

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA

CAMPUS JABOATÃO DOS GUARARAPES

Pós-Graduação em Gestão e Qualidade em Tecnologia da Informação e
Comunicação

HENNING BARLY SUMMER DE MELO

**UMA PROPOSTA PARA O USO DE PROGRESSIVE WEB APPS EM
AMBIENTES DE COMPUTAÇÃO MÓVEL EM NUVEM**

Jaboatão dos Guararapes

2020

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
CAMPUS JABOATÃO DOS GUARARAPES

Pós-Graduação em Gestão e Qualidade em Tecnologia da Informação e
Comunicação

HENNING BARLY SUMMER DE MELO

**UMA PROPOSTA PARA O USO DE PROGRESSIVE WEB APPS EM
AMBIENTES DE COMPUTAÇÃO MÓVEL EM NUVEM**

Trabalho de Conclusão de Curso apresentado como requisito parcial para obtenção do diploma de Pós-Graduação *Lato Sensu* em Gestão e Qualidade em Tecnologia da Informação e Comunicação do IFPE Campus Jaboatão dos Guararapes, sob orientação do professor Nilson Cândido de Oliveira Júnior.

Jaboatão dos Guararapes

2020

FICHA CATALOGRÁFICA

M528p Melo, Henning Barly Summer de.

Uma proposta para o uso de progressive web apps em ambientes de computação móvel em nuvens / Henning Barly Summer de Melo; orientador Prof. Ms. Nilson Cândido. Jaboatão dos Guararapes: IFPE, 2020.

79f.; il.

Trabalho de Conclusão de Curso (Especialização em Gestão e Qualidade em Tecnologia da Informação e Comunicação) – IFPE - Campus Jaboatão dos Guararapes.

Inclui Referências.

1. Tecnologia da Informação e Comunicação 2. Computação móvel 3. Aplicativos móveis. 4. Desempenho - Avaliação. I. Melo, Henning Barly Suumer de. II. IFPE. III. Título.

CDD 005.3

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
CAMPUS JABOATÃO DOS GUARARAPES

Pós-Graduação em Gestão e Qualidade em Tecnologia da Informação e
Comunicação

HENNING BARLY SUMMER DE MELO

Trabalho de Conclusão de Curso
julgado adequado para obtenção do título
de Pós-Graduação *Lato Sensu* em Gestão
e Qualidade em Tecnologia da Informação
e Comunicação, defendida e aprovada por
unanimidade em 18/02/2020 pela Banca
Examinadora.

Orientador:

Prof. Me. Nilson Cândido de Oliveira Júnior
Campus Jaboatão dos Guararapes – IFPE

Banca Examinadora:

Prof. Dr. Francisco do Nascimento Júnior
Membro Interno – Campus Jaboatão dos Guararapes – IFPE

Prof. Dr. Luciano de Souza Cabral
Membro Interno – Campus Jaboatão dos Guararapes – IFPE

Resumo

O aumento no uso de aplicações móveis está diretamente associado ao desenvolvimento constante da Computação Móvel (CM), a exemplo dos *smartphones*, *tablets* e *notebooks*. No entanto, o uso de aplicações nesses dispositivos acaba sendo um problema para suas limitações de processamento, memória RAM, armazenamento, conexão de rede e, principalmente, bateria. Para mitigar essas limitações, uma área conhecida como Computação Móvel em Nuvem ou *Mobile Cloud Computing* – MCC tem apresentado algumas alternativas. Uma delas está na descarga computacional ou *Computing Offloading*, que consiste em fazer o envio completo ou parcial da execução de processos ou tarefas do dispositivo móvel para um recurso computacional remoto com maior poder de processamento. No entanto, a eficiência dessa descarga da execução em ambiente MCC pode ter relação direta com o tipo de aplicação utilizada pelo dispositivo móvel. Diante disso, o objetivo deste trabalho consiste em propor uma metodologia para o uso de Aplicações Web Progressivas ou *Progressive Web Apps* (PWA) para fazer o *offloading* de tarefas na Computação Móvel em Nuvem. A partir da metodologia proposta, foram desenvolvidos dois estudos de caso, envolvendo a detecção de faces e o cálculo da sequência Fibonacci. A avaliação desses estudos de caso foi feita através da execução de diferentes cargas de trabalho, usando a aplicação PWA em diferentes dispositivos móveis e através do *offloading* em containers Docker, utilizando as redes 3G, 4G e WiFi. Os resultados da avaliação foram obtidos a partir da métrica de tempo de execução das tarefas, observando-se que, em alguns casos, a execução local dessas tarefas é melhor que a execução na nuvem; enquanto em outros, a nuvem é uma melhor opção de processamento. A aplicação PWA desenvolvida ainda foi analisada utilizando a ferramenta de inspeção de PWA *LightHouse*, obtendo a nota máxima de 100 para o quesito performance da aplicação.

