

UM ESTUDO SOBRE A INTEGRAÇÃO ENTRE SERVIÇOS DA PLATAFORMA FIREBASE E PWA¹

Gabriel Gradaschi Heinz²

José Antônio Oliveira de Figueiredo³

RESUMO

O presente trabalho apresenta um estudo sobre tecnologias disponíveis para o desenvolvimento WEB, em especial a Progressive WEB Application e sua aplicação como um estudo de caso integrado com a plataforma Firebase. A motivação para este trabalho parte da iniciativa de que aplicações híbridas, que é o caso das PWA's, estão cada vez mais comuns no dia-a-dia. Com isso, são apresentadas informações a respeito das características delas e dos serviços que trabalham junto dela. A partir do estudo verificou-se que as Progressive WEB Application trazem funcionalidades interessantes para o desenvolvimento WEB Mobile, em conjunto dos recursos de terceiros, principalmente, a plataforma Firebase. Em suma, este trabalho pretende mostrar as vantagens das PWA's e do Firebase para o desenvolvimento de aplicativos híbridos.

Palavras-chave: Progressive WEB Application. Aplicativos Híbridos. FireBase.

INTRODUÇÃO¹

Vivemos com a constante dinâmica da WEB que a cada dia se torna mais acessível, adequando-se às novas necessidades que surgem diariamente, especialmente no que se refere a ter acesso a informações em qualquer lugar. Assim, para que sua utilização possa ser melhor aproveitada, foram desenvolvidas

¹ Trabalho de Conclusão de Curso (TCC) apresentado ao Curso de Tecnologia em Sistemas para Internet do Instituto Federal Sul-rio-grandense, Câmpus Passo Fundo, como requisito parcial para a obtenção do título de Tecnólogo em Sistemas para Internet, na cidade de Passo Fundo, em 2021.

² Aluno do curso de Tecnologia de Sistemas para Internet no Instituto Federal de Educação, Ciência e Tecnologia Sul-rio-grandense de Passo Fundo (IFSUL). E-mail: manchaheinz@gmail.com.

³ Orientador, professor do Instituto Federal de Educação, Ciência e Tecnologia Sul-rio-grandense de Passo Fundo (IFSUL). E-mail: jose.figueiredo@ifsul.edu.com.br

novas aplicações que aprimoram a experiência e capacidade dos aplicativos móveis de se adaptarem.

De acordo com Ilhasoft (2016), empresa brasileira de desenvolvimento, estamos na era mais digital de todos os tempos e vivendo um grande avanço tecnológico diariamente, com novas tecnologias, novas soluções e novos dispositivos. Dentre essa tecnologia, é possível encontrar ferramentas que facilitam o cotidiano das pessoas: os aplicativos. Atualmente, eles são os responsáveis pela maioria de nossa interação digital, e alguns são considerados indispensáveis tanto para o trabalho quanto para a vida social, como o WhatsApp ou o Facebook, por exemplo.

O desenvolvimento de Progressive WEB Application (PWA), une os serviços disponibilizados por aplicações WEB tradicionais programas desenvolvidos para serem acessados através de navegadores, utilizando tecnologias da WEB, como HTML, CSS e JavaScript. Com as funcionalidades dos aplicativos nativos, programas desenvolvidos especificamente para uma plataforma, capazes de executar e explorar as funcionalidades da plataforma para o qual foram desenvolvidos, possibilitando reduzir os custos de distribuição e manutenção. Ao tornar uma aplicação WEB em uma aplicação PWA, aumentamos a duração de tempo do usuário em cada sessão no navegador, além de aumentar a taxa de conversão de novos usuários.

O aplicativo proposto neste trabalho, é um fórum online, onde os usuários podem criar postagens com suas dúvidas para serem respondidas por outros usuários do aplicativo. Ele possibilita o cadastro de postagens, categorias e imagens, todas elas armazenadas no Firebase. As postagens possuem os campos título, referente ao nome dela, categoria, uma forma de filtrar as postagens por uma categoria em específico e a descrição, onde o usuário irá descrever o assunto da postagem. As categorias, possuem um cadastro simples, tendo apenas o campo título, onde o usuário informa o nome da categoria, caso as que existem não se adequam a postagem. Por fim, a aplicação possibilita o upload de imagens armazenadas no dispositivo do usuário.

Neste trabalho é apresentado, além de um estudo sobre Progressive Web Application e a plataforma Firebase, também será apresentado o desenvolvimento do aplicativo ForumOnline, uma PWA integrada com a plataforma Firebase com o objetivo de compreender o funcionamento dessas duas tecnologia, visto que as PWAs estão cada vez mais presentes no cotidiano.

O artigo está dividido como segue: seção 2 tem-se um referencial teórico tanto da Progressive WEB Application, como conceitos e características, quanto da plataforma Firebase, com conceitos e serviços; seção 3 é dedicada à documentação da aplicação, descrevendo os métodos e serviços utilizados e, por fim; seção 4 são apresentados as considerações finais.

2 MATERIAIS E MÉTODOS

Nesta seção são apresentados alguns dos conceitos que embasam o desenvolvimento do trabalho.

2.1 Progressive WEB Application

Os Progressive WEB Apps (PWAs) são aplicativos baseados nas tecnologias utilizadas na WEB - HTML, CSS e JavaScript -, sendo executados no browser dos dispositivos móveis (CECCONI, 2018).

Em dispositivos móveis, funcionalidades como - câmera, geolocalização e trabalhar offline - só eram disponíveis com desenvolvimento de aplicativos híbridos ou nativos. Em virtude da evolução que a tecnologia WEB está sofrendo, principalmente nos navegadores, tornou-se possível acessar diversas funcionalidades e recursos de hardware usando basicamente tecnologias voltadas para WEB (HTML, CSS e Javascript) e além de possibilitar a utilização delas em desktop, qualificando estas aplicações como PWA (CECCONI, 2018).

As PWAs são, de certa forma, semelhantes à aplicativos WEB, nativos e híbridos, entretanto não totalmente. A listagem a seguir, mostra um resumo sobre cada tipo de aplicativo:

- Aplicativos WEB: são aplicativos que rodam em cima dos navegadores. Estão disponíveis através de uma URL e tem acesso restrito aos recursos do dispositivo.
- Aplicativos nativos: são aplicativos desenvolvidos para uma plataforma específica. Tem acesso a todos os recursos disponíveis no dispositivo, no entanto o custo de desenvolvimento e manutenção é maior que outros aplicativos pela necessidade de escrever um código para cada plataforma, além de se requerer mais tempo e mais desenvolvedores. Podem ser disponibilizados em lojas virtuais.

- Aplicativos híbridos: são aplicativos desenvolvidos utilizando tecnologias WEB, com comportamento que se assemelha aos aplicativos nativos. Escrevendo um código apenas é possível executar a aplicação em diferentes plataformas. Tem acesso a alguns recursos do dispositivo e também podem ser disponibilizados em lojas virtuais.
- Aplicativos PWA: são aplicativos desenvolvidos para mais de uma plataforma como aplicativos híbridos, com uma experiência mais próxima da oferecida por aplicativos nativos. São executados por meio do browser, além de poderem acessar recursos do dispositivo móvel através do navegador. E podem ser hospedados na WEB como aplicativos WEB. Em suma, uma PWA é uma junção das principais características de cada aplicativo.

2.1.1 Características das PWA's

As PWAs têm como objetivo proporcionar ao usuário a melhor experiência possível. Elas devem ser: progressivas, responsivas, trabalhar off-line, ser semelhantes a aplicativos, permitirem atualizações constantes, seguras, fáceis de se encontrar, engajadas, instaláveis e linkáveis (LEPAGE, 2019).

- **Progressivas:** Baseia-se em utilizar a técnica *progressive enhancement* ao desenvolver a PWA. Ao usar esse preceito na prática, no decorrer do desenvolvimento, aplicando em camadas as funcionalidades disponíveis na WEB, estará garantido que a aplicação funcionará em qualquer navegador para qualquer usuário. Navegadores modernos com suas atualizações em dia, entregam uma gama de funcionalidades maior e melhor que suas versões desatualizadas e navegadores antigos terão limitações para executar algumas funcionalidades (CECCONI, 2018).
- **Responsividade:** A noção de *Responsive WEB Design* (RWD) por meio de uma matéria escrita por Ethan Marcotte em meados de 2010, começou a se espalhar. Nesta matéria é dito que utilizando o RWD permite aos desenvolvedores projetar interfaces que se adequam a diversos ambientes e interações do usuário (MARCOTTE, 2010). Neste ambiente, o RWD pretende

organizar as páginas WEB conforme seu tamanho, disposição da tela, cores e alterações geradas a partir do usuário (CECCONI, 2018).

- **Trabalhar offline:** Com a intenção de melhorar a experiência do usuário em caso onde não haja conexão, criou-se o termo *Offline First*. O conceito de *Offline First* é um conceito que objetiva desenvolver a aplicação a partir do cenário com o menor grau de conexão. Segundo Cecconi (2018), casos onde a conectividade é baixa, instável ou que não haja conectividade são frequentes.

Cecconi (2018) diz que, não há necessidade de resolver o problema da falta de conexão de uma só vez, o desenvolvedor pode resolver em partes armazenando as informações do usuário em cache local. Em casos onde a falta de conexão é total, o usuário é informado que algumas funcionalidades da aplicação não poderão ser utilizadas, no entanto toda e qualquer ação do usuário durante esse período, pode ser armazenada no banco de dados ou servidor responsável pela aplicação no momento que a mesma possuir conexão.

Esta ação se baseia em criar caches de páginas de dados acessados pelo usuário durante o período que a aplicação ficou offline. Uma aplicação PWA conta com um recurso chamado *Service Work*, que são scripts que o navegador executa em segundo plano, onde os recursos não necessitam de uma página WEB ou da interação de um usuário. Tais recursos, ao serem utilizados nas aplicações, podem administrar as páginas WEB com eventos iniciados por essas (GAUNT, 2019).

Conforme Gaunt (2019) os *Service Workers* funcionam separados das páginas WEB fazendo um meio de campo entre o usuário e o servidor. Caso a conexão com a Internet falhe ou a aplicação fique *offline*, os *Service Workers* têm a responsabilidade de armazenar os dados, se eles estiverem registrados. Outro caso, o usuário pode fechar as guias do navegador ou minimizar o navegador, no entanto os *Service Workers* ainda mantêm a conexão, não interrompendo requisições que virão a ser feitas (CECCONI, 2018).

- **Semelhante a aplicativos:** Nas aplicações que rodam em dispositivos móveis, os aplicativos PWA devem oferecer ao usuário mecanismos de interação semelhantes a um aplicativo nativo. Para oferecer essa experiência ao usuário, é usado a arquitetura de *shell* (LEPAGE, 2019).

A arquitetura *shell* se baseia em arquivos HTML, CSS e JavaScript que são indispensáveis para uma resposta rápida ao usuário. Ao efetuar o primeiro acesso, esses arquivos são armazenados em cache local para que nos acessos posteriores, o tempo entre o acesso e a apresentação seja o menor possível, mesmo que a aplicação esteja offline. Assim, essa arquitetura separa a interface do usuário dos dados que a aplicação carregará para a mesma quando necessário (LEPAGE, 2019).

- **Atualizações constantes:** Na prática, se resume em usar as funções disponibilizadas pelos *Service Workers* para oferecer aos usuários versões mais atuais da aplicação, mesmo que o usuário não faça uma ação em relação a isso.

Terá uma versão da aplicação armazenada em cache e quando for disponibilizada uma versão mais atual, será responsabilidade do *Service Workers* administrar e mostrá-la para o usuário. Caso o usuário esteja fazendo uso da aplicação no instante que a atualização for ofertada, a versão mais recente ficará aguardando um momento para ser ativada. Durante o período em que o usuário estiver usando a aplicação, este estará ainda acessando a versão antiga. Em suma, os *Service Workers* têm capacidade de rodar várias versões diferentes ao mesmo tempo. Para que o usuário possa acessar a versão mais atualizada da aplicação, a guia ou o browser devem ser fechados, assim a versão desatualizada será descartada e a versão atualizada poderá ser acessada (CECCONI, 2018).

- **Segurança:** Para que seja possível efetuar registros e usar um *Service Workers*, completo ou só algumas funcionalidades, é necessário que a aplicação possua um certificado de segurança HTTPS. O HTTPS é usado para determinar uma conexão que seja segura a partir do navegador até um servidor e o seu objetivo é criptografar os dados enviados, impedindo o roubo ou a alteração desses dados (CECCONI, 2018).

- **Fáceis de encontrar e instaláveis:** Ambas as características possuem semelhanças em comum, por esta razão podem ser reunidas. A partir da criação de um *WEB App Manifest*, podemos definir semelhanças entre essas duas características. O manifesto de aplicativo WEB é um arquivo JSON, onde são estabelecidas as informações referentes a aplicação (GAUNT; KINLAN, 2019).

Gault e Kinlan ressaltam que, nesse arquivo deve ter dados que façam os mecanismos de busca e os dispositivos reconhecerem o site WEB como uma PWA. Ao criar esse manifesto, ele possibilita ao desenvolvedor:

- Escolher um ícone e um nome para aplicação;
 - Possibilitar a adição na aplicação, na tela inicial do dispositivo, como se estivesse instalado;
 - Administrar as cores, disposição da tela e a maneira que a aplicação deve se comportar ao ser iniciada pelo usuário.
-
- **Engajados:** O envio de notificações para usuários é outra funcionalidade oferecida pelos *Service Workers*, com o intuito de manter o usuário utilizando a aplicação. As notificações são geradas na própria aplicação local a qual deve estar em execução, ou enviadas por meio de um servidor, caso o usuário estiver com a aplicação fechada (GAUNT, 2019).
-
- **Linkáveis:** Da mesma forma que WEBSITE, as PWAs também são executadas em cima dos navegadores, seja de smartphones ou desktop, criando URLs. Para cada uma das páginas do aplicativo é necessário ter uma URL diferente, podendo ser acessada e compartilhada por qualquer um que tenha conexão com a Internet (CECCONI, 2018).

2.2 - Firebase

Segundo Werlton (2018), o Firebase é uma plataforma de desenvolvimento da Google para desenvolvedores iniciantes e profissionais, responsável por conservar e sincronizar informações em tempo real. Disponibiliza uma gama de soluções de desenvolvimento, como o *Cloud Storage*, que é um banco de dados não

relacional; o *Authentication*, que é usado para autenticar contas em aplicativos e o *Hosting*, que possibilita o desenvolvedor hospedar sua aplicação na plataforma.

Para utilizar o Firebase o usuário deve ter um projeto criado para conectá-lo à plataforma. Após a criação, o usuário será informado sobre configurações, arquivos e objetos que serão adicionados ao projeto local, os quais ele dará ou não permissão (FIREBASE, 2018).

As vantagens que o Firebase oferece são:

- A construção do *back-end* é simples, rápida e econômica;
- A hospedagem da aplicação pode ser feita através da conta que o usuário tem no Google;
- A integração entre os serviços é de fácil entendimento para o desenvolvedor;

As desvantagens que a plataforma apresenta são:

- O usuário deve confiar os dados utilizados pela aplicação à empresa que oferece o serviço, nesse caso Google LLC (Google Inc);
- Algumas das funções oferecidas pela plataforma pode sofrer limitações;
- O usuário depende dos serviços oferecidos pela empresa, caso a mesma descontinue os serviços, o usuário deve migrar a aplicação.

2.2.1 Realtime Database

Segundo Werliton (2018), o Realtime Database é um banco de dados NoSQL que possui hospedagem na WEB. As informações armazenadas são em formato JSON (JSON, 2018) e sincronizadas instantaneamente com todos os usuários que estiverem conectados. Além de estar disponível para uso caso não haja conexão por meio de em cache local, como o cache do navegador.

Esse banco de dados oferece regras de segurança para designar a estrutura dos dados armazenados e quais dados podem ser lidos e gravados. Através do *Firebase Authentication*, o programador define quem pode acessar os dados armazenados, quais dados serão acessados e a forma de acesso (FIREBASE, 2018).

2.2.2 Firebase Cloud Storage

O *Firestore* é um serviço para armazenamento de dados simples, prático e econômico. Utilizando os SDKs de desenvolvimento do Firebase, o upload e download de arquivos em aplicativos é feito por meio da segurança do Google, não importando a qualidade da rede. Os SDKs são usados para armazenar imagens, vídeos e áudios gerados pelo usuário. Para ter acesso a esses arquivos, é utilizado o *Google Cloud Storage* no lado do servidor (FIREBASE, 2018).

Os arquivos gerados pelo usuário são armazenados no Google Cloud Storage e acessados através do Firebase e do Cloud Storage. Utilizando SDKs do Firebase o upload e download dos arquivos podem ser feitos a partir de usuários com dispositivos móveis, tornando mais flexível a transferência dos arquivos. O Google Cloud Platform fornece processos para filtragem de imagens e conversão de vídeos. O Cloud Storage possui regras de segurança as quais permitem controlar o acesso a arquivos individuais ou em grupo. Desta forma, o desenvolvedor pode deixar os arquivos de forma pública ou privada (FIREBASE, 2018).

2.3 - Ionic Framework

Desenvolvido com finalidade de auxiliar os desenvolvedores de aplicativos móveis. O Ionic é um framework de código aberto para desenvolvimento de aplicativos híbridos de alta qualidade e desempenho, fazendo uso de tecnologias já conhecidas e utilizadas na WEB como HTML, CSS e JavaScript (IONIC FRAMEWORK, 2019).

Uma aplicação escrita em Ionic tem o propósito de ser acessada independentemente da plataforma do dispositivo móvel, como: IOS nativo, Android, Windows Phone e Desktop, cuja norma é “Write once, run anywhere”, escrevendo apenas uma base de código o app pode operar em qualquer plataforma. Para que isso ocorra, é usado o Angular para a organização da aplicação e através de plugins o Apache Cordova pode acessar o hardware do dispositivo (Fernandes, 2019).

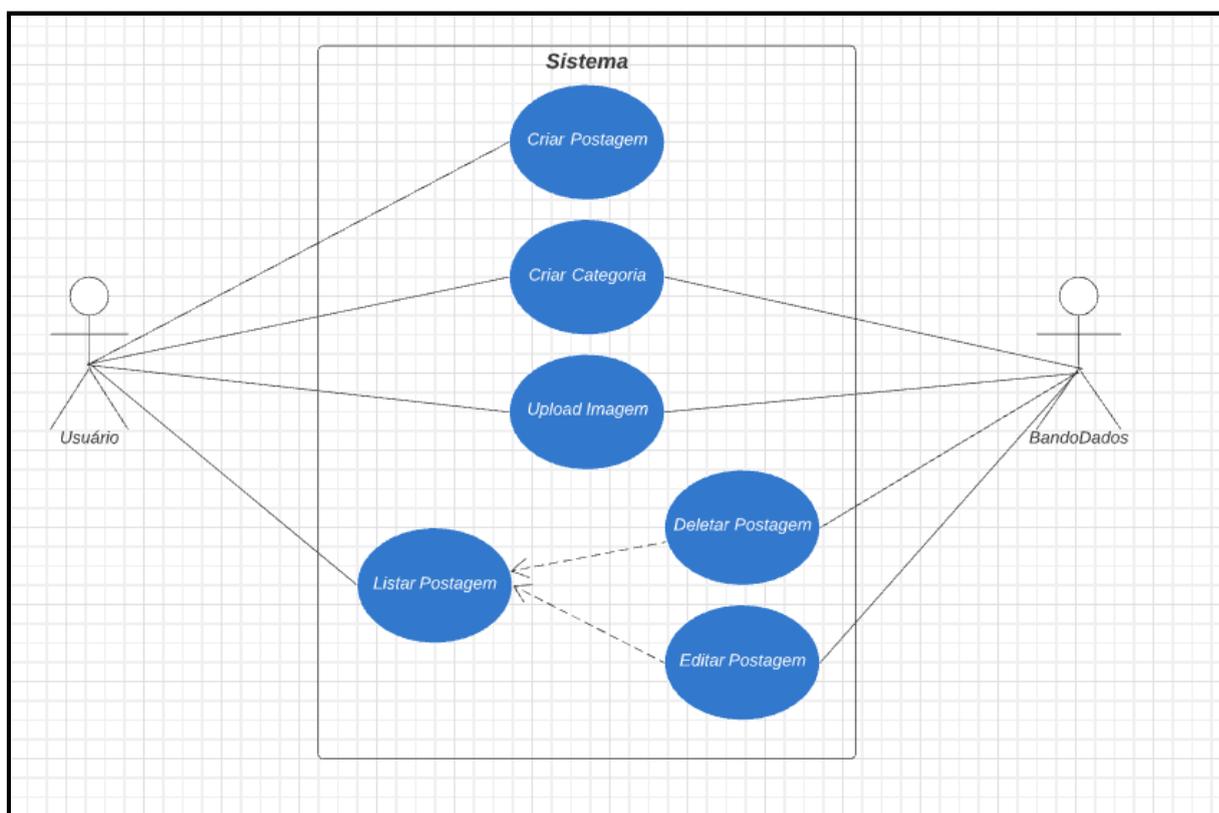
3 O aplicativo ForumOnline

Nesta seção serão apresentadas as etapas do desenvolvimento do aplicativo ForumOnline, que tem por objetivo ser um ambiente de estudo sobre a integração entre serviços da plataforma Firebase e sistemas PWA.

O principal recurso utilizado para o desenvolvimento do App foi a biblioteca *AngularFirestore*, da qual os métodos *valueChanges*, que obtém o valor mais recente do campo usado para construção dos formulários e *snapshotChanges*, que tem função semelhante ao *valueChanges*, porém é utilizado para exclusão e alteração de informações. A integração ocorre através da credencial gerada quando é criado um projeto no Firebase, ela pode ser obtida em “visão geral do projeto”, no ícone “WEB” e em “registrar aplicativo”.

A Figura 1 apresenta os casos de uso dessa aplicação. A seguir uma descrição destes casos de uso.

Figura 1: Diagrama de caso de uso da aplicação



Fonte: Autor, 2021

Caso ListarPostagem: quando o usuário acessar essa tela, o Firebase envia para o aplicativo todas as postagens armazenadas. A tela mostra ao usuário todas as postagens informando o título e a categoria postagem.

Caso CriarCategoria: quando o usuário acessar essa tela, ele informa o título da categoria, o front-end armazena essa informação no objeto ‘categoria’ e envia o

para o back-end, que executa a método *addCategoria* para gravar essa categoria no firebase.

Caso CriarPostagem: quando o usuário acessar essa tela, ele informa o título, categoria e descrição da postagem. o Front-end armazena essas informações no objeto 'postagem', que executa um método *addPostagem* para gravar essa categoria no Firebase.

Caso DeletarPostagem: quando o usuário acessar a tela inicial, ele deve mover a postagem para o lado esquerdo, assim aparecerá um botão que executa a método *deletePostagem*, excluindo essa postagem do firebase.

Caso EditarPostagem: quando o usuário acessar a tela inicial, ele deve clicar em uma postagem, assim conseguindo editar as informações da postagem. Ao salvar as informações será executado o método *savePostagem*.

UploadImagem: nesse caso de uso o usuário faz o upload de uma imagem armazenada na memória de seu dispositivo. Ao acessar o usuário deve clicar em 'upload file', abrindo uma nova janela para selecionar a imagem que será salva. Ao selecionar a imagem será chamada a função *uploadFile* que criará uma coleção para armazenar as imagens salvas, caso a coleção não exista. Após isso, será chamada a método *addImagemtoDB* responsável por gravar as imagens no Firebase.

3.1 Página principal

Nesta tela, o usuário pode visualizar todas as postagens cadastradas no aplicativo. A página principal é apresentada na Figura 2.

Figura 2: Tela inicial do aplicativo ForumOnline.



Fonte: Autor, 2021

Ao selecionar uma postagem e arrastá-la para o lado esquerdo, o usuário tem a opção de excluir a postagem selecionada. A Figura 3 demonstra esse cenário.

Figura 3: Tela com ação de remover a postagem



Fonte: Autor, 2021

Ao selecionar uma postagem e clicar nela, o usuário será redirecionado à tela de cadastro de postagens como diferencial, que pode alterar as informações da postagem. A Figura 4 demonstra esse cenário.

Figura 4: Tela onde o usuário pode editar a postagem.



← Postagens

Titulo:lol2

Informe a categoria: ▼

descricao
lol2

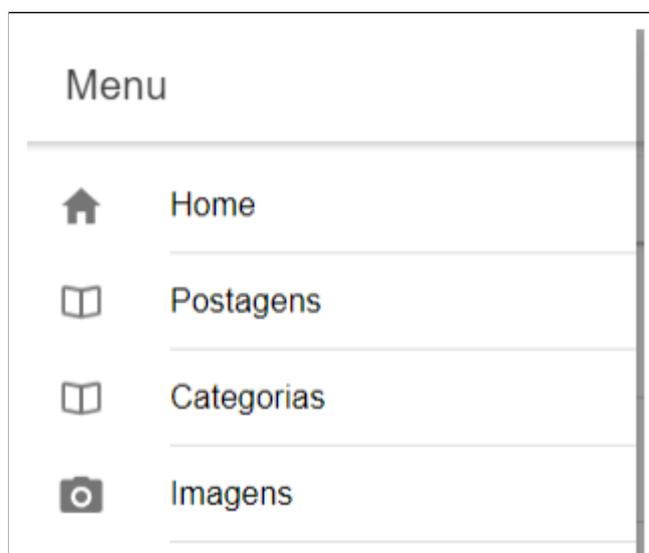
SALVAR

Fonte: Autor, 2021

3.2 Página de postagem

O usuário pode, por meio do menu lateral (canto superior esquerdo), acessar a página de cadastro das postagens, chamada Postagens. Para cadastrar uma postagem é necessário informar o título, a categoria através de uma barra de seleção e a descrição, tendo toda a informação a respeito da postagem. A Figura 5 demonstra esse cenário.

Figura 5: Menu lateral



Fonte: Autor, 2021

Após preencher todos os campos, clique no botão “salvar” que chamará a função *savePostagem* que por sua vez chamará o método *addPostagem*, responsável por gravar as informações da postagem em uma nova coleção no banco de dados. A Figura 6 demonstra esse cenário.

Figura 6: Tela de cadastro de postagens

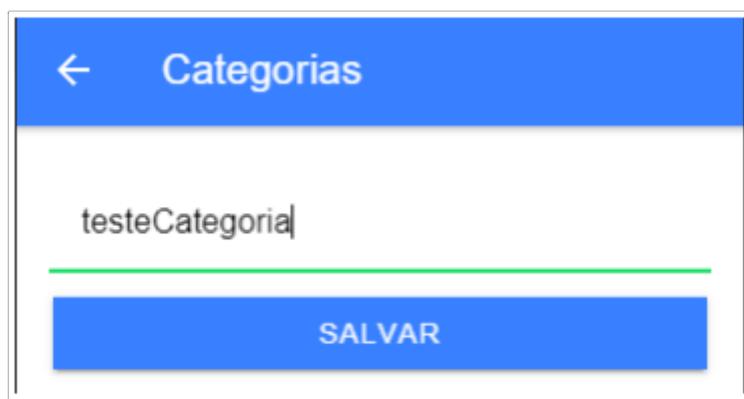
A screenshot of the "Postagens" registration screen. The screen has a blue header with a back arrow and the title "Postagens". Below the header, there are four input fields: "Titulo: banana", "Informe a categoria: comida" (with a dropdown arrow), "descricao" (with the text "banana boa" entered), and a "SALVAR" button at the bottom.

Fonte: Autor, 2021

3.3 Página de categorias

O usuário, por meio do menu lateral (Figura 5), pode acessar a página de cadastro de categorias. De modo semelhante a página de cadastro de postagens, é necessário apenas informar o nome da categoria. A Figura 7 apresenta esta tela.

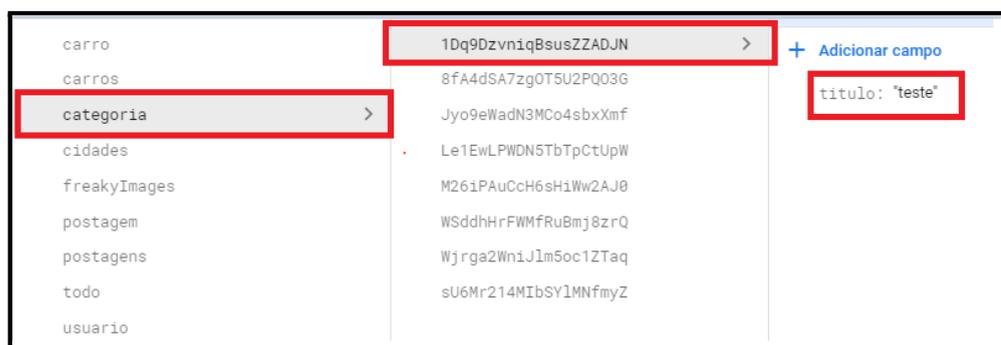
Figura 7: Tela de cadastro de categorias



Fonte: Autor, 2021

Para gravar a categoria no banco de dados, clique no botão “salvar” que chamará a função *saveCategoria* que por sua vez chamará o método *addCategoria*, responsável por gravar a categoria informada no banco de dados. A Figura 8 demonstra esse cenário.

Figura 8: Tela de armazenamento do Firebase



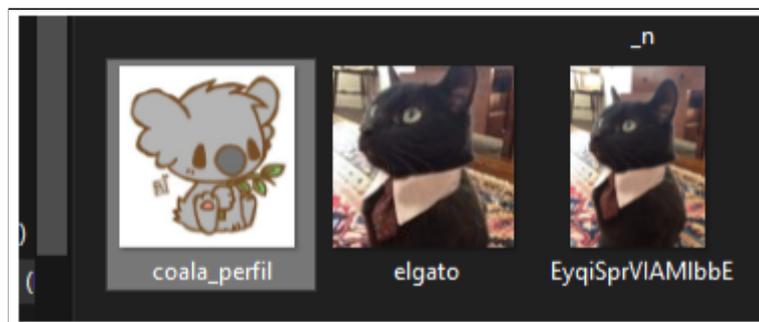
Fonte: Autor, 2021

3.4 Pagina de imagens

O usuário, por meio do menu lateral, pode acessar uma tela para salvar imagens no banco de dados. Ao acessar essa tela, o usuário irá navegar até o botão "escolher arquivo", abrindo uma nova tela para selecionar uma imagem que esteja

armazenada no HD do dispositivo (desktop ou mobile). A Figura 9 demonstra esse cenário.

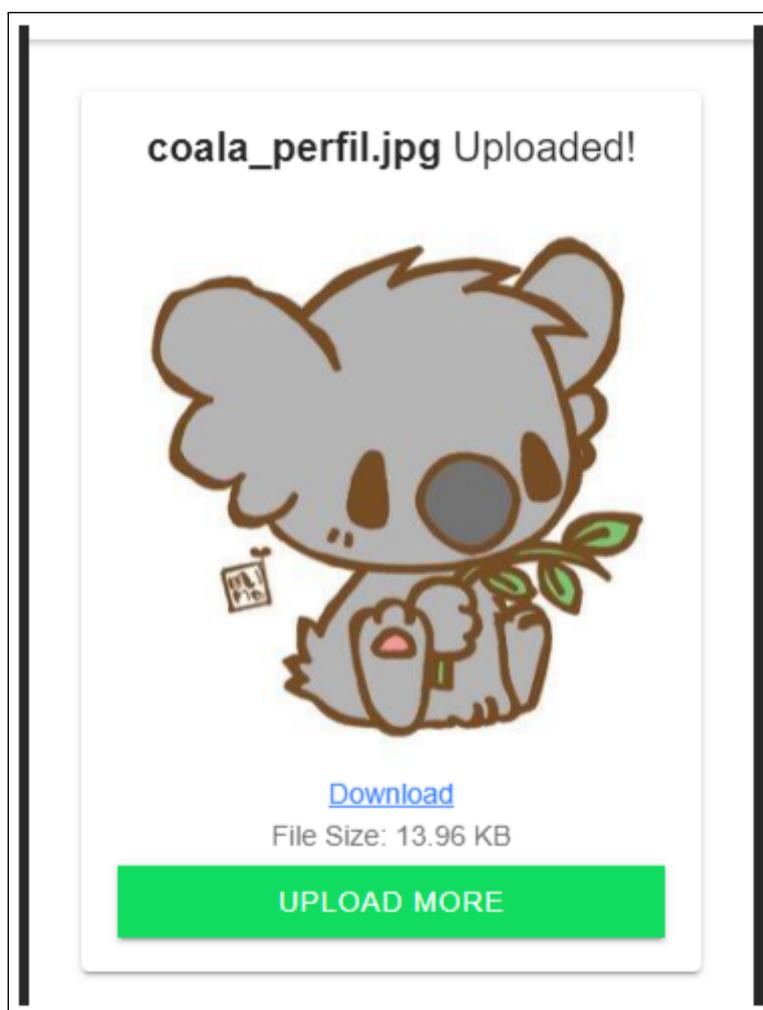
Figura 9: Tela de arquivos do usuário



Fonte: Autor, 2021

Ao selecionar a imagem que será salva no banco de dados, deve clicar em “upload more” chamando a função *addImageDB*, responsável por gravar a imagem no banco de dados, a Figura 10 demonstra esse cenário.

Figura 10: Tela que informa ao usuário que a imagem foi salva



Fonte: Autor, 2021

4 CONSIDERAÇÕES FINAIS

Com o desenvolvimento deste trabalho percebeu-se que as Progressive WEB Application, além de ser uma nova metodologia para o desenvolvimento de softwares, é uma alternativa eficiente e de fácil aprendizado, o Ionic mostrou-se muito eficaz, por ser semelhante ao javascript, ele facilitou o desenvolvimento dessa tecnologia.

A respeito do Firebase, ele facilitou e agilizou o processo de desenvolvimento do aplicativo, pois as configurações para integrar o aplicativo a ele são feitas apertando um botão, além de disponibilizar ao desenvolvedor um passo-a-passo didático para fazer a integração.

Como trabalho futuro pretende-se desenvolver o sistema de login dos usuários, via email e senha e/ou redes sociais, incluir uma barra de pesquisa para facilitar a

busca pelas postagens, e o funcionamento offline do cadastro das postagens, categorias e imagens.

ABSTRACT

The actual text shows a study on technologies available for WEB development, in particular the Progressive WEB Application and its application as a case study integrated with the Firebase platform. The motivation to this text art of the initiative that hybrid applications, which is the case of PWA's, are increasingly common day-to-day. Thereby information about their characteristics is presented and the services that work with it. From the study it was found that Progressive WEB Applications bring interesting features to WEB Mobile development, together with the resources of other services, principally, the Firebase platform. As a case study for technology applications the design and development of an online posting system was developed.

Keywords: Progressive WEB Application. Hybrid Applications. FireBase

REFERÊNCIAS

CECONI, Laura. **EXPERIÊNCIA DO USUÁRIO EM PROGRESSIVE WEB APPS**. 2018. Disponível em: <<https://repositorio.ucs.br/xmlui/bitstream/handle/11338/4780/TCC%20Laura%20Ceconi.pdf?sequence=1&isAllowed=y>>. Acesso em: 20 abr. 2021.

FERNANDES, Marcelo Felipe Guarani. **Udyat: um aplicativo híbrido com o intuito de assegurar o usuário no meio em que vive**.

GAUNT, Matt. **Adicionar Notificações Push a um app da WEB**. 2019. Disponível em: <<https://developers.google.com/WEB/fundamentals/codelabs/push-notifications/>>. Acesso em: 20 abr. 2021.

GAUNT, Matt. Introdução aos service workers. 2019. Disponível em: <<https://developers.google.com/WEB/fundamentals/primers/service-workers/?hl=pt-br>>. Acesso em: 20 abr. 2021.

GAUNT, Matt; KINLAN, Paul. **O manifesto do aplicativo WEB**. 2019. Disponível em: <<https://developers.google.com/WEB/fundamentals/WEB-app-manifest>>. Acesso em: 20 abr. 2021.

ILHASOFT. **O que são apps nativos e como eles funcionam**. 2016. Disponível em: <https://www.ilhasoft.com.br/o-que-sao-apps-nativos-e-como-eles-funcionam/>>. Acesso em: 17 abr. 2021

IONIC FRAMEWORK. **Core Concepts**. Disponível em: <<https://ionicframework.com/docs/intro/concepts>>. Acesso em: 20 abr. 2021.

IONIC FRAMEWORK. **What is Ionic Framework?: WEB Standards-based**. 2019. Disponível em: <<https://ionicframework.com/docs/intro>>. Acesso em: 20 abr. 2021.

LEPAGE, Pete. **Your First Progressive WEB App**. 2019. Disponível em: <<https://developers.google.com/WEB/fundamentals/codelabs/your-first-pwapp/>>. Acesso em: 20 abr. 2021.

MARCOTTE, Ethan. **Responsive WEB Design**. 2010. Disponível em: <<https://alistapart.com/article/responsive-WEB-design/>>. Acesso em: 20 abr. 2021.