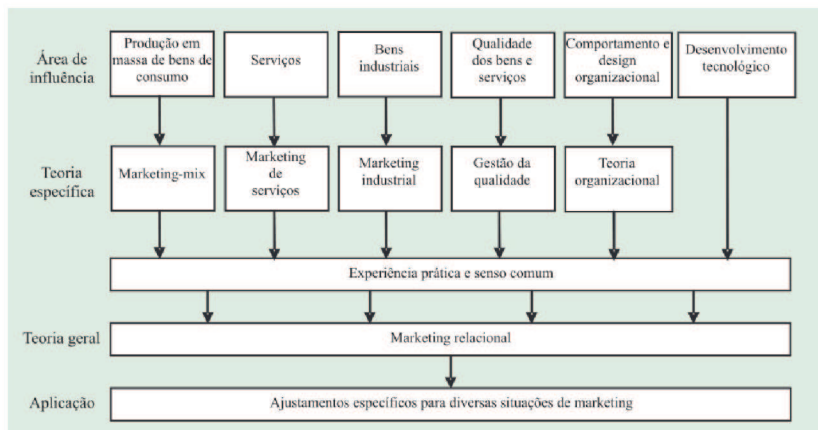
Nesse capítulo são apresentadas os conceitos e tecnologias utilizadas no projeto.

2.1 MARKETING RELACIONAL

A maioria das teorias do marketing foram desenvolvidas a partir do mercado de produtos de grande consumo. Dessa forma, analisar marketing associado aos mercados de serviços sugere uma perspectiva diferente da habitual. Nesse contexto houve o advento do conceito de Marketing Relacional.

Segundo Antunes Joaquim e Rita (2008), pesquisadores consideram o marketing relacional como um conjunto de contribuições e abordagens multi-disciplinares, as quais são influenciadas por áreas associadas ao comportamento da organização, ao desenvolvimento de serviços e as tecnologias utilizadas. O marketing relacional é conhecido como uma evolução ao marketing de serviço e à gestão da qualidade, através da Figura 1, Gummesson (1998) representa sua evolução e desdobramentos que levaram à teoria do marketing relacional e suas aplicações.

Figura 1 – Evolução do Marketing Relacional



Fonte: Gummesson (1998)

Na década de 90 pesquisadores como Christopher (1994) estu-

davam abordagens do marketing relacionado a canais de distribuição, onde o foco estava associado ao desenvolvimentos de relações com o consumidor. O comportamento do consumidor tornou-se de fato nos últimos anos um objeto de ampla atenção na literatura e prática empresarial devido ao excesso de capacidade produtiva por parte das empresas, à maior intensidade da concorrência e a ampliação de preocupação com a qualidade por parte dos consumidores e fornecedores.

Segundo Gummesson (1998) os pilares da teoria devem ser os seguintes:

- A relação: o marketing deve estar orientado para a criação, manutenção e desenvolvimento de relações com os clientes;
- A interatividade das partes: as relações entre vendedores e clientes para a criação e entrega mútua de valor exigem um estreito e intenso processo de comunicação entre ambos;
- O longo prazo: para criar, manter e desenvolver as relações é necessário um longo espaço temporal.

2.1.1 Perspectiva do Marketing Relacional

O marketing tradicional tem como característica principal prospectar clientes, no entanto, essa característica é definida por apenas uma parte das perspectivas do marketing relacional, pois nele também são adicionadas ações para manter e consolidar a relação com o cliente a longo prazo. A realidade orientada a criação de valor desenvolveu um marketing personalizado que satisfaça as necessidades dos consumidores. Uma evolução da abordagem orientada ao produto para orientada à recursos.

A organização se habilita com estratégias para criação de valor na satisfação do cliente atuando na retenção do cliente com um maior nível de compromisso (ANTUNES JOAQUIM E RITA, 2008). Abaixo itens que definem as atividades e estrutura do Marketing Relacional:

- Ênfase na retenção dos clientes;
- Ênfase no valor do cliente;
- Visão a longo prazo;
- Grande ênfase no serviço ao cliente;
- Contato contínuo com o cliente;

- Elevado nível de compromisso com os clientes;

- Qualidade é uma preocupação de todos.

Segundo Toledo (2015) o custo e efeito de aquisição e retenção de clientes no resultados das empresas adquirem um peso significativo na composição de valor de mercado da instituição. Empresas que possuem uma boa gestão do valor do cliente contribuem com o lucro a longo prazo, pois concentram seus esforços e receitas para definir, analisar e captar os segmentos de clientes mais atrativos para o negócio.

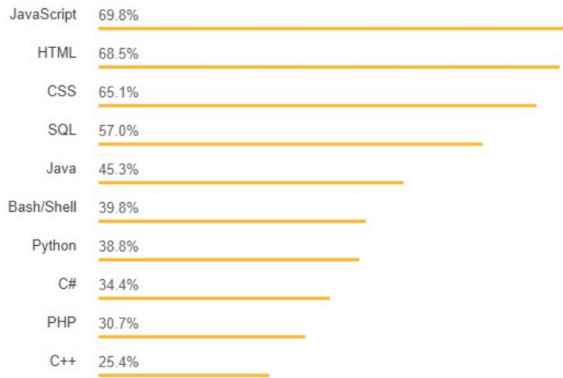
O processo de gestão do valor compreende um conjunto integrado e interdependente de atividades estratégicas e operacionais que reflete um sistema de decisões. Essa filosofia direciona as ações de marketing, moldando a criação e entrega de valor para os clientes, o qual culmina com a criação de valor para a marca (TOLEDO, 2015).

2.2 JAVASCRIPT

Criada em 1995 pela Netscape Communications Corporation, o *JavaScript* foi desenvolvido inicialmente com o nome de *Mocha*, posteriormente passou a se chamar *Live Script* para finalmente ser lançado com o nome atual no navegador *Netscape*. A denominação da linguagem, se deve a similaridades com a sintaxe do Java, embora as duas linguagens não tenham nenhuma outra relação além desta.

Visando implementar uma tecnologia de processamento modo cliente a linguagem de programação que tem como objetivo possuir uma sintaxe dinâmica, fácil de entender e poderosa. O *JavaScript* começou como uma pequena linguagem de *script* para uso no Netscape Communicator, quando a *Web* ainda era uma nova invenção (WOHLGETHAN, 2018). Atualmente, o *JavaScript* é a linguagem de programação mais usada no mundo, segundo estudo publicado pelo site *Stack Overflow*, mostrado na Figura 2.

Figura 2 – Tecnologias populares *Stack Overflow*.



Fonte: *Stack OverFlow*. Overflow (2018)

O *JavaScript* permite criar pequenos programas embutidos no próprio código de uma página HTML, é capaz de gerar valores, processar alguns dados, fazer validações de formulários, alterar e criar valores de elementos em HTML. A maioria das funcionalidades são executadas diretamente no computador cliente, evitando a troca de dados e informações com o servidor, desta forma, o tempo de processamento depende apenas do processamento local do lado do cliente, não dependendo assim de latência da rede, ou seja, o tempo do momento de envio de um pacote pela rede até o recebimento no destino, tempo ofensor em processamentos principalmente associados a navegação na web na década de 90.

A popularidade do *JavaScript* decorre de sua natureza dinâmica, diversas bibliotecas podem ser baixadas em tempo de execução, todos os navegadores modernos permitem a execução de *scripts* nas páginas Web, possibilitando modificações de maneira dinâmica nas informações apresentadas. Sua flexibilidade trás pontos de atenção, em razão de seus poucos mecanismos de proteção ou ocultação de informações, conseqüentemente abrem vulnerabilidades de segurança, como *scripts* entre sites e ataques de *SQL Injection*.

2.2.1 Conceitos *JavaScript*

Nessa subseção são descritos alguns conceitos básicos da linguagem em tópicos que caracterizam seu potencial de utilização na reali-

dade atual segundo Grill (2008).

- Linguagem compilada x Linguagem script: de forma resumida é possível diferenciar linguagens compiladas através dos passos necessários para que o programa torne-se executável. Em uma linguagem compilada os códigos fontes da aplicação são traduzidos em linguagens de máquina, desta maneira o programa pode ser executado, no entanto para alterar parte do código fonte a aplicação precisará ser novamente compilada para atualizar o executável gerado anteriormente. Nas linguagens interpretadas em *scripts*, como é o caso do JavaScript, o código fonte já está pronto para ser executado, tornando desnecessária a criação de arquivos estáticos;
- Tipagem dinâmica: conhecido como *Duck Typing* linguagens do tipo dinâmica são diferentes das linguagens estáticas. Nesse tipo de linguagem as variáveis podem assumir qualquer tipo ou objeto conforme o valor recebido na variável. Se a variável "conceito" receber um valor *String* automaticamente ela se comportará como uma variável do tipo *String* no código fonte da aplicação;
- Funções de ordem superior: são funções capazes de receberem uma ou mais funções como parâmetros ou argumentos. Podem ter uma função como saída. Dessa maneira é possível criar *function factories*, que nada mais são que funções que a partir de outras funções simples são capazes de realizar ações mais complexas;
- Programação *Client-side versus Server-side*: a linguagem originou com o formato *client-side*, ou seja, roda no computador cliente. Nesse formato o código-fonte é enviado ao computador do cliente e só lá é interpretado e executado, dependendo apenas da capacidade de processamento do cliente. Diferente de programas em linguagens *Server-side*, os quais são executados no lado do servidor, aumentando o tempo de execução das funcionalidades.
- Segurança: por ser executada no lado do cliente, o JavaScript precisa ter algumas restrições e boas práticas visando evitar composição de códigos maliciosos que possam causar danos ao usuário.

2.3 APLICAÇÃO WEB PROGRESSIVA

Aplicativos Nativos e Web são dois tipos principais de aplicações. Aplicativos nativos dependem de plataformas como Android, iOS, Windows e são criados usando linguagens de programação específicas, enquanto os aplicativos Web são plataformas em formato de sites e são executados por um navegador, geralmente escritos em HTML5. Os aplicativos nativos têm acesso total ao hardware do dispositivo, enquanto os aplicativos da Web ainda carecem de algumas limitações.

Nos últimos anos, o uso de aplicativos móveis tiveram um o crescimento em comparação à web móvel. Empresas e desenvolvedores muitas vezes viam a necessidade de criar aplicativos móveis nativos para superar as limitações que as plataformas Web impõem aos dispositivos móveis. Em contrapartida os aplicativos nativos precisam de mais recursos e tempo para desenvolver, manter e distribuir as soluções. Para a evolução da tecnologia, havia a necessidade de uma plataforma que pudesse combinar os recursos e as experiências dos aplicativos nativos com a Web.

PWA é o termo que combina os principais benefícios apresentados pelas duas frentes propostas. Foi introduzido pela primeira vez na conferência de desenvolvedores da Google em maio de 2016 em San Francisco pelo engenheiro do Google Chrome, Alex Russell. O termo foi utilizado para descrever a nova geração de aplicativos da web que são carregados em sites comuns, mas abordam a maioria dos recursos que faltavam nos aplicativos da Web. O combinação de uso do *Service Worker*, *shell* para aplicativos, *Web App Manifest* e notificação por push fizeram o PWA parecer e se comportar como um aplicativo nativo. Como o PWA é um aplicativo da web, ele pode ser usado em todas as plataformas, reduzindo o custo e os recursos e aumentando a acessibilidade (THAKUR, 2018).

Segundo Warcholinski (2018) Muitas empresas líderes adotaram o PWA e lançaram seus próprios aplicativos da Web com funcionalidades nativa, incluindo o AliExpress, que relatou um aumento de 104% nas taxas de conversão para novos usuários, e o Twitter, que observou aumento de 75% nos tweets e uma redução de 20% na taxa de rejeição.

2.3.1 Vantagens *versus* Desvantagens

Todas as principais vantagens e desvantagens citadas a seguir relativo ao *Progressive Web Apps* decorrem do fato do PWAs mesclar a

conveniência e o alcance da Web com a funcionalidade dos aplicativos móveis nativos (WARCHOLINSKI, 2018).

2.3.1.1 Vantagens

- Modo *offline*: Podem ser armazenados pela cache do navegador e acessados mesmo quando offline. Empresas de venda online, por exemplo, conseguem disponibilizar o acesso para navegação em seus produtos mesmo quando o usuário não está conectado na Internet, aumentando o engajamento do potencial usuário;
- Melhor desempenho: Devido ao serviço desempenhado do *Service Worker*, que será apresentado na subseção 2.3.2.1, o desempenho oferecido pelo PWA é superior aos aplicativos tradicionais;
- Instalação ou atualizações manuais não são necessárias: O usuário não precisa atualizar manualmente para novas versões ou correções de bugs. O benefício também atinge o fornecedor, tendo em vista que a atualização automática pode reduzir as solicitações de suporte ao cliente que ainda possuem uma versão desatualizada do aplicativo. Diversas atualizações para correções não impactam o usuário final;
- Redução no uso de dados móveis: Em países emergentes os dados móveis são mais caros do que nos países desenvolvidos. Os PWAs são muito menores que os aplicativos móveis e exigem muito menos largura de banda do que os aplicativos tradicionais da Web, porque podem aproveitar muito melhor o cache;
- Independem das lojas de aplicativos: Para empresas menores e desenvolvedores independentes que não desejam ter o custo recorrente do Google ou Apple a tecnologia é um ponto positivo. No entanto, para o lado usuário nem sempre é visto como uma vantagem, tendo em vista que ele confia mais em aplicações disponíveis em lojas virtuais do que através de PWAs.

2.3.1.2 Desvantagens

- Compatibilidade com *iOS*: após a versão 11.3 é possível executar PWAs em dispositivos Apple, no entanto em dispositivos mais antigos não há essa possibilidade. A Apple não permite o acesso

do PWAs em alguns recursos importantes como: *Touch ID*, *Face ID*, *Bluetooth*, *Beacons*, dentre outros;

- PWAs não podem fazer tudo: Eles não podem fazer tudo o que os aplicativos móveis podem fazer. Por serem escritos em JavaScript, eles não são tão eficientes em relação ao consumo de bateria como os aplicativos escritos em linguagens nativas, como *Swift*. O desempenho também não é tão efetivo como o desempenho de aplicativos nativos, isso ocorre pelo fato do JavaScript ser uma linguagem de programação de *thread* única.