Palavras-chave: Computação Móvel em Nuvem, *Offloading*, Aplicações Móveis, *Progressive Web Apps*, Avaliação de Desempenho, Detecção de Faces, Sequência Fibonacci

Abstract

The increased use of mobile applications is directly associated with the constant development of Mobile Computing (CM), such as smartphones, tablets and notebooks. However, the use of applications on these devices turns out to be a problem for their processing limitations, RAM, storage, network connection and especially battery. To mitigate these limitations, an area known as Mobile Cloud Computing (MCC) has presented some alternatives. One of these is Computing Offloading, which consists of sending all or part of the execution of processes or tasks from the mobile device to a remote computing resource with higher processing power. However, the efficiency of this discharge from running in MCC environment can be directly related to the type of application used by the mobile device. Therefore, the objective of this work is to propose a methodology for the use of Progressive Web Applications or Progressive Web Apps (PWA) to offload tasks in Mobile Cloud Computing. Based on the proposed methodology, two case studies involving face detection and Fibonacci sequence calculation were developed. These case studies were evaluated by performing different workloads using the PWA application on different mobile devices and offloading in Docker containers using the 3G, 4G and WiFi networks. The results of the evaluation were obtained from the task execution time metric, noting that in some cases local execution of these tasks is better than cloud execution; while in others, cloud is a better processing option. The developed PWA application was further analyzed using the LightHouse PWA inspection tool, obtaining a maximum score of 100 for the application performance.

Keywords: *Mobile Cloud Computing, Offloading, Mobile Applications, Progressive Web Apps, Performance Evaluation, Face Detection, Fibonacci Sequence*

Sumário

Índice de Figuras	9
Tabela de Símbolos e Siglas	12
Agradecimentos	13
1. INTRODUÇÃO	14
1.1. Delimitação do Tema	17
1.2. Motivação e Justificativa	17
1.3. Objetivos e Contribuições	19
1.3.1. Objetivo Geral	19
1.3.2. Objetivos Específicos	19
1.3.3. Contribuições Esperadas	19
1.4. Estrutura do Trabalho	20
2. REFERENCIAL TEÓRICO E TRABALHOS RELACIONADOS	21
2.1. Computação Móvel	21
2.2. Computação em Nuvem	26
2.3. Computação Móvel em Nuvem	29
2.4. Progressive Web Apps	32
2.5. Avaliação de Aplicações na Computação Móvel em Nuvem	35
2.6. Trabalhos Relacionados	36
3. DESENVOLVIMENTO E AVALIAÇÃO DE PWA NA MCC	39
3.1. Planejamento – Definir objetivos da aplicação PWA na MCC	40
3.2. Modularização – Selecionar módulos de execução local e remoto	41
3.3. Implementação – Desenvolver arquivos da aplicação PWA na MCC	42
3.4. Inspeção – Analisar aspectos da aplicação PWA na MCC	45
3.5. Avaliação – Avaliar o desempenho da aplicação PWA na MCC	47
4. AVALIAÇÃO DA METODOLOGIA	49
4.1. Planejamento – Definir objetivos da aplicação PWA na MCC	49
4.2. Modularização – Selecionar módulos de execução local e remoto	50
4.3. Implementação – Desenvolver arquivos da aplicação PWA na MCC	52
4.4. Inspeção – Analisar aspectos da aplicação PWA na MCC	56
4.5. Avaliação – Avaliar a aplicação PWA na MCC	59
5. Resultados	66
5.1. Resultados Detecção de Faces	66

5.2. Resultados Cálculo do Fibonacci.....	70
6. CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS.....	73
6.1. Considerações Finais	73
6.2. Limitações.....	74
6.3. Trabalhos Futuros	74
REFERÊNCIAS	75

