

Desenvolvimento de uma aplicação PWA para adoção de animais

Deivid Hallan Galiani¹

<https://orcid.org/0009-0009-9822-8305>

Felipe Diniz Dallilo²

<https://orcid.org/0000-0002-0885-0304>

Fabiana Florian³

<https://orcid.org/0000-0002-9341-0417>

RESUMO

A popularização dos smartphones impulsionou o mercado de aplicativos móveis, porém, para que os usuários possam utilizar aplicativos nativos, é necessário que os desenvolvedores os disponibilizem nas plataformas específicas, como a Google Play Store e a App Store. Além disso, é preciso que os dispositivos tenham memória suficiente e os aplicativos sejam atualizados frequentemente. Nesse contexto, os Progressive Web Apps (PWA) surgem como uma alternativa, proporcionando uma experiência semelhante a aplicativos nativos, executáveis em qualquer navegador de dispositivos móveis ou desktop. Os PWAs apresentam uma evolução em relação à web tradicional, incorporando recursos que aproximam suas funcionalidades aos aplicativos nativos. Além disso, oferece facilidade de uso e maior acessibilidade aos usuários, ao mesmo tempo em que reduz as dificuldades de desenvolvedores em disponibilizar seus aplicativos para download. Para explorar os recursos e benefícios do PWA, foi desenvolvido uma aplicação para adoção de animais, permitindo aos usuários acessar e interagir com a plataforma mesmo em locais com baixa conectividade, além de receber atualizações e notificações relevantes em tempo real, mantendo-os atualizados sobre atividades na plataforma.

Palavras-chave

Progressive Web Apps; Offline; Desenvolvimento Multiplataformas; Smartphone.

Submetido em: 07/06/2023 – Aprovado em: 26/06/2023 – Publicado em: 06/07/2023

1 Graduando do Curso de Sistemas de Informação da Universidade de Araraquara – UNIARA. Araraquara-SP. E-mail: dhgaliani@uniara.edu.br

2 Orientador. Docente Curso de Sistemas de Informação da Universidade de Araraquara- UNIARA. Araraquara-SP. E-mail: fddallilo@uniara.edu.br

3 Coorientador. Economista e Advogada. Docente Curso de Sistemas de Informação da Universidade de Araraquara- UNIARA. Araraquara-SP. E-mail: fflorian@uniara.edu.br



Development of a PWA application for animal adoption

ABSTRACT

The popularity of smartphones has driven the mobile app market, but for users to be able to use native apps, developers need to make them available on specific platforms such as the Google Play Store and the App Store. Additionally, devices must have sufficient memory, and apps need to be regularly updated. In this context, Progressive Web Apps (PWAs) emerge as an alternative, providing a native-like experience that can be executed in any mobile or desktop browser. PWAs represent an evolution from traditional web applications by incorporating features that bring them closer to native apps. Moreover, they offer ease of use and greater accessibility to users while reducing the challenges faced by developers in making their apps accessible. To explore the features and benefits of PWAs, an animal adoption application was developed, enabling users to access and interact with the platform even in low-connectivity areas. They also receive real-time updates and relevant notifications, keeping them informed about activities on the platform.

Key-words

Progressive Web Apps; Offline; Multiplatform Development; Smartphone.

1 INTRODUÇÃO

Os dispositivos móveis, especialmente os *smartphones*, seguem em ascensão, a Pesquisa Nacional por Amostra de Domicílios (PNAD) apontou que em 2020, 99,2% dos domicílios brasileiros com conexão com a internet, utilizava pelo menos um *smartphone* para este fim (PNAD, 2020, p. 7). Com essa popularização, diversas empresas passaram a almejar suas próprias aplicações móveis nativas, ou seja, aplicativos que são desenvolvidos para sistema operacional específico, como *Android* e *iOS*, tais aplicativos propõem facilidade na rotina dos usuários, seja de forma interativa nas redes sociais, ou para obter acesso a notícias, pedir uma refeição, entre outras funcionalidades (FORTUNATO e BERNARDINO, 2018).

Os aplicativos nativos oferecem acesso total aos recursos do dispositivo, como câmera, GPS e acelerômetro, proporcionando uma experiência do usuário excelente em termos de interface, desempenho e confiabilidade. No entanto, aplicativos nativos são dependentes de plataforma e são construídos usando linguagens de programação específicas, como Java para *Android*, C# para *Windows* e Swift para *iOS*, esses aplicativos podem ser baixados e instalados somente por meio de lojas de aplicativos específicas, como *App Store* da Apple ou o *Google Play*, e não podem ser implantados em várias plataformas, pois não há reaproveitamento de código, sendo necessário desenvolver aplicativos diferentes para cada tipo de dispositivo (THAKUR, 2018).

Além do tempo que as empresas levam para desenvolverem e publicarem nas lojas de aplicativos, ainda há o risco da baixa aceitação dos usuários, muitas vezes evitam ocupar a memória de seus *smartphones* com vários aplicativos (ANTUNES, 2019). A revista *Panorama Mobile Time/Opinion Box* (PAIVA, 2021) entrevistou 1.986 usuários de *smartphones*, observando que 39% dos entrevistados instalaram um aplicativo nas últimas 24 horas e 35% disseram que no mesmo período de tempo desinstalaram o aplicativo.

Nesse contexto, o PWA surge como um recurso inovador, uma vez que não depende das lojas oficiais, como a *Google Play Store* e *App Store* e nem do sistema do dispositivo móvel (TRINDADE e AFFINI, 2018). E ainda, tem sua usabilidade semelhante a um aplicativo nativo utilizado em *smartphone*, *tablet*, *laptop* ou *desktop*, mas com um desenvolvimento mais simples a partir de alguns recursos, como o HTML, CSS e Javascript, trazendo menores manutenções, baixo custo e tempo reduzido (LEPAGE, 2020).