2.3.2 Visão Geral da Arquitetura

2.3.2.1 *Service Worker*

Service Worker é um script executado em segundo plano no navegador. É um serviço do *JavaScript* executado separadamente da página da web, portanto, não pode acessar o DOM (Document Object Model), interface de programação para documentos HTML, diretamente. Dessa maneira, ele se comunica com a página da web por meio de uma interface. É através desse serviço que os recursos como notificações e funcionalidade offline são exercidos (FRANSSON; DRIAGUINE, 2017).

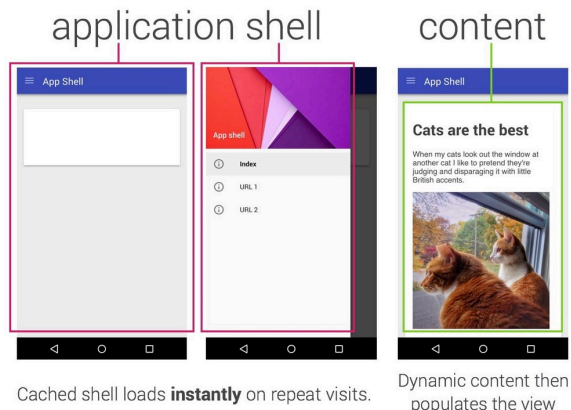
Segundo Biorn-Hansen e Gronli (2017) o *Service Worker* é responsável pela maioria das principais características associadas aos aplicativos da web progressivos. Um PWA não pode funcionar corretamente em navegadores sem o seu suporte. O *Worker* é registrado na primeira página de uma acesso do usuário. Pode ser usado para lidar com tarefas como sincronização em segundo plano, mecanismos de cache de dados e aplicativos, assim como interceptação de solicitações de rede.

2.3.2.2 *Application Shell*

O *shell* do aplicativo é definido pelo grupo Google como o mínimo de HTML, CSS e JavaScript que alimentam uma interface do usuário. Eles listam três critérios para o shell: tempo de carregamento rápido, cache e exibição de conteúdo dinâmico. Os dados são extraídos de APIs externas.

O shell do aplicativo pode ser considerado como um pacote de código que o desenvolvedor publicaria em uma loja de aplicativos se estivesse criando um aplicativo nativo. É a carga necessária para o desenvolvimento, mas pode não ser a história toda. Ele mantém a interface do usuário local e gera conteúdo dinamicamente por meio de uma API.

Figura 3 – Exemplo do funcionamento do *Application Shell*



Fonte: Google Developers - developers.google.com.

2.4 VUE

Vue.js se trata de um framework JavaScript progressivo para construção de interfaces de usuário e as famosas *Single Page Applications (SPA)*. O framework foi originalmente inventado por Evan You, um ex-funcionário da Google, que além de trabalhar nos projetos de Vue.js, também atuou no projeto MeteorJs, um outro framework Javascript, enquanto ainda atuava na Google. Evan ainda teve a oportunidade de atuar em projetos importantes de Angular, framework já consagrado na época, mas arriscou investir num projeto próprio, que seria uma espécie de aperfeiçoamento do mesmo.

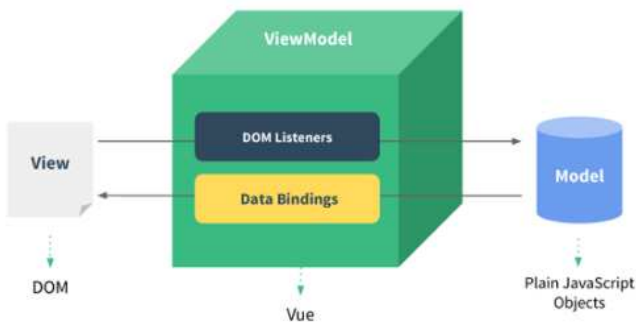
Desde então o Vue.js vem ganhando cada vez mais popularidade entre a comunidade de desenvolvedores. Dentre os motivos que contribuem para essa crescente, pode-se destacar o fato de ele ser muito mais leve e simples que outros frameworks mais populares, como o React e o Angular, além de ainda, ser uma espécie de combinação de

características desses frameworks.

2.4.1 Visão Geral da Arquitetura

O Vue.js conta com a capacidade de um *Data Binding Reativo*, ou seja, ele torna extremamente simples a sincronização entre os dados e o *Domain Object Model (DOM)*, como mostra a Figura 4.

Figura 4 – Vue - Visão geral



Fonte: <https://v1.vuejs.org/guide/overview.html>

Explicando melhor a ilustração, pode-se dizer que é inserida uma determinada variável no template HTML através de um sintaxe especial do Vue, que faz com que, depois do *data binding*, ela seja automaticamente atualizada a cada alteração do estado desse mesmo dado.

Uma segunda grande característica do Vue é seu sistema de componentes. Esse sistema é na verdade uma abstração que permite a construção de uma grande aplicação através da junção de pequenas partes, que chamamos de componentes. Esse conceito é bastante útil no desenvolvimento de um projeto, uma vez que torna os componentes reutilizáveis por toda a aplicação. Uma grande vantagem em se tratando de escalabilidade.

Além das características citadas, o Vue conta com uma série de outras peculiaridades, como as propriedades computadas, os templates, as diretivas, o roteamento, os watchers e muitas outras. Talvez elas não sejam exclusivas do framework, mas certamente ajudaram no processo de torná-lo um dos frameworks mais completos e populares do

momento.

Para melhor gerenciar essas propriedades, Callum Macrae (2018) explica que há diversas bibliotecas que funcionam em harmonia com Vue e são mantidas pelas mesmas pessoas que cuidam do próprio Vue. Para tratar do roteamento, por exemplo, existe o `vue-router`. Para o gerenciamento de estados e compartilhamento de dados entre componentes numa armazenagem global, existe o `vuex`. Já para testar componentes Vue, existe o `vue-test-utils`.

2.5 FIREBASE

Em 2014, Google completou a aquisição de uma empresa em São Francisco chamada Firebase Inc. Firebase contempla uma agrupamento de soluções de desenvolvimento destinado a acelerar integração de recursos em nuvem com aplicativos web e móveis. Após a compra a empresa realizou uma série de alterações combinando os serviços do *Google Cloud* (KHAWAS; SHAH, 2018).

Segundo Smyth (2017) Firebase é considerado uma plataforma de desenvolvimento web.

A plataforma, conhecida por ser um *Backend-as-a-Service (BaaS)*, oferece serviços como banco de dados em tempo real, armazenamento em nuvem, autenticação, relatórios de falhas, aprendizado de máquina, configuração remota e hospedagem de arquivos estáticos. Seu banco de dados não relacional (NoSQL), armazena os dados em formato *JavaScript Object Notation JSON* Smyth (2017).

Conforme Smyth (2017) após as dependências adicionadas ao aplicativo, os dados não estruturados podem ser adicionados ao banco de dados pelo seguinte código:

```
//Write a message to the Firebase Database
```

```
database= FirebaseDatabase.getInstance();
DatabaseReference myRef = database.getReference("node");
myRef.setValue ("Hello, World");
```

Arquivos como imagens, áudio, vídeo podem ser armazenados no aplicativo. Os dados são altamente protegidos e robustos, dando continuidade do último ponto em caso de erro de rede.

Aplicativos móveis geralmente precisam de serviços de *back-end* para trabalhar com a interface do usuário de *front-end*. Isso significa que é necessário desenvolvimento de um *back-end* e servidores para

trabalhar com aplicativos móveis. O Firebase é um serviço de nuvem para impulsionar *back-ends* dos aplicativos. Além de fornecer o suporte para armazenamento de dados e autenticação do usuário, após a integração do aplicativo móvel ao Firebase, não é preciso escrever código de back-end ou gerenciar a infraestrutura.

2.5.1 *Firestore User Authentication*

A maioria dos aplicativos precisam de informações de autenticação para identificação do perfil do usuário e garantia da proteção dos dados. Sem alguma maneira de identificar um usuário para outro, seria impossível para o aplicativo saber quais dados e configurações pertencem a uma pessoa. De modo geral as opções de autenticação são endereço de e-mail e senha, muitas vezes permitindo o uso de dados de plataformas terceiras como Facebook, Google ou Twitter (SMYTH, 2017).

Segundo Kumar (2018) apesar da necessidade de autenticação do usuário para aplicativos a função pode ser substancialmente mais complexa do que parece. A autenticação não deve ser exclusivamente prover segurança e confiabilidade. Deve-se levar em consideração a necessidade de alteração de dados por parte dos usuários, esquecimento de senhas e incorporação com API de terceiros. O banco de dados deve ser implementado e armazenado com segurança e permitir gerenciar as operações CRUD (*Create, Read, Update and Delete*).

O *Firestore Authentication* permite a criação do login com todos os requisitos necessários com poucas linhas de código. Uma vez que o usuário está autenticado no Firebase o usuário terá um *User ID* que pode ser usado para a inserção de dados e integração com outros serviços. A lista suportada pelo *Firestore Authentication* prevê autenticações com Facebook, Google, Twitter, GitHub, número de telefone, e-mail e senha.

Existem duas formas de autenticação no *Firestore* disponíveis, uma envolvendo o uso do *FirestoreUI Auth* e outra abordagem de mais baixo nível usando *SDK Firestore*. Na prática, envolvem o uso de algumas coleção de classes.

2.5.2 *Firestore Realtime Database*

O *Firestore Realtime Database* é um banco de dados hospedado na nuvem. Todos os dados são armazenados em um objeto JSON e sincronizados em tempo real com os usuários conectados, através do compartilhamento de uma única instância do *Realtime Database*.

Seus principais recursos são:

- Tempo real;
- Offline;

A figura 5 exemplifica um objeto JSON para armazenamento dos dados de um aplicativo de comércio eletrônico. Os produtos são um mapa do ID do produto para suas propriedades. Os clientes são um mapa do ID do cliente para suas propriedades. Os pedidos são um mapa do ID do pedido para suas propriedades.

Figura 5 – Exemplo de estrutura JSON.

```
{
  "products": {
    "00001": {
      "name": "iPhone 6s plus",
      "price": 613.50
    },
    "00002": {
      "name": "LG Nexus 5",
      "price": 187.99
    }
  },
  "customers": {
    "00001": {
      "firstName": "Alex",
      "lastName": "Cheng",
      "email": "alex@example.com"
    }
  },
  "orders": {
    "00001": {
      "customer": "00001",
      "items": {
        "00001": true,
        "00002": true
      },
      "shippingCost": 5.00
    }
  }
}
```

Fonte: Cheng (2018).

Conforme exemplo apresentado por Cheng (2018) cada valor de dados no banco possui um caminho exclusivo para identificá-lo. O objeto JSON pode ser comparado a um diretório de arquivos, o caminho descreve como navegar da raiz para um valor de dados. Por exemplo, o caminho do preço do produto com o ID 00001 é `/products/00001/price`. O caminho `/pedidos` representa todos os pedidos. O caminho é usado ao ler ou gravar dados, além de configurar regras de segurança.

instância da classe `firebase.database.Database`:

```
let database = firebase.database();
```

Após a utilizar o objeto `database()` é possível usar o método de referência para buscar os dados de um caminho específico. O resultado será um objeto `firebase.database.Reference`.

2.5.2.3 Escrever Dados

O objeto de referência também pode ser usado para escrever dados no banco. O método `set()` é utilizado para substituir todos os dados do *path*, utilizando parâmetro na função, é possível selecionar os dados específicos que devem ser alterados, dessa maneira os objetos são sincronizados de maneira unitária com o servidor.

Figura 8 – Escrevendo no Banco Firebase

```
let ref = database.ref('products');

ref.child('00001').set({
  "name": "New iPhone 6s plus",
  "price": 699.99
});

ref.child('00001').update({
  "price": 639.99
});
```

Fonte: Cheng (2018)

2.6 ENCERRAMENTO

O conjunto de conceitos e tecnologias utilizadas é capaz de suprir as principais necessidades da proposta da aplicação. Temos desde dados relevantes que nos fornecem um contexto sobre o universo em que a aplicação se engloba, como é o caso do Marketing Relacional, até

bibliotecas, linguagens e ferramentas de desenvolvimento que melhor se adequam às necessidades e características da aplicação, como é o caso do Vue.js e o Firebase, que tratam de toda a parte backend e frontend da aplicação.

3 SOLUÇÕES EXISTENTES

Neste capítulo serão apresentadas aplicações existentes com objetivo similar ao do proposto neste projeto.

3.1 ZYGO FERRAMENTA DE FIDELIZAÇÃO

A Zygo é uma empresa de tecnologia que tem como objetivo ajudar bares e restaurantes a fidelizar os clientes (ZYGO, 2020). Criado por ex-donos de restaurantes, a Zygo tem como propósito garantir uma forma de trazer clientes que geram retornos crescentes sobre o investimento em marketing, através de uma consultoria especializada e uma tecnologia de criação de campanhas e clube de benefícios.

O produto da empresa está associado ao uso de tecnologia para promover restaurantes através de relacionamentos especiais com os clientes utilizando os conceitos do marketing relacional. A aplicação é *White Label*, conceito nascido da ideia de oferecer facilidades a marcas e clientes, terceirizando a estrutura de produtos e serviços através de modelos prontos. Este modelo é frequentemente utilizado por segmentos de empresas que desejam investir em serviços de tecnologia sem precisar investir financeiramente no desenvolvimento próprio.

Neste contexto, a aplicação não permite a integração entre os pontos de fidelidade. A cada novo plano de fidelidade com bares ou restaurantes da região, mesmo o usuário já havendo cadastro em outro estabelecimento que possui o sistema de fidelidade do fornecedor Zygo, o usuário precisará realizar um novo procedimento de cadastro na plataforma.

Figura 9 – Portal Fidelidade



Fonte: Portal de fidelidade restaurante utilizando ferramenta Zygo.