Índice de Figuras

Figura 1 – Exemplo de um SoC e seus componentes.....	23
Figura 2 – Principais Dispositivos da Computação Móvel	26
Figura 3 – Comparação entre a arquitetura de contêineres e virtualizada	27
Figura 4 – Composição da Computação em Nuvem	28
Figura 5 – Visão geral de um ambiente de Computação Móvel em Nuvem.....	30
Figura 6 – Funcionalidades acessíveis a partir de um PWA	33
Figura 7 – Metodologia para uso de PWA em ambientes de MCC	39
Figura 8 – Principais tipos de aplicações da Computação Móvel em Nuvem	40
Figura 9 – Exemplo da execução de tarefas em um ambiente de MCC.....	42
Figura 10 – Exemplo do Módulo de Execução no cache do <i>Service Worker</i>	43
Figura 11 – Exemplo das definições de um arquivo <i>Web Manifest</i> de um PWA	44
Figura 12 – Exemplo da ferramenta de análise de qualidade de PWA	46
Figura 13 – Resultados da análise do PWA com a ferramenta <i>LightHouse</i>	46
Figura 14 – Algoritmo recursivo para calcular números do Fibonacci	52
Figura 15 – Arquivos de cache do <i>Service Worker</i> da aplicação PWA	53
Figura 16 – Exemplo do arquivo <i>Web Manifest</i> da aplicação PWA.....	54
Figura 17 – Estrutura da aplicação remota para execução do <i>offloading</i>	55
Figura 18 – Utilização da ferramenta Ngrok para gerar URLs externas	56
Figura 19 – Resultados da análise da aplicação PWA com o <i>LightHouse</i>	57
Figura 20 – Aplicação PWA sendo acessada e instalada nos dispositivos	63
Figura 21 – Execução da tarefa de detecção de face na aplicação PWA	65
Figura 22 – Execução da tarefa de cálculo Fibonacci na aplicação PWA	65
Figura 23 – Tempo de execução imagens de Dimensão 1 na rede 3G, 4G e WiFi..	66
Figura 24 – Tempo de execução imagens de Dimensão 2 na rede 3G, 4G e WiFi..	67

Figura 25 – Tempo de execução imagens de Dimensão 3 na rede 3G, 4G e WiFi..68

Figura 26 – Tempo de execução números do Grupo 1 nas redes 3G, 4G e WiFi70

Figura 27 – Tempo de execução números do Grupo 2 nas redes 3G, 4G e WiFi71

Figura 28 – Tempo de execução números do Grupo 3 nas redes 3G, 4G e WiFi72

Índice de Tabelas

Tabela 1 – Relação dos tipos de infraestrutura de Nuvens	24
Tabela 2 – Relação dos tipos de infraestrutura de Nuvens	29
Tabela 3 – Relação das plataformas e aplicações móveis	33
Tabela 4 – Características que definem uma aplicação PWA	34
Tabela 5 – Exemplo de um plano de avaliação	48
Tabela 6 – Resultados da análise da aplicação pela ferramenta LightHouse.....	58
Tabela 7 – Tarefas das cargas de trabalhos da avaliação.....	60
Tabela 8 – Dispositivos móveis e servidor remoto da avaliação.....	60
Tabela 9 – Aplicação da equação (1) no planejamento da avaliação.....	64

Tabela de Símbolos e Siglas

API	<i>Application Programming Interface</i>
CPU	<i>Central Processing Unit</i>
CSS	<i>Cascading Style Sheets</i>
GSM	<i>Global System for Mobile Communications</i>
HTTP	<i>Hypertext Transfer Protocol</i>
HTTPS	<i>Hypertext Transfer Protocol Security</i>
HTML	<i>Hypertext Markup Language</i>
IaaS	<i>Infrastructure as a Service</i>
ISP	<i>Internet Service Provider</i>
JSON	<i>JavaScript Object Notation</i>
LTE	<i>Long Term Evolution</i>
MCC	<i>Mobile Cloud Computing</i>
MVC	<i>Model View Controller</i>
NIST	<i>National Institute of Standards and Technology</i>
NPM	<i>Node Package Manage</i>
PDA	<i>Personal Digital Assistent</i>
PaaS	<i>Platform as a Service</i>
PWA	<i>Progressive Web App</i>
RAM	<i>Random Aleatory Memory</i>
SaaS	<i>Software as a service</i>
SDK	<i>Software Development Kit</i>
SSD	<i>Solid State Disk</i>
SoC	<i>System on a Chip</i>
URL	<i>Uniform Resource Locator</i>
VM	<i>Virtual Machine</i>
Wi-Fi	<i>Wireless Fidelity</i>
1G	<i>Primeira Geração de redes móveis</i>
2G	<i>Segunda Geração de redes móveis</i>
3G	<i>Terceira Geração de redes móveis</i>
4G	<i>Quarta Geração de redes móveis</i>
5G	<i>Quinta Geração de redes móveis</i>

Agradecimentos

“Antes de agradecer, eu gostaria de lembrar todas as pessoas que, independentemente das mais cruéis e injustas condições a que foram submetidos, jamais deixaram de lutar e persistir em seus objetivos e sonhos”

Gostaria de começar agradecendo ao IFPE pela oportunidade de realizar esta especialização, bem como a todos os professores que compartilharam seus conhecimentos ao longo do curso de GQTIC. Em especial, gostaria de agradecer ao professor Roberto pela compreensão e todo o apoio para que continuasse com o curso.