O PWA tem se mostrado benéfico para os desenvolvedores e empresas por combinar as funções e experiências de aplicativos nativos com as páginas da *web*, permitindo dessa forma, que os usuários instalem a aplicação no dispositivo e ainda tenham acesso a todos os recursos oferecidos, trazendo mais rapidez e eficiência aos usuários (LEPAGE, 2018).

O presente trabalho tem o objetivo de explorar as características da tecnologia PWA e seus benefícios aos usuários da *web*, além do desenvolvimento de uma aplicação

voltada para a adoção de animais, como cães, gatos e outros animais utilizando as tecnologias envolvidas em um PWA, fornecendo aos usuários uma plataforma de manuseio intuitivo e fácil, que permita a adoção responsável de animais de estimação, com informações detalhadas sobre cada animal. Além disso, os usuários podem visualizar anúncios de animais disponíveis para adoção, filtrando por cidade, raça, espécie e sexo. Usuários registrados também podem manter anúncios de seus próprios animais para adoção.

A aplicação contará com recursos de cache, que armazena os arquivos essenciais de seu funcionamento (HTML, JavaScript e CSS), garantindo que os usuários possam acessar e interagir com a plataforma mesmo em locais com baixa conectividade

2 REVISÃO DE LITERATURA

A expressão *Progressive Web App* (PWA) foi discutida pela primeira vez por Russell em 2015, que definiu PWA como “aplicações desenvolvidas por meio de um conjunto combinado de conceitos e tecnologias *web*”. Pontes (2018) aponta que as tecnologias *web* de desenvolvimento de um PWA são as mesmas utilizadas na criação de páginas da internet, entre elas estão o HTML, CSS e JavaScript. Conforme Ceconi (2018), tais tecnologias são executadas nos navegadores de dispositivos *mobiles*, apresentando todos os recursos que um navegador dispõe, como envio de notificações, instalação e o funcionamento *offline*. Além da funcionalidade, a aparência também se torna semelhantes aos aplicativos nativos, de forma é que se torna possível instalar o PWA e ter um ícone na tela inicial do dispositivo móvel, facilitando à busca e o acesso à aplicação (TRINDADE e AFFINI, 2018).

Os aplicativos móveis nativos para Android ou Iphone, apresentam diversas funcionalidades, tais como a capacidade de enviar notificações por push, funcionar offline e serem carregados na tela inicial do dispositivo móvel. Em contraste, historicamente, os aplicativos da web móveis acessados por meio de um navegador móvel não ofereciam essas funcionalidades. No entanto, com o surgimento dos aplicativos da web progressivos, esses recursos foram adicionados por meio da implementação de novas APIs da web, conceitos de design atualizados e novos termos e definições específicos da web progressiva (TANDEL, 2018).

Segundo Richard e LePage (2020), os PWA são aplicações *web* que sofreram um aprimoramento progressivo, com novos recursos ativados nos navegadores, usando o *service workers* e o *manifest file*, a aplicação se torna confiável podendo ser instalada por qualquer indivíduo em todo dispositivo multiplataforma com uma única base de código.

2.1 Características Do Pwa

Conforme Lepage (2018), para que o PWA seja eficiente ele deve apresentar algumas características:

1. Progressivo – qualquer usuário pode aplicá-lo em seu navegador de preferência, mesmo em versões mais antigas que não oferecem suporte para instalação de PWAs;
2. Responsivo – é suportado por qualquer dispositivo e se adapta a diferentes tamanhos de tela, por exemplo, desktop, smartphone e tablet;
3. Funcionamento offline: parte do conteúdo pode ficar disponível mesmo sem conexão com a internet;
4. Semelhante a aplicativos nativos - layout funcional com interação e navegação no mesmo estilo dos aplicativos;
5. Atual – com service worker a aplicação se mantém atualizada, sem a necessidade de ação do usuário
6. Seguro – utiliza HTTPS para garantir a segurança das informações transmitidas entre o usuário e a aplicação;
7. Descobrível - Podem ser indexadas por ferramentas de busca;
8. Envolvimento – proporciona o engajamento por meio de ferramentas como as notificações push;
9. Instalável - concede a opção de "salvar" os aplicativos na tela inicial dos dispositivos sem precisar acessar a loja de aplicativos;
10. Linkável – pode ser compartilhado facilmente via URL.

Para Fortunato e Bernardino (2018), uma das principais vantagens do PWA, é centralizar todo o processo de desenvolvimento em uma única aplicação que atenda todos os dispositivos, sem a necessidade de desenvolver um aplicativo para cada tipo de dispositivo móvel (*Android, iOS, Windows Phone*), obtendo dessa forma, um menor custo final de desenvolvimento.

Reshetilo e Opanasenko (2017) apontam para três vantagens dos PWAs: A primeira também diz respeito ao custo e tempo de desenvolvimento, pois não há necessidade de várias equipes para o desenvolvimento de aplicativos. A segunda está relacionada ao processo de aquisição de um novo usuário devido à facilidade de navegação e de instalação que é sugerida quando o usuário estiver navegando no site naturalmente. Uma vez aceito, o processo de instalação começa sem sair da página ou visitar a loja de aplicativos, o aplicativo é instalado em segundos com um ícone acessível na tela inicial.