3.2 DOTZ

O Dotz é um programa de benefícios presente em diferentes empresas de diversos ramos, como supermercados, bancos, farmácias, livrarias e até postos de gasolina. A aplicação está disponível tanto através da plataforma *web* como por aplicativos nativos disponibilizados em suas respectivas lojas virtuais. A ideia central da aplicação é a moeda digital dotz. A cada compra feita em estabelecimentos parceiros o usuário acumula Dotz para trocar por produtos ou serviços disponibilizados na própria plataforma. Os pontos acumulados em cada compra variam conforme a taxa de proporção definida pelo próprio estabelecimento, podendo ser 1 dotz a cada 1 real, como 1 dotz a cada 10 reais.

A identificação do usuário na aplicação ocorre através da inserção do seu número de CPF ou da apresentação do seu cartão dotz de fidelidade. Ao acumular o número mínimo de pontos necessários o usuário consegue trocar determinada pontuação em dotz por prêmios disponibilizados ou desconto em compras.

A pontuação acumulada em dotz é única e independente do estabelecimento em que foi realizada a compra, assim como as premiações disponibilizadas fazem parte do catálogo dotz oferecido pela aplicação, e não apenas pelo que o estabelecimento parceiro oferece.

Figura 10 – Site oficial Clube Dotz

The screenshot shows the 'Ofertas Relâmpago Dotz' section of the website. The header includes the 'dotz' logo, navigation links (SOBRE DOTZ, GANHE, TROQUE, CLUBE DOTZ), a search bar, a 'Fale conosco' icon, and 'Entrar' and 'Cadastrar' buttons. The main content area features four product offers, each with a countdown timer and a discount percentage:

Produto	Desconto	Preço Original (DZ)	Preço com Dotz (DZ)
Carregador Multilaser Wireless Concept Past sem fio para	46%	22.004	11.679
Conjunto 03 Necessary Diamante Feminina - Jacki Design	61%	12.435	4.807
Conjunto de facas com 4 peças cabo marrom	45%	9.681	5.292
Panela Pipoqueira 3,4L em Alumínio - MSR	45%	9.055	4.979

Fonte: dotz.com.br

3.3 ENCERRAMENTO

Após o levantamento de algumas das ferramentas que atualmente são utilizadas no mercado foi possível observar que existem oportunidades de melhorias com diferenciais significativos. A principal motivação e diferencial mapeado no projeto está associado a unificação dos cartões de fidelidade em uma única aplicação.

Diferente do Dotz, que possui uma pontuação única e permite aos usuários o resgate de cupons em estabelecimentos distintos ao que foi pontuado, e também diferente do Zygo, onde a pontuação é única por estabelecimento mas os cartões fidelidade são disponibilizados em aplicações distintas, sem integração, para a aplicação proposta os pontos serão gerados de forma unitária, por estabelecimento, mas possibilitando ao usuário cliente acompanhar de forma unificada todos os pontos gerados em diferentes estabelecimentos, sem que seja possível utilizar os pontos em estabelecimentos diferentes dos que originaram a pontuação, o que reforça junto aos estabelecimentos parceiros o objetivo da estratégia de Marketing Relacional: reter clientes.

Outro ponto em destaque está aliado a facilidade relacionada à utilização da aplicação, sejam pelas características da tecnologia PWA ou também pelas funcionalidades da aplicação, como por exemplo o cadastro rápido e fácil com apenas os dados de e-mail e CPF ou os benefícios associados ao simulador disponibilizado, o qual permite auxiliar os estabelecimentos no desenvolvimento de programas de fidelidade mais assertivos.

4 SOLUÇÃO PROPOSTA

Para auxiliar os clientes na gestão de seus pontos de fidelidade assim como impulsionar os estabelecimentos parceiros com a utilização de estratégias bem sucedidas, será projetada e desenvolvida uma aplicação *web* que implemente os conceitos resultantes da fundamentação apresentada.

A aplicação deve ser desenvolvida utilizando-se da metodologia de uma aplicação *web* progressiva, o *PWA*, usufruindo assim de seus principais benefícios tais como velocidade, responsividade, confiabilidade e interação envolvente.

Através de sua utilização será possível pontuar a carteira de um cliente, cadastrar um cupom, simular e disponibilizar programas de fidelidade, gerenciar pontos obtidos como cliente e quando possível, resgatar cupons.

A análise da aplicação será feita com base na fundamentação teórica através de uma lista de requisitos e os diagramas UML de classes e casos de uso. Seu desenvolvimento será em duas etapas: regras de negócio e interface gráfica.

O cronograma do projeto foi desenvolvido visando a liberação de um MVP, produto mínimo viável, e as atividades foram divididas entre *sprints* com o objetivo de validar a ideia do serviço com usuários de aplicativos, bares e restaurantes. Após finalizada as etapas essenciais serão executados testes que validam e avaliam a aplicação desenvolvida.

4.1 ESPECIFICAÇÕES

Nesta seção serão descritos a análise e o projeto da aplicação, com a apresentação das funcionalidades implementadas e os recursos.

4.1.1 Regras de Negócio

Nesta subseção serão apresentadas as regras de negócio do projeto.

4.1.1.1 Usuários

- RN001 - Existência de usuários: serão considerados usuários todos que possuem acesso ao aplicativo através do e-mail e senha cadastrados;
- RN002 - Consistência de usuários: só será permitido o cadastro de dois usuários com o mesmo CPF e e-mail caso eles sejam de diferentes tipos (cliente e estabelecimento);
- RN003 - Acesso ao sistema: Somente usuários cadastrados poderão acessar as funcionalidades da aplicação;
- RN004 - Editar perfil: O usuário poderá realizar alterações em seu cadastro, exceto no campo CPF;
- RN005 - Cadastro de clientes: O cadastro de um novo cliente só poderá ser realizado por um administrador de estabelecimento e deve obrigatoriamente informar um e-mail e CPF válidos, sendo o CPF a sua primeira senha de acesso;
- RN006 - Carteira do cliente: A carteira do cliente deve armazenar todas as pontuações acumuladas em cada restaurante bem como apresentar os cupons que estejam disponíveis para resgate ou utilização;
- RN007 - Resgate de cupom: Ao resgatar um cupom, caso o cliente tenha o número mínimo de pontos necessários o código do cupom será gerado e a carteira do cliente terá a pontuação descontada.

4.1.1.2 Estabelecimento

- RN008 - Cadastro de estabelecimentos: O cadastro de um novo estabelecimento só poderá ser realizado por um administrador da aplicação;
- RN009 - Cadastro de administradores de Estabelecimento: O cadastro de um administrador de estabelecimento só poderá ser realizado por um administrador da aplicação;
- RN010 - Estabelecimento disponível: Para que um estabelecimento fique disponível aos clientes ele deve possuir no mínimo um administrador, um programa de fidelidade e um cupom ativos.

4.1.1.3 Estabelecimentos - Programas de Fidelidade

- RN011 - Programa de fidelidade: Para que um programa de fidelidade esteja ativo e válido é necessário possuir no mínimo uma taxa de pontuação cadastrada.

4.1.1.4 Estabelecimentos - Produtos

- RN012 - Cadastro de produto: Para cadastrar um novo produto deve ser informado no mínimo sua descrição;
- RN013 – Imagens: as imagens submetidas na aplicação deverão estar em um dos formatos jpg, jpeg, png ou bmp.

4.1.1.5 Estabelecimentos - Cupons

- RN014 - Cadastro de cupom: Para que seja possível cadastrar um cupom ele deve ser vinculado a um produto, informada sua pontuação de resgate e definido se o cupom trata-se de um prato principal, acompanhamento ou bebida;
- RN015 - Validação de cupom: Para que um cupom seja utilizado é obrigatório a validação do CPF responsável e que o código do cupom esteja com a situação disponível. Caso positivo, o código é removido da carteira do cliente e o cupom volta a ficar disponível para resgate.

4.1.1.6 Estabelecimentos - Pontuar Clientes

- RN016 - Consistência Pontuação cliente: Para que um cliente pontue é obrigatório informar um CPF cadastrado e que haja um programa de fidelidade vigente para a data do registro;
- RN017 - Registro de pontuação: Ao registrar a pontuação de um cliente o mesmo recebe o acréscimo dos pontos em sua carteira para o restaurante em questão. Os pontos obtidos só poderão ser utilizados no estabelecimento responsável pela origem da pontuação.

4.1.2 Requisitos Funcionais

Nos tópicos a seguir serão apresentados os requisitos funcionais elicitados para a aplicação. Requisitos classificados com prioridade essencial devem estar na versão inicial do projeto.

- RF001 – Cadastrar estabelecimento.
Descrição: O usuário administrador da aplicação poderá cadastrar um novo estabelecimento.
Entrada e pré-condições: Nome, Endereço, Categoria, Logo.
Saídas e pós-condições: Um novo estabelecimento é cadastrado na aplicação.
Prioridade: Essencial.
- RF002 – Cadastrar conta de estabelecimento.
Descrição: O usuário administrador da aplicação poderá cadastrar um novo administrador de estabelecimento.
Entrada e pré-condições: Nome, CPF, Estabelecimento, E-mail, Senha.
Saídas e pós-condições: Uma conta de usuário administrador de estabelecimento é criada na aplicação.
Prioridade: Essencial.
- RF003 – Cadastrar conta de cliente.
Descrição: O usuário estabelecimento poderá cadastrar um novo cliente.
Entrada e pré-condições: CPF, E-mail.
Saídas e pós-condições: Uma conta de usuário cliente é criada na aplicação.
Prioridade: Essencial.
- RF004 – Acessar aplicação.
Descrição: A aplicação deve permitir o acesso a usuários através de e-mail e senha previamente cadastrados, sejam eles clientes ou administradores de estabelecimentos.
Entrada e pré-condições: E-mail, senha.
Saídas e pós-condições: O usuário tem acesso a página inicial da aplicação com acesso a funcionalidades restritas.
Prioridade: Essencial.
- RF005 – Recuperar senha.
Descrição: A aplicação deve permitir que o usuário recupere sua senha em caso de esquecimento.

Entrada e pré-condições: CPF.

Saídas e pós-condições: O usuário receberá um e-mail com link para redefinição de senha.

Prioridade: Importante.

- RF006 – Redefinir senha.

Descrição: A aplicação deve permitir que o usuário redefina sua senha ao acessar o link de recuperação recebido em seu e-mail.

Entrada e pré-condições: O usuário receberá um e-mail com link para redefinição de senha, Nova senha, Confirmação de nova senha, CPF.

Saídas e pós-condições: O usuário tem sua senha redefinida.

Prioridade: Importante.

- RF007 – Cadastrar programa de Fidelidade.

Descrição: O usuário estabelecimento poderá cadastrar um ou mais programas de fidelidade conforme o dia da semana, turno e clientes conhecidos ou novos.

Entrada e pré-condições: Taxa de pontuação, Pontuação 1º cliente, Dia da semana, Hora início, Hora fim.

Saídas e pós-condições: O estabelecimento disponibiliza o programa de fidelidade aos clientes.

Prioridade: Essencial.

- RF008 – Editar programa de Fidelidade.

Descrição: O usuário estabelecimento poderá alterar o programa de fidelidade a qualquer momento.

Entrada e pré-condições: Programa de fidelidade a ser alterado, Campos a serem alterados.

Saídas e pós-condições: O programa de fidelidade é atualizado.

Prioridade: Essencial.

- RF009 – Cadastrar produto.

Descrição: O usuário estabelecimento poderá cadastrar produtos a serem utilizados nos cupons e no simulador.

Entrada e pré-condições: Descrição, Foto (opcional), Valor Custo (opcional), Valor Venda (opcional).

Saídas e pós-condições: O produto é disponibilizado para seleção nas telas cadastro de cupom e simulador de cupom.

Prioridade: Importante.

- RF010 – Editar produto.

Descrição: O usuário estabelecimento poderá alterar e inativar o

produto.

Entrada e pré-condições: Produto a ser alterado, Campos a serem alterados.

Saídas e pós-condições: O produto e cupons não resgatados vinculados a ele são atualizados.

Prioridade: Importante.

- RF011 – Cadastrar cupom.
 Descrição: O usuário administrador de estabelecimento poderá cadastrar um ou mais cupons de produtos.
 Entrada e pré-condições: Produto, Pontuação de resgate.
 Saídas e pós-condições: O estabelecimento disponibiliza o cupom aos clientes.
 Prioridade: Essencial.

- RF012 – Editar cupom.
 Descrição: O usuário administrador um estabelecimento poderá alterar e inativar o cupom.
 Entrada e pré-condições: Cupom a ser alterado, Campos a serem alterados.
 Saídas e pós-condições: O cupom é atualizado.
 Prioridade: Essencial.

- RF013 – Pontuar cliente.
 Descrição: O usuário administrador de estabelecimento poderá acrescentar pontos à carteira de um cliente de acordo com o programa de fidelidade vigente.
 Entrada e pré-condições: CPF cliente, Valor compra, Programa de fidelidade vigente.
 Saídas e pós-condições: A carteira do cliente com o restaurante é acrescida de pontos. A pontuação é registrada no histórico de compras do cliente.
 Prioridade: Essencial.

- RF014 – Resgatar cupom.
 Descrição: O usuário cliente poderá resgatar um cupom disponibilizado pelo restaurante após acumular o número mínimo de pontos necessários em sua carteira.
 Entrada e pré-condições: Pontuação mínima.
 Saídas e pós-condições: A carteira do cliente com o restaurante tem os pontos do cupom reduzidos. A carteira do cliente com o restaurante é acrescida de pontos. A pontuação é registrada no

histórico de compras do cliente. O cupom do cliente é gerado.
Prioridade: Essencial.

- RF015 – Validar cupom.
Descrição: O usuário administrador de estabelecimento poderá validar o cupom resgatado por um cliente.
Entrada e pré-condições: Cupom válido, CPF cliente.
Saídas e pós-condições: O cupom do cliente é inserido em seu histórico de cupons utilizados. O cupom volta a ficar disponível para resgate do cliente.
Prioridade: Essencial.
- RF016 – Simular programa de fidelidade.
Descrição: O usuário administrador de estabelecimento poderá simular diferentes cenários de vendas para calcular o percentual sugerido do programa de fidelidade cadastrado.
Entrada e pré-condições: Produtos, Total de venda, Valor custo, Valor venda, Quantidade média de venda.
Saídas e pós-condições: Pontuação sugerida de *cashback* no programa.
Prioridade: Essencial.
- RF017 – Simular cupom.
Descrição: O usuário administrador de estabelecimento poderá simular diferentes cenários de vendas para calcular a pontuação sugerida ao cupom.
Entrada e pré-condições: Produto, Valor custo, Valor venda, Quantidade média de venda.
Saídas e pós-condições: Pontuação sugerida para troca do produto.
Prioridade: Essencial.
- RF018 – Atualizar perfil.
Descrição: O usuário poderá atualizar os seus dados cadastrais incluindo a senha de acesso.
Entrada e pré-condições: Dados a serem alterados, Usuário a ser alterado.
Saídas e pós-condições: O usuário tem seus dados atualizados.
Prioridade: Essencial.