Também agradeço aos colegas de turma que sempre mantiveram o respeito uns com os outros, deixo meu desejo de boa sorte a todos e todas que, assim como eu, reconhecem a dificuldade que foi chegar até o final de mais essa etapa. Agradeço ainda toda paciência do meu professor, orientador e coordenador Nilson Cândido por ter dado todo apoio possível no desenvolvimento deste trabalho de conclusão de curso. Foi uma satisfação tê-lo como professor e como orientador nesse projeto.

Por fim, agradeço a minha família e principalmente a minha princesa Hannah que me inspirou sempre nos momentos mais complicados, muito obrigado filha. Ainda agradeço todas as pessoas que sempre acreditaram e atuaram ao longo dos anos para que eu chegasse até aqui, meu muito obrigado. Por fim, não tenho palavras suficientes para agradecer o carinho, o amor e a dedicação e todo o apoio daquela que fez esse momento ser de fato possível: Evelin Elenin, obrigado por tudo!

1. INTRODUÇÃO

O uso de *smartphones* tem permitido que bilhões de pessoas no mundo utilizem as mais diferentes aplicações e sistemas antes só possíveis de serem acessados a partir de um terminal fixo, como um *desktop* (SARWAR; SOOMRO, 2013). Essa mudança de paradigma foi dada a partir da primeira década deste século, com o lançamento do *smartphone* da *Apple*, conhecido como *iPhone*¹. No entanto, só a partir do surgimento do sistema operacional *Android*² é que o uso dos *smartphones* ganhou popularidade entre os usuários, tornando-se mais acessível financeiramente, graças à plataforma *Android*, que foi desenvolvida para ser gratuita e acessível pelas fabricantes de *smartphones* para utilização em seus dispositivos (KADU *et al.*, 2012).

Assim, por ser uma plataforma aberta e sem custos, o *Android* foi rapidamente adotado por diversos fabricantes de *smartphones*, tendo como grande destaque a sul coreana *Samsung*, que entrou no mercado para competir com o *iPhone* da já consolidada *Apple* (KADU *et al.*, 2012). Atualmente, o mercado de *smartphones* está sendo disputado por diferentes fabricantes, o que permite aos consumidores escolher entre os mais variados modelos que são lançados periodicamente para atender aos mais diferentes perfis de usuários do mercado (JUNIOR *et al.*, 2019). No entanto, mesmo com o avanço no desenvolvimento de dispositivos com maiores capacidades computacionais, um ponto considerado crítico e limitador para o avanço da computação móvel é a bateria (FERNANDO *et al.*, 2013).

Isso está atrelado ao fato de que o desenvolvimento de baterias mais eficientes e com maior durabilidade não tem acompanhado os avanços computacionais dos *smartphones*, que incluem o aumento do processamento (CPU), da memória (RAM) e do armazenamento (SSD) (AKHERFI *et al.*, 2018). Essa melhoria computacional acaba elevando o preço desses dispositivos, devido aos altos investimentos para projetar e fabricar novos SoC's (*System on a Chips*) ou Sistema em um Chip, responsáveis por integrar componentes essenciais em um *smartphone* (LONGO *et al.*, 2015). No entanto, o aumento da capacidade computacional de um dispositivo móvel,

¹ <https://www.apple.com/br/iphone/>

² <https://www.android.com>

como número de CPUs, acabará refletindo em seu consumo energético, sendo um problema para as limitadas capacidades energéticas das baterias utilizadas nesses dispositivos que dependendo do uso não dura nem um dia inteiro (LO'AI *et al.*, 2016).