E a terceira vantagem do PWA é que quando o aplicativo precisa ser melhorado, seja utilizando um recurso ou adicionando um novo, as versões são atualizadas automaticamente, assim que o usuário estiver *online*. Isso se deve ao processo de

desenvolvimento do PWA que é centralizado no mesmo código, o que afeta tanto o site, quanto todas as plataformas que o PWA usa (RESHETILO e OPANASENKO, 2017).

2.2 Tecnologias Envolvidas No PWA

Com a finalidade de compreender as características do PWA, é necessário considerar alguns fatores. O desenvolvimento de PWA é possível por meio de diferentes tecnologias, o presente estudo irá abordar algumas delas.

Service workers: são módulos de *software* escritos em *JavaScript* que é executado em segundo plano, fornecendo algumas funcionalidades, por exemplo, verifica o *status* da rede de dados, verifica se há versões mais recentes dos arquivos, permite a disponibilização *offline*, fornece notificações *push* e permite acesso a geolocalização por meio do GPS do dispositivo (GRONER, 2018).

Manifest file: é um arquivo no formato JSON, com informações sobre a aplicação PWA, como o ícone de abertura da aplicação, título, cor de tema e cor da barra de notificação do dispositivo (MELO 2020).

HTML: HyperText Markup Language é uma linguagem de marcação que informa a estrutura dos documentos e auxilia na construção de páginas da *web* usando recursos de hipermídia que possibilitam que os usuários realizem interações (BORTOLOSSI, 2012).

HTTPS: *Hyper Text Transfer Protocol Secure* é um protocolo de rede que criptografa os dados solicitados, garantindo maior segurança aos usuários (PONTES, 2018).

JavaScript: é uma linguagem de programação que quando combinado com o HTML, pode realizar diversas funções, como processamento de dados, alteração e criação de elementos HTML (imagem ao clicar, realizar funções matemáticas e criar caixas de diálogo), todo o processamento ocorre diretamente sem troca de informações com o servidor (FORTUNATO, 2015).

CSS: *Cascade Style Sheets* é um recurso para controlar o estilo das páginas HTML, atribuindo cores e formato aos elementos do site, todas as ações podem ser modificadas através do JavaScript (GRILLO e FORTES, 2008).

API: *Application Programming Interface* são interfaces que fornecem definições e protocolos para gestão e armazenamento de dados em bancos de dados. É possível usar qualquer linguagem backend para criar uma API que permita aos usuários acessar apenas o código HTML, proporcionando maior segurança para o sistema. Isso ocorre porque os usuários podem interagir com o servidor sem expor o código-fonte e se conectar a vários bancos de dados.

3 DESENVOLVIMENTO DO PROJETO

Para o desenvolvimento da aplicação web para adoção de animais de estimação, utilizou-se a tecnologia PWA em conjunto com o framework Angular 14 e a biblioteca angular-pwa. O objetivo é de oferecer uma experiência ao usuário similar à de uma aplicação nativa.

Para o backend, foi criada uma API em NodeJS com o framework Adonis 5, que se comunica com um banco de dados relacional PostgreSQL. Para permitir o upload e visualização de imagens dos anúncios pelos usuários do sistema, foi utilizado o Amazon S3 Bucket. Dessa forma, as imagens são armazenadas em um servidor externo.

Para melhorar a interação com os usuários, foram adicionadas notificações push, utilizando a API swPush do Angular-pwa. Dessa forma, os usuários inscritos em uma determinada cidade receberão notificações em segundo plano quando novos animais forem disponibilizados para adoção em sua cidade. Essa funcionalidade visa aumentar a visibilidade dos animais disponíveis para adoção e aumentar o engajamento dos usuários.

O desenvolvimento do sistema seguiu as seguintes etapas:

Análise: nesta etapa foram definidos os requisitos do sistema, bem como os casos de uso.

Projeto: nesta etapa foi definida a arquitetura do sistema, bem como os módulos, componentes e serviços necessários para implementar as funcionalidades requeridas.

Implementação: nesta etapa foi realizada a codificação do sistema utilizando o framework Angular e a biblioteca RxJS, e um servidor backend utilizando o framework Adonis 5 em Node.js, se comunicando com um banco de dados PostgreSQL.

3.1 Ferramentas

As ferramentas utilizadas para o desenvolvimento do sistema foram:

Visual Studio Code: editor de código-fonte com suporte para TypeScript, HTML, CSS e outras linguagens.

Node.js: ambiente de execução JavaScript que permite executar aplicações no lado do servidor.

NPM: gerenciador de pacotes JavaScript que permite instalar e gerenciar dependências externas.

Angular CLI: interface de linha de comando que permite gerar, testar e implantar aplicações Angular.

Amazon S3: serviço de armazenamento de objetos oferecido pela Amazon Web Services (AWS), projetado para armazenar e recuperar grandes quantidades de dados, como imagens, vídeos, arquivos de áudio e documentos.

PostgreSQL: sistema gerenciador de banco de dados relacional de código aberto

3.2 Escopo

A partir da análise das necessidades do sistema, foram levantados os requisitos funcionais do projeto apresentado no Quadro 1.