4.1.3 Diagrama de Classes

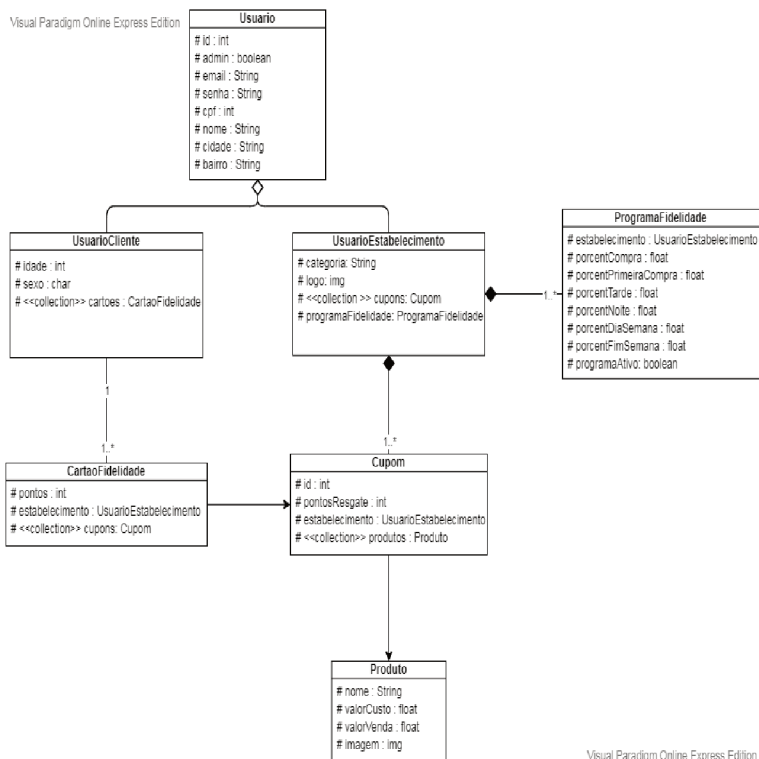
Na Figura 11 é apresentada a versão do diagramas de classes proposto neste trabalho, onde as classes `UsuarioEstabelecimento` e `UsuarioCliente` são agregados da classe `Usuário`, diferenciadas por seus atributos que servirão para interfaces distintas na aplicação desenvolvida.

Os usuários Clientes possuem uma coleção de um ou mais cartões fidelidade, obtidos durante a pontuação em um estabelecimento. Caso seja a primeira compra do cliente no estabelecimento o cartão é adicionado a sua carteira seguido da pontuação relativa ao programa de fidelidade vigente no ato da compra. Conforme regra de negócio RN005 todos os clientes são cadastros através de um estabelecimento, desta forma, toda classe `UsuarioCliente` deve estar relacionada a no mínimo um cartão fidelidade.

Após acumular determinada quantidade de pontos o cliente pode resgatar um cupom disponibilizado pelo restaurante, desde que sua pontuação disponível seja maior ou igual ao mínimo necessário para o seu resgate. Os cupons gerados pelo estabelecimento para obtenção de benefícios estão associados aos produtos previamente cadastrados. Ao liberar um novo cupom de troca o estabelecimento precisa informar obrigatoriamente a pontuação de resgate do produto cadastrado.

Todos os cupons disponíveis são cadastrados pelo `UsuarioEstabelecimento`, assim como os programas de fidelidade vigentes. É obrigatório que um estabelecimento esteja associado a no mínimo um programa de fidelidade e um cupom, tendo em vista tratar-se da essência da aplicação. Os programas de fidelidade são cadastrados por uma taxa, representada em percentual, aplicada sobre o valor da compra efetuada. Caso seja de interesse do estabelecimento é possível definir diferentes taxas para clientes recorrentes e novos clientes, dias de semanas e fins de semana, e por fim, compras no período diurno e compras no período noturno.

Figura 11 – Diagrama de Classes



Fonte: elaborado pelos autores

4.1.4 Casos de Uso

A Figura 12 reproduz os casos de uso associados ao projeto. Três atores foram identificados no desenvolvimento da solução. As atividades serão desdobradas em funcionalidades do aplicativo. O Ator Administrador é responsável por cadastrar o estabelecimento e inserir as informações requisitos para a liberação do associado. Após a liberação do estabelecimento o Administrador tem a função de atribuir um usuário responsável do estabelecimento para que seja realizado os cadastros específicos dos produtos e promoções.

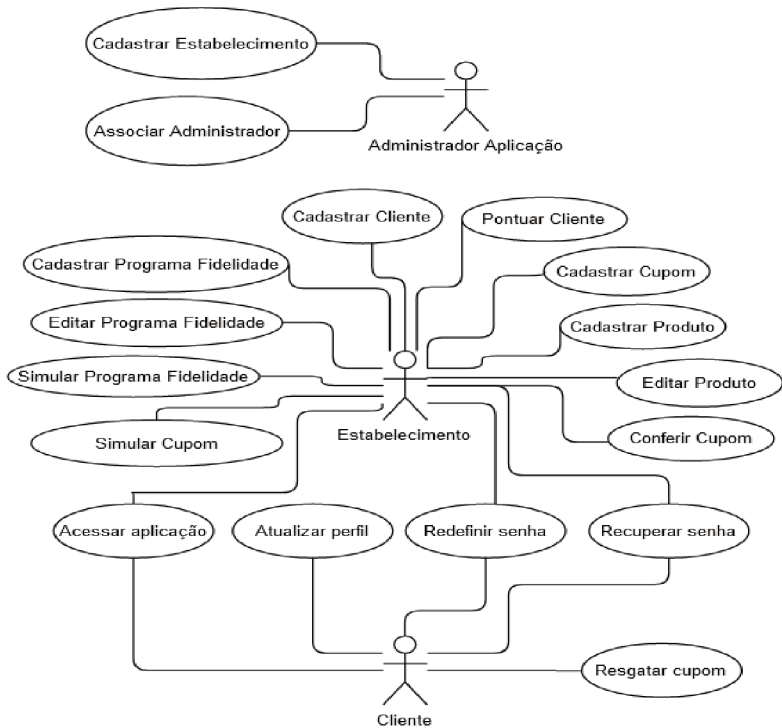
O Ator Estabelecimento, usuário responsável pela unidade de negócio, fica encarregado pelo *setup* básico do estabelecimento como: ca-

dastrar cupons, cadastrar e editar produtos, cadastrar, simular e editar programas de fidelidade. Algumas das atividades associadas ao Estabelecimento possuem interface com o Ator cliente, dentre elas: Pontuar, utilizada no momento do pagamento da compra do usuário. Resgatar Cupom: relacionado ao momento que ao Ator Usuário contém pontos suficientes para utilizar em uma promoção vigente no estabelecimento.

Os programas de fidelidade cadastrados pelo Ator Estabelecimento serão os responsáveis pelos benefícios recebidos de *cashback* pelo Ator Cliente no momento da pontuação. O programa de fidelidade pode ser editado a qualquer momento. O Ator estabelecimento pode criar um ou muitos programas de fidelidade parametrizando o benefício por dia da semana, por exemplo, conforme sua estratégia de marketing associado a campanha executada. No momento da simulação do programa de fidelidade o Ator Estabelecimento tem a possibilidade de utilizar seus produtos previamente cadastrados para obter uma pontuação sugerida no programa proposto com base na média do total de vendas do produto.

O Ator Usuário pode realizar outras funcionalidades como, por exemplo, atualizar seu cadastro, redefinir e recuperar sua senha assim como acessar a aplicação para ver sua carteira em estabelecimentos onde seu CPF já foi utilizado. Algumas funcionalidades como, recuperar senha, atualizar perfil estão também disponíveis para o Ator Estabelecimento, tendo em vista que o ator é um objeto do tipo usuário, conforme apresentado na subseção 4.1.3.

Figura 12 – Casos de Uso



Fonte: elaborado pelos autores

4.1.5 Persistência de Dados e Integração

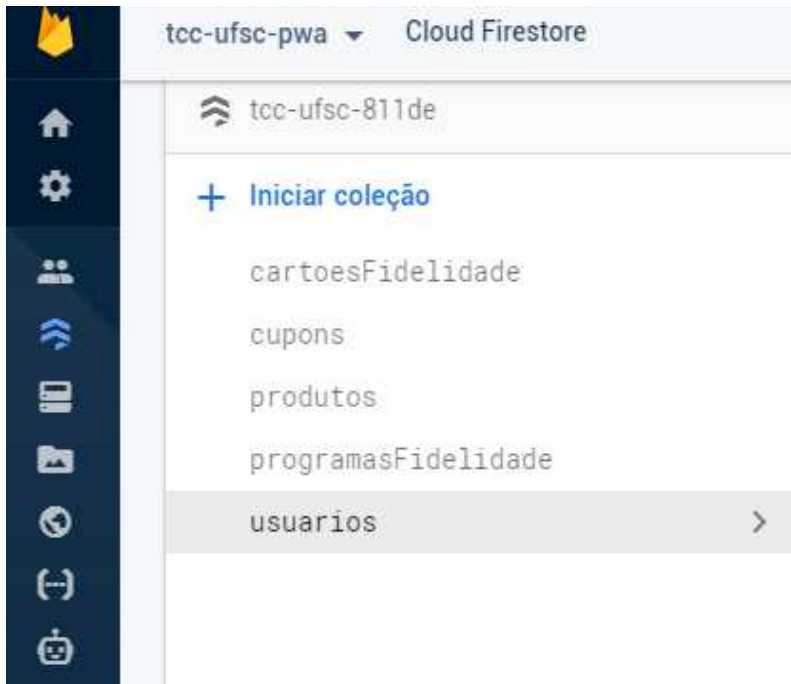
Toda a estrutura da base de dados foi desenvolvida através da plataforma da Google denominada *Firebase*. Nela, há uma parte específica para tratar o *database*, denominada de *Firestore*.

O *Cloud Firestore* é um banco de dados NoSQL orientado a documentos. Ao contrário de um banco de dados relacional, não há tabelas nem linhas registradas. Sua estrutura é composta por dados armazenados em documentos e organizados em coleções. Cada documento pode conter subcoleções e objetos aninhados, podendo ser objetos simples, como strings, ou objetos complexos, como listas.

Os dados da aplicação estão organizados seguindo a modelagem

do diagrama de classes, no entanto, com as adaptações ao banco de dados orientado a documentos. Nas Figuras 13 e 14 é possível visualizar uma representação gráfica para a estrutura dos objetos e sua representação na interface *Firebase*.

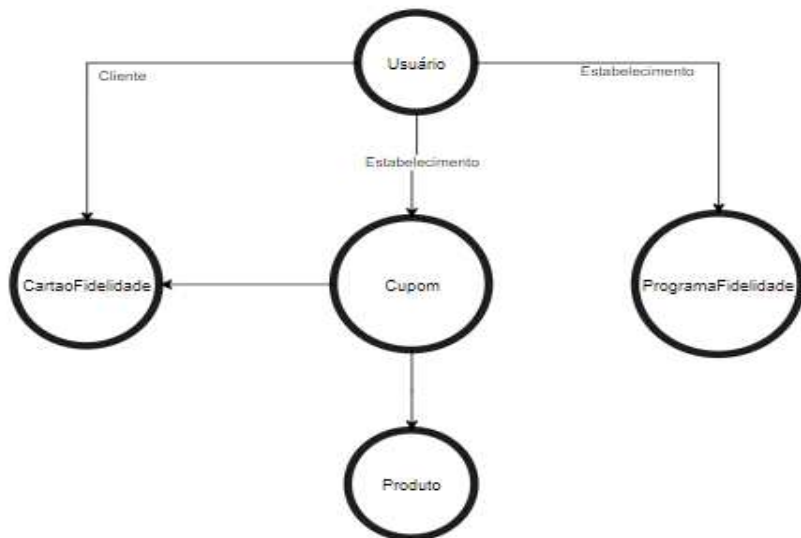
Figura 13 – Base de Dados na interface Firebase.



Fonte: elaborado pelos autores

O processo de integração da base de dados com a aplicação foi realizada seguindo a documentação fornecida pela própria Google. Como passo inicial é necessário criar uma conta no site da biblioteca. A conta gratuita possui limitação máxima de armazenagem e requisições conforme a Figura 15.

Figura 14 – Representação estrutura base de dados.



Fonte: elaborado pelos autores

Figura 15 – Preços Firebase

Autenticação		
Phone Auth - EUA, Canadá e Índia	10 mil/mês	US\$ 0,01/verificação
Phone Auth - todos os outros países	10 mil/mês	US\$ 0,06/verificação
Outros serviços de autenticação	✓	✓
Cloud Firestore		
Dados armazenados	1 GiB no total	US\$ 0,18/GiB
Saída de rede	10 GiB/mês	Google Cloud pricing
Gravações de documento	20 mil/dia	US\$ 0,18/100 mil
Leituras de documento	50 mil/dia	US\$ 0,06/100 mil
Exclusões de documento	20 mil/dia	US\$ 0,02/100 mil

Fonte: <https://firebase.google.com/pricing>

Para este projeto optou-se pela utilização da conta gratuita, considerando o limite disponível como suficiente para utilização da aplicação. Caso haja interesse em escalonar o projeto é necessário a criação de um plano de conta personalizada.

Além dos serviços da base de dados, o *Cloud Firestore*, foi utilizado também um módulo específico para facilitar a implementação do processo de autenticação fornecido pelo Firebase, o *Firebase Authentication*.

Depois de criada a conta, é necessário iniciar um novo projeto através da interface *Web*. Ao finalizar o processo, o serviço fornece uma série de parâmetros, como a chave da API, identificadores e endereço do banco de dados. Esses parâmetros são posteriormente utilizados no código fonte para configurar a conexão entre a aplicação e a API, tornando desta forma possível realizar as requisições de leitura e escrita na base de dados, bem como utilizar o processo de autenticação fornecido pela aplicação e alimentar a API com dados para coleta de informação através do *Analytics*, conforme a Figura 16.

Figura 16 – Configuração Inicial Firebase

```
31
32 // Your web app's Firebase configuration
33 const firebaseConfig = {
34   apiKey: "[REDACTED]",
35   authDomain: "tcc-ufsc-6954c.firebaseio.com",
36   databaseURL: "https://tcc-ufsc-6954c.firebaseio.com",
37   projectId: "tcc-ufsc-6954c",
38   storageBucket: "tcc-ufsc-6954c.appspot.com",
39   messagingSenderId: "[REDACTED]",
40   appId: "[REDACTED]",
41   measurementId: "[REDACTED]"
42 };
43 // Initialize Firebase
44 firebase.initializeApp(firebaseConfig);
45 firebase.analytics();
```

Fonte: elaborado pelos autores

4.2 DESENVOLVIMENTO

Após a instalação do Vue.js, através do comando “vue create nomeProjeto”, um rascunho de aplicação foi criado contendo toda a

estrutura base da aplicação, como alguns recursos dos *websites* (*assets*), componentes, *routers* e *views*. Após a criação da base de dados, seguindo o conceito de desenvolvimento orientado a componentes, ampliou-se o projeto em módulos, ou seja, componente a componente, fazendo com que a cada etapa de módulo, uma funcionalidade específica fosse concluída, como exemplo: para a concepção do módulo de registro de usuário foi criado e implementado toda a parte de cadastro, bem como a conclusão da conexão com a base de dados e seus respectivos registros e a instalação das bibliotecas necessárias para o funcionamento do mesmo. Concluída essa etapa, avançamos para a etapa de controle de acesso (*login* e *logout*) e posteriormente todas as outras restantes. A Figura 17 ilustra um trecho de código responsável pelo *login*.

Figura 17 – Trecho de código do método login

```
    },
  },
  methods: {
    login() {
      firebase.auth().signInWithEmailAndPassword(this.email, this.password).then(data => {
        firebase.firestore().collection('users').doc(data.user.uid).get().then((user => {
          user.data().admin ? router.push('home-admin') : router.push('home')
        })
      ), reject => {
        this.$root.$emit('changeToast', { message: reject.message, type: 'error' })
      })
    }
  }
}
```

Fonte: elaborado pelos autores

4.2.1 Tornando a aplicação uma PWA

A aplicação WEB tem seus primeiros passos de configuração para PWA nas etapas iniciais. Ao criar o projeto em Vue.js, seleciona-se a opção de suporte para PWA fornecido pelo próprio instalador do Node. Ao selecionar essa opção, o projeto é pré-configurado com os arquivos e configurações essenciais para suporte a essa tecnologia, como é o caso do *Service Worker*, que pode ser definido como um *script* que os navegadores executam em segundo plano que possibilitam recursos que não precisam de uma página Web ou de interação do usuário, ou ainda, permitir o uso da aplicação em modo offline.

Finalizada a etapa inicial é necessário criar o arquivo *manifest.json*, responsável por disponibilizar a aplicação de modo que ela seja instalável na tela inicial de um dispositivo sem o uso das lojas de aplicativos, como Google Play e Apple Store por exemplo. O arquivo fornece informações sobre a aplicação WEB em formato JSON, como seu nome, nome abreviado, ícone, versão, descrição, entre outros. Esse arquivo foi adicionado ao *head* da página inicial, disponibilizando então suas informações à aplicação, que por sua vez, terá esses dados compartilhados com o dispositivo ao realizar a instalação. Em outras palavras, ao instalar a aplicação na tela inicial de um dispositivo, além dela ficar disponível até mesmo offline, esta já deve possuir todas as informações pré-definidas, como o ícone da aplicação e o seu nome, o que não acontece num link de tela inicial ou bookmark, por exemplo.

Na Figura 18 é possível visualizar um trecho de código do arquivo em questão, onde foram definidas algumas informações relevantes para a etapa de adequação PWA, seguidos da sua inclusão no head da página inicial, etapa fundamental para a integração dessas informações com a aplicação.

Figura 18 – Trecho de código do arquivo manifest.json

```

{
  "name": "Food wallet",
  "short_name": "FW",
  "theme_color": "#408A87",
  "icons": [
    {
      "src": "https://tcc-ufsc-6954c.web.app/img/icons/android-chrome-192x192.png",
      "sizes": "192x192",
      "type": "image/png"
    },
    {
      "src": "https://tcc-ufsc-6954c.web.app/img/icons/android-chrome-512x512.png",
      "sizes": "512x512",
      "type": "image/png"
    },
    {
      "src": "https://tcc-ufsc-6954c.web.app/img/icons/android-chrome-maskable-192x192.png",
      "sizes": "192x192",
      "type": "image/png",
      "purpose": "maskable"
    },
    {
      "src": "https://tcc-ufsc-6954c.web.app/img/icons/android-chrome-maskable-512x512.png",
      "sizes": "512x512",
      "type": "image/png",
      "purpose": "maskable"
    }
  ],
  "start_url": "https://tcc-ufsc-6954c.web.app/",
  "display": "standalone",
  "background_color": "#000000",
  "url": "https://tcc-ufsc-6954c.web.app/",
  "lang": "",
  "screenshots": [],
  "description": "Aplicativo pontos de fidelidade unificado."
}

