DESENVOLVIMENTO DE PROGRESSIVES WEB APPS E ANÁLISE COMPARATIVA COM APLICAÇÕES MÓVEIS NATIVAS

Pedro da Cruz Ferreira da Silva Neto[1]
João Frederico Roldan Viana[2]
André Barros Pereira[3]
Márcia Terezinha Tonieto[4]

RESUMO

Este trabalho resulta do estudo sobres as *Progressives Web Apps* (*PWA*). Partindo do ponto que desenvolver aplicações nativas para dispositivos móveis, demanda o desenvolvimento de várias aplicações diferentes uma para cada Sistema Operacional (SO), verificou-se que dependendo dos recursos nativos que uma aplicação pode vir à utilizar, o desenvolvimento de uma aplicação *PWA* possa ser o mais indicado. Para tanto, o objetivo deste estudo é apresentar as características das *PWA*, suas vantagens e desvantagens e como criar uma aplicação *PWA*.

Palavras-chave: PWA, Aplicativos, Mobile.

This paper result of the *PWA* study. Since native mobile applications require the development of different applications for each operating system, it has been found that depending on the native resources that an application may use, the development of a *PWA* application may be the most suitable. Therefore, the aim of this study is to present the characteristics of *PWA*, its advantages and disadvantagesand how to create a *PWA* application.

Palavras-chave: PWA, Applications, Mobile.

1. INTRODUÇÃO

O uso de dispositivos móveis ocupa grande parte da vida das pessoas. Verificaros smartphones várias vezes por dia tornou-se uma rotina para a maioria delas. Durante anos, a única maneira das empresas atingirem os usuários de dispositivos móveis era criando um aplicativo móvel nativo ou híbrido. Hoje, porém, com abordagem de *Progressive Web App* (PWA) tornou-se uma solução alternativa para empresas de qualquer tamanho para atrair usuários móveis ativos.

Diante deste cenário, é necessário avaliar se uma aplicação para dispositivos móveis irá fazer uso de um aplicativo nativo, ou se uma abordagem com PWA é o suficiente para atender os requisitos.

Este trabalho tem como objetivo geral abordar as PWA, apresentando suas vantagens e

[1] Bacharel em Sistemas de Informação – Faculdade Lourenço Filho – E-mail: cruz_pedro@live.com. Versão adaptada do trabalho de conclusão de curso.
[2] Mestre em Computação Aplicada pela Universidade Estadual do Ceará - UECE- E-mail: jfredrv@gmail.com
[3] Mestre em Engenharia de Sistemas e Computação pela Universidade Federal do Rio de Janeiro – RJ. E-mail: andrebarrospereira@gmail.com
[4] Mestre em Computação pela Universidade Estadual do Ceará - UECE- E-mail: marciatonieto@flf.edu.br

desvantagens com relação a aplicativos nativos, com o intuito de servir como base para saber qual abordagem usar ao desenvolver uma aplicação. Como objetivos específicos têm-se: abordar conceitualmente e funcionalmente as PWA, os impactos que elas podem causar no desenvolvimento como um todo, já que pode-se está presenciando o fim de aplicativos nativos, e por fim configurar um ambiente de desenvolvimento onde será criado uma PWA, fazendo uso de tecnologias como IndexedDb, Firebase e JavaScript.

2. ARQUITETURAS WEB

Uma aplicação *web* é um *software* que roda no lado do cliente, ou seja, na máquina do próprio usuário. Com essa abordagem é necessário apenas que o cliente possua um *browser* (navegador), independente de seu sistema operacional paraacessar uma aplicação *web*. Com isso os desenvolvedores não precisam desenvolverdiversas versões de uma mesma aplicação para sistemas operacionais diferentes. Tendo isso em vista, este capítulo visa apresentar um breve histórico da evolução da arquitetura *web*, apresentará o modelo de multicamadas e por fim demonstrará seus estilos arquitetônicos.

2.1 EVOLUÇÃO DA COMUNICAÇÃO

A Internet mudou completamente quando Tim Berners-Lee, a fim de solucionaros problemas de comunicações da Organização Europeia para a Pesquisa Nuclear (CERN), propôs que se usasse um sistema de comunicação em hipertexto. Esse sistema acabou agradando os gerentes do CERN, e logo depois, ele foi implantado com o nome de *World Wide Web (WWW)*. Todos os serviços da *Internet* se renderamao poder da *Web* e à Linguagem de Marcação de Hipertexto (HTML), que a sustenta. Até o serviço de correio eletrônico, campeão de tráfego na *Internet* por muitos anos, que por muito tempo exigia aplicações específicas, separadas do *browser*, hoje é lido dentro de um *browser*, através de páginas HTML(ROCHA,1999).

A medida que o HTML foi ganhando fama, quem a usava não queria apenas uma simples apresentação de texto estruturada, queriam usar cores, imagens e técnicas de design mais avançado. Suas páginas HTML acabavam ficando com vários códigosde estilos. Porém a manutenção seria o maior problema dessa abordagem, se um componente mudar de estilo em apenas uma página a alteração seria simples, mas se fosse em várias páginas, a manutenção já seria bastante penosa. Misturar estilo eestrutura não era mais interessante, e foi assim que em 1995, Hakon Wium Lie e Bert Bos apresentaram a proposta do *Cascading Style Sheets* (CSS) que logo foi apoiada pela *World Wide Web Consortium* (W3C). A ideia era, utilizar HTML somente para estruturar o *website* e a tarefa de apresentação fica com o CSS disposto em umarquivo separado com o sufixo .css ou no próprio HTML demarcado pelas tags <style>(PEREIRA,2009).

O ano de 1995 é muito importante para a *internet*, a Netscape Communications apresenta o *Javascript*, uma linguagem leve, interpretada e baseada em objetoscom funções de primeira classe, mais conhecida como a linguagem de *script* parapáginas *Web*, mas usada também em vários outros ambientes sem *browser* como*node.js*, *Apache CouchDB* e *Adobe Acrobat*. O *Javascript* é uma linguagem de script scriptmulti-paradigma, baseada em protótipo que é dinâmica, e suporta estilos de programação orientado a objetos, imperativo e funcional (MDN,2018), e possibilitou

que os desenvolvedores pudessem criar componentes dinâmicos, e assim possibilitandouma melhora na *interface* do usuário. Com o *javascript* a Internet acabou se tornandomais performática e produtiva, porque os dados não precisavam mais ser enviados aoservidor para se gerar toda a página HTML. Juntos o *Javascript*, HTML e CSS são astrês tecnologias mais populares para a produção de conteúdo na internet (DEVSARAN,2016).

Já no ano de 1996 a Macromedia lançaria o *Flash*, que é uma plataforma multimídia de desenvolvimento para aplicações que contenham animações, áudioe vídeo, bastante utilizada na construção de anúncios publicitários e páginas *web* interativas. O *Flash* pode manipular vetores e gráficos para criar textos animados, desenhos, imagens e até *streaming* de áudio e vídeo pela internet. O *Flash* ganhou bastante popularidade entre os programadores e desenvolvedores *web* por permitir um desenvolvimento rápido de aplicações com alta qualidade e integração transparente com outras categorias de conteúdo (CIPOLI,2018).

Com o *Flash* os programadores puderam dar ao usuário uma experiência maisrica através de animações de áudio e vídeo. O *Flash* por muitos anos foi usado para criação de jogos e aplicativos interativos para dispositivos móveis, mas seu uso foi diminuindo gradualmente.

No ano de 2008 o primeiro rascunho público do HTML5 é lançado pelo Web Hypertext Application Technology Working Group (WHATWG) (THOMS,2017) essa tecnologia ganhou os holofotes, foi quando em 2010 o CEO da Apple Inc., Steve Jobs emitiu uma carta pública intitulada "Reflexões sobre o Adobe Flash", onde ele conclui que odesenvolvimento do HTML5 tornaria o *Flash* não mais necessário para exibir qualquer conteúdo *web*. Depois disso o fim do *Flash* era questão de tempo, muitos *browsers* já não o utilizavam mais, e em julho de 2017, em comunicado oficial a Adobe, anunciou encerrar definitivamente o suporte para sua tecnologia Flash até 2020. A companhia disse precisar destes últimos anos para incentivar criadores de conteúdo edesenvolvedores a utilizarem novas plataformas, principalmente formatos de código aberto e que funcionem também em celulares e *tablets* (CANALTECH,2017).

Em 1999, o conceito de aplicação *web* apareceu na linguagem Java. Mais tarde, em 2005, o Ajax foi apresentado por Jesse James Garrett em seu artigo "Ajax: uma nova abordagem para a aplicação Web". Esta nova técnica de desenvolvimento,possibilitou a criação de aplicações *web* assíncronas, ou seja, o fluxo do código não éinterrompido até que a resposta seja obtida. Ao invés disso, após realizar a requisição, a resposta é obtida em um momento posterior, de forma independente, e tratada por uma função (chamada função de *callback*). Com isso era possível enviar dados parao servidor e recuperá-los sem interferir na navegação de uma página específica, não precisando baixar a página inteira (MARQUES,2016).

2.2 MODELO EM CAMADAS

- **Modelo de uma camada** - Sistemas centralizados ou de arquitetura uni processada, utilizado na década de 1960, era caracterizado por manter todos os recursos do sistema (banco de dados, regras de negócios e interfaces de usuário) em computadores de grande porte, os mainframes que tinha a responsabilidade de realizar todas as tarefas e processamentos (GRANATYR,2007). As aplicações eram escritas em um único módulo ou camada monolítica, com programas e dados firmemente entrelaçados.Tal integração

estreita entre programa e dados dificultava a evolução e reuso de componentes. Cada desenvolvedor de aplicação escolhia como estruturar e armazenar os dados e usava-se frequentemente técnicas para minimizar o caro armazenamento e manipulação em memória principal (MARTINS,2012).

- **Modelo de duas camadas** – Cliente-servidor dos anos de 1980 à 1990, período da origem de *softwares* para gerenciamento de banco de dados relacionais (SGBD), a utilização de rede locais interligando microcomputadores departamentais,o início do paradigma da programação orientada a objetos, e a redução dos custos do hardware, tornando possível a massificação da computação pessoal (MARTINS,2012). Dentro deste contextoque, o modelo de duas camadas surgiu com o objetivo de dividir a carga de processamento entre o servidor e as máquinas clientes (GRANATYR,2007). Na camada do cliente, é onde o usuário interage com o sistema, ela é responsável por prover uma *User Interface* (UI) agradável e de fácil acesso para que ousuário possa manipular o sistema. Mas apesar de prover a UI, a camada do clientenão se restringe a isso, regras de negócio também podem ser implementadas, assim diminuindo a complexidade no servidor.

Com o modelo de duas camadas foi possível que *softwares* de terceiros tivessem acesso ao banco de dados, ou seja, com isso os *softwares* poderiam ser diferentes para cada usuário ou setor, afim de melhor atender às suas necessidades. O modelo pode apresentar algumas desvantagens, como sua aplicação é dividida em partes, acaba acarretando em um *software* mais complexo, com novos cenários a serem tratados. A comunicação do cliente com o servidor se dá por meio da rede, com isso dados sensíveis serão trafegados na rede eprecisam de um cuidado maior com a criptografia (ROCHA,1999).

- **Modelo de Multicamadas** - Cliente-servidor de múltiplas camadas, se consolida com a popularização da internet e melhoria das tecnologias de redes. Ele tem como propósito que uma aplicação cliente não realizasse comunicação direta com o bancode dados, no meio do caminho haveria uma ou mais camadas, que elas sim se comunicam com o banco de dados. Distribuir o processamento da aplicação em várias máquinas, evitando a sobrecarga sobre uma única camada, melhor o desempenho e compartilhar recursos, utilizando-os como se fossem recursos locais, característica conhecida como transparência de uso. Com isso a aplicação pode ser dividida em pequenos pedaços, com possibilidade de aumento de escala e expansão da rede sem perda de qualidade, melhora a robustez em função da distribuição dos serviços em mais de uma máquina, e muitos outros benefícios (MARTINS,2012).

O modelo de multicamadas apresentou diversas vantagens em relação ao modelo de duas camadas. Destacamos:

- Clientes Leves;
- Facilidade de Redistribuição;
- Modularização;
- Economia de conexões no servidor;
- Independência de localização;
- Escalabilidade.

2.2.2 Estilo Arquitetônico

- **Arquitetura de Micro serviços** - Segundo (FOWLER,2014 "É uma abordagem que desenvolve uma aplicação única como uma suíte de pequenos serviços, cada um rodando em seu próprio processo e se comunicando com mecanismos leves, geralmente uma Interface de Programação de Aplicações (API) de recurso Protocolo de Transferência de Hipertexto (HTTP)".

- **Arquitetura Monolítica** - Segundo (SANTOS,2017), "Uma aplicação monolítica é aquele tipo de aplicação na qual toda a base de código está contida em um só lugar, ou seja, todas as funcionalidades estão definidas no mesmo bloco". Esse bloco geralmente se divideem três partes: apresentação, negócio e dados.

Assim, uma aplicação monolítica é separada em pequenas aplicações autônomas, ou seja, devem possuir um sistema de *deploy* automático e independente,além de que cada aplicação possui um conjunto de regras de negócio específico.

3. EVOLUÇÃO DOS DISPOSITIVOS MÓVEIS

Atualmente os dispositivos móveis são imprescindíveis na vida das pessoas, na medida que os dispositivos móveisforam evoluindo e novas funcionalidades foram aparecendo como calendário, rádio e jogos. Mas, foi quando a Apple lançou seu primeiro smartphone, o Iphone, com seu próprio Sistema Operacional (SO) o IOS, e a Google lançou o seu SO para smartphones, o Android, que os dispositivos móveis ganhariam a fama. Com esses sistemas abriu-se um leque de oportunidades para empresas e programadores independentes desenvolverem os mais diversos aplicativos.

O marco da telefonia móvel se deu no ano de 1973, quando dois engenheiros da Motorola, realizaram a primeira ligação da história feita a partir de um telefone móvel, o modelo era o então, DynaTAC como mostra a Figura 1. Com seus poucos mais de 30cm de tamanho e peso de quase 1kg, o seu lançamento ocorreu no ano de 1983, 10 anos após a demonstração (RIGUES,2016).

A primeira geração de telefones móveis, teriam as mesmas características doDynaTAC, não sendo tão portáteis, mas era o primeiro passo e a tendência seria queesses aparelhos evoluíssem e aumentasse suas funcionalidades (JORDAO,2009).

Figura 1 – Motorola DynaTAC

Fonte: (PINTEREST,2019)

Figura 2 – IBM Simon

Fonte: (RIGUES,2016)

No ano de 1994, a IBM lança o Simon, um aparelho que reunia diversas funcionalidades, dentre elas: Assistente Pessoal Digital (PDA) e Fax, e foi um grande passopara o desenvolvimento dos dispositivos móveis, sendo o precursor da tela *touchs-creen*, mas suas medidas ainda eram muito grandes e acabou não sendo tão popular, vendendo apenas 50 mil unidades (RIGUES,2016). A Figura 2 mostra o Simon.

Na figura 3 temos o Nokia Communicator 9000 que foi um aparelho que fez bastante sucesso. Lançado em 1996, foi o primeiro dispositivo móvel comercializado com acesso à internet, além disso reunia diversas funcionalidades como, FAX, SMS,e-mail, agenda de contatos, calculadora, bloco de anotações e outras mais. Suas medidas ainda eram grandes, mas possuía um o recurso de *flip*, em que tinha uma segunda tela e um teclado QWERTY (RIGUES,2016).

Em janeiro de 2007, Steve Jobs em uma apresentação da Apple, lança umsmartphone que mudaria tudo, o IPhone. Com design minimalista, o IPhone dispensouo uso de teclas físicas, tudo seria a base do toque na tela como mostra a Figura 4 (JORDAO,2009).

Figura 3 – Nokia 9000

Fonte: (PRODNOTE,2015)

Figura 4 – Primeiro IPhone

Fonte: (RIGUES,2016)

A Apple desenvolveu um SO próprio para o dispositivo, o IOS, um sistema operacional de fácil interação para o usuário que o tornou bastante popular. Além disso, com o IPhone surgiram também os aplicativos, inicialmente vindos apenas de fábrica,e depois comprados pela loja virtual, a App Store (RIGUES,2016).

Em 2008 surgiria um novo SO que entraria na concorrência com o IOS da Apple, o smartphone T-Mobile G1, era lançado com um SO desenvolvido pela Google,o Android, ele também trazia uma loja de aplicativos, a Android Market que depois passou a ser chamada como é hoje, Google Play. Com o tempo o Android ganhou espaço e hoje a maioria dos smartphones usam o SO da Google (MICROLINS,2017).

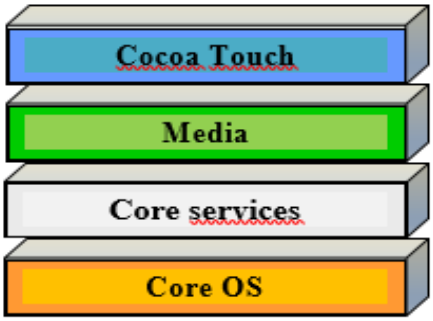
a. IOS

Segundo (GARCIA,2019) "O iOS é um sistema operacional desenvolvido pelaApple que pode ser encontrado no iPhone, iPad e iPod Touch da empresa, visto que os notebooks da empresa utilizando o MacOS e os relógios inteligentes o watch OS", é um software proprietário, portanto só funciona em aparelhos da própria Apple e é um SOpensado para dispositivos *touch screen.*

A arquitetura do IOS é dividida em quatro camadas como mostra a Figura 5. As camadas inferiores do sistema fornecem os serviços fundamentais nos quais todosoftware depende, as camadas

subsequentes contêm serviços e tecnologias mais sofisticadas que se baseiam nas camadas abaixo (DEVELOPER APPLE,2015).

Figura 5 – Arquitetura IOS



Fonte: (Baseado em DEVELOPER APLLE , 2015)

1. Cocoa Touch - A camada que inclui tecnologias para criar a interface do usuáriode um aplicativo, responder a eventos do usuário e gerenciar o comportamentodo aplicativo (DEVELOPER APPLE,2015).

2. Media - A camada que abrange tecnologias especializadas para reprodução,

gravação e edição de mídia audiovisual e para renderização e animação degráficos 2D e 3D (DEVELOPER APPLE,2015).

3. Core Services - A camada que contém muitos serviços e tecnologias fundamentais que variam da contagem automática de referência e comunicação de rede de baixo nível à manipulação de strings e formatação de dados (DEVELOPER APPLE,2015).

4. Core OS - A camada define interfaces de programação relacionadas ao hardware e à rede, incluindo interfaces para executar tarefas de computação de altodesempenho na CPU e GPU de um computador (DEVELOPER APPLE,2015).

b. ANDROID

De acordo com (BARROS,2015) "Android é o sistema operacional móvel do Google. Presente em múltiplos aparelhos de diversas fabricantes, como Samsung,Motorola, LG, e Sony, é a plataforma mobile mais popular do mundo. É conhecido porser baseado no núcleo do Linux, ter um código aberto e uma série de possibilidadesde personalização. "O Android é um SO que diferente do IOS, tem seu código fonte aberto, e com isso, diversas empresas utilizam o com versões customizadas em seusprodutos.

O Android é uma pilha de software composto por seis camadas exemploficado na Figura 6. A base do Android é uma versão modificada do kernel do Linux, que prover vários serviços essenciais, como segurança, rede e gerenciamento de memóriae processos, além de uma camada de abstração de hardware para as outras camadasde software (DEVELOPER ANDROID,2019).

Figura 6 – Arquitetura Android

Fonte: (Próprio autor)

1. Linux Kernel - O core do Android é o kernel do linux, com ele o Android rea- proveita os recursos de segurança do linux e que as fabricantes de dispositivos desenvolvam utilitários de hardware para um kernel conhecido (DEVELOPER ANDROID,2019).

2. Camada de abstração de hardware (HAL) - A HAL consiste em bibliotecas, que implementam uma interface para um componente especifico de hardware, como o Bluetooth (DEVELOPER ANDROID,2019).

3. Android Runtime - Virtual Machine personalizada, projetada para garantir que várias instâncias sejam executadas com eficiência em um único dispositivo (DE-VELOPER ANDROID,2019).

4. Bibliotecas C/C++ nativas - Responsável por fornecer suporte para os principais recursos do Android (DEVELOPER ANDROID,2019).

5. Estrutura da Java API - Conjunto de bibliotecas que ajudam a criar aplicações Android, fornecendo recursos reutilizáveis do sistema (DEVELOPER ANDROID,2019).

6. Aplicativos do sistema - Camada composta por aplicativos incorporados ou quais- quer outros aplicativos de terceiros que foram instalados no dispositivo (DEVELO- PER ANDROID,2019).

4. CONHECENDO AS PROGRESSIVES WEB APPS

Progressives Web Apps (PWA) é uma tecnologia emergente do Google, um conceito relativamente novo no mundo dos dispositivos móveis e da internet. De acordo com o Google Developers (GOOGLE DEVELOPERS,2019b):

> "As *Progressive Web* Apps fornecem uma experiência instalável, semelhante a um aplicativo, em computadores e dispositivos móveis que são criados e entregues diretamente pela Web. Eles são aplicativos da webque são rápidos e confiáveis. E o mais importante, são aplicativos da web que funcionam em qualquer navegador."

PWAs são desenvolvidas usando certas tecnologias e abordagens para criar aplicações que aproveitam os recursos dos dispositivos móveis nativos e de aplicativos da Web, essencialmente é uma mistura de aplicações web e mobiles nativas. O (GOOGLE DEVELOPERS,2019b) definiu um *checklist* a ser seguido para seconsiderar uma aplicação como uma PWA são elas:

• Progressivo - Funciona para qualquer usuário, independentemente do navegador escolhido, pois é criado com aprimoramento progressivo como princípio funda- mental.

• Responsivo - Se adéqua a qualquer formato: desktop, celular ou tablet.

• Independente de conectividade - Aprimorado com *serviceworkers* para trabalhar *off-line* ou em redes de baixa qualidade.

• Semelhante a aplicativos - Parece com aplicativos para os usuários, com interações e navegação de estilo de aplicativos, pois é compilado no modelo de *shell* de aplicativo.

• Atual - Sempre atualizado graças ao processo de atualização do *serviceworker*.

• Seguro - Fornecido via HTTPS para evitar invasões e garantir que o conteúdo não seja adulterado.

• Descobrível - Pode ser identificado como "aplicativo" graças aos manifestos W3Ce ao escopo de registro do *serviceworker*, que permitem que os mecanismos depesquisa os encontrem.

• Reenvolvente - Facilita o reengajamento com recursos como notificações *push*.

• Instalável - Permite que os usuários "guardem" os aplicativos mais úteis em suastelas iniciais sem precisar acessar uma loja de aplicativos.

• Linkável - Compartilhe facilmente por URL, não requer instalação complexa.

4.2 CARACTERÍSTAS

4.2.1 Confiável

Quando iniciado a partir da tela inicial do usuário, os *serviceworkers* permitem que uma PWA seja carregado instantaneamente, independentemente do estado da rede (GOOGLE DEVELOPERS,2019b).

Os *service workers* são *scripts* que o navegador roda por debaixo dos panos, separado de uma página Web, possibilitando que recursos sejam acessados mesmo sem uma interação do usuário ou de uma página Web. O *serviceworker* possui funcionalidades como *Push Notifications,* que é basicamente uma notificação que o usuário recebe sem requisitar e Sincronização em Segundo Plano, que é uma API que permite que a aplicação adie ações até que o usuário tenha uma conectividade com a Internet estável. Isso é útil para garantir que uma ação feita pela o usuário, seja realmente realizada (GOOGE DEVELOPERS,2019a). O *serviceworker* é uma API interessante para os desenvolvedores já que permite configurar experiências *off-line*. Os s*erviceworkers* possuem algumas características importantes:

- É executado em uma *thread* separada do navegador, portanto, não possui acesso ao Modelo de Objeto de Documento (DOM) diretamente.
- É um *proxy* de rede programável, portanto, permite gerenciar as solicitações derede da página.
- É encerrado quando ficar ocioso e reiniciado quando for necessário.
- Os *serviceworkers* utilizam promessas para retornar os dados de suas funções.

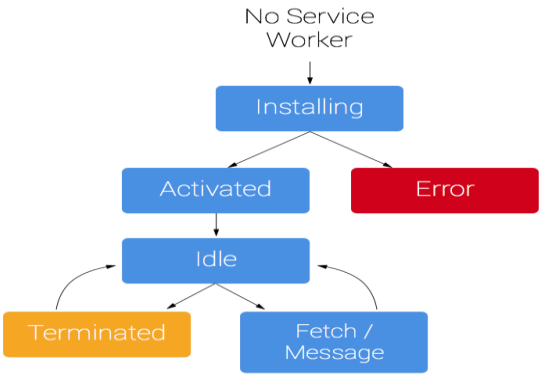
O *Serviceworker* possui um ciclo de vida à parte da página da Web e funciona em uma estrutura já determinada. Entendendo como cada evento ocorre e suas respostas é o ideal para se ter a melhor forma de atualizar e guardar os arquivos. Primeiramente é necessário registrar o *serviceworker* na aplicação, isso pode ser feitovia um arquivo JavaScript (JS) pode se adicionar uma verificação, antes do registro, para verificar se o navegador suporta o uso de *serviceworker*, após

registrar o *serviceworker* o navegador inicia a etapa de instalação em segundo plano (JUSTEN,2018).

Na etapa de instalação, é onde se normalmente é armazenado recursos estáticos em cache. Se todos os recursos forem salvos em cache corretamente, o *serviceworker* estará instalado. Se no momento de guardar os recursos, ocorrer alguma falha,a etapa de instalação não será finalizada corretamente, e portanto, o *serviceworker* não será ativado. Finalizando a etapa de instalação, é iniciada a fase de ativação, é nesse evento onde se gerencia o cache da aplicação e deleta coisas antigas de versões anteriores (GOOGE DEVELOPERS,2019a).

Após a etapa de ativação, o *serviceworker* gerenciará as páginas dentro do seu escopo, com exceção da página que registrou o *serviceworker*, onde ela só será controlada se for carregada novamente. Enquanto o *serviceworker* estiver controlando aspáginas, ele poderá assumir dois estados: tratando de eventos de busca e mensagensque as páginas possam gerar, ou encerrado, para economizar memória do dispositivo (GOOGE DEVELOPERS,2019a). A Figura 7 mostra uma versão minimalista do ciclo de vida do *serviceworker*,em sua primeira instalação.

Figura 7 – Ciclo de Vida do Service Worker



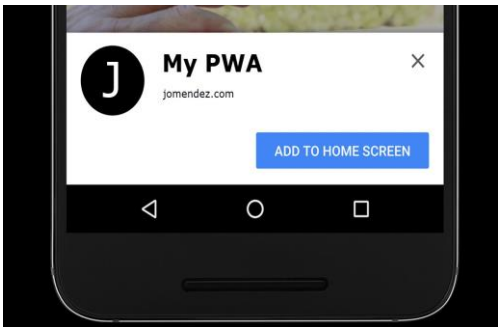
Fonte:(GOOGE DEVELOPERS,2019a)

4.2.2 Rápido

A maioria dos dados das PWAs é salvo no armazenamento do dispositivo no primeiro acesso. A próxima vez que o usuário acessá-la, a aplicação fará o downloadde poucos dados. Este é um recurso útil para pessoas que possuem conexão com a internet limitada. O aplicativo é mais confiável do que apenas um site e permite o envio de notificações para seus usuários, mesmo depois que o aplicativo for fechado. Uma vez armazenado em um dispositivo, leva muito menos tempo para ser reativo do que um site comum que precisa buscar e carregar tudo de novo (WILSON; MADSEN, 2010). As PWAs são instaláveis e podem ser exibidos na tela inicial do usuário, sem anecessidade de uma loja de aplicativos, como Google Play Store ou Apple Store. Elasoferecem uma experiência imersiva em tela cheia com a ajuda do arquivo de manifestodo aplicativo web.

O Manifesto do Aplicativo Web, é um arquivo no formato Notação de Objetos JavaScript (JSON), que permite controlar como a aplicação irá aparecer e como ela será iniciada, ela conterá também informações relevantes a aplicação, assim fazendocom que o navegador entenda que a aplicação é uma PWA e assim o navegador apresentará uma mensagem para o usuário para que se possa instalar o app na telainicial do *smartphone* como é possível ver na Figura 8 (GOOGLE DEVELOPERS, 2019a).

Figura 8 – Adicionar a tela inicial



Fonte:(JOMENDEZ,2018)

Existem algumas configurações importantes disponíveis no arquivo de manifesto, como:

- *background color* - Define a cor de fundo esperada para a aplicação. Esse valorrepete o que já está disponível no CSS do site, mas pode ser usado pelos navegadores para desenhar a cor de fundo de um atalho quando o manifesto está disponível antes de a folha de estilo ser carregada. Isso cria uma transição suaveentre o carregamento da aplicação e o carregamento do conteúdo do site (MDN,2019a).
- *Description* - Fornece uma descrição geral do que a aplicação representa
- *Lang* - Especifica o idioma principal para as propriedades, "name"e "short name"
- *Dir* - Especifica a direção do texto principal para os membros "name", "short name"e "description". Juntamente com o membro "lang", ajuda na exibição correta dos idiomas da direita para a esquerda (MDN,2019a).
- *Display* - Define o modo de exibição da aplicação.
- *Icons* - Especifica uma lista de arquivos de imagem que podem servir como ícones de aplicativo, configurando de acordo com a resolução de tela do dispositivo.
- *Related applications* - Define uma lista de aplicativos nativos que podem ser
- instalados ou acessíveis via plataforma externa, como Google *Play Store*, esses aplicativos podem ser versões alternativas da aplicação PWA(MDN,2019a).
- *Short name* - Fornece um nome curto legível para o aplicativo. Isso é feito quandonão há espaço suficiente para exibir o nome completo da aplicação, como as telas iniciais dos dispositivos.
- *Theme color* - Define a cor do tema padrão para a aplicação. Isso às vezes afeta omodo como o sistema operacional o exibe.

4.2.3 Fácil Instalação e Atualização

Como já foi apresentado, para adicionar ou instalar uma aplicação PWA, os usuários simplesmente precisam abrir seu site progressivo em um navegador em seudispositivo móvel. Em algum momento, eles serão solicitados a adicionar o aplicativo em sua tela inicial, ou os usuários podem adicionar o próprio PWA no menu do navegador.Bem simples em comparação a aplicativos nativos, em que o usuário necessita ir a uma plataforma de aplicativos e instalar (CODICA,2019).

Quanto às atualizações, os usuários de aplicações PWA não precisam atualizar seu aplicativo toda vez que for lançada uma nova versão, eles sempre terão o mais novo.

4.2.4 Custos Reduzidos de Desenvolvimento e Suporte

Não precisa criar uma solução diferente para cada plataforma, pois o mesmo PWA funciona no *Android* e no *iOS* e cabe em qualquer dispositivo. Devido à atuali- zação fácil e simultânea, existirá apenas uma versão do aplicativo circulando. O que significa que não se gastará custos extras suportando várias versões (CODICA,2019).

4.2.5 Rápido, Leve e Seguro

Com o PWA implementado, a aplicação pode ser carregada em poucos segun-dos, o que é possível graças aos dados armazenados em cache pelo *Service Worker*.Sendo menores em tamanho do que os aplicativos móveis nativos, as PWA são maisleves e eficientes e usam menos capacidade e dados do dispositivo. Ao mesmo tempo,eles fornecem uma experiência de usuário próxima à de um aplicativo nativo. Além disso todas as aplicações PWA funcionam via HTTPS, o que significa segurança extrae nenhum acesso não autorizado aos dados (JUSTEN,2018).

4.2.6 Não necessita de nenhuma loja de aplicativos

Para publicar uma aplicativo nativo para dispositivo móvel é necessário enviá-lo para alguma loja de aplicativos, como, Google Play Store ou Apple Store, e que ainda vai passar por um processo de análise do aplicativo. Com uma PWA, não há necessidade de aguardar o término do período de moderação e como já mencionado,para instalar a aplicação progressiva, os usuários só precisarão abrir o website e clicarem "Adicionar à tela inicial". Embora se comportando como um aplicativo nativo, a PWA ainda seria uma página da Web, clicável e compartilhável, e também é indexada pelo Google (CODICA,2019).

4.2.7 Customizável

Se o usuário estiver sem conexão e quiser acessar novas páginas que não foram guardadas em cache, a aplicação não irá travar, mas mostrará uma mensagempersonalizada, como na Figura 9.

Figura 9 – Telas Customizadas

Fonte:(CODICA,2019)

4.3 DESVANTAGENS

4.3.1 Funcionalidades Limitadas

As PWA ainda são sites e ainda não suportam todas as funcionalidades que osaplicativos nativos podem oferecer. Elas têm acesso limitado a recursos dos dispositivos. Além disso, as PWA podem oferecer um nível de notificação menos personalizado secomparado aos aplicativos nativos.

4.3.2 Limitações para o iOS

No momento, ainda há uma lacuna entre as PWA para Android e iOS. Embora os PWAs estejam disponíveis para usuários do iOS, nem todos os recursos que funcionam em dispositivos Android são oferecidos para iOS, no entanto, existe um movimentona equipe por trás do iOS que aos poucos estão implementando recursos para serem usados via PWA, e provavelmente os usuários do iOS irão receber as mesmas funcionalidades das aplicações progressivas disponíveis no Android (FIRTMAN,2018).

5. COMPARATIVO PWA E APLICAÇÃO NATIVA

As PWA provaram ser muito úteis e foram apresentados diversos recursos disponíveis em dispositivos móveis, em que as PWA dão suporte. No entanto, eles nãoestão aqui para tomar o lugar dos aplicativos nativos, mas para corrigir alguns problemascomo, a compatibilidade entre plataformas e funcionamento em baixa conectividade eaté mesmo *offline.*

Tendo isso em vista, esse capítulo irá abordar, o que são as aplicações para dispositivos móveis nativas, e uma comparação levando em consideração as vantagens e desvantagens entre desenvolver uma aplicação PWAe um aplicativo nativo.

5.1 APLICAÇÃO NATIVA

Os aplicativos nativos são desenvolvidos para um dispositivo específico ou umaplataforma, como Android e iOS, e podem interagir e aproveitar os recursos fornecidospor esse dispositivo específico. Esses aplicativos oferecem acesso mais rápido a diferentes recursos do dispositivo, como câmera, microfone, agenda, localização e etc. Para usa-lós é necessário realizar a instalação do aplicativo no dispositivo desejado, normalmente os aplicativos se encontram nas lojas das plataformas, mas também é possível instalar via pacotes externos (CLUSTOX,2018).

Ao desenvolver uma aplicação nativa, é necessário muitas vezes usar linguagens de programação específicas para a plataforma, elas permitem ao desenvolvedor ter acesso às funcionalidades de API e hardware do dispositivo. As linguagens deprogramação disponíveis para desenvolver aplicativos nativos, varia para cada plataforma (CLUSTOX,2018). Abaixo alguns exemplos das linguagens de programação disponíveis:

- Java - Linguagem oficial do sistema operacional Android, usado para criar aplica-tivos nativos para a plataforma.
- Kotlin - Linguagem mais recente e semelhante ao Java, também para criar aplica-tivos nativos para Android.

- Objective-C - Linguagem principal para criar *software* para dispositivos iOS.
- Swift - Linguagem lançada da Apple para a criação de *software* para o iOS, propõe-se a ser mais simples de implementar do que o Objective-C.

5.2 COMPARATIVO PWA E APLICAÇÃO NATIVA

5.2.1 Criação da aplicação e Distribuição

- **Aplicação Nativa**: O desenvolvimento de uma aplicação móvel nativa para Android e iOS pode vir a ser necessárias duas equipes, uma para cada sistema. Mesmo que os aplicativos de ambos os sistemas sejam desenvolvidos ao mesmo tempo,ainda demorará mais para garantir que a funcionalidade seja a mesma para os dois aplicativos. Tudo isso significa tempo e custos consideráveis necessários para criar um aplicativo.

  O envio e aprovação via App Stores é uma parte separada da distribuição do aplicativo móvel nativo. O produto terá que passar por um período de moderação,pela equipe da plataforma, o que geralmente leva algum tempo. Para a Google Play Store, pode levar algumas horas, enquanto na Apple App Store pode levarde dois a quatro dias. Esse tempo de moderação, acaba impactando no atraso da distribuição da aplicação.

- **A criação de uma PWA** requer apenas uma equipe de desenvolvimento da Web,pois na verdade é um site, embora com alguns recursos nativos dos dispositivosmóveis. A validação pelas lojas não é necessária, pois se está criando um *website*. Não é necessário enviar a aplicação para nenhuma loja nem esperar que ele sejaaprovado. Depois que a PWA é construída e publicada na Web, ela está pronto para ser usada.

5.2.2 Instalação

Aplicação Nativa: Basicamente o fluxo de instalação é, ir em uma loja de aplicativos, procurar a aplicação e realizar o download, e após isso, a instalação no dispositivo móvel, e por fim o aplicativo estará disponível para uso.

PWA: Com a PWA é necessário abrir um navegador, acessar o site da aplicação, irá ser apresentado uma mensagem, perguntando se quer adicionar a aplicaçãopara a tela inicial do dispositivo, confirmando, a aplicação irá ser instalada.

5.2.3. Engajamento do Usuário

Uma das ferramentas de engajamento mais poderosas é a Push Notifications. São mensagens entregues por meio de um aplicativo instalado nos dispositivos, nos dispositivos móveis ou nos desktops dos usuários, para alertar seus usuários sobre novas chegadas de ações, vendas ou outras notícias.

Aplicação Nativa: Em aplicativos móveis nativos, a disponibilidade do recurso de notificações por *push* não depende do sistema operacional ou do modelo dodispositivo. Os usuários os receberão independentemente desses fatores.

PWA: Nas PWA, as *Push Notifications* também estão disponíveis, mas apenas para o Android. Isso é possível graças aos *service workers* eles podem enviar notificações quando uma aplicaçãoPWAnão está em execução.

5.2.4 Funcionamento Offline

Aplicação Nativa: Quando estamos falando sobre o modo *offline* do aplicativo nativo, assumimos que ele opera da mesma maneira que na conexão. O ponto éque um aplicativo nativo mostra o conteúdo e a funcionalidade que ele conseguiuarmazenar em *cache* quando a conexão ainda estava lá. Isso está disponível devido ao armazenamento local e à sincronização suave de dados com a nuvem.

PWA: Nas aplicações PWA, os usuários também podem aproveitar o modo offline. Quando ativadas, as páginas mostram o conteúdo pré-carregado ou carregado,que é fornecido com os *service workers*. No entanto, o modo offline nas PWA é um pouco mais lento em comparação a um aplicativo móvel nativo, pois ele é implementado de forma diferente. Ao mesmo tempo, a diferença entre os dois tipos de aplicativos não é tão drástica.

6. ESTUDO DE CASO

Descrevemos aqui o desenvolvimento de uma aplicação com suporte à PWA. Basicamente o App consiste em 3 telas. Será descrita a aplicação que utiliza alguns recursos de uma PWA, como, funcionamento *offline*, câmera do dispositivo, localização atual, notificações e *background sync*. Para isso foram utilizadas algumas ferramentas como *Firebase*, *Workbox* e *IndexedDB*.

6.1 FERRAMENTAS UTILIZADAS

6.1.1 Firebase

De acordo com (TREINAWEB,2017) "O Firebase é uma plataforma do Googleque contém várias ferramentas e uma excelente infraestrutura para ajudar desenvol-vedores web e mobile a criar aplicações de alta qualidade e performance". Para odesenvolvimento do App, serão utilizados alguns recursos disponibilizados pelo firebase:

- **Realtime Database**: Banco de dados *No SQL* hospedado em nuvem.
- **Storage**: Banco de dados utilizado para guardar arquivos de mídia, como imagens,vídeos e áudio.
- **Notifications**: Serviço de envio de notificações para usuários conectados.

6.1.2 IndexedDB

O *indexedDB* é um banco de dados disponibilizado pelo *browser*, segundo (MDN,2019b) " *IndexedDB* é uma API para armazenamento *client-side* de quantidades significantes de informações e buscas com alta performance por índices". *IndexedDB* éa solução para grande porção de dados estruturados.

6.1.3 Workbox

O *workbox* é uma ferramenta, que vai nos ajudar a manipular, de forma mais simples, o *serviceworker* da aplicação. Segundo (GOOGE DEVELOPERS,2019b) "O *workbox* é um conjunto de bibliotecas e módulos que facilitam o armazenamento em *cache* e aproveitam ao máximo os recursos usados para criar umaPWA".

6.2 CONHECENDO O APLICATIVO

Para esse projeto foi desenvolvido um aplicativo que realiza postagens. Cada uma dessas postagem deve possuir além da imagem, uma descrição e a localizaçãode onde foi realizado tal evento. O aplicativo desenvolvido têm três telas:

- **Tela Inicial**: É apresentação do aplicativo, ela reúne todas as postagens feitas pelos os usuários, e um botão de ação, para criar uma nova postagem no aplicativo.Como é visto na Figura 10.
- **Tela de Criação de Postagens**: AFigura 11 mostra como é a tela que possibilitaao usuário criar uma postagem.
- **Tela de Ajuda**: É uma página para auxiliar o usuário na navegação do aplicativo.Ainda existem botões de ação que simulariam o entrar em contato com a equipede SAC, como mostrado na Figura 12.

Figura 10 -Tela Inicial Figura 11- Tela de Criação de Post Figura 12-Tela de Ajuda

Fonte: (Elaboradas Pelo Autor)

6.3 CONFIGURAÇÕES INICIAS DO APLICATIVO COM PWA

6.3.1 Arquivo de Manifesto

O manifesto da aplicação é configurado por meio de um arquivo de texto, que possibilita adicionar informações relevantes sobre o aplicativo. Com o arquivo demanifesto configurado a aplicação agora pode ser adicionada a tela inicial (MDN,2019a). exemplificado na figura 13.

6.3.2 Service Worker

Como o *service worker* é um *script* do navegador que executa em segundo plano. No primeiro momento vamos apenas configura-lo para fazer cache de alguns recursos úteis a nossa aplicação, como, arquivos *javascript*, css, imagens e páginas html do aplicativo. Para configurar o *service worker*, foi criado um arquivo *javascript*,que utiliza a biblioteca externa chamada *workbox*. O *Workbox* ajudará a configurar o *service worker* de maneira mais fácil, já que ele abstrai, muito código (GOOGE DEVELOPERS,2019b).

Figura 13 – Adicionando App

Fonte: (Elaborado pelo autor)

6.4 RECURSOS AVANÇADOS DO SERVICE WORKER

6.4.1 Cache das Postagens

Com o *service worker* configurado, um recurso importante do aplicativo é alistagem de postagens, é possível configurar o *service worker*, para fazer o *cache* daspostagens a medida que forem sendo carregadas, assim tornando o aplicativo funcionalmesmo em modo *offline*. No arquivo de configuração do *service worker*, criaremos uminterceptador para a rota de listagem de Postagens e faremos o tratamento para guardaros dados em cache quando necessário. Na configuração acima, os dados retornados da requisição são guardados no *IndexedDB*, na tabela 'posts'. Quando a requisição é retornada, os dados da tabela 'posts' são limpos, para então ser salvos, afim de mantermos sempre dados validos no *IndexedDB*.

6.4.2 Ações Customizadas

O *service worker* nos permite customizar ações de tratamento, quando nossa aplicação estiver *offline*, e um recurso interessante que será adicionado, é a exibiçãode uma página customizada, quando o usuário tenta acessar um recurso *offline* e queainda não foi cacheado pelo *service worker*, também conhecida como *fallback page* como mostra a Figura 14.

6.4.3 Sincronização de Dados em Background

Um recurso interessante que o aplicativo possuí é o envio de postagens, mesmoo usuário estando *offline*, o *service worker*, possui um evento de *sync* onde nele é possível verificar se o aplicativo está com acesso a internet e realizar uma determinada ação (GOOGE DEVELOPERS,2019a).

Ao cadastrar uma postagem, será salvo no *IndexedDB*, o body da requisição para o servidor na tabela 'sync-posts', após isso será registrado um evento 'sync', com a tag'sync-new-posts' no *service worker*, para que ele trate-o logo mais.

Para o *service worker* tratar o evento que acabou de ser transmitido, é necessárioimplementar uma função. Para isso verificaremos se o evento transmitido terá a tag ’sync-new-posts’, que foi passada previamente, e assim podemos tratar os dados da forma que quisermos, nesse caso, vai ser chamada os dados do *IndexedDB*, referente a tabela ’sync-posts’ e enviaremos os dados para o servidor *Firebase*, e por fim, limpamos os dados da tabela ’sync-posts’, que acabaram de ser salvos.

6. 5 NOTIFICAÇÕES

Notificações é uma ferramenta importante para informar o usuário sobre um determinado evento, tendo isso em vista, o aplicativo desenvolvido vai enviar notificações para os usuários quando for criada uma nova postagem. Para a criação do fluxo de notificação foi usado o *Firebase* e configurações no arquivo do *service worker*. Num primeiro momento é necessário verificar se o usuário deseja receber notificações,por isso, o aplicativo possui um botão que habilita as notificações, momo mostra a Figura 15. Este botão só estará disponível se o browser possuir suporte as notificações, para isso foi adicionado uma verificação, em que esconde o botão caso as notificações não estiverem disponíveis. O *browser* tendo suporte as notificações, é necessário a permissão do usuáriopara podermos utiliza-las.

O usuário dando permissão para o aplicativo usar as notificações, podemos configura-las e criar uma inscrição no *Firebase*, essa inscrição vai auxiliar no envio das notificações, como uma identificação para o dispositivo do usuário e enviar a notificação para eles. Para essa ação, foi criada uma função que verifica se o dispositivo já possuiuma inscrição. No retorno da requisição para o *Firebase*, se tudo ocorrer bem, é chamada uma função que exibe para o usuário uma notificação, pré-configurada. A Figura 15, mostra o exemplo, de como será exibida a notificação:

Figura 14 - Tela *Offline* Figura 15 - Habilitar Notificações Figura 16 - Notificação Pré Configurada

Fonte: (Elaboradas pelo autor)

Por fim, para exibir as notificações, é necessário, configurar o *serviceworker*. O *serviceworkere* possui o evento ’push’, onde é configurada uma função para o tratamento das notificações.

7. CONCLUSÃO

Analisando os resultados apresentados, as PWA não chegaram para substituiras aplicações nativas, essa não é a sua proposta, mas sim facilitar o desenvolvimentode aplicações web, com recursos nativos de um dispositivo móvel, em que normalmente só era possível com aplicações nativas, e vão além disso, implementando recursos como o funcionamento *offline* e tratamentos customizados. O uso ou não da PWA vaidepender exclusivamente do negócio em que a aplicação vai ser direcionada. Caso elavá possuir diversas funcionalidades nativas, o ideal é desenvolver uma aplicação semo uso da PWA, entretanto, se uma PWA possuir tais recursos, é recomendado o uso da PWA, por facilitar o desenvolvimento.

Por ser algo muito recente a maioria dos livros que abordam sobre as PWA começaram a serem publicados em meados de 2016, oferecendo na maioria das vezes uma visão mais geral do assunto e com pouco foco na parte prática, com isso sefez necessário procurar conteúdo nas mais variáveis fontes, como sites, artigos e no próprio *GitHub*. Com o intuito de definir quais seriam os objetivos desse trabalho, coma pesquisa bibliográfica foi possível entender melhor a dimensão que as PWA podemter no futuro.

Vale salientar que entre os objetivos definidos nesse trabalho, a configuração do ambiente de desenvolvimento foi o mais complexo. Grande parte do material disponível na Internet estava incompleto e não mostrava como configurar, por completo, o ambiente. Outra ocorrência comum era o aparecimento de erros no ambiente durante a execução de algum comando nos *serviceworkers*.

As PWA apresentam uma proposta de melhorar o engajamento do usuário, porexemplo, quando se utiliza um App mobile nativo, tem todo um passo à passo para começar a usá-lo. Primeiro tem que ir na loja de aplicativos do dispositivo móvel, fazero download do App esperar a conclusão da instalação e após isso poder usá-lo, já comuma PWA o usuário basta abrir um navegador, de sua preferência, e acessar a URL do App, e pronto, já está disponível para utilização. Além do que, não é necessário esperar dias para que o novo App esteja disponível na loja de App, por exemplo, na Google Play.

REFERÊNCIAS

BARROS, T. *Android*. 2015. Disponível em:<https://www.techtudo.com.br/tudo-sobre/android.html>.

CIPOLI, P. *O que é o Adobe Flash?* 2018. CanalTech. Disponível em:<https://canaltech.com.br/software/O-que-e-o-Adobe-Flash/>. Acesso em: 31 jul 2018.

CLUSTOX. *Aplicação Nativa*. 2018. Disponível em:<https://www.clustox.com/blog/native-app-vs-pwa-comparative-study>.

DEVELOPER ANDROID. *Android Arquitetura*. 2019. Disponível em:<https://developer.android.com/guide/platform>.

DEVELOPER APPLE. *IOS Arquitetura*. 2015. Disponível em:<https://developer.apple.com/library/archive/documentation/MacOSX/Conceptual/OSX_Technology_Overview/CocoaApplicationLayer/CocoaApplicationLayer.html#//apple_ref/doc/uid/ TP40001067-

CH274-SW1>.

DEVSARAN. *From History of Web Application Development*. 2016. Disponível em:<https://www.devsaran.com/blog/history-web-application-development>. Acesso em: 31jul 2018.

FIRTMAN, M. *PWA IOS*. 2018. Disponível em:<https://medium.com/@firt/progressive-web-apps-on-ios-are-here-d00430dee3a7>.

FOWLER, M. *Microservices*. 2014. Disponível em: <https://martinfowler.com/articles/microservices.html>.

GARCIA, R. *IOS*. 2019. Disponível em:<https://www.apptuts.com.br/tutorial/ipad/o-ue-e-ios/>.

GOOGLE DEVELOPERS. *Service Workers*. 2019. Disponível em:<https://developers.google.com/web/fundamentals/primers/service-workers/>.

GOOGLE DEVELOPERS. *Workbox*. 2019. Disponível em: <https://developers.google.com/web/tools/workbox>.

GOOGLE DEVELOPERS. *Manifesto*. 2019. Disponível em:<https://developers.google.com/web/fundamentals/web-app-manifest>.

GOOGLE DEVELOPERS. *PWA: Definição*. 2019. Disponível em:<https://codelabs.developers.google.com/codelabs/your-first-pwapp>.

GRANATYR, J. *Introdução ao Modelo Multicamadas*. 2007. Disponível em:<https://www.devmedia.com.br/introducao-ao-modelo-multicamadas/5541>.

JOMENDEZ. *PWA Add Home Screen*. 2018. Disponível em:<http://www.jomendez.com/2018/06/05/add-home-screen-pwas/>.

JORDAO, F. *História Mobile 2*. 2009. Disponível em:<https://www.tecmundo.com.br/celular/2140-historia-a-evolucao-do-celular.htm>.

JUSTEN, W. *PWA Overview*. 2018. Disponível em:<https://willianjusten.com.br/como-fazer-seu-site-funcionar-offline-com-pwa/>.

MARQUES, R. *Ajax com jQuery: Trabalhando com requisições as-síncronas*. 2016. Disponível em:<https://www.devmedia.com.br/ajax-com-jquery-trabalhando-com-requisicoes-assincronas/37141>.

MARTINS, C. *Aplicações corporativas multicamadas - Partes 1 e 2*. 2012. Disponível em:<https://www.devmedia.com.br/aplicacoes-corporativas-multicamadas-partes-1-e-2/ 26545>.

MDN. *Web Docs - JavaScript*. 2018. Disponível em:<https://developer.mozilla.org/ pt-BR/docs/Web/JavaScript>. Acesso em: 31 jul 2018.

MDN. *Arquivo de Manifesto*. 2019. Disponível em:<https://developer.mozilla.org/pt-BR/docs/Web/Manifest>.

MDN. *IndexedDB*. 2019. Disponível em:<https://developer.mozilla.org/pt-BR/docs/IndexedDB>.

MICROLINS. *Historia Android*. 2017. Disponível em:<https://www.microlins.com.br/noticias/tecnologia/linha-do-tempo-a-evolucao-dos-celulares>.

PEREIRA, A. *A origem do CSS, um pouco da história*. 2009. Disponível em:<https://www.devmedia.com.br/a-origem-do-css-um-pouco-da-historia/15195>.

PINTEREST. *Motorola Dynatac*. 2019. Disponível em: <https://br.pinterest.com/pin/

346636502552324402/>.

PRODNOTE. *Nokia 9000*. 2015. Disponível em: <https://prodnote.wordpress.com/2015/03/12/nokia-9000-communicator-throwback/>.

RIGUES, R. *História Mobile*. 2016. Disponível em: <https://blog.meuquantum.com.br/ veja-evolucao-do-smartphone/>.

ROCHA, H. L. S. da. *Arquitetura WEB*. [S.l.]: EP:SF-Rio/D99, 1999.

SANTOS, L. *Microserviços: dos grandes monólitos às peque- nas rotas*. 2017. Disponível em: <https://medium.com/trainingcenter/microserviços-dos-grandes-monólitos-às-pequenas-rotas-adb70303b6a3>.

THOMS, N. *A look back at HTML5*. 2017. FastHosts. Disponível em:<https://www.fasthosts.co.uk/blog/websites/look-back-html5>.

TREINAWEB. *Firebase*. 2017. Disponível em:<https://www.treinaweb.com.br/blog/firebase-descubra-no-que-esta-plataforma-pode-te-ajudar/>.