Quadro 1 – Requisitos Funcionais

RF1	Permitir que os usuários se cadastrem no sistema
RF2	Permitir que os usuários criem anúncios de animais para adoção
RF3	Permitir que usuários não registrados visualizem os anúncios disponíveis
RF4	Permitir que os usuários filtrem anúncios por cidade e características dos animais
RF5	Enviar notificações push para usuários inscritos em determinada cidade
RF6	Armazenar imagens dos animais em um servidor externo
RF7	Comunicar com um banco de dados relacional PostgreSQL
RF8	Oferecer uma experiência ao usuário similar à de uma aplicação nativa
RF9	O sistema deve ser acessível em diferentes dispositivos e navegadores
RF10	O sistema deve proteger as informações pessoais dos usuários e seus dados de acesso

Fonte: Elaborado pelo autor

3.3 Passos Do Desenvolvimento

Para a criação do projeto PWA em angular, foram definidas as seguintes etapas:

- Criação do projeto utilizando a ferramenta Angular CLI
- Adição da biblioteca angular-pwa, que fornece à aplicação um Service Worker básico configurado e gerado os arquivos **ngsw-config.json** e o arquivo **manifest.json**.
- Configuração do arquivo **ngsw-config.json**, que contém instruções de como o service worker deve armazenar em cache e atualizar ativos, lidar com solicitações de navegação e responder a notificações push.
- Configuração do arquivo **manifest.json**, que é usado para informar aos navegadores e aos dispositivos sobre as características e recursos da aplicação web. Ele define informações como ícone do aplicativo, cor de tema, nome da aplicação, entre outros dados importantes da aplicação.

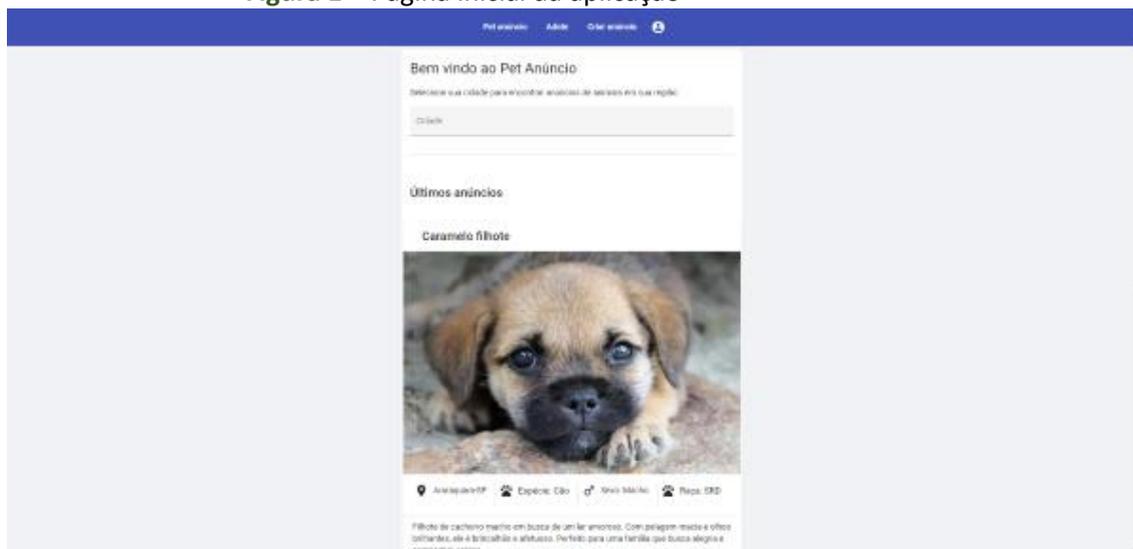
4 FUNCIONAMENTO DA APLICAÇÃO

A aplicação PWA para adoção de animais oferece uma ampla gama de funcionalidades que têm como objetivo facilitar o processo de adoção. Por meio dessa aplicação, os usuários podem explorar diversas funcionalidades, incluindo a visualização de animais disponíveis para adoção, bem como acessar detalhes completos sobre cada um deles, tais como idade, raça, sexo e fotos dos animais. Com uma interface intuitiva e de fácil utilização, a aplicação proporciona uma experiência completa e funcional, assegurando que o processo de adoção seja eficiente e gratificante para todos os envolvidos.

4.1 Página Inicial

A página inicial (Figura 1) é o ponto central da aplicação, onde os usuários podem começar a explorar os animais disponíveis para adoção. Ela exibe um campo de busca por cidade, permitindo que os usuários encontrem animais disponíveis em uma área específica. Além disso, a página inicial mostra os últimos anúncios publicados, oferecendo aos usuários uma visão rápida dos animais que estão procurando um lar.