```

Fonte: elaborado pelos autores

Figura 19 – Trecho de código do arquivo index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,initial-scale=1.0">
    <link rel="icon" href="<%= BASE_URL %>favicon.ico">
    <link rel="manifest" href="/manifest.json">
    <title><%= htmlWebpackPlugin.options.title %></title>
```

Fonte: elaborado pelos autores

5 RESULTADOS OBTIDOS

Após o desenvolvimento das primeiras funcionalidades essenciais da aplicação, a ferramenta foi disponibilizada através de seu endereço *Web* para que assim os usuários tivessem acesso a sua utilização. Estabelecimentos fictícios foram criados com intuito de simular os requisitos essenciais do projeto.

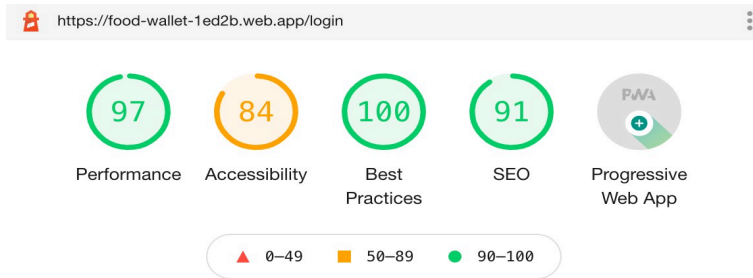
Sua elaboração final resultou em uma aplicação capaz de gerenciar programas de fidelidade para estabelecimentos comerciais, bem como o controle e utilização dos pontos obtidos por usuários clientes em todos os estabelecimentos participantes, de forma unificada, através de sua *Food Wallet*.

A aplicação foi dividida em dois modos de uso, o modo estabelecimento/administrador, que dá acesso às configurações e gerenciamento de um estabelecimento em questão, e o modo usuário, que é a visão do cliente do estabelecimento onde é possível controlar seus pontos e cupons.

O *deploy* da aplicação foi realizado usando os serviços de hospedagem e domínio disponibilizados pela própria conta Firebase. O acesso a aplicação, foi disponibilizado para teste através do endereço: <https://food-wallet-1ed2b.web.app>. O acesso ao modo estabelecimento é possível utilizando o login de acesso admin@admin.com e senha admin123, enquanto para o modo cliente o acesso está disponível através do login user@user.com e senha user123.

Para fins de métrica de qualidade da aplicação desenvolvida, foi utilizado a ferramenta da Google denominada de Lighthouse, que já é disponibilizada por padrão na aba de desenvolvedor do próprio Google Chrome. Após geração do relatório, foi possível constatar que, de um modo geral, a aplicação apresenta bons índices de performance, acessibilidade, boas práticas, SEO e PWA, como mostra a figura 20.

Figura 20 – Métricas de Qualidade.



Fonte: elaborado pelos autores

5.1 INTERFACE ESTABELECIMENTO ADMINISTRADOR

Para cadastrar um novo administrador de estabelecimento é preciso haver uma solicitação aos proprietários da aplicação, que por sua vez, estão responsáveis pela criação de um novo usuário liberando o perfil em questão. A liberação do Administrador proprietário foi realizada através do banco de dados durante o escopo desse projeto, tendo em vista que o aplicativo no presente momento foi manuseado apenas pelos autores do projeto. A imagem a 21 ilustra a tela de início, tanto para os usuários comuns quanto para os restaurantes administradores.

Figura 21 – Tela de login.

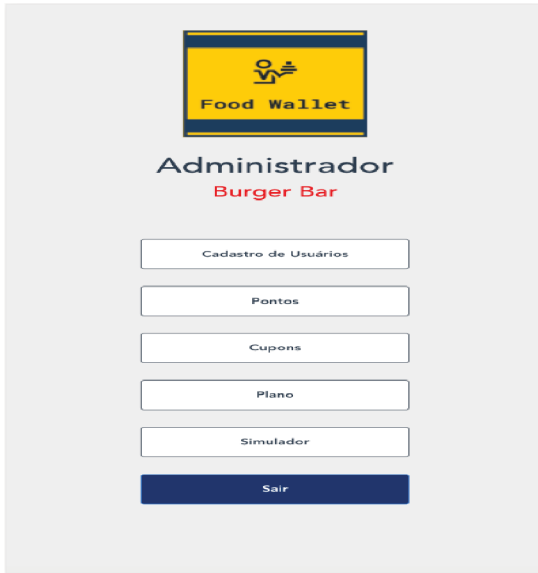


A screenshot of the login page for 'Food Wallet'. At the top center is a yellow rectangular logo with a stylized figure holding a fork and knife, with the text 'Food Wallet' below it. Below the logo is the heading 'A sua carteira de pontos!' in a bold, dark font. Underneath the heading are two white input fields: the first is labeled 'E-mail' and the second is labeled 'Senha'. Below these fields is a dark blue button with the text 'Entrar' in white.

Fonte: elaborado pelos autores

Após efetuado a criação do usuário de perfil restaurante administrador, deve-se usar as credenciais geradas para realizar o *login*. Com o acesso concluído, a aplicação disponibilizará o painel de gerenciamento ao estabelecimento associado conforme como ilustra a Figura 22.

Figura 22 – Tela inicial Administrador Restaurante.



Fonte: elaborado pelos autores

Na tela inicial do restaurante administrador é possível realizar cadastros de usuários, inserir novos pontos aos clientes, cadastrar cupons de desconto, configurar o plano com as taxas de pontuação do restaurante e fazer simulações de planos de fidelidade, de forma que a aplicação auxilie na criação de bons programas. Todas as telas citadas são formulários do tipo *input submit*, simplificando a usabilidade do usuário com o uso de formulários curtos. Na Figura 23, 24 e 25 é possível visualizar ilustrações que mostram o funcionamento de cada uma delas.

Figura 23 – Tela Gerenciamento Administrador Restaurante.

The image displays two mobile application screens for restaurant management. Both screens have a header with the text 'Administrador Burger Bar'.

The left screen, titled 'Cadastro de Usuários', features two input fields: 'E-mail' and 'CPF'. Below these fields are two buttons: a dark blue 'Salvar' button and a grey 'Voltar' button.

The right screen, titled 'Pontos', features three input fields: 'CPF', 'CPF', and 'Código'. Below these fields are three buttons: a dark blue 'Pontuar' button, a dark blue 'Validar' button, and a grey 'Voltar' button.

Fonte: elaborado pelos autores

Figura 24 – Gerenciamento de cupons e planos de fidelidade

Administrador Burger Bar

Meus Cupons

Nome	Valor	Tipo	
Cheese Burger	250	Prato P.	✖
Regrigerante lata	100	Bebida	✖
Suco lata	50	Bebida	✖

Adicionar Cupom

Tipo de item: Prato Principal Aperitivo Bebida

Administrador Burger Bar

Plano

Pontuação por Vendas:

Pontuação para 1º Cliente:

Pontuação por Vendas - Final de Semana:

Pontuação por Vendas - Período Noturno:

Fonte: elaborado pelos autores

Figura 25 – Simulador de Cupons

The image shows a mobile application interface for an administrator. At the top, it says 'Administrador Burger Bar' and 'Simulador'. Below this, there are two input fields: 'Preço Cobrado (R\$)' and 'Custo Real (R\$)'. A dark blue button labeled 'Calcular Lucro' is positioned below the second field. Underneath, there is a light blue input field for 'Lucro Atual (R\$)'. Below that is a slider control for 'Lucro Aceitável pela Venda', which is currently set to 'R\$0'. A second dark blue button, 'Calcular Redução', is located below the slider. At the bottom of the form is a grey button labeled 'Voltar'.

Fonte: elaborado pelos autores

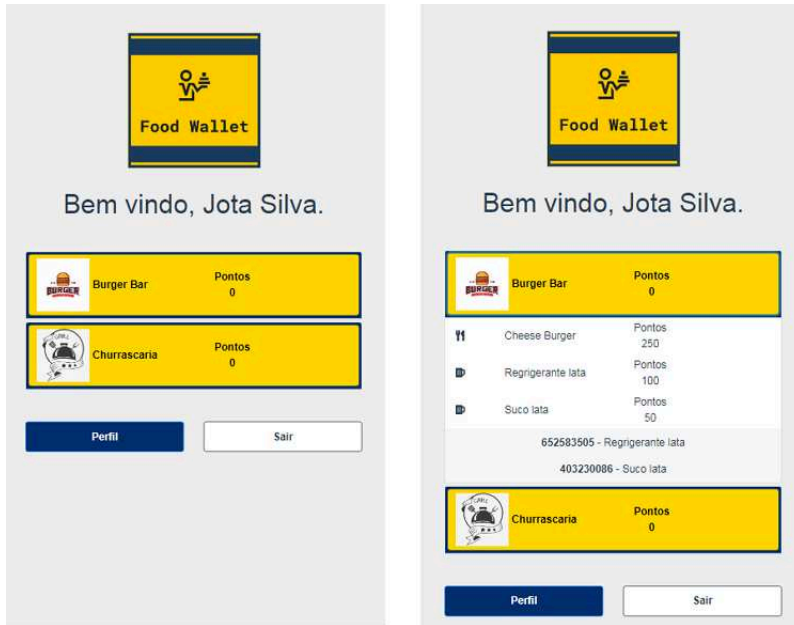
5.2 INTERFACE CLIENTE USUÁRIO

Com relação ao cliente do estabelecimento, o mesmo terá seu acesso criado pelo próprio restaurante/administrador, que por sua vez fará o cadastro na tela de cadastro de usuários, inserindo o e-mail do cliente e seu CPF, que num primeiro momento, servirá como senha de primeiro acesso.

Com as credenciais em mãos - e-mail e senha/CPF - o usuário

poderá entrar no sistema quando as inserir nos campos de login através do link disponibilizado. Feito isso, a tela inicial do usuário mostra a sua carteira de pontos, ou seja, a lista de estabelecimentos que ele foi cadastrado e a quantidade de pontos que o mesmo possui em cada um deles. De acordo com essa quantidade de pontos, é possível trocá-los ou não pelos pratos oferecidos por cada estabelecimento. A imagem à seguir mostra como está definida a tela inicial.

Figura 26 – Tela Inicial Usuário Cliente.



Fonte: elaborado pelos autores

Caso o usuário tenha o número mínimo de pontos estabelecidos para o cupom, será disponibilizado um botão de resgate ao lado do valor da pontuação. Ao executar seu resgate será gerado em sua carteira os códigos para cada cupom disponível, conforme imagem anterior. Para utiliza-lo o cliente deve apresentar o código gerado ao estabelecimento em questão, que por sua vez se encarregará de validar o mesmo em sua tela de gerenciamento.

Com o objetivo de incentivar os clientes a compartilharem alguns de seus dados particulares, que podem ser utilizados para tomadas de decisão dos estabelecimentos, há um campo específico para a atualiza-

ção dos seus dados cadastrais.

Para realizar essa atualização o cliente precisa clicar no botão denominado Perfil, no campo inferior esquerdo da tela inicial. A tela de atualização cadastral é composta por um formulário simples, que a princípio possui apenas 3 campos; nome, sexo e data de nascimento, além da opção de troca de senha, que encontra-se oculta, mas pode ser habilitada selecionando o campo Mudar Senha. A imagem 27 ilustra a atualização dos dados cadastrais.

Figura 27 – Tela Dados Cadastrais usuário cliente.

Dados Cadastrais

Nome Completo:
Jota Silva

Sexo: Masculino Feminino Não informar

Data de Nascimento:
01/01/1990

Mudar Senha ▾

Nova Senha

Confirmar Senha

Salvar

Voltar

Fonte: elaborado pelos autores

Por fim, pode-se notar que a aplicação foi desenvolvida visando, principalmente a simplicidade e usabilidade dos usuários, no entanto sem deixar de considerar o pré-requisito de unificação dos programas de pontos entre diferentes estabelecimentos, fator que foi definido como objetivo específico para elaboração dessa aplicação.

6 CONCLUSÃO

A estratégia de marketing relacional afiliado ao uso da tecnologia promovem o mercado de serviços na realidade atual. No desenvolvimento desse trabalho observou-se o cumprimento do objetivo geral e específicos do projeto, onde a aplicação *web* progressiva pode contribuir com o gerenciamento unificado e inteligente dos programas de fidelidades de bares e restaurantes. Através do *Food Wallet* os estabelecimentos conseguem testar métodos de pontuações e cupons para disponibilizar benefícios para seus clientes sem haver necessidade de grandes investimentos de desenvolvimento de aplicativos nativos. O formato da aplicação possibilita a digitalização do segmento de serviços com investimento relativamente baixo em tecnologia.

Os envios de notificações *push* para os clientes divulgando novos cupons e promoções gerando uma maior interação entre o cliente e fornecedor se mostraram requisitos relevantes para a construção da visão de relacionamento ao longo prazo, retenção dos clientes, aumento no nível de compromisso e contato com o cliente, todas perspectivas importantes no conceito de marketing de relacionamento.

Com base nos testes realizados com a aplicação foi possível evidenciar que as desvantagens do *PWA* não impactaram negativamente o uso da ferramenta, tendo em vista que o desempenho do aplicativo e as incompatibilidades de utilização de recursos, como por exemplo, *Face ID* ou *Touch ID*, não revelaram essenciais para o uso da ferramenta. A possibilidade de visualização dos dados armazenados em cache no navegador, assim como o desempenho do uso da plataforma e facilidade de inserir novas versões em produção sem haver impacto no usuário final se mostraram diferenciais do software.

Durante o uso da aplicação foi observado alguns pontos de melhorias de captação de dados que favorecem a relação de cliente fornecedor, ampliando os dados captados e conhecimento extraídos a partir das informações inseridas pelos clientes, possibilitando uma maior assertividade nas ações dos fornecedores. Para o acréscimo de clientes na carteira e retenção dos clientes já conhecidos também foram levantados como itens de melhoria. Havendo a possibilidade de destinar algumas visões gráficas dos usuários cadastrados na plataforma aos estabelecimentos independente de haver associação do cliente com o estabelecimento específico.

A aplicação do escopo de liberação dos dados dos usuários cadastrados para restaurantes distintos pode ser implementado no escopo

desse projeto como trabalho futuro, no entanto deve-se ser observado os direitos do titular conforme a lei geral de proteção de dados, tendo em vista que o compartilhamento, mesmo de maneira anônima, precisa ser autorizado pelo usuário que pode revogar a liberação de seus dados a qualquer momento.

6.1 TRABALHOS FUTUROS

Durante os testes da aplicação alguns requisitos de captação de dados estruturais para a realização de simulações de benefícios de maneira mais assertiva foram identificados. No modelo proposto a aplicação não registra e armazena os produtos que foram vendidos pelo estabelecimento, havendo uma oportunidade de melhoria de automação e geração de estatística no simulador em caso de integração dos dados com o sistema de venda do fornecedor.

Outra oportunidade, que promove o marketing relacional, identificado no desenvolvimento do projeto, está associado a captação de indicadores através das promoções e mensagens *push* enviadas do estabelecimento para os clientes já cadastrados. Ações de engajamento com o cliente podem ser melhor quantificadas caso o estabelecimento tenha conhecimento do resultado obtido a partir de notificações específicas enviadas ao cliente. Integrações com CRM também podem ser citadas como um auxílio na melhoria da obtenção dessas informações.

A aplicação também torna-se uma fonte de geração estratégica da relação do valor com o cliente com a inclusão na ferramenta de algumas informações de indicadores epidemiológicos e comportamentais dos usuários que adquirem produtos no estabelecimento ou na aplicação como um todo. As informações captadas dos usuários podem ser um aliado na construção do relacionamento cliente x fornecedor, pois disponibiliza ao fornecedor dados que facilitam a promoção de produtos e dias específicos.

Com a inclusão das funcionalidades o estabelecimento tem a oportunidade de acompanhar os indicadores de engajamento, conversão além de realizar interações com o cliente promovendo novos formatos de relacionamentos direcionados para o perfil de cliente atual e obtenção de novos perfis da clientes.

Outras oportunidades que podem ser obtidas em trabalho futuro estão associadas aos métodos de pagamento. A aplicação pode ser integrada a o novo meio de pagamento eletrônico do Brasil, o Pix. Desta forma, o usuário com uma conta habilitada pode realizar a transação de

pagamento imediatamente utilizando os números do telefone, CPF ou endereço de e-mail do usuário, facilitando assim o processo de utilização do aplicativo.

A aplicação também pode ser integrada a outros métodos de pagamento utilizando a tecnologia de *blockchain*, tecnologia responsável por transações seguras e imutáveis para a gravação de transações e rastreamento de ativos, dentre outras soluções seguras que promovem facilidades no contexto de métodos de pagamentos.

Por fim, sugere-se também como trabalho futuro o desenvolvimento de uma interface para o usuário administrador da aplicação, responsável este pelo cadastro de estabelecimentos e que no atual projeto é realizada apenas via inserção direta no banco de dados.

APÊNDICE A - Código-Fonte

O código-fonte da aplicação está disponível no endereço <https://github.com/agarceznunes/food-wallet/tree/master>

APÊNDICE B - Artigo

Aplicação Web Progressiva para o gerenciamento unificado de pontos de fidelidade entre bares e restaurantes

Alexandre Garcez Nunes e Luís Eduardo Vieira Tarouco Bueno

8 de dezembro de 2020

Resumo

Com a disseminação da tecnologia nos mais diversos setores, como é o caso do setor gastronômico, notamos uma marcante presença no uso de aplicações de software que visam fomentar o lucro e a qualidade dos produtos e serviços prestados por estabelecimentos comerciais, bem como fornecer e facilitar informações a seus clientes. Muitos desses estabelecimentos não possuem uma certa clareza sobre suas fraquezas, não conseguindo, desta forma, alcançar todo o potencial possível. Nesse sentido, um dos pontos importantes a ser levado em consideração, é o que esse estabelecimento pode e deve fazer para fidelizar e conquistar novos clientes.

No mercado gastronômico atual, podemos encontrar uma diversidade de programas de fidelidade que prometem mitigar esse problema, todavia, por serem muitos e dos mais diversos, acabam pecando na usabilidade e simplicidade, criando uma espécie de resistência ao seu uso pelos próprios clientes.

Pensando nisso, o presente projeto apresenta como solução o desenvolvimento de uma Aplicação *Web* Progressiva responsável por gerenciar e disponibilizar programas de fidelidade a clientes, assim como permitir aos usuários de bares e restaurantes o controle de todos os seus pontos de fidelidade obtidos. A aplicação visa, em primeiro lugar, a simplicidade, tanto com relação aos usuários clientes de estabelecimentos gastronômicos quanto aos gestores destes estabelecimentos, ambos com acesso a plataforma por diferentes interfaces.

Palavras-chaves: PWA. *Vue*. *Firebase*. Restaurantes. Marketing Relacional. Fidelidade. Cupom.

1 Introdução

Com o surgimento de novas tecnologias e o aprimoramento das já existentes, projetos e aplicações disseminaram em diversos modelos de negócios afetando não só a maneira como os produtos e serviços são aplicados na sociedade mas também na forma com que as pessoas realizam negócios.

Para o Desenvolvimento da Sociedade da Informação, segundo o Centro Regional de Estudos para o Desenvolvimento da Sociedade da Informação, em sua Pesquisa sobre o uso das tecnologias de informação e comunicação nas empresas brasileiras - TIC Empresas

2019, 98% das empresas utilizaram internet nos últimos 12 meses, sendo o mercado de alimentação e alojamento o menos expressivo com 91% de utilização (BRASIL, 2020).

Utilizar a internet como plataforma de negócio ou como apoio para resolver problemas e atrair clientes tem sido cada vez mais crucial para a competitividade e sustentabilidade das empresas. Aprimorar a relação de lealdade e oferecer experiências bem sucedidas ao invés de apenas bons produtos e serviços tem feito a diferença na escolhas dos clientes diante a tantas opções de mercado.

Estratégia muito utilizada no relacionamento empresa-cliente e que possui como principal objetivo estabelecer relações mutuamente satisfatórias de longa duração, o Marketing Relacional é definido por Berry Parasuraman (1991) como um conceito que tem por objetivo atrair clientes e manter relações duradouras, é o processo que utiliza os recursos da empresa para criar a melhor experiência possível e o máximo valor para o cliente, englobando um conjunto de atividades de Marketing orientadas para o estabelecimento de relações a longo prazo, com base nas promessas mútuas de confiança entre a empresa e o cliente.

Considerando isso, este projeto busca explorar o Marketing Relacional junto as empresas do setor de gastronomia, como bares e restaurantes, oferecendo uma solução tecnológica que incentive e auxilie as empresas na construção de boas estratégias de Marketing Relacional e forneça aos clientes aproximação com a rede de benefícios além da facilidade em pontuar e gerenciar em um único local seus pontos obtidos nos diferentes estabelecimentos frequentados.

Para isto, é proposto o desenvolvimento de uma Aplicação Web Progressiva, disponível para navegação multiplataforma. A aplicação visa uma experiência confiável e semelhante a de um aplicativo nativo, permitindo o acesso em qualquer dispositivo através de um navegador de internet. De acordo com Richard (2020) as PWAs são caracterizadas como rápidas, respondendo instantaneamente as interações dos usuários; confiáveis, com carregamento rápido independente da qualidade da rede; e envolvente, com interações e navegação semelhante aos aplicativos nativos, possibilitando notificações push e inserção de um atalho da aplicação na tela inicial do dispositivo móvel.

Neste projeto a utilização do PWA se constitui principalmente pela disponibilidade a todos os públicos, independente do dispositivo, pela possibilidade de utilização sem baixar a aplicação, pela alternativa em progressivamente instalar a aplicação após a aprovação do seu conteúdo, e por fim, pelo engajamento promovido após a aproximação e fácil acessibilidade fornecida aos usuários. Todas essas características apresentadas vão ao encontro da aplicação desenvolvida.

No decorrer do trabalho serão apresentadas as principais características e vantagens do Marketing de Relacionamento, apresentando estratégias habitualmente utilizadas por empresas e a forma como a aplicação incorporará essas estratégias em sua solução.

Na sequência serão apresentadas as tecnologias que em conjunto com as estratégias de mercado trarão benefícios aos usuários da aplicação, sejam eles estabelecimentos ou clientes.

E por fim, o projeto da solução desenvolvida contemplado por sua análise, recursos utilizados no desenvolvimento da aplicação e seus resultados obtidos.

2 Marketing Relacional

A maioria das teorias do marketing foram desenvolvidas a partir do mercado de produtos de grande consumo. Dessa forma, analisar marketing associado aos mercados de serviços sugere uma perspectiva diferente da habitual. Nesse contexto houve o advento do conceito de Marketing Relacional.

Segundo [Antunes Joaquim e Rita \(2008\)](#), pesquisadores consideram o marketing relacional como um conjunto de contribuições e abordagens multi-disciplinares, as quais são influenciadas por áreas associadas ao comportamento da organização, ao desenvolvimento de serviços e as tecnologias utilizadas. O marketing relacional é conhecido como uma evolução ao marketing de serviço e à gestão da qualidade.

A organização se habilita com estratégias para criação de valor na satisfação do cliente atuando na retenção do cliente com um maior nível de compromisso ([ANTUNES JOAQUIM E RITA, 2008](#)). Abaixo itens que definem as atividades e estrutura do Marketing Relacional:

- Ênfase na retenção dos clientes;
- Ênfase no valor do cliente;
- Visão a longo prazo;
- Grande ênfase no serviço ao cliente;
- Contato contínuo com o cliente;
- Elevado nível de compromisso com os clientes;
- Qualidade é uma preocupação de todos.

3 Aplicação Web Progressiva

Aplicativos Nativos e Web são dois tipos principais de aplicações. Aplicativos nativos dependem de plataformas como Android, iOS, Windows e são criados usando linguagens de programação específicas, enquanto os aplicativos Web são plataformas em formato de sites e são executados por um navegador, geralmente escritos em HTML5. Os aplicativos nativos têm acesso total ao hardware do dispositivo, enquanto os aplicativos da Web ainda carecem de algumas limitações.

Nos últimos anos, o uso de aplicativos móveis tiveram um crescimento em comparação à web móvel. Empresas e desenvolvedores muitas vezes viam a necessidade de criar aplicativos móveis nativos para superar as limitações que as plataformas Web impõem aos dispositivos móveis. Em contrapartida os aplicativos nativos precisam de mais recursos e tempo para desenvolver, manter e distribuir as soluções. Para a evolução da tecnologia, havia a necessidade de uma plataforma que pudesse combinar os recursos e as experiências dos aplicativos nativos com a Web.

Aplicação Web Progressiva é o termo que combina os principais benefícios apresentados pelas duas frentes propostas. Foi introduzido pela primeira vez na conferência de desenvolvedores da Google em maio de 2016 em San Francisco pelo engenheiro do Google Chrome, Alex Russell. O termo foi utilizado para descrever a nova geração de aplicativos

da web que são carregados em sites comuns, mas abordam a maioria dos recursos que faltavam nos aplicativos da Web. O combinação de uso do *Service Worker*, *shell* para aplicativos, *Web App Manifest* e notificação por push fizeram o PWA parecer e se comportar como um aplicativo nativo. Como o PWA é um aplicativo da web, ele pode ser usado em todas as plataformas, reduzindo o custo e os recursos e aumentando a acessibilidade (THAKUR, 2018).

Segundo Warcholinski (2018) Muitas empresas líderes adotaram o PWA e lançaram seus próprios aplicativos da Web com funcionalidades nativa, incluindo o AliExpress, que relatou um aumento de 104% nas taxas de conversão para novos usuários, e o Twitter, que observou aumento de 75% nos tweets e uma redução de 20% na taxa de rejeição.

4 Vue

Vue.js se trata de um framework JavaScript progressivo para construção de interfaces de usuário e as famosas *Single Page Applications (SPA)*. O framework foi originalmente inventado por Evan You, um ex-funcionário da Google, que além de trabalhar nos projetos de Vue.js, também atuou no projeto Meteor.js, um outro framework Javascript, enquanto ainda atuava na Google. Evan ainda teve a oportunidade de atuar em projetos importantes de Angular, framework já consagrado na época, mas arriscou investir num projeto próprio, que seria uma espécie de aperfeiçoamento do mesmo.

Desde então o Vue.js vem ganhando cada vez mais popularidade entre a comunidade de desenvolvedores. Dentre os motivos que contribuem para essa crescente, pode-se destacar o fato de ele ser muito mais leve e simples que outros frameworks mais populares, como o React e o Angular, além de ainda, ser uma espécie de combinação de características desses frameworks.

4.1 Visão Geral da Arquitetura

O Vue.js conta com a capacidade de um *Data Binding Reativo*, ou seja, ele torna extremamente simples a sincronização entre os dados e o *Domain Object Model (DOM)*, como mostra a Figura 1.

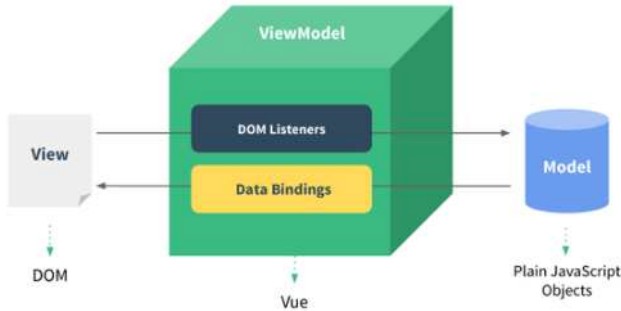
5 Firebase

Em 2014, Google completou a aquisição de uma empresa em São Francisco chamada Firebase Inc. Firebase contempla uma agrupamento de soluções de desenvolvimento destinado a acelerar integração de recursos em nuvem com aplicativos web e móveis. Após a compra a empresa realizou uma série de alterações combinando os serviços do *Google Cloud* (KHAWAS; SHAH, 2018).

Segundo Smyth (2017) Firebase é considerado uma plataforma de desenvolvimento web.

A plataforma, conhecida por ser um *Backend-as-a-Service (BaaS)*, oferece serviços como banco de dados em tempo real, armazenamento em nuvem, autenticação, relatórios de falhas, aprendizado de máquina, configuração remota e hospedagem de arquivos estáticos. Seu banco de dados não relacional (NoSQL), armazena os dados em formato *JavaScript Object Notation JSON* Smyth (2017).

Figura 1 – Vue - Visão geral



Fonte: <https://v1.vuejs.org/guide/overview.html>.

Conforme Smyth (2017) após as dependências adicionadas ao aplicativo, os dados não estruturados podem ser adicionados ao banco de dados pelo seguinte código:

```
//Write a message to the Firebase Database  
  
database= FirebaseDatabase.getInstance;  
DatabaseReference myRef = database.getReference("node");  
myRef.setValue ("Hello, World");
```

Arquivos como imagens, áudio, vídeo podem ser armazenados no aplicativo. Os dados são altamente protegidos e robustos, dando continuidade do último ponto em caso de erro de rede.

Aplicativos móveis geralmente precisam de serviços de *back-end* para trabalhar com a interface do usuário de *front-end*. Isso significa que é necessário desenvolvimento de um *back-end* e servidores para trabalhar com aplicativos móveis. O Firebase é um serviço de nuvem para impulsionar *back-ends* dos aplicativos. Além de fornecer o suporte para armazenamento de dados e autenticação do usuário, após a integração do aplicativo móvel ao Firebase, não é preciso escrever código de *back-end* ou gerenciar a infraestrutura.

6 Solução Proposta

Para auxiliar os clientes na gestão de seus pontos de fidelidade assim como impulsionar os estabelecimentos parceiros com a utilização de estratégias bem sucedidas, será projetada e desenvolvida uma aplicação *web* que implemente os conceitos resultantes da fundamentação apresentada.

A aplicação deve ser desenvolvida utilizando-se da metodologia de uma aplicação *web* progressiva, o *PWA*, usufruindo assim de seus principais benefícios tais como velocidade, responsividade, confiabilidade e interação envolvente.

Através de sua utilização será possível pontuar a carteira de um cliente, cadastrar um cupom, simular e disponibilizar programas de fidelidade, gerenciar pontos obtidos como cliente e quando possível, resgatar cupons.

A análise da aplicação será feita com base na fundamentação teórica através de uma lista de requisitos e os diagramas UML de classes e casos de uso. Seu desenvolvimento será em duas etapas: regras de negócio e interface gráfica.

O cronograma do projeto foi desenvolvido visando a liberação de um MVP, produto mínimo viável *MVP Minimum Viable Product*, e as atividades foram divididas entre *sprints* com o objetivo de validar a ideia do serviço com usuários de aplicativos, bares e restaurantes. Após finalizada as etapas essenciais serão executados testes que validam e avaliam a aplicação desenvolvida.

7 Diagrama de Classes

Na Figura 2 é apresentada a versão do diagramas de classes proposto neste trabalho, onde as classes *UsuarioEstabelecimento* e *UsuarioCliente* são agregados da classe *Usuário*, diferenciadas por seus atributos que servirão para interfaces distintas na aplicação desenvolvida.

Os usuários *Clientes* possuem uma coleção de um ou mais cartões fidelidade, obtidos durante a pontuação em um estabelecimento. Caso seja a primeira compra do cliente no estabelecimento o cartão é adicionado a sua carteira seguido da pontuação relativa ao programa de fidelidade vigente no ato da compra, ao contrário, apenas a pontuação do cliente é acrescida em seu cartão já existente. Conforme regra de negócio RN005 todos os clientes são cadastros através de um estabelecimento, desta forma, toda classe *Usuario-Cliente* deve estar relacionada a no mínimo um cartão fidelidade.

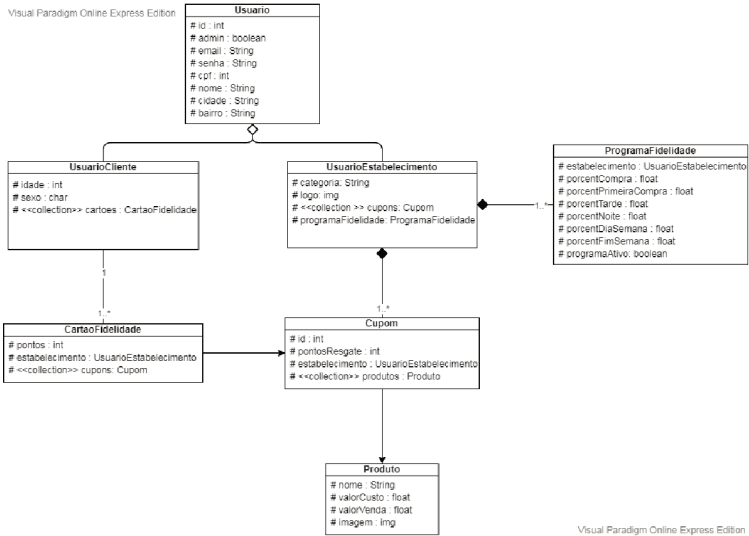
Após acumular determinada quantidade de pontos o cliente pode resgatar um cupom disponibilizado pelo restaurante, desde que sua pontuação disponível seja maior ou igual ao mínimo necessário para o seu resgate. Os cupons gerados pelo estabelecimento para obtenção de benefícios estão associados aos produtos previamente cadastrados. Ao liberar um novo cupom de troca o estabelecimento precisa informar obrigatoriamente a pontuação de resgate do produto cadastrado.

Todos os cupons disponíveis são cadastrados pelo *UsuarioEstabelecimento*, assim como os programas de fidelidade vigentes. É obrigatório que um estabelecimento esteja associado a no mínimo um programa de fidelidade e um cupom, tendo em vista tratar-se da essência da aplicação. Os programas de fidelidade são cadastrados por uma taxa, representada em percentual, aplicada sobre o valor da compra efetuada. Caso seja de interesse do estabelecimento é possível definir diferentes taxas para clientes recorrentes e novos clientes, dias de semanas e fins de semana, e por fim, compras no período diurno e compras no período noturno.

8 Casos de Uso

A Figura 3 reproduz os casos de uso associados ao projeto. Três atores foram identificados no desenvolvimento da solução. As atividades serão desdobradas em funcionalidades do aplicativo. O *Ator Administrador* é responsável por cadastrar o estabelecimento e inserir as informações requisitos para a liberação do associado. Após a liberação

Figura 2 – Diagrama de Classes



Fonte: Elaborado pelos autores

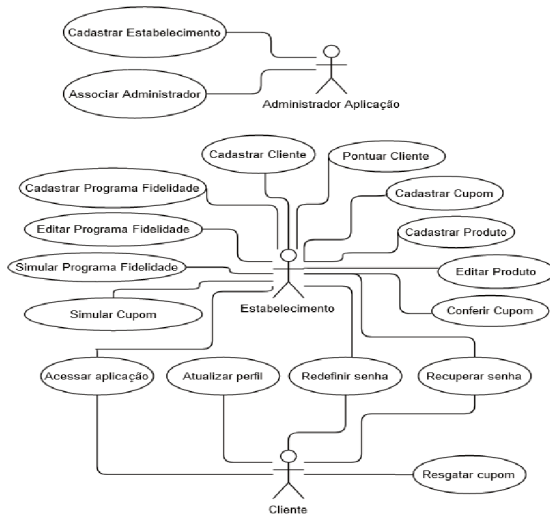
do estabelecimento o Administrador tem a função de atribuir um usuário responsável do estabelecimento para que seja realizado os cadastros específicos dos produtos e promoções.

O Ator Estabelecimento, usuário responsável pela unidade de negócio, fica encarregado pelo *setup* básico do estabelecimento como: cadastrar cupons, cadastrar e editar produtos, cadastrar, simular e editar programas de fidelidade. Algumas das atividades associadas ao Estabelecimento possuem interface com o Ator cliente, dentre elas: Pontuar, utilizada no momento do pagamento da compra do usuário. Resgatar Cupom: relacionado ao momento que ao Ator Usuário contém pontos suficientes para utilizar em uma promoção vigente no estabelecimento.

Os programas de fidelidade cadastrados pelo Ator Estabelecimento serão os responsáveis pelos benefícios recebidos de *cashback* pelo Ator Cliente no momento da pontuação. O programa de fidelidade pode ser editado a qualquer momento. O Ator estabelecimento pode criar um ou muitos programas de fidelidade parametrizando o benefício por dia da semana, por exemplo, conforme sua estratégia de marketing associado a campanha executada. No momento da simulação do programa de fidelidade o Ator Estabelecimento tem a possibilidade de utilizar seus produtos previamente cadastrados para obter uma pontuação sugerida no programa proposto com base na média do total de vendas do produto.

O Ator Usuário pode realizar outras funcionalidades como, por exemplo, atualizar seu cadastro, redefinir e recuperar sua senha assim como acessar a aplicação para ver sua carteira em estabelecimentos onde seu CPF já foi utilizado. Algumas funcionalidades como, recuperar senha, atualizar perfil estão também disponíveis para o Ator Estabelecimento, tendo em vista que o ator é um objeto do tipo usuário, conforme apresentado na subseção 7.

Figura 3 – Casos de Uso



Fonte: Elaborado pelos autores

9 Resultados Obtidos

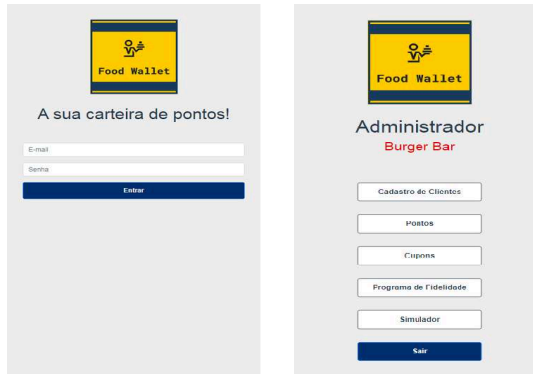
9.1 Interface Estabelecimento Administrador

Para cadastrar um novo administrador restaurante é preciso haver uma a solicitação aos proprietários da aplicação, que por sua vez, estão responsáveis pela criação de um novo usuário liberando o perfil em questão. A liberação do Administrador proprietário foi realizada através do banco de dados durante o escopo desse projeto, tendo em vista que o aplicativo no presente momento foi manuseado apenas pelos autores do projeto. A figura a 4 ilustra a tela de início, tanto para os usuários comuns quanto para os restaurantes administradores.

Após efetuado a criação do usuário de perfil restaurante administrador, deve-se usar as credenciais geradas para realizar o *login*. Com o acesso concluído, a aplicação disponibilizará o painel de gerenciamento ao estabelecimento associado conforme como ilustra também a figura 4.

Na tela inicial do restaurante administrador é possível realizar cadastros de usuários, inserir novos pontos aos clientes, cadastrar cupons de desconto, configurar o plano com as taxas de pontuação do restaurante e fazer simulações de rendimento conforme o plano configurado. Todas as telas citadas são formulários do tipo *input submit*, simplificando a usabilidade do usuário com o uso de formulários curtos. Na Figura 5 e 6 é possível visualizar ilustrações que mostram o funcionamento de cada uma delas.

Figura 4 – Tela de login e menu do estabelecimento.



Fonte: Elaborado pelos autores

Figura 5 – Tela Gerenciamento Administrador Restaurante.



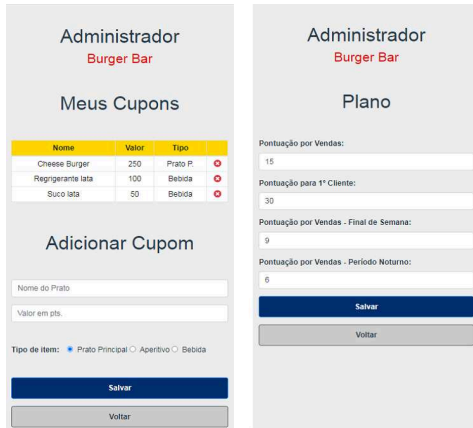
Fonte: Elaborado pelos autores

9.2 Interface Usuário Cliente

Com relação ao cliente do estabelecimento, o mesmo terá seu acesso criado pelo próprio restaurante/administrador, que por sua vez fará o cadastro na tela de cadastro de usuários, inserindo o e-mail do cliente e seu CPF, que num primeiro momento, servirá como senha de primeiro acesso.

Com as credenciais em mãos - e-mail e senha/CPF - o usuário poderá entrar no sistema quando as inserir nos campos de login através do link disponibilizado. Feito isso, a tela inicial do usuário mostra a sua carteira de pontos, ou seja, a lista de estabelecimentos que ele foi cadastrado e a quantidade de pontos que o mesmo possui em cada um deles. De

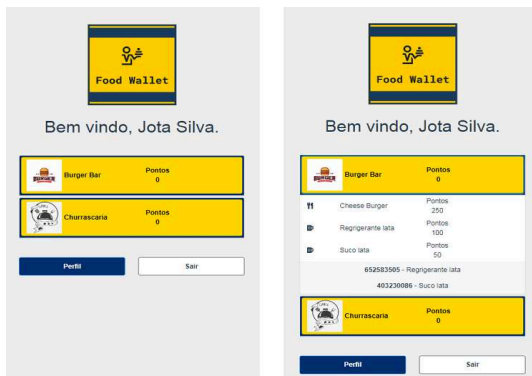
Figura 6 – Tela Gerenciamento Administrador Restaurante.



Fonte: Elaborado pelos autores

acordo com essa quantidade de pontos, é possível trocá-los ou não pelos pratos oferecidos por cada estabelecimento. A figura 7 apresenta como está definida a tela inicial.

Figura 7 – Tela Inicial Usuário Cliente.



Fonte: Elaborado pelos autores

Caso o usuário tenha o número mínimo de pontos estabelecidos para o cupom, será disponibilizado um botão de resgate ao lado do valor da pontuação. Ao executar seu resgate será gerado em sua carteira os códigos para cada cupom disponível, conforme imagem anterior. Para utilizá-lo o cliente deve apresentar o código gerado ao estabelecimento em questão, que por sua vez se encarregará de validar o mesmo em sua tela de gerenciamento.

Com o objetivo de incentivar os clientes a compartilharem alguns de seus dados particulares, que podem ser utilizados para tomadas de decisão dos estabelecimentos,

há um campo específico para a atualização dos seus dados cadastrais. O projeto foi implementado de modo que os estabelecimentos forneçam uma pontuação extra para cada usuário que completar a atualização destes dados. Essa pontuação pode ser definida pelo restaurante na tela de definição de planos.

Para realizar essa atualização o cliente precisa clicar no botão denominado Perfil, no campo inferior esquerdo da tela inicial. A tela de atualização cadastral é composta por um formulário simples, que a princípio possui apenas 3 campos; nome, sexo e data de nascimento, além da opção de troca de senha, que encontra-se oculta, mas pode ser habilitada selecionando o campo Mudar Senha.

10 Conclusão

A estratégia de marketing relacional afiliado ao uso da tecnologia promovem o mercado de serviços na realidade atual. No desenvolvimento desse trabalho observou-se o cumprimento do objetivo geral e específicos do projeto, onde a aplicação *web* progressiva pode contribuir com o gerenciamento unificado e inteligente dos programas de fidelidades de bares e restaurantes. Através do *Food Wallet* os estabelecimentos conseguem testar métodos de pontuações e cupons para disponibilizar benefícios para seus clientes sem haver necessidade de grandes investimentos de desenvolvimento de aplicativos nativos. O formato da aplicação possibilita a digitalização do segmento de serviços com investimento relativamente baixo em tecnologia.

Os envios de notificações *push* para os clientes divulgando novos cupons e promoções gerando uma maior interação entre o cliente e fornecedor se mostraram requisitos relevantes para a construção da visão de relacionamento ao longo prazo, retenção dos clientes, aumento no nível de compromisso e contato com o cliente, todas perspectivas importantes no conceito de marketing de relacionamento.

Com base nos testes realizados com a aplicação foi possível evidenciar que as desvantagens do *PWA* não impactaram negativamente o uso da ferramenta, tendo em vista que o desempenho do aplicativo e as incompatibilidades de utilização de recursos, como por exemplo, *Face ID* ou *Touch ID*, não revelaram essenciais para o uso da ferramenta. A possibilidade de visualização dos dados armazenados em cache no navegador, assim como o desempenho do uso da plataforma e facilidade de inserir novas versões em produção sem haver impacto no usuário final se mostraram diferenciais do software.

Durante o uso da aplicação foi observado alguns pontos de melhorias de captação de dados que favorecem a relação de cliente fornecedor, ampliando os dados captados e conhecimento extraídos a partir das informações inseridas pelos clientes, possibilitando uma maior assertividade nas ações dos fornecedores. Para o acréscimo de clientes na carteira e retenção dos clientes já conhecidos também foram levantados como itens de melhoria. Havendo a possibilidade de destinar algumas visões gráficas dos usuários cadastrados na plataforma aos estabelecimentos independente de haver associação do cliente com o estabelecimento específico.

A aplicação do escopo de liberação dos dados dos usuários cadastrados para restaurantes distintos pode ser implementado no escopo desse projeto como trabalho futuro, no entanto deve-se ser observado os direitos do titular conforme a lei geral de proteção de dados, tendo em vista que o compartilhamento, mesmo de maneira anônima, precisa ser autorizado pelo usuário que pode revogar a liberação de seus dados a qualquer momento.

Referências

ANTUNES JOAQUIM E RITA, P. O marketing relacional como novo paradigma: uma análise conceptual. *Revista de Gestão dos Países de Língua Portuguesa*, scielopt, v. 7, p. 36 – 46, 04 2008. ISSN 1645-4464. Disponível em: <http://www.scielo.mec.pt/scielo.php?script=sci_arttextpid=S1645-44642008000200005nrm=iso>. Citado na página 3.

BRASIL, C. G. da Internet no. *Pesquisa Sobre o Uso das Tecnologias de Informação e Comunicação nas Empresas Brasileiras*. Núcleo de Informação e Coordenação do Ponto BR, 2020. Disponível em: <<https://www.cetic.br/media/docs/publicacoes>>. Citado na página 2.

KHAWAS, C.; SHAH, P. Application of firebase in android app development-a study. *International Journal of Computer Application*, v. 179, 2018. Citado na página 4.

RICHARD, P. L. S. 2020. Disponível em: <<https://web.dev/what-are-pwas/>>. Acesso em: 07/12/2020. Citado na página 2.

SMYTH, N. *Firebase Essentials - Android Edition*. CreateSpace Independent Publishing Platform, 2017. ISBN 9781546660330. Disponível em: <<https://books.google.com.br/books?id=9i4tDwAAQBAJ>>. Citado 2 vezes nas páginas 4 e 5.

THAKUR, P. *Evaluation and Implementation of Progressive Web Application*. Metropolia Ammattikorkeakoulu, 2018. Disponível em: <<http://www.theseus.fi/handle/10024/142997>>. Citado na página 4.

WARCHOLINSKI, M. What are the advantages and disadvantages of progressive web apps? 2018. Disponível em: <<https://brainhub.eu/blog/advantages-disadvantages-progressive-web-apps/>>. Acesso em: 19 jun. 2020. Citado na página 4.

REFERÊNCIAS

- ANTUNES JOAQUIM E RITA, P. O marketing relacional como novo paradigma: uma análise conceptual. *Revista de Gestão dos Países de Língua Portuguesa*, scielo, v. 7, p. 36 – 46, 04 2008. ISSN 1645-4464. <[http://www.scielo.mec.pt/scielo.php?script=sci_arttext&pid = S1645 – 44642008000200005&nrm = iso](http://www.scielo.mec.pt/scielo.php?script=sci_arttext&pid=S1645-44642008000200005&nrm=iso)>.
- BIORN-HANSEN, T. A. M. A.; GRONLI, T.-M. Progressive web apps: The possible web-native unifier for mobile development. 2017.
- BRASIL, C. G. da Internet no. *Pesquisa Sobre o Uso das Tecnologias de Informação e Comunicação nas Empresas Brasileiras*. Núcleo de Informação e Coordenação do Ponto BR, 2020. <<https://www.cetic.br/media/docs/publicacoes>>.
- CHENG, F. *Build Mobile Apps with Ionic 4 and Firebase*. Apress, 2018. ISBN 978-1-4842-3774-8. <<https://www.academia.edu/42089943>>.
- CHRISTOPHER, A. P. M. *Relationship Marketing*. Elsevier Butterworth-Heinemann, 1994. <<https://books.google.com.br>>.
- FRANSSON, R.; DRIAGUINE, A. *Comparing Progressive Web Applications with Native Android Applications : An evaluation of performance when it comes to response time*. 2017. 58 p.
- GRILL, R. P. d. M. F. F. D. N. *Aprendendo JavaScript*. Universidade de São Paulo, 2008. <<https://web.icmc.usp.br/SCATUSU/RT/nd72.pdf>>.
- GUMMESSON, E. Implementation requires a relationship marketing paradigm. *Journal of the Academy of Marketing Science*, v. 26, n. 3, p. 242–249, 1998. <<https://doi.org/10.1177/0092070398263006>>.
- KHAWAS, C.; SHAH, P. Application of firebase in android app development-a study. *International Journal of Computer Application*, v. 179, 2018.
- KUMAR, A. *Mastering Firebase for Android Development: Build real-time, scalable, and cloud-enabled Android apps with Firebase*. Packt Publishing, 2018. ISBN 9781788624251. <<https://books.google.com.br/books?id=RMNiDwAAQBAJ>>.

OVERFLOW, S. Instant loading web apps with an application shell architecture. 2018.

<<https://insights.stackoverflow.com/survey/2018technology-frameworks-libraries-and-tools>>. Acessado em 24 jun. 2020.

RICHARD, P. L. S. 2020. <<https://web.dev/what-are-pwas/>>. Acessado em 07/12/2020.

SMYTH, N. *Firebase Essentials - Android Edition*. CreateSpace Independent Publishing Platform, 2017. ISBN 9781546660330.

<<https://books.google.com.br/books?id=9i4tDwAAQBAJ>>.

THAKUR, P. *Evaluation and Implementation of Progressive Web Application*. Metropolia Ammattikorkeakoulu, 2018.

<<http://www.theseus.fi/handle/10024/142997>>.

TOLEDO, S. L. d. A. M. G. L. *Valor Para o Cliente e Valor do Cliente Conceitos e Implicações para o Processo de Marketing*.

Desenvolvimento em Questão, 2015. <<https://books.google.com.br>>.

WARCHOLINSKI, M. What are the advantages and disadvantages of progressive web apps? 2018. <<https://brainhub.eu/blog/advantages-disadvantages-progressive-web-apps/>>. Acessado em 19 jun. 2020.

WOHLGETHAN, E. Supporting web development decisions by comparing three major javascript frameworks:angular, react and vue.js. 2018.

<<https://edoc.sub.uni-hamburg.de/haw/volltexte/2018/4350>>.

Acessado em 29 jun. 2020.

ZYGO. Quem somos. 2020. <<https://zygotecnologia.com/quem-somos>>. Acessado em 01 nov. 2020.