Assim, o aspecto energético, em termos de melhores baterias, é considerado um problema atual e não trivial para os avanços da computação móvel nos próximos anos (JIAO *et al.*, 2013). É importante destacar ainda que os dispositivos móveis, quando comparados com os computadores convencionais, possuem outras limitações, como o número de CPU, o tamanho da memória RAM e a capacidade de armazenamento de informações (RAHIMI *et al.*, 2014). Desta forma, essa é uma questão bastante relevante dentro da computação móvel, devido às demandas provenientes do uso crescente de aplicativos que exigem dispositivos com melhores capacidades computacionais e energéticas (GAD-ELRAB *et al.*, 2017).

A grande utilidade no uso dos *smartphones* está na mobilidade que estes proporcionam aos seus usuários, permitindo utilizar as mais diversas aplicações, como redes sociais, mensageiros, jogos, vídeos e outros (SILVA, J. S. *et al.*, 2017). No entanto, existe no dispositivo móvel uma característica que o torna realmente interessante quanto à praticidade na realização das mais diversas atividades: a capacidade de comunicação. Através de um *smartphone*, o usuário pode utilizar diferentes tecnologias de redes sem fio, como o WiFi, 3G, 4G e o mais recente 5G (EWALDSSON, 2016). Com isso, o dispositivo móvel poderá estar conectado sempre com a internet e assim utilizar os mais diferentes tipos de serviços, a partir de qualquer lugar que ofereça cobertura para a rede sem fio utilizada por ele.

Essa capacidade de se conectar através de redes sem fio, como o WiFi, torna possível transferir algumas tarefas do dispositivo móvel para serem executadas remotamente em computadores com maior poder computacional, como um servidor ou nuvem (GUO *et al.*, 2016). Assim, considerando a tríade, computação móvel, redes de comunicação e nuvens computacionais, surgiu uma área de estudo conhecida como *Mobile Cloud Computing* (MCC) ou Computação em Nuvem Móvel (RAHIMI *et al.*, 2014). A MCC visa desenvolver estudos e soluções para tornar os dispositivos móveis mais eficientes quanto ao uso de recursos considerados limitados, como CPU, memória RAM, armazenamento interno e, principalmente, bateria (XU *et al.*, 2015).

Uma alternativa para tornar o dispositivo móvel eficiente dentro da MCC é através do descarregamento computacional ou *computing offloading* (KUMAR *et al.*, 2013). Essa abordagem visa transferir do dispositivo móvel para um computador remoto a execução de métodos e tarefas, com objetivo de diminuir o uso dos recursos computacionais do dispositivo. Isso permite que o processamento seja realizado fora do dispositivo móvel sem consumir os seus limitados recursos computacionais e energéticos (JIAO *et al.*, 2013). Porém, essa transferência requer o uso de redes sem fio pelo dispositivo móvel, como a 3G, 4G, 5G e WiFi, mas, nem sempre, essas redes estarão disponíveis ou com uma qualidade de conexão satisfatória.

Diante disso, ao realizar o *offloading* em MCC através de uma conexão de rede com baixa qualidade, é possível que a execução seja menos eficiente que essa mesma execução quando realizada no próprio dispositivo móvel (FERNANDO *et al.*, 2013). Com isso, é importante observar que, ao enviar uma tarefa através do *offloading* para ser executada em um recurso computacional remoto, esse envio poderá ser afetado por diferentes aspectos da conexão de rede do dispositivo móvel, como a latência e largura de banda (RAJ, 2017). Desta forma, o *offloading* nem sempre trará benefícios para o dispositivo móvel, visto que, além da rede, sua eficiência também dependerá do tipo de aplicação móvel utilizada, da capacidade de processamento do servidor remoto e ainda da complexidade da tarefa ou do método que será executado no ambiente da computação móvel em nuvem (LI, Y. *et al.*, 2017).

Para isso, uma das formas de analisar quando o *offloading* é uma alternativa de execução capaz de tornar o dispositivo móvel mais eficiente em ambientes de MCC está na avaliação de desempenho de sistemas (ADS) ou *Performance Evaluation System* (PES) (JAIN, 1991). Através da avaliação de desempenho, é possível analisar quando o *offloading* é considerado uma boa estratégia de execução em relação a uma execução feita no próprio dispositivo móvel. Deste modo, pode-se avaliar uma aplicação na Computação Móvel em Nuvem utilizando métricas, como o tempo de execução e consumo de energia e assim determinar a relevância que esse tipo de aplicação pode ter quando utilizada em ambientes da MCC (LILJA, 2000).

Desta forma, é importante realizar a avaliação dos diferentes tipos de aplicações que podem ser utilizadas na MCC, sendo uma opção ainda a ser explorada

a de Aplicações Web Progressivas ou *Progressive Web Apