



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ
BACHARELADO EM ENGENHARIA DE SOFTWARE

ANTONIO CARLOS SERAFIM DOS REIS

UM ESTUDO COMPARATIVO ENTRE MODELOS DE DESENVOLVIMENTO DE
APLICAÇÕES MÓVEIS

QUIXADÁ

2019

ANTONIO CARLOS SERAFIM DOS REIS

UM ESTUDO COMPARATIVO ENTRE MODELOS DE DESENVOLVIMENTO DE
APLICAÇÕES MÓVEIS

Monografia apresentada no curso de Engenharia de Software da Universidade Federal do Ceará, como requisito parcial à obtenção do título de bacharel em Engenharia de Software. Área de concentração: Computação.

Orientador: Prof. Dr. Marcio Espíndola
Freie Maia

QUIXADÁ

2019

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

R298e Reis, Antônio Carlos Serafim dos.
Um estudo comparativo entre modelos de desenvolvimento de aplicações móveis / Antônio Carlos Serafim dos Reis. – 2019.
39 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de Engenharia de Software, Quixadá, 2019.
Orientação: Prof. Dr. Marcio Espíndola Freie Maia.

1. Smartphones. 2. Aplicativos móveis. 3. Progressive Web App. I. Título.

CDD 005.1

ANTONIO CARLOS SERAFIM DOS REIS

UM ESTUDO COMPARATIVO ENTRE MODELOS DE DESENVOLVIMENTO DE
APLICAÇÕES MÓVEIS

Monografia apresentada no curso de Engenharia de Software da Universidade Federal do Ceará, como requisito parcial à obtenção do título de bacharel em Engenharia de Software. Área de concentração: Computação.

Aprovada em: ___/___/___

BANCA EXAMINADORA

Prof. Dr. Marcio Espíndola Freie Maia (Orientador)
Universidade Federal do Ceará – UFC

Prof. Dr. Jefferson de Carvalho Silva
Universidade Federal do Ceará - UFC

Prof. Me. Victor Aguiar Evangelista de Farias
Universidade Federal do Ceará - UFC

À meus pais.

AGRADECIMENTOS

A Deus por ter me proporcionado ao longo desses anos ter saúde e força para superar as dificuldades.

Aos meus pais, pela educação que me proporcionaram desde o início da minha vida, pelo amor, incentivo apoio incondicional.

Aos meus irmãos e namorada por sempre terem me ajudado e motivado pra seguir em frente.

A todos que direta ou indiretamente fizeram parte da minha formação, um muito obrigado a todos.

“Se pude enxergar mais longe, é porque me apoiei nos ombros de gigantes.”

(Isaac Newton)

RESUMO

Com a popularização dos *smartphones*, as aplicações mobiles surgiram com diversos gêneros e uma enorme gama de funcionalidades. Com isso, a forma de desenvolver essas aplicações também se mostrou versátil apresentando diversos modelos de desenvolvimento. Este trabalho visa a comparação das formas de desenvolvimento de aplicação mobiles Nativa, Híbrida e PWA. O objetivo principal é comparar tais formas de desenvolvimento através da criação de uma aplicação simples com os modelos citados. As métricas de comparação são categorizadas em métricas que correspondem ao desenvolvimento de uma aplicação e ao desempenho que a aplicação criada apresenta. Com a aplicação criada são concluídos os cenários onde cada vertente se sobressai, as principais características de cada uma e uma análise comparativa baseado nos resultados obtidos.

Palavras-chave: Smartphones. Aplicações Móveis. Nativo. Híbrido. PWA.

ABSTRACT

With the popularization of smartphones, mobile applications have come up with many genres and a huge range of features. Thus, the way to develop these applications also proved versatile with several development modes. This work aims to compare the forms of application development native, hybrid and PWA. The main objective is to compare such forms of development by creating a simple application with both models. Comparison metrics are categorized into metrics that correspond to the development of an application and the performance that the created application presents. With the application created the scenarios where each stands out, the main characteristics of each one and a comparative analysis based on the obtained results.

Keywords: Smartphones. Mobile applications. Native. Hybrid. PWA.

LISTA DE FIGURAS

Figura 1 – Modelos de desenvolvimento de aplicações móveis	13
Figura 2 – Comparativo dos diferentes tipos de formas de desenvolvimento	16
Figura 3 – Diagrama de uma chamada de uma API PWA.	21
Figura 4 – Diagrama de uma chamada de uma API Nativa.	23
Figura 5 – Diagrama de uma chamada de uma API híbrida.	24
Figura 6 – Passos Metodológicos.	26
Figura 7 – Fluxo de trabalho da aplicação desenvolvida.	27
Figura 8 – Modelos de desenvolvimento de aplicações móveis	30

LISTA DE QUADROS

Quadro 1 – Comparações deste trabalho com os trabalhos relacionados.	18
Quadro 2 – Comparação entre ferramentas dos modelos de desenvolvimento.	28
Quadro 3 – Configuração dos ambientes utilizados para realização dos testes na aplicação desenvolvida.	29
Quadro 4 – Métricas de Desenvolvimento.	32
Quadro 5 – Métricas de Desempenho.	33

LISTA DE ABREVIATURAS E SIGLAS

PWA	<i>Progressive Web Apps</i>
SDK	<i>Software Development Kit</i>

SUMÁRIO

1	INTRODUÇÃO	12
2	TRABALHOS RELACIONADOS	15
2.1	Progressive Web Apps: the Definite Approach to Cross-Platform Development (MAJCHRZAK; BIØRN-HANSEN; GRØNLI, 2018) . . .	15
2.2	Estudo de Frameworks Multiplataforma Para Desenvolvimento de Aplicações Mobile Híbridas (PREZOTTO;BONIATI, 2014)	16
2.3	An Evaluation of Cross-Platform Frameworks for Multimedia Mobile Applications Development (FERREIRA et al., 2018)	17
2.4	Comparação deste trabalho com trabalhos relacionados	18
3	FUNDAMENTAÇÃO TEÓRICA	19
3.1	Aplicações PWA (Progressive Web Apps)	19
3.1.1	<i>Funcionamento do PWA</i>	20
3.2	Aplicações Nativas	21
3.2.1	<i>Funcionamento das Aplicações Nativas</i>	22
3.3	Aplicações Híbridas	23
3.4	Métricas de avaliação	25
4	PROCEDIMENTOS METODOLÓGICOS	26
4.1	Definição da Aplicação Desenvolvida	26
4.2	Desenvolvimento da Aplicação	27
4.3	Aplicação das Métricas de avaliação	28
5	RESULTADOS	30
5.1	Aplicação desenvolvida	30
5.2	Avaliação de desenvolvimento	31
5.3	Avaliação de desempenho	33
6	CONSIDERAÇÕES FINAIS	35
	REFERÊNCIAS	37

1 INTRODUÇÃO

Atualmente os dispositivos móveis, como *tablet's* e *smartphone's*, são aparelhos que fazem intermédio de mais da metade do consumo de mídia digital, representando 67% de todo o tempo gasto com o uso de mídia digital nos Estados Unidos (LIPSMAN; LELLA, 2017). A cada dia que passa, maior é a demanda de criar aplicações móveis que suprem as necessidades gerais dos usuários, como: ter um bom desempenho e que responda as ações do usuário rapidamente, capacidade de funcionar offline, layout responsivo que se adéque às diferentes resoluções de dispositivos existentes, engajamento do usuário e etc. No ano de 2016 o uso de internet através de dispositivos como *smartphone* e *tablet* ultrapassou o uso através de *desktop* pela primeira vez a nível mundial (SIMPSON, 2016).

Atualmente, os três modelos de desenvolvimento móvel mais conhecidos são nativo, híbrido e o *Progressive Web Apps* (PWA). No desenvolvimento nativo é necessário desenvolver a aplicação especificamente para cada plataforma, sendo necessários dois times diferentes para atender as plataformas Android e IOS. Para o desenvolvimento desse tipo de aplicação é utilizado um *Software Development Kit* (SDK), que permite acessar um conjunto de funcionalidades específicas para a plataforma (KUMAR, 2014). Já o desenvolvimento híbrido combina o melhor das aplicações nativas e de tecnologia web. Essas aplicações são construídas utilizando HTML5, CSS e JavaScript. Uma vez construídas, elas são executadas em contêineres de aplicações nativas que é o que proporciona o acesso a pacotes nativos (KHO'I; JAHID, 2016). Uma aplicação PWA é uma aplicação web aprimorada com algumas tecnologias que permitem um comportamento nativo no dispositivo móvel, e ainda sendo possível ser executada em navegador de *desktop* (FRANSSON; DRIAGUINE, 2017). Para uma aplicação ser considerada PWA é necessária ter características como: progressiva, responsiva, independente de conectividade, instalável, segura, atualizável, detectável, engajável, semelhante a aplicativos, vinculável (LEPAGE, 2018).

Apesar de não existir uma grande variedade de modelos de desenvolvimento móvel disponíveis, não é uma tarefa fácil definir qual o modelo de desenvolvimento móvel — dentre nativo, híbrido e *Progressive Web App* (PWA) — seria o mais apropriado a ser utilizado por equipes de desenvolvimento. Pelo fato de cada um desses modelos desenvolvimento citados apresentarem características diferentes, existem situações onde um modelo específico seria o mais adequado em relação a outro dependendo das necessidades.

Existem pontos positivos e negativos para os critérios geralmente levados em consideração ao se criar uma aplicação móvel. É preciso conhecer bem as necessidades do negócio e os critérios mais importantes para que se escolha bem o melhor modelo de desenvolvimento a ser utilizado.

Com a competitividade imposta pelo mercado, é de extrema relevância que a equipe de desenvolvimento móvel escolha o modelo de desenvolvimento que melhor se adéque às necessidades da aplicação a ser desenvolvida. Atualmente, aplicações *PWA* tem ganho destaque como metodologia de desenvolvimento de aplicações móveis, baseado em um conjunto de padrões defendidos pelo *Google Web Fundamentals*, que busca minimizar a lacuna existente induzindo funcionalidades como suporte offline, sincronização de *background* e instalação na tela inicial do dispositivo (MAJCHRZAK; BIØRN-HANSEN; GRØNLI, 2018). Além disso, empresas como Twitter, AliExpress e Expedia já utilizam desse modelo de desenvolvimento (MAJCHRZAK; BIØRN-HANSEN; GRØNLI, 2018). Outro ponto positivo tem sido a plataforma iOS disposta a fornecer um bom suporte para essas aplicações, com a implementação de duas funcionalidades essenciais para *PWA*: *Service Workers* e *Web App Manifest* (FIRTMAN, 2018). Na Figura 1 são vistos os modelos de desenvolvimento de aplicações móveis.

Figura 1 – Modelos de desenvolvimento de aplicações móveis



Fonte: Kadaptada de (CABRERA, 2018)

Com esse cenário descrito, o principal objetivo desse trabalho é avaliar as diferenças entre o desenvolvimento de aplicações nativas, híbridas e *PWA*. Para alcançar esse objetivo, será implementada uma aplicação utilizando os três diferentes modelos de desenvolvimento móvel, que permita realizar uma comparação entre os mesmos. A aplicação escolhida para o desenvolvimento apresenta funções de carregamento de imagem e redirecionamento para texto, que atualmente reflete parte das funções de aplicações móveis em geral. Além disso, são dispostas métricas de avaliação para comparar os modelos de desenvolvimento. Ao final

serão identificados os cenários que aplicações de cada tipo de desenvolvimento tem o melhor rendimento comparado aos demais.

Portanto, o trabalho presente se estende por duas etapas principais onde, a etapa inicial consiste em implementar uma aplicação com cada modelo de desenvolvimento para aplicações móveis. A etapa seguinte compara as dificuldades, métricas de desempenho e de desenvolvimento entre os modelos de implementação utilizados.

Os demais capítulos estão ordenados do seguinte modo: no Capítulo 2 são apresentados os trabalhos relacionados a este, que em geral comparam abordagens de modelos de desenvolvimento de aplicações móveis; no Capítulo 3 está descrita toda a fundamentação teórica essencial para o entendimento deste trabalho; no Capítulo 4 é exposto detalhadamente os procedimentos metodológicos deste trabalho; o Capítulo 5 apresenta os resultados obtidos; por fim, o Capítulo 6 as considerações finais deste trabalho.

2 TRABALHOS RELACIONADOS

Neste capítulo são apresentados trabalhos relacionados que contribuem para o presente trabalho. Essa contribuição ocorre em relação ao escopo dos trabalhos que serão apresentados, como em (MAJCHRZAK; BIØRN-HANSEN; GRØNLI, 2018), (PREZOTTO; BONIATI, 2014) e (FERREIRA et al., 2018).

2.1 Progressive Web Apps: the Definite Approach to Cross-Platform Development (MAJCHRZAK; BIØRN-HANSEN; GRØNLI, 2018)

Em (MAJCHRZAK; BIØRN-HANSEN; GRØNLI, 2018) é questionado se *PWA* seria uma alternativa de modelo de desenvolvimento que poderia unificar o desenvolvimento multiplataforma. Com base em características que definem desenvolvimento multiplataforma e de *PWA*, foi examinado a possibilidade de *PWA* ser a tecnologia unificadora para desenvolvimento móvel. Além disso, em (MAJCHRZAK; BIØRN-HANSEN; GRØNLI, 2018) foram usadas as seguintes etapas para demonstrar o que seria um desenvolvimento móvel unificador. Primeiramente foram definidos os requisitos que definiriam um desenvolvimento móvel como unificador:

- Uma interface do usuário que pode ser construída a partir de elementos típicos que proveem um layout diferente.
- A possibilidade de prover um controle de fluxo e várias formas de interação com o aplicativo
- Uma aparência que se alinhe com as diretrizes de design e usabilidade
- A possibilidade de definir tipos de dados e executar operações básicas
- Mapear campos de formulários, realizar validação e possibilidade de persistir dados
- Meios de reagir a eventos e mudança de estados
- Possibilidade de acessar funcionalidades comuns de hardware
- Um grau suficiente de escalabilidade
- Uma performance considerada próxima a aplicações nativas
- Um ótimo grau de manutenibilidade de aplicações
- Suporte a sistemas backend
- Nível básico de segurança integrada ao framework

Com base em definições teóricas de *PWA*, os autores do trabalho (MAJCHRZAK; BIØRN-HANSEN; GRØNLI, 2018) concluíram que *PWA* já implementa muitos requisitos considerados importantes para unificar o desenvolvimento multiplataforma, mas ainda assim é muito cedo para afirmar isso de maneira mais evidente. Neste trabalho também se busca uma fundamentação teórica dos modelos de desenvolvimento móvel para avaliá-los com base em uma lista de métricas.

2.2 Estudo de Frameworks Multiplataforma Para Desenvolvimento de Aplicações Mobile Híbridas (PREZOTTO;BONIATI, 2014)

Em (PREZOTTO;BONIATI, 2014) é realizado um estudo comparativo e uma análise entre *frameworks* de desenvolvimento híbridos, com foco em características e desempenho de cada uma abordada. Além disso, também é realizando um comparativo entre *frameworks* de UI (*User Interface*). Para realizar essa atividade ele aborda um estudo de caso, desenvolvendo uma aplicação de diário de classe escolar em diversos *frameworks* e realizou a disponibilização em plataformas distintas. A justificativa para o desenvolvimento da aplicação é dada por ser uma aplicação útil na área de educação.

Além disso, (PREZOTTO;BONIATI, 2014) faz uma exploração das formas de desenvolvimento, comparando de forma teórica as formas de desenvolvimento de aplicações móveis. Na Figura 2 é exposto o gráfico comparativo das formas de desenvolvimento de aplicações móveis encontrados em (PREZOTTO;BONIATI, 2014).

Figura 2 – Comparativo dos diferentes tipos de formas de desenvolvimento

	Acesso ao Dispositivo	Performance	Tempo Desenvolvimento	App Store	Multi Plataforma
Nativo	Sim	Sim	Caro	Sim	Não
Web	Parcial	Sim*	Ótimo	Não	Sim
Híbrido	Sim	Sim*	Ótimo*	Sim	Sim*

Fonte: (PREZOTTO; BONIATI, 2014)

Após o desenvolvimento da aplicação, é feita a comparação da mesma dos recursos desenvolvidos em diferentes plataformas. Os testes foram realizados em dispositivos reais e emuladores. Também foram utilizadas sistemas operacionais de dispositivos móveis diferentes para a comparação da execução da aplicação criada.

2.3 An Evaluation of Cross-Platform Frameworks for Multimedia Mobile Applications Development (FERREIRA et al., 2018)

Em (FERREIRA et al., 2018) apresentam um estudo sobre o uso de *frameworks* no desenvolvimento de aplicações *cross-platform* e buscam analisar o atual estado de maturidade dos mesmos. Esse estudo foi realizado desenvolvendo uma aplicação em três diferentes *frameworks* (PhoneGap, Titanium e Sencha) assim como em plataformas nativas (Android e IOS) para ser realizada uma prova de conceito. Os dois pontos principais do estudo de (FERREIRA et al., 2018) é saber se aplicações *cross-platform* dispõem de recursos suficientes para o desenvolvimento de aplicações móveis que utilizam sensores e recursos multimídia assim como em relação ao desempenho das aplicações comparada às versões nativas. Algumas das métricas de comparação utilizadas por (FERREIRA et al., 2018) são:

- Linguagem;
- *LoC e NoF*(Número de linhas e funções do código);
- Tempo médio de Inicialização;
- Dificuldades;
- Documentação.

Como resultado foi visto que a maior parte das funcionalidades foram implementadas em todas as versões. É destacado que os *frameworks* utilizados apresentam um tempo de execução maior comparado ao nativo. Com isso, foi concluído que tais aplicações *cross-platform* ainda não são recomendadas para casos em que o tempo de execução de algoritmos é essencial.

Este trabalho também busca utilizar de métricas para uma comparação entre os modelos de desenvolvimento. Mas diferente do estudo proposto por (FERREIRA et al., 2018), esse estudo inclui PWA entre os modelos de desenvolvimento móvel e tem comparação mais teórica e qualitativa para atingir o resultado.

2.4 Comparação deste trabalho com trabalhos relacionados

No Quadro 1, são destacadas as principais diferenças entre o presente trabalho com os trabalhos relacionados.

Quadro 1 – Comparações deste trabalho com os trabalhos relacionados.

Métricas de Comparação	Presente Trabalho	(MAJCHRZAK; BIØRN-HANSEN; GRØNLI, 2018)	(PREZOTTO; BONIATI, 2014)	(FERREIRA et al., 2018)
<i>Objetivo Principal</i>	Comparação dos métodos de Desenvolvimento	Avaliação de Desenvolvimento PWA	Comparação de frameworks de desenvolvimento híbrido	Comparação de frameworks no desenvolvimento de aplicações cross-platform
<i>Nativo</i>	Sim	Não	Não	Não
<i>Híbrido</i>	Sim	Não	Sim	Sim
<i>PWA</i>	Sim	Sim	Não	Não

Fonte: Elaborado pelo autor.

3 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados os principais conceitos aplicados neste trabalho. Na Seção 3.1 é apresentado o histórico e o funcionamento do modelo de desenvolvimento aplicações PWA. Na Seção 3.2 é apresentado o histórico e o funcionamento do modelo de desenvolvimento de aplicações nativas. Na Seção 3.3 é apresentado o histórico e o funcionamento do modelo de desenvolvimento de aplicações híbridas. Na Seção 3.4 são apresentadas as métricas de avaliação para a comparação dos modelos de desenvolvimento.

3.1 Aplicações PWA (Progressive Web Apps)

O modelo de desenvolvimento PWA foi proposto em 2015 no trabalho de (RUSSELL, 2015) descrevendo novas funções que são suportadas pelos *browsers* atuais. No geral, esses *browsers* permitem que os usuários tenham a atualização de aplicativos web para aplicativos de execução nativa do sistema operacional utilizado. Segundo (LEPAGE, 2018) e de (RUSSELL, 2015), algumas das características principais de PWA são:

- Progressivo: Funciona para qualquer usuário, independentemente do navegador escolhido;
- Responsivo: Se adapta bem visualmente aos diferentes tipos de resolução de tela;
- Independente de conectividade: Aprimorado com o service worker para trabalhar offline ou em redes de baixa qualidade.
- Semelhante a aplicativos: Parece com aplicativos para os usuários, com interação e navegação de estilo de aplicativos.
- Compartilhável: Compartilhado facilmente por uma URL, não requer uma instalação complexa;
- Seguro: Fornecido via HTTPS para evitar invasões e garantir que o conteúdo não seja adulterado;
- Engajável: Facilita o reengajamento com recursos como notificações push;
- Instalável: Permitem que os usuários guardem os aplicativos na tela inicial sem precisar acessar uma loja de aplicativos;
- Atualizado: Sempre atualizado devido ao processo de atualização do service worker;
- Detectável: Capaz de ser identificado como aplicativo devido ao manifesto que permitem com que mecanismos de pesquisa o encontrem.

3.1.1 Funcionamento do PWA

Aplicação PWA é uma aplicação web aprimorada que utiliza algumas tecnologias que permitem um comportamento nativo no dispositivo através de APIs do navegador (FRANSSON;DRIAGUINE, 2017). Nesse tipo de aplicação, faz-se o uso essencialmente de características chaves como *Service Worker* e *Application Shell*. Pelo fato desse tipo de aplicação necessitar apenas de um navegador, isso possibilita sua execução através de qualquer dispositivo que tenha um navegador instalado, possibilitando uma grande variedade de plataformas suportadas. Essa mescla de melhores práticas das funcionalidades da web e característica que faz com que a aplicação seja mais semelhante a aplicativos faz com que se possa unir a melhor experiência da web e características de aplicativos.

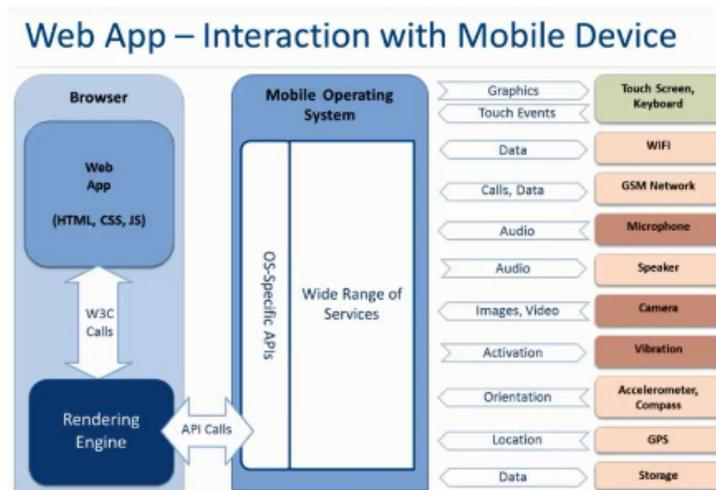
O *Service Worker* é um script JS responsável por executar no *background* separado do navegador. Esse script é capaz de responder a eventos que são executados dentro da aplicação, isso permite que se tenha um maior controle do cache da aplicação em geral. Através do *service worker* é possível armazenar o conteúdo necessário para a aplicação funcionar razoavelmente mesmo sem acesso à internet e permitir a experiência do usuário mesmo offline. Assim, o usuário precisa acessar a aplicação apenas a primeira vez para que todo conteúdo necessário seja armazenado no cache, isso faz com que não haja requisições de conteúdo toda vez que o usuário entrar na aplicação. Essa funcionalidade proporciona uma flexibilidade maior para decidir qual é o melhor cenário para buscar informações externas através de requisições quando o acesso à internet estiver disponível.

O *Service Worker* também é usado como uma maneira de melhorar o desempenho da aplicação no que se refere a entrega e apresentação do conteúdo mais rápido. Isso acontece pelo fato do próprio cache poder fornecer o conteúdo necessário para a execução da aplicação sem ser necessário qualquer tipo de requisição externa de conteúdo. Um outro fator bastante importante na melhoria de desempenho é o *Application Shell* que é o mínimo de HTML, CSS e Javascript necessário para que seja apresentada a interface do usuário (MISHRA, 2016). Combinando esses dois conceitos é possível obter o conteúdo necessário a ser mostrado e também a estrutura básica do layout na qual esse conteúdo será exibido de uma maneira mais otimizada.

Progressive Web Apps se fundamentam essencialmente em fazer o uso de funcionalidades disponibilizadas pelo navegador para atingir as principais características que definem um PWA. A aplicação consegue ser executada em diferentes plataformas pelo fato de

ser necessário apenas o uso do navegador para usá-la. Apesar de existirem algumas diferenças e limitações nos navegadores por parte principalmente do iOS, isso é algo que vem mudando ao longo dos últimos anos. A partir da versão iOS 11.3 será possível fazer o uso de service worker e Web App manifest na plataforma (FIRTMAN, 2018). Na Figura 3 é explorado como uma aplicação PWA funciona em um dispositivo móvel.

Figura 3 – Diagrama de uma chamada de uma API PWA.



Fonte: (SANTOS, 2014)

3.2 Aplicações Nativas

As aplicações nativas são desenvolvidas para uma única plataforma, com a capacidade de executar e explorar todas as funções que tal plataforma em questão dispõe. Ou seja, os aplicativos que são criados exclusivamente para iOS, Android e/ou Windows Phone, entre outras, conseguem acessar todo o potencial do dispositivo através da própria arquitetura do sistema operacional, como câmera, calendário, álbum de fotos, GPS, entre outros. É dito que um aplicativo nativo é todo programa construído sob medida para uma única plataforma, com o intuito de funcionar sob medida para o dispositivo em conjunto com suas especificidades. O desenvolvimento de aplicações móveis nativas teve início juntamente com a popularização dos *smartphones* na segunda metade da década passada e se transformando numa tendência para a década seguinte e até o presente momento (MONDE, 2011).

3.2.1 *Funcionamento das Aplicações Nativas*

Aplicações nativas são focadas em desenvolver uma aplicação para uma plataforma específica, seja IOS ou Android. Como consequência, esse tipo de aplicação é executada apenas para a plataforma específica a qual ela foi desenvolvida. Nesse modelo de desenvolvimento móvel o código base muda completamente de uma plataforma para outra, cada plataforma tem ferramentas como linguagem de programação e IDE completamente divergente. Nesse tipo de aplicação faz-se o uso de SDK (Software Development Kit) que proporciona usar um conjunto de funcionalidades disponíveis na plataforma. Algumas dessas funcionalidades são mencionadas por (KUMAR, 2014):

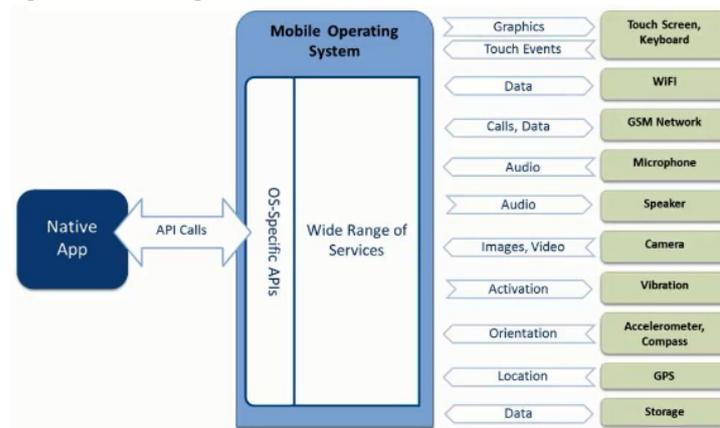
- Multi-touch: Gestos compostos de interface do usuário;
- API gráficas rápidas: gráficos extremamente rápidos;
- Animação fluida: Altamente interativos e uso intenso de algoritmos computacionais;
- Componentes embutidos (nativos): Câmera, geolocalização e lista de contatos;
- Interação: Capaz de integrar com outros aplicativos na tela inicial;
- Flexibilidade: Responde a comandos nativos como controle de volume;
- Documentação: Uma quantidade muita alta de material disponível para IOS e Android.

Um fator que geralmente é levado em consideração quando se fala sobre desenvolvimento nativo é o desempenho. Nesse tipo de abordagem existe um ambiente mais apropriado para a execução da aplicação, é possível seguir guias do próprio sistema operacional para uma melhor experiência do usuário. A grande vantagem desse modelo é que é permitido utilizar de capacidades integradas do próprio dispositivo, não existe uma camada adicional para o acesso dessas funcionalidades nativas do dispositivo (ABED, 2016). Um dos motivos do ótimo desempenho por parte das aplicações nativas também citados por (ABED, 2016) é que a própria estrutura da aplicação é mais rápida e confiável. Isso acontece pelo fato de que o conteúdo, estrutura e componentes visuais já estarem embutidos no próprio dispositivo e proporcione um carregamento instantâneo.

Em aplicações nativas também é possível ter suporte de armazenar dados para serem usados quando offline. De acordo com (VIJAYAJUMAR, 2017), existem diversas formas de lidar com capacidade offline e sincronização de dados em aplicações nativas. Existe a possibilidade de sincronização de dados inicial que é realizada assim que um novo dado é requisitado e guardado localmente. Outra possibilidade bastante importante é a capacidade de sincronizar dados locais

para servidores, essa funcionalidade é bastante útil em ocasiões onde é necessário que a aplicação seja capaz de armazenar dados locais durante o uso da aplicação e envie os dados previamente salvos uma vez que a conexão com a internet é restabelecida. Essa sincronização é capaz pelo fato da aplicação receber um *callback* do próprio sistema operacional para que a sincronização seja realizada. Na Figura 4 é explorado como uma aplicação nativa funciona em um dispositivo móvel.

Figura 4 – Diagrama de uma chamada de uma API Nativa.



Fonte: (SANTOS, 2014)

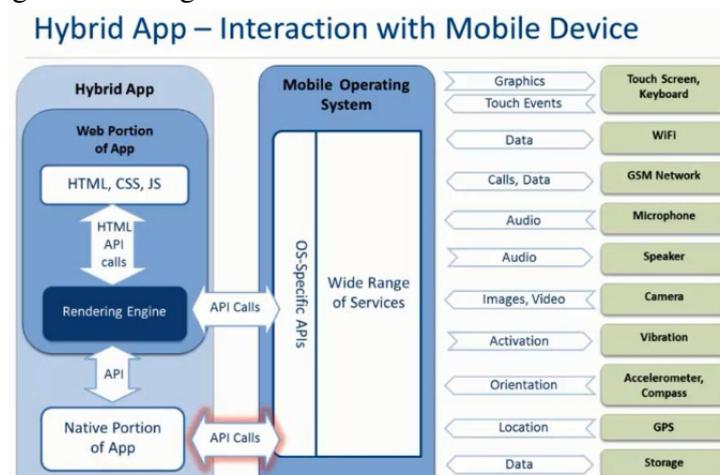
3.3 Aplicações Híbridas

Desenvolvimento híbrido utiliza essencialmente de HTML, CSS e JavaScript para desenvolver as aplicações, uma vez criada com essas tecnologias a aplicação é executada em containers de aplicações nativas que é o que proporciona o acesso a pacotes nativos (KHO'I; JAHID, 2016). Para ter uma maior facilidade geralmente se faz o uso de frameworks como Ionic, Xamarim e NativeScript, o dispositivo é capaz de executar tais aplicações em uma web view, que age como um navegador para ser capaz de mostrar o código de uma aplicação híbrida. Com o código baseado em tecnologia web, é possível utilizar do mesmo código base para gerar aplicações para diferentes plataformas com um esforço relativamente mínimo. (HOLZER; ONDRUS, 2012) consideram que aplicações híbridas estão em um nível intermediário entre aplicações nativas e web apps em relação a portabilidade do código. Em aplicações híbridas os componentes nativos usados também sofrem de algumas limitações assim como aplicações PWA, por outro lado os componentes web se beneficiam em relação a portabilidade da mesma maneira que web apps.

Em termos de desempenho, aplicações híbridas trazem uma certa falta de confiança pelo fato de aplicações nativas usarem componentes embutidos. Aplicações híbridas fazem o uso de elementos de interface de usuário que não são nativos e isso afeta a experiência geral que o usuário tem sobre o desempenho da aplicação. De acordo com (ABED, 2016), em aplicações híbridas os usuários tendem a criticar mais essas aplicações por questões relacionadas ao desempenho e elementos de interface de usuário. Aplicações híbridas tem um desempenho considerado mediano devido ao uso de elementos nativos que fazem com que o desempenho da aplicação seja melhorado consideravelmente. Por outro lado, existem os elementos web que são considerados mais devagar que fazem com que esse tipo de aplicação perca em performance (HOLZER; ONDRUS, 2012). Outro fator bastante importante sobre desempenho em aplicações híbrida é o fato dessas aplicações serem renderizadas em web views. O problema de desempenho surge devido aos diferentes tipos de navegadores e suportes dados aos mesmos em diferentes sistemas operacionais (LINDENBERG, 2011).

Em aplicações híbridas é possível utilizar diferentes alternativas para tornar uma aplicação com funcionalidades offline. Assim como em web app, é possível utilizar do localStorage que vem como padrão nos navegadores para armazenar dados localmente no dispositivo, nesse tipo de armazenamento é usado uma chave para guardar e buscar um valor. Outra alternativa seria fazer o uso do banco de dados SQLite que proporciona . Uma grande vantagem de usar SQLite é em relação ao nível de segurança que esse banco possibilita, é possível por exemplo encriptar arquivos contidos dentro do banco de dados usando SQLCipher (KHO'I;JAHID, 2016). Na Figura 5 é explorado como uma aplicação híbrida funciona em um dispositivo móvel.

Figura 5 – Diagrama de uma chamada de uma API híbrida.



Fonte: (SANTOS, 2014)

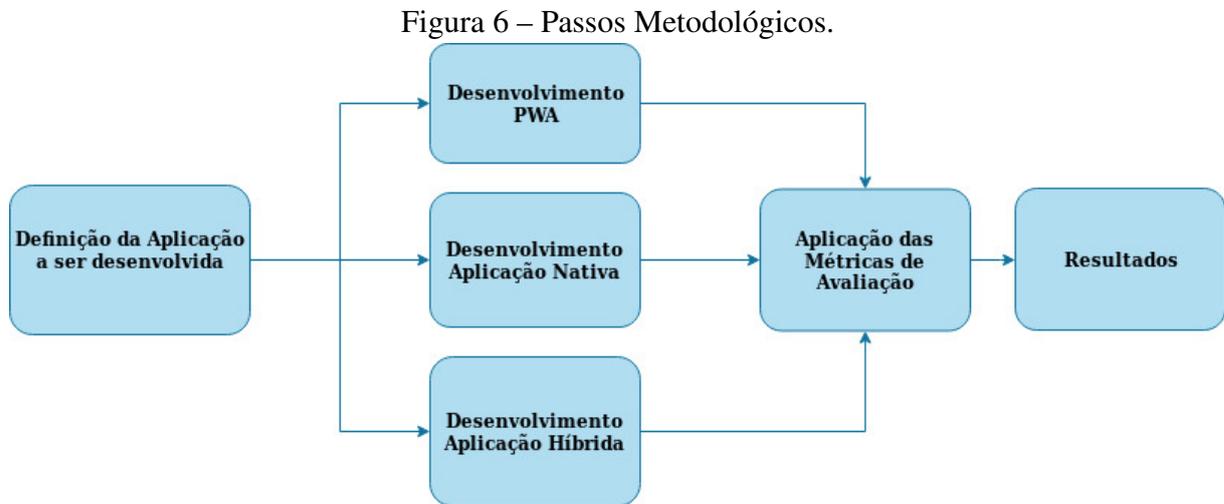
3.4 Métricas de avaliação

Para avaliar os modelos de desenvolvimento citados nas sessões anteriores, serão utilizadas métricas subjetivas e objetivas a fim de avaliação. As métricas utilizadas nesse trabalho, baseando-se nos trabalhos relacionados e em demais trabalhos acadêmicos (MAJCHRZAK; BIØRN-HANSEN; GRØNLI, 2018), (PREZOTTO; BONIATI, 2014), (FERREIRA et al., 2018) e (MISHRA, 2016), serão:

- **Métricas de Desenvolvimento:** são métricas utilizadas para avaliar a dificuldade de implementação das aplicações em cada modelo de desenvolvimento proposto assim como o resultado alcançado.
 1. Curva de aprendizagem: que medirá as dificuldades encontradas durante o desenvolvimento com cada abordagem. De forma objetiva, ela é uma representação do nível médio cognitivo de aprendizagem para uma determinada atividade ou ferramenta;
 2. Comunidade: onde será verificado todo o suporte existente da comunidade ativa de desenvolvimento para cada metodologia utilizada;
 3. Documentação: será verificada a clareza da documentação, se atende as necessidades do desenvolvedor e se utiliza exemplos práticos.
 4. *LoC* e *NoF*: *LoC* corresponde a quantidade de linhas utilizadas na codificação da aplicação, e *NoF*, o número de funções utilizadas.
- **Métricas de Desempenho:** são métricas utilizadas para verificar e validar o desempenho das aplicações criadas, as utilizadas nesse trabalho serão:
 1. Carregamento de conteúdo, onde serão verificadas o comportamento e a fluidez do carregamento dos conteúdos presente na aplicação;
 2. Uso de memória, que de maneira mais direta medirá a quantidade de memória a aplicação demanda do dispositivo.
 3. Capacidade Offline, que é uma forma de verificar o funcionamento da aplicação em estado offline. As métricas de avaliação de desempenho serão importantes também nessa fase.

4 PROCEDIMENTOS METODOLÓGICOS

Neste capítulo são apresentados todos os passos da execução deste trabalho. Na Figura 6 pode ser visto todo o fluxo de trabalho que foi empregado. Inicialmente foi realizado um estudo de caso sobre a aplicação a ser desenvolvida (Seção 4.1), logo após foi realizada a implementação da aplicação definida, utilizando cada modelo de desenvolvimento proposto por esse trabalho e também realizados casos de teste (Seção 4.2), por fim foram submetidas aos testes de comparação entre as abordagens utilizando as métricas descritas (Seção 4.3).



Fonte: Elaborada pelo autor.

4.1 Definição da Aplicação Desenvolvida

Nesta Seção é definida a aplicação a ser desenvolvida, assim como os requisitos necessários para gerar uma avaliação e comparação entre as abordagens de desenvolvimento propostas.

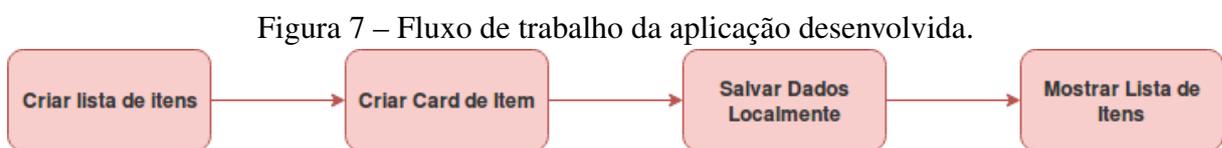
Para definir a aplicação a ser desenvolvida, foi feito um estudo de caso com foco nas funcionalidades que contribuem para as métricas de avaliação deste trabalho e que mais são utilizadas nas aplicações móveis atuais. Como o objetivo do trabalho se trata de comparar os modelos de desenvolvimento, a aplicação escolhida fornece, de forma quantitativa e qualitativa, um bom cenário de avaliação para comparar os modelos aqui propostos.

A aplicação escolhida se trata de um ambiente com grande demanda de carregamento de imagens e com redirecionamento para telas informativas individuais de cada instancia. Com essa aplicação é possível medir a as dificuldades encontradas para cada

abordagem de desenvolvimento e o suporte que a comunidade de cada modelo dispõe. O desempenho das aplicações deve ser medido verificando o carregamento de conteúdo na aplicação assim como o uso de memória e a capacidade do funcionamento offline da aplicação.

4.2 Desenvolvimento da Aplicação

O fluxo de trabalho da aplicação desenvolvida, foi idealizado pensando nos cenários de utilização a qual é submetida na etapa de testes e comparação das métricas de avaliação. Na Figura workflow pode ser visto o fluxo de execução/trabalho da aplicação.



Fonte: Elaborada pelo autor.

Inicialmente é necessário criar uma lista de itens, onde a criação de cada item é individual, apresentando os atributos que cada item deve ter. Em seguida é criado o card individual do item criado, que é onde apresenta a descrição individual de cada item. Após isso os dados do item criado são salvos localmente. Essa etapa é importante para alguns cenários de utilização que foram utilizados na etapa de testes. Por fim, a aplicação apresenta a lista com todos os itens criados e salvos localmente.

A seguir é descrito o processo de desenvolvimento da aplicação definida em cada um dos modelos de desenvolvimento propostos por esse trabalho. Para auxiliar na compreensão das diferenças do desenvolvimento entre as abordagens, são demonstradas as seguintes ferramentas para cada modelo de desenvolvimento:

- Ambiente de Desenvolvimento;
- Plataforma de execução da aplicação;
- Sistema de controle de versão;
- Linguagens de Programação utilizadas;

Com essas premissas utilizadas, o quadro 2 evidencia a diferença da implementação da aplicação definida. Observando apenas para o kit de ferramentas de cada um dos modelos, é possível notar que há uma certa flexibilidade ao desenvolver uma aplicação PWA ou utilizar um desenvolvimento híbrido, pois as plataformas de execução são diversas.

Quadro 2 – Comparação entre ferramentas dos modelos de desenvolvimento.

Modelos de Desenvolvimento	Ambiente de Desenvolvimento	Plataforma de Execução da Aplicação	Sistema de controle de Versão	Linguagem de Programação
<i>PWA</i>	VSCode	Android, iOS, Windows, Macbook (qualquer plataforma que tenha um navegador)	Git	Typescript
<i>Híbrido</i>	VSCode	Android e iOS	Git	Typescript
<i>Nativo</i>	Android Studio	Android	Git	Java

Fonte: Elaborado pelo autor.

4.3 Aplicação das Métricas de avaliação

Para realizar a aplicação das métricas de avaliação, se baseando nas definições do atual trabalho, foram adotadas formas diretas que medem quantitativamente cada métrica utilizada. Definidas na fundamentação teórica deste trabalho, as métricas utilizadas são:

- Métricas de Desenvolvimento:
 1. Curva de aprendizagem;
 2. Comunidade;
 3. Documentação;
 4. *LoC* e *NoF*.
- Métricas de Desempenho:
 1. Carregamento de conteúdo;
 2. Uso de memória;
 3. Capacidade Offline.

Parte das métricas de desenvolvimento, como a curva de aprendizagem e documentação, são métricas subjetivas e que os trabalhos presentes na literatura quantificam de diversas formas. A forma adotada para comparar os modelos de desenvolvimento nessas métricas, apresenta uma discussão apontando as diferenças e semelhanças entre elas.

Para medir o desempenho da aplicação foi avaliado como carregamento de conteúdo de cada abordagem flui além de verificação de uso de memória das aplicações. Também é verificada a capacidade offline das aplicações criadas. A configuração do ambiente é mostrada no quadro 3.

Quadro 3 – Configuração dos ambientes utilizados para realização dos testes na aplicação desenvolvida.

<i>Configurações</i>	<i>Ambiente 1</i>
Sistema Operacional	Android 9 One Pie
Processador	4x 2.2 GHz Kryo 260 + 4x 1.8 GHz Kryo 260
Chipset	Snapdragon 660 Qualcomm SDM660
GPU	Adreno 512
RAM	4 GB

Fonte: Elaborado pelo autor.

Para verificar esses dados de cada aplicação foi utilizado o gerenciador de serviços em execução que o ambiente Android disponibiliza nas versões de desenvolvedor.

Para avaliar o desempenho de memória das aplicações desenvolvidas, são propostos cenários de teste da aplicação que servem para medir de maneira objetiva desempenho de cada versão da aplicação. Para isso são listados os seguintes cenários de utilização de cada aplicação desenvolvida:

- CT01: Consumo de memória na inicialização da aplicação;
- CT02: Consumo de memória na utilização das funcionalidades da aplicação;
- CT03: Memória cache utilizada pelas funcionalidades da aplicação;

Com esses cenários é possível comparar de forma objetiva o funcionamento da aplicação com cenários diferentes de execução e que serão carregadas em tempo de execução e monitorar seu consumo de inicialização.

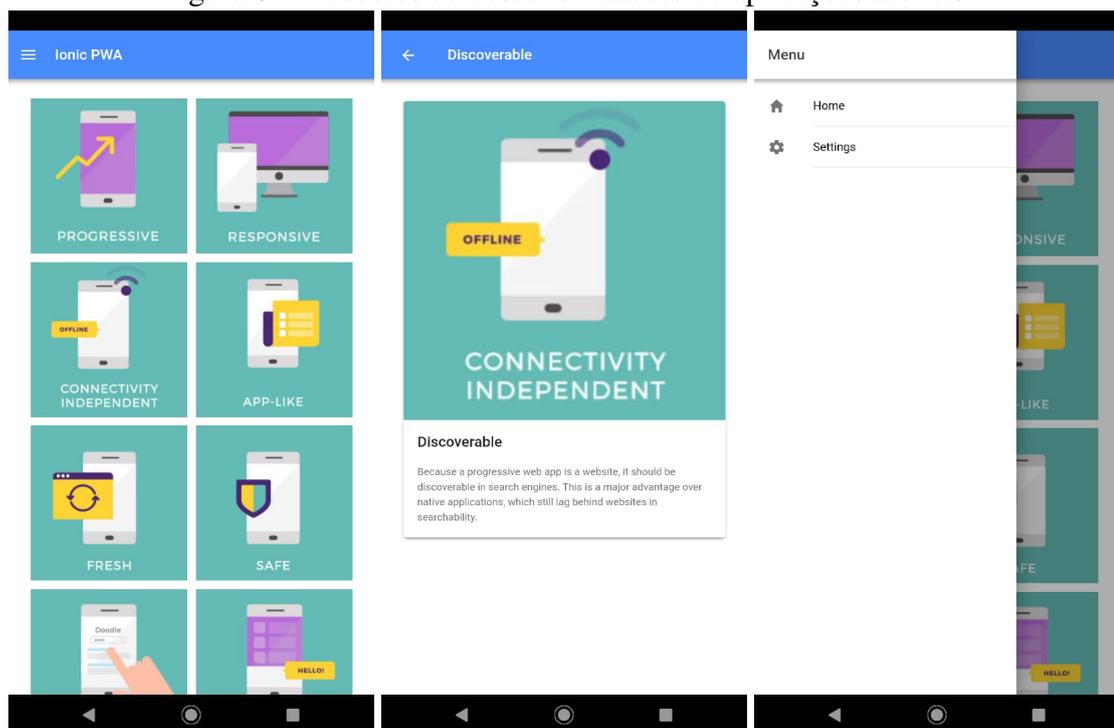
5 RESULTADOS

Nesta Seção são apresentados os resultados obtidos durante o trabalho. É importante enfatizar que o principal objetivo do trabalho é comparar os modelos de desenvolvimento de aplicações móveis com foco no desenvolvimento da aplicação e no desempenho apresentado.

5.1 Aplicação desenvolvida

Na Figura 8 são apresentadas as principais telas da aplicação desenvolvida utilizando PWA. As aplicações desenvolvidas de forma nativa e híbrida apresentam a mesma interface. A tela principal exibe a lista com os itens adicionados ao aplicativo, tais que ao serem clicados redirecionam para a página com sua respectiva descrição.

Figura 8 – Modelos de desenvolvimento de aplicações móveis



Fonte: elaborada pelo autor

Como o objetivo do trabalho é sintetizar os resultados de desempenho e de desenvolvimento da aplicação em diferentes modelos de desenvolvimento não houve preocupação sobre a interface de usuário da aplicação. A interface é simples, listando todos os itens como cards lado a lado em uma lista. Como visto na imagem da aplicação, a distribuição dos itens apresenta uniformidade e um espaçamento suficiente para diferenciar os itens listados.

5.2 Avaliação de desenvolvimento

Para avaliar a fase de desenvolvimento da aplicação são levadas em considerações as seguintes métricas:

- Métricas de Desenvolvimento:
 1. Curva de aprendizagem;
 2. Comunidade;
 3. Documentação;
 4. *LoC* e *NoF*.

Para avaliar a curva de aprendizagem, são comparados as principais dificuldades encontradas em cada um dos modelos de desenvolvimento e o tempo levado para desenvolver a aplicação em cada um deles.

Para avaliar a comunidade de cada modelo de desenvolvimento, são comparados o número de repositórios encontrados no github utilizando cada método e o número de questões no stackoverflow. Para buscar os repositórios foi utilizado uma busca no github inserindo o termo referente ao tópico do mesmo. Outros trabalhos utilizam métricas subjetivas a partir das análises feitas.

Para avaliar a documentação de cada modelo de desenvolvimento, são levadas em questão a clareza identificada, a qualidade e a quantidade de exemplos reais além do suporte dado caso haja alguma dúvida. Como resultado é dada uma nota subjetiva partindo da perspectiva do autor sendo Insuficiente, Regular, Bom e Ótimo.

Para avaliar o LoC e o NoF de cada modelo de desenvolvimento, são levadas em consideração apenas as principais funções da aplicação diretamente ligadas a home. Para fazer isso foi utilizado plugin LineCount do VsCode. O Quadro 4 apresenta resultados obtidos.

Quadro 4 – Métricas de Desenvolvimento.

Métricas de Desenvolvimento	Nativo	Híbrido	PWA
Principais Dificuldades	Sem experiência prévia. As APIs de banco de dados para salvar imagens e textos locais	Como foi utilizado o mesmo código base do PWA, não houve dificuldades.	Aplicar a capacidade offline utilizando service worker em algumas funcionalidades da aplicação como imagens e textos
Tempo de Desenvolvimento	12h	6h	6h
Nº Repositórios GitHub	56,749	18,853	3,435
Nº Questões StackOverflow	1,219,670	136,589	2,517
Clareza Documentação	Ótimo	Ótimo	Ótimo
Exemplos Práticos	Ótimo	Ótimo	Bom
Suporte	Ótimo	Ótimo	Bom
LoC	1148	772	772
NoF	42	17	17

Fonte: Elaborado pelo autor.

As considerações para os modelos de desenvolvimento na avaliação de desenvolvimento são:

- **Nativo:** foi o que exigiu mais tempo na etapa de desenvolvimento pela ausência de experiência prévia. No entanto, é visto que a comunidade existente é a maior e a mais sólida atualmente com bons resultados na clareza da documentação e na qualidade de exemplos práticos. Comparados aos demais modelos de desenvolvimento, o *LoC* e o *NoF* foi mais extenso;
- **Híbrida e PWA:** os resultados foram bem semelhantes pois havia experiência prévia. A comunidade do modelo Híbrido se mostrou maior que o PWA. A qualidade dos exemplos práticos e o suporte do modelo híbrido também foi julgado melhor que o modelo PWA. O *LoC* e o *NoF* de ambas foi igual, que mostra uma equivalência prática na etapa de desenvolvimento destes modelos.

5.3 Avaliação de desempenho

Para avaliar o desempenho da aplicação são levadas em considerações as seguintes métricas:

- Métricas de Desempenho:
 1. Carregamento de conteúdo;
 2. Uso de memória;
 3. Capacidade Offline;
 4. *LoC e NoF*.

Na avaliação do carregamento de conteúdo das aplicações foi verificado o comportamento de cada aplicação em relação a disposição dos itens e de sua fluidez. Com isso, para chegar em um resultado é dada uma nota subjetiva partindo da perspectiva do avaliador sendo Insuficiente, Regular, Bom e Ótimo.

Para realizar a avaliação do uso de memória da aplicação, foram feitas as verificações do consumo de memória pelo próprio gerenciador de serviços em execução do Android através do modo desenvolvedor. Com isso os Cenários de utilização CT01, CT02 e CT03 foram realizados para obter o valor de consumo de memória das aplicações.

A capacidade offline da aplicação da aplicação, diz respeito sobre o funcionamento da aplicação sem que haja rede. Com isso foram feitos os testes de operacionalidade em cada aplicação ao ter o ambiente offline a rede. Como é um critério subjetivo, para chegar a um resultado partindo da perspectiva do avaliador, se deu notas de Não Operacional ou Operacional, que são auto explicativas. No Quadro 5 são vistos os resultados obtidos nos testes de desempenhos propostos.

Quadro 5 – Métricas de Desempenho.

Modelos de Desenvolvimento	Carregamento de Conteúdo	CT01	CT02	CT03	Capacidade Offline
Nativo	Ótimo	5.8 Mb	6.6 Mb	57.12 Mb	Operacional
Híbrido	Bom	6.5 Mb	7.1 Mb	64.29 Mb	Operacional
PWA	Bom	7.1 Mb	8.8 Mb	70.94 Mb	Operacional

Fonte: Elaborado pelo autor.

As considerações para os modelos de desenvolvimento na avaliação de desempenho são:

- **Nativo:** foi o modelo de desenvolvimento que mais se sobressaiu em relação ao desempenho do uso de memória do ambiente a qual é executado. Como desenvolver dessa forma ocasiona o uso componentes diretamente, sem utilização de um ambiente de compilação de instruções, era esperado. A Capacidade Offline desse modelo teve o status operacional por conseguir mostrar todo o conteúdo ao ser executado. O carregamento de conteúdo se mostrou superior em relação aos demais modelos de desenvolvimento por carregar com mais rapidamente e com mais fluidez o conteúdo da aplicação;
- **Híbrida e PWA:** os resultados, assim como na avaliação de desenvolvimento, foram semelhantes. O carregamento de conteúdo de ambas apresentou uma fluidez e rapidez considerável e bastante similar. Nos cenários de teste, o modelo Híbrido se mostrou superior, consumindo menos memória em todos os casos. No CT02 o modelo PWA demandou uma quantidade de memória bem superior aos demais modelos. Em ambos os modelos a capacidade offline também foi verificada apresentando operacionalidade com eficiência.

Todas as formas de desenvolver uma aplicação mobile apresentaram pontos positivos e negativos. Para a criação de uma aplicação mobile, o desenvolvedor tem a liberdade de escolher entre os modelos de desenvolvimento baseado na necessidade real de um projeto, ou, apoiado no conhecimento prévio existente de uma equipe. Apesar de desenvolver de forma nativa mostrar melhor desempenho e um suporte maior, aplicações que exigem mais rapidez no desenvolvimento e facilidade de compreensão da tecnologia talvez se saiam melhor utilizando um modelo Híbrido ou PWA. É válido notar que o projeto teste é uma aplicação mobile simples e que em um cenário de complexidade maior de uso de componentes e outros atributos de cada modelo de desenvolvimento pode revelar uma outra realidade de desempenho e complexidade. Para aplicações mais simples que demandam mais desempenho, sem a preocupação com a dificuldade e tempo de implementação, é recomendado a utilização do modelo de desenvolvimento Nativo. Caso não haja essa demanda maior de desempenho, desenvolver de forma Híbrida ou utilizando PWA, seja o mais recomendado.

6 CONSIDERAÇÕES FINAIS

A popularização dos *smartphones* fez com que surgissem as aplicações mobiles de diversos gêneros e com uma grande gama de funcionalidades. O cenário de como construir uma aplicação foi se moldando e mostrando as necessidades e prioridades. As formas de realizar o desenvolvimento dessas aplicações também foram aumentando e seguindo a demanda dos desenvolvedores.

Este trabalho teve como objetivo comparar os modelos de desenvolvimento realizando a criação de uma aplicação mobile utilizando os modelos de desenvolvimento Nativo, Híbrido e PWA. Esses modelos de desenvolvimento são atualmente a forma de como desenvolver uma aplicação mobile, cada um com suas características de semelhanças e diferenças.

Assim como os trabalhos acadêmicos realizados com essa vertente, a comparação das formas de desenvolver vai desde métricas que são de interesse do desenvolvedor, a métricas de desempenho da aplicação criada. Os resultados mostram que todos os modelos de desenvolvimento executam bem sua função, e que cada modelo de desenvolvimento tem sua particularidade. Enquanto o modelo Híbrido foi semelhante ao modelo PWA, o modelo de desenvolvimento Nativo foi a mais distinta.

A aplicação desenvolvida para obtenção dos resultados apresenta funcionalidades mais simples fazendo com que, em um possível trabalho futuro, conte com o desenvolvimento de uma aplicação mais complexa para verificar os resultados apresentados, afim de encaixar o cenário onde cada modelo de desenvolvimento realmente apresenta mais vantagens.

REFERÊNCIAS

- ABED, R. **Hybrid Vs Native Mobile Apps? The Answer Is Clear**. [S.l], 2016. Disponível em: <https://ymedialabs.com/hybrid-vs-native-mobile-apps-the-answer-is-clear/>. Acesso em: 17 mai. 2018.
- CABRERA, J. M. C. **Existen diferentes formas de desarrollar Apps: híbridas, nativas y web responsive**. 2018. Disponível em: <https://webappdesign.es/apps-hibridas-nativas-y-web-responsive/>. Acesso em: 27 jun. 2019.
- FERREIRA, C. M.; PEIXOTO, M. J.; DUARTE, P. A.; TORRES, A. B.; JUNIOR, M. L. S.; ROCHA, L. S.; VIANA, W. **An Evaluation of Cross-Platform Frameworks for Multimedia Mobile Applications Development**. [S.l.]: IEEE, 2018. 1206–1212 p.
- FIRTMAN, M. **Progressive Web Apps on iOS are here**. [S.l], 2018. Disponível em: <https://medium.com/@firt/progressive-web-apps-on-ios-are-here-d00430dee3a7/>. Acesso em: 15 mai. 2018.
- FRANSSON, R.; DRIAGUINE, A. **Comparing progressive web applications with native android applications: an evaluation of performance when it comes to response time**. [S.l: s.n], 2017. p. 1-58.
- HOLZER, A.; ONDRUS, J. Mobile app development: Native or web? in: **Proc. Workshop eBus.(WeB)**. [S.l.: s.n.], 2012., 2012.
- KHO'I, F. M.; JAHID, J. **Comparing Native and Hybrid Applications with focus on Features**. [S.l:s.n], 2016, p. 1-47.
- KUMAR, P. **Analysis of Native and Cross-Platform Methods for Mobile Application Development**. 2014. [S.l]: Tavant Technologies. White Paper, p. 1-9..
- LEPAGE, P. **Seu primeiro Progressive Web App**. [S.l], 2018. Disponível em: <https://developers.google.com/web/fundamentals/codelabs/your-first-pwapp/?hl=pt-br/>. Acesso em: 17 mai. 2018.
- LINDBERG, O. **10 steps to better hybrid apps**. [S.l], 2011. Disponível em: <https://medium.com/net-magazine/10-steps-to-better-hybrid-apps-e8e33831ea5e/>. Acesso em: 17 mai. 2018.
- LIPSMAN, A.; LELLA, A. **The 2017 US Mobile App Report**. [S.l.]: COMSCORE White Paper, 2017.
- MAJCHRZAK, T. A.; BIØRN-HANSEN, A.; GRØNLI, T.-M. Progressive web apps: the definite approach to cross-platform development. **Proceedings of the 13th International Conference on Web Information Systems and Technologies – WEBIST**, p. 344–351. , [S.l], 2018.
- MISHRA, R. Progressive webapp: Review. **International Research Journal of Engineering and Technology**. V. 3, n. 6, p. 2–3, [S.l], 2016.
- MONDE, L. **Les ventes mondiales de smartphones en hausse de 74 %**. [S.l], 2011. Disponível em: <https://www.lemonde.fr/technologies/article/2011/08/12/les-ventes-mondiales-desmartphones-en-hausse-de-741558812651865.html>. Acesso em: 15 jul. 2019.

PREZOTTO, E. D.; BONIATI, B. B. Estudo de frameworks multiplataforma para desenvolvimento de aplicações mobile híbridas. **Anais do EATI.** , ano, v. 4, p. 72–79, [S.l], 2014.

RUSSELL, A. **Progressive web apps: escaping tabs without losing our soul.** Infrequently Noted, [S.l], 2015.

SANTOS, H. M. B. d. **Conceção e desenvolvimento de aplicações móveis: estágios na empresa network electronics 2000 consultadoria e redes informáticas, lda.** Tese (Doutorado). [S.l: s.n], 2014.

SIMPSON, R. **Mobile and tablet internet usage exceeds desktop for first time worldwide 2016.** 2016. [S.l]. Disponível em: <http://gs.statcounter.com/press/mobile-and-tablet-internet-usageexceeds-desktop-for-first-time-worldwide/>. Acesso em: 16 mai. 2018.

VIJAYAJUMAR. **Handling Offline Capability And Data Sync In Native Mobile Apps.** [S.l], 2017. Disponível em: <https://www.ideas2it.com/blog/offline-sync-native-apps/>. Acesso em: 17 mai. 2018.