Figura 1 – Página inicial da aplicação

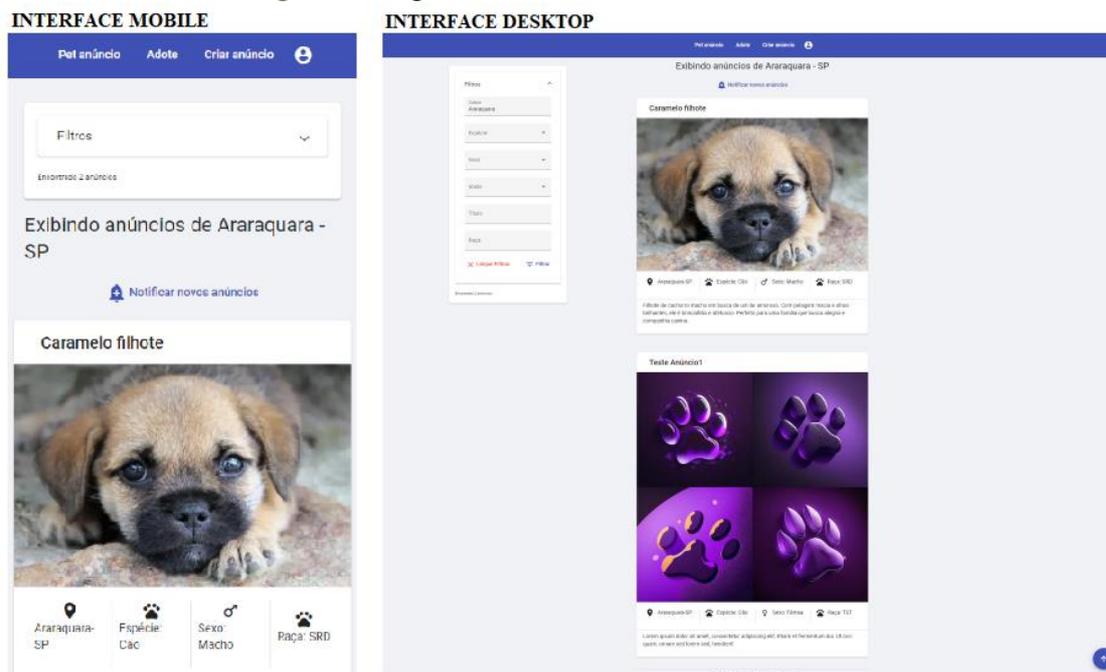


Fonte: Elaborado pelo autor

4.2 Página De Anúncios

A página de anúncios (Figura 2) é a funcionalidade central do sistema, oferecendo aos usuários uma maneira eficiente de encontrar animais disponíveis para adoção. Nessa página, os usuários podem aplicar filtros personalizados, como cidade, idade, sexo e espécie, para refinar sua busca e encontrar animais que atendam às suas preferências. Além disso, a página de anúncios apresenta um recurso adicional: o botão "Notificar Novos Anúncios". Ao utilizar esse botão, os usuários podem optar por receber notificações push sempre que novos anúncios forem publicados, mantendo-se atualizados sobre os animais disponíveis para adoção.

Figura 2 – Página de anúncios



Fonte: Elaborado pelo autor

4.3 Página De Criação Do Anúncio

A funcionalidade "Criar Anúncio" (Figura 3) consiste em um formulário de cadastro que permite aos usuários publicarem anúncios de animais disponíveis para adoção. Esse formulário abrange diversas informações sobre o animal, incluindo título do anúncio, espécie, raça, sexo, idade, descrição, cidade, telefone de contato e imagens do animal. Ao preencher esse formulário, os usuários podem compartilhar informações essenciais sobre os animais. Esses anúncios, uma vez preenchidos e enviados, são publicados na aplicação, permitindo que sejam visualizados por potenciais adotantes interessados.

Figura 3 – Página Criar/Editar anúncio

The image shows a web form titled "Editar anúncio" (Edit ad) with a blue header bar containing navigation links: "Pet anúncio", "Adote", "Criar anúncio", and a user profile icon. The form fields are as follows:

- Título do anúncio ***: Caramelo filhote
- Especie ***: Cão
- Raça ***: SRD
- Sexo ***: Macho
- Idade ***: Filhote
- Descrição ***: Filhote de cachorro macho em busca de um lar amoroso. Com pelagem macia e olhos brilhantes, ele é brincalhão e afetuoso.
- Cidade ***: Araraquara
- Telefone ***: (11)11111-1111

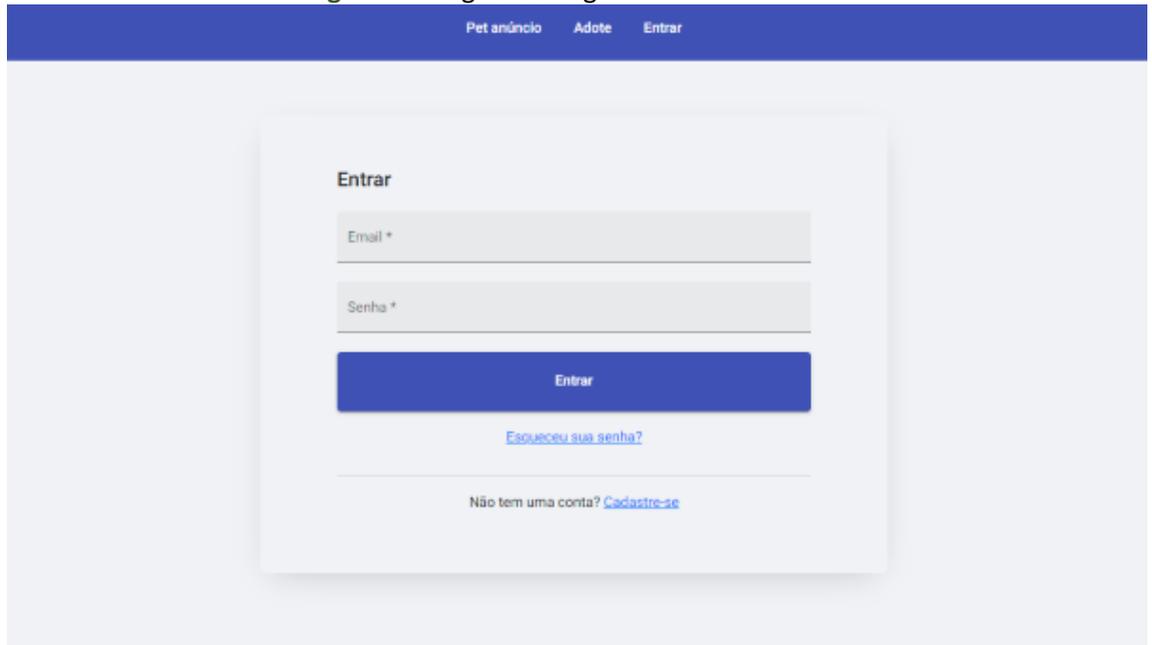
Below the text fields is an image selection section with a blue button labeled "Selecionar imagens". Underneath, there are two image slots, each showing a filename ("1.jpeg" and "1-2.jpeg") and a trash icon for deletion. At the bottom of the form is a large blue "Salvar" (Save) button.

Fonte: Elaborado pelo autor

4.4 Página De Login

A funcionalidade de login (Figura 4) é o ponto de acesso dos usuários ao sistema, onde eles podem autenticar suas credenciais. Nessa funcionalidade, existem campos designados para inserção do email e senha, que garantem a segurança dos dados e possibilitam a identificação individual de cada usuário. Ao realizar o login, os usuários obtêm acesso às funcionalidades restritas do sistema, como a capacidade de criar anúncios e interagir de forma mais completa com a plataforma.

Figura 4 – Página de Login

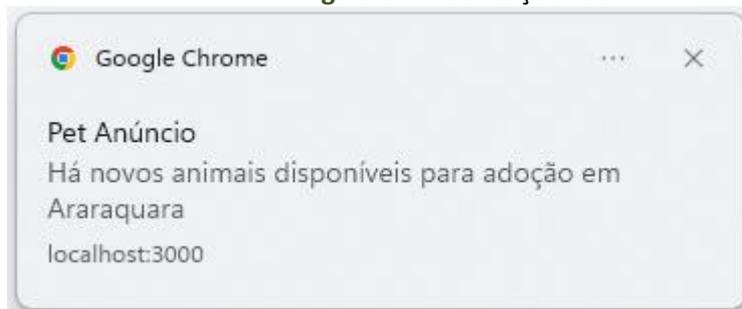


Fonte: Elaborado pelo autor

4.5 Notificações Push

A funcionalidade de notificações push (Figura 5) foi desenvolvida para manter os usuários sempre atualizados sobre novos anúncios na cidade de seu interesse. Os usuários têm a opção de se inscrever para receber notificações push e, dessa forma, são alertados sempre que novos anúncios forem publicados na cidade em que estão interessados, mesmo quando a plataforma não estiver aberta em seu navegador. Essa abordagem proativa garante que os usuários sejam prontamente informados sobre os novos anúncios, aumentando a interação com a plataforma.

Figura 5 – Notificação Push



Fonte: Elaborado pelo autor

5 RESULTADOS E DISCUSSÃO

Algumas plataformas populares já utilizam o PWA, como por exemplo, o Twitter, AliExpress, Uber e Pinterest. Kerssens (2019), a rede social Twitter transformou

o site em um PWA e viu um aumento no engajamento, uma redução no uso de dados e um aumento de 75% nos *tweets* enviados. O varejista *online* AliExpress relata que os usuários passam 74% mais tempo no aplicativo (GOOGLE, 2017). Enquanto o Uber, tempo de carregamento do PWA passou a ser inferior a 3 segundos mesmo em locais com conexão de internet ruim, pois o tamanho da instalação é mínimo (CROLL, 2017).

O site de mídia social Pinterest também decidiu lançar um PWA, e publicou um estudo de caso, a empresa relatou um aumento de 40% no tempo gasto em 5 minutos e um aumento de 44% na receita de anúncios gerada pelo usuário, o estudo inclui que os tamanhos de instalações das aplicações aplicativos PWA são muito menores quando comparadas aos aplicativos nativos (OSMANI, 2017).

Alguns dos sites mais famosos se renderam às vantagens do PWA, como a aplicação tem seu *layout* semelhante ao aplicativo nativo, o usuário não encontra dificuldades em acessar qualquer *menu*, o fato de funcionar com baixa conectividade é um ponto positivo, especialmente no caso do Uber, pois o usuário pode acessar a aplicação em qualquer local que ele esteja e solicitar o automóvel.

Corroborando com essa discussão alguns autores também desenvolveram o PWA, Ceconi (2018), realizou um trabalho em formato de rede social, denominada Wanderlust, podendo ser acessada pelo link: <https://wanderlust-pwa.firebaseio.com/>.

A aplicação tem seu enfoque na área do Turismo e disponibiliza “Diário de Viagens e mapas” baseado em depoimentos e compartilhamentos de amigos ou seguidores, facilitando a busca de pontos turísticos interessantes.

A avaliação ocorreu a partir da ferramenta LightHouse, que de acordo com o Google Developers (2017), é uma ferramenta de auditoria automatizada de averiguação de desempenho de *Websites* que gera relatórios com possíveis falhas e soluções, entre outras verificações. Na aplicação deste trabalho, os resultados apontaram uma média em *desktop* de 94,8 de satisfação, enquanto para *mobile* foi de 95,4% (CECONI, 2018).

Os resultados são positivos, já que a aplicação é capaz de funcionar com baixa conectividade, o que permite que o usuário tenha acesso à essa rede social a qualquer momento e em diferentes locais mesmo em redes móveis. Além disso, essa funcionalidade otimiza o tempo do usuário, pois a aplicação é interativa e possibilita a consulta de recomendações de melhores lugares para viajar, de acordo com o nicho de cada turista.

SANTOS et al. (2021), desenvolveu um trabalho sobre PWA para geolocalização de preços de mercadorias, utilizando o método de *crowdsourcing*. Esse termo se refere à obtenção de trabalho, informações ou opiniões coletivas através do envio de dados pela internet, oferecendo recompensas em dinheiro, descontos, cupons, produto gratuitos, entre outras opções, ou de forma voluntária. O uso do *crowdsourcing* se apresenta como uma alternativa para manter o engajamento dos usuários, incentivando-os a contribuir com informações, avaliações e recomendações de produtos disponíveis na aplicação.

Na metodologia em um primeiro momento foi construído uma API, e posteriormente foi desenvolvido o PWA, as recompensas atuam por meio de categorias evolutivas dos Perfis de usuários, ou seja, os padrões de bonificação vão mudando de acordo com evolução do perfil do usuário que começa em “Iniciante”, os níveis vão subindo conforme o usuário participa das interações. Os autores concluíram que o protótipo do PWA desenvolvido é uma ferramenta capaz de auxiliar o consumidor a economizar, cerca de 100% dos usuários recomendam o aplicativo e 92% utilizam a internet como principal meio de pesquisa de preço.

Santos et al. (2020) traz uma proposta inovadora, que difere das aplicações de supermercados já disponíveis, as interações auxiliam os usuários a encontrarem os melhores produtos, além de outras informações, tornando a proposta mais atrativa, e assim retém mais os usuários. Nesse caso, o PWA se mostra mais uma vez uma ótima opção, devido ao seu baixo custo os supermercados menores também podem ter acesso a essa aplicação, sem precisar pagar ou esperar para que sua aplicação seja disponibilizada em alguma loja de aplicativos, e assim atrair mais clientes.

Carvalho (2019) desenvolveu um protótipo de PWA de Boletim Cadastral Imobiliário (BCI) tanto para novos registros, quanto para renovação de registros de imóveis existentes na Cidade de Monte Carmelo - MG. O site pode ser visualizado no domínio: <https://mcboletim.orgfree.com/>. Para a construção do PWA, primeiramente foi estruturado o banco de dados, a partir de dados coletados junto aos proprietários e informações do imóvel (endereço completo) e da construção (água e esgoto, pintura, instalações elétricas).

Posteriormente, são criadas páginas HTML e PHP que possuem caixa de consulta e de login, páginas privadas para o administrador com mapa da localização do imóvel, e uma página com campo de cadastro para calcular o UPTU com base nas respostas coletadas.

Para o teste qualitativo, foram formuladas cinco perguntas (1. A proposta do Site/PWA foi útil?; 2. O Site/PWA está claro e objetivo?; 3. Utilizaria o site/PWA?; 4. Você já conhecia o PWA?; 5. Em que o PWA contribui para o melhoramento do Site?) que foram respondidas por oito voluntários, sendo dois funcionários da prefeitura, um funcionário de uma empresa de agrimensura e cinco usuários comuns.

O autor conclui que o PWA alcança o seu objetivo, pois os usuários se mostraram bastante satisfeitos. O PWA por ser acessível pode ser desenvolvido para qualquer cidade, desde as maiores até as menores, trazendo grandes facilidades para as prefeituras coletarem os dados e proporcionando mais clareza em seus cálculos, além de favorecer para que as atualizações sejam realizadas, com a própria contribuição dos usuários, que também são beneficiados, pois a aplicação contribui para pesquisas rápidas, até mesmo em imóveis que ficam em áreas com baixa cobertura de internet.

6 CONCLUSÃO

O PWA é uma proposta que combina recursos diversos, permitindo que seja acessado por qualquer dispositivo móvel, como smartphones, que são amplamente utilizados atualmente. Essa é uma vantagem significativa, já que grande parte da população possui um smartphone, porém nem todos têm acesso à internet de alta qualidade. O PWA consegue contornar esses obstáculos, funcionando com baixa conectividade, o que possibilita que os usuários continuem utilizando a aplicação mesmo em ambientes com restrições de conexão. Isso contribui para uma maior inclusão e alcance dos usuários.

Outro ponto importante é que o PWA oferece menor burocracia e desenvolvimento mais aberto em comparação às aplicações nativas. Essa abordagem resulta em menores custos de desenvolvimento, facilitando a distribuição da aplicação. Além disso, o PWA é projetado para reduzir os problemas de atualizações constantes, processamento e uso de recursos de armazenamento, fatores que muitas vezes levam os usuários a desinstalar os aplicativos nativos. Com o PWA, essas questões são mitigadas, proporcionando uma experiência mais leve, eficiente e independente de conexão.

Considerando os benefícios das tecnologias envolvidas no PWA, foi possível desenvolver uma aplicação para adoção de animais, acessível aos usuários independentemente do dispositivo utilizado e dos níveis de conectividade disponíveis. Através da utilização dos service workers, os usuários desfrutam de uma experiência similar à de um aplicativo nativo. Isso significa que a aplicação pode ser instalada na tela inicial do dispositivo, permitindo um acesso rápido e direto, além de funcionar offline ou em ambientes com baixa conectividade. Dessa forma, os usuários têm a comodidade de utilizar a aplicação de forma fluida e eficiente, aproveitando todos os recursos oferecidos para promover a adoção de animais.

Além disso, as notificações push desempenham um papel importante nesse contexto, permitindo que os usuários sejam prontamente informados sobre novos anúncios de animais, mesmo quando não estão utilizando a plataforma. Isso aumenta a interação e a participação ativa dos usuários na busca por um animal de estimação para adoção.

A combinação de funcionalidades offline e notificações push torna o desenvolvimento de uma aplicação PWA uma escolha altamente vantajosa para alcançar um público mais amplo e promover a adoção responsável de animais. Essa abordagem proporciona uma solução econômica, pois não é necessário desenvolver e publicar a aplicação em várias plataformas diferentes. Com o PWA, é possível alcançar uma ampla disseminação da aplicação, uma vez que ela pode ser acessada diretamente através do navegador, eliminando a necessidade de downloads e instalações. Isso resulta em um maior potencial de retenção de usuários, já que não há barreiras adicionais para o acesso à aplicação. Dessa forma, o desenvolvimento de uma aplicação

PWA oferece não apenas benefícios técnicos, mas também vantagens econômicas e maior alcance de público.

REFERÊNCIAS

- ANTUNES, Ana. PWA: O que são Progressive Web Apps e por que usar? Technical Report. 2019 Disponível em: <https://gobacklog.com/blog/progressive-web-apps/>. Acesso em: 25 maio de 2022.
- BORTOLOSSI, Humberto José. Criando conteúdos educacionais digitais interativos em matemática e estatística com o uso integrado de tecnologias: GeoGebra, JavaView, HTML, CSS, MathML e JavaScript. **Revista do Instituto GeoGebra Internacional de São Paulo**, v. 1, n. 1, p. XXXVIII-XXXVI, 2012.
- CARVALHO, Lucas Henrique Vicentini Viana de et al. Desenvolvimento de um protótipo em PWA para fins de atualização cadastral no município de Monte Carmelo -MG. 2019.
- CECONI, Laura. Experiência do usuário em Progressive Web Apps. 2019.
- CROLL, A. Building m. uber: Engineering a High-Performance Web App for the Global Market. 2017. Disponível em: <https://eng.uber.com/muber>. Acesso em 1 jun. 2022.
- DEVELOPERS, Google. Your first Progressive web app. Disponível em: <https://developers.google.com/web/fundamentals/codelabs/your-first-pwapp>. Acesso em: 12 de maio de 2022.
- FURTADO, Daniel A. Programação para Internet: Introdução ao Javascript e MOD. 2015. Disponível em: <http://www.daniel.prof.ufu.br/teaching/PPI-SI/PPI-Modulo4-JavaScript.pdf>. Acesso em: 01 de junho de 2022.
- FORTUNATO, David; BERNARDINO, Jorge. Aplicativos web progressivos: uma alternativa aos aplicativos móveis nativos. In: **2018 13ª Conferência Ibérica de Sistemas e Tecnologias de Informação (CISTI)**. IEEE, 2018. p. 1-6. Disponível em: <https://ieeexplore.ieee.org/abstract/document/8399228>. Acesso em: 08 maio de 2022.
- GOOGLE, developers. Auditar apps da Web com o Lighthouse. Google Developers, 26 set. 2017. Disponível em: <https://developers.google.com/web/tools/lighthouse>. Acesso em: 04 maio de 2022.
- GOOGLE. AliExpress. 2017. Disponível em: <https://developers.google.com/web/showcase/2016/aliexpress>. Acesso em: 05 de junho de 2022.
- GRILLO, Filipe Del Nero; FORTES, RENATA PONTIN DE MATTOS. Aprendendo JavaScript. São Carlos-SP, 2008. Disponível em: http://conteudo.icmc.usp.br/CMS/Arquivos/arquivos_enviados/BIBLIOTECA_113_ND

[72.pdf](#). Acesso em: 01 de junho de 2022.

GRONER, L. Progressive Web Apps (PWA) e a evolução da Web, 2018. Disponível em: <https://imasters.com.br/front-end/progressive-web-apps-pwa-e-evolucao-da-web>
Acesso em 17 de abril de 2022.

GRONER, Loiane. Learning JavaScript Data Structures and Algorithms: Write complex and powerful JavaScript code using the latest ECMAScript. Packt Publishing Ltd, 2018.

KERSSENS, Tjarco. Applicability of progressive web apps in mobile development. 2019.

LEPAGE, Pete. Seu primeiro Progressive Web App. Disponível em Acesso em 17 de novembro de 2019.

LEPAGE, Pete.; RICHARD, Sam. What are Progressive Web Apps? Web.dev, 6 jan. 2020. Disponível em: <https://web.dev/what-are-pwas>. Acesso em: 19 de abril de 2022.

MELO, Henning Barly Summer de et al. Uma proposta para o uso de progressive web apps em ambientes de computação móvel em nuvens. 2020.

OSMANI, Addy. "A pinterest progressive web app performance case study. recurso en línea]. A Medium Corporation. 2017. Disponível em: <https://medium.com/dev-channel/apinterest-progressive-web-app-performance-case-study-3bd6ed2e6154>. Acesso em 1 jun. 2022.

PAIVA, Fernando. Panorama Mobile Time/Opinion Box. 2021. Uso de Apps no Brasil, p. 4, 2021. Disponível em: <https://www.mobiletime.com.br/pesquisas/uso-de-apps-no-brasil-dezembro-de-2021>. Acesso em: 25 maio de 2022.

PNAD contínua. Acesso à internet e à televisão e posse de telefone móvel celular para uso pessoal, 2018. Rio de Janeiro: IBGE, 2020. 12 p.

PONTES, Guilherme. **Progressive Web Apps: Construa aplicações progressivas com React**. Casa do Código, 2018

RESHETILO, K; OPANASENKO, S. Progressive Web Apps vs Native: Which Is Better for Your Business? 2017. Disponível em: <https://www.technology.org/2017/07/28/progressive-web-apps-vs-native-which-is-etter-for-your-business>. Acesso em: 26 maio de 2022.

SANTOS, Felipe Nascimento et al. Um Progressive web app para geolocalização de preços de mercadorias por crowdsourcing. 2021.

TANDEL, Sayali; JAMADAR, Abhishek. Impact of progressive web apps on web app

development. **International Journal of Innovative Research in Science, Engineering and Technology**, v. 7, n. 9, p. 9439-9444, 2018.

THAKUR, P. Evaluation and implementation of progressive web application. Metropolia Ammattikorkeakoulu, 2018. Disponível em:
<https://www.theseus.fi/handle/10024/142997>. Acesso em: 10 maio de 2022.

TRINDADE, Patrícia Esteves; AFFINI, Letícia Passos. APONTAMENTO A CERCA DO PROGRESSIVE WEB APPS. 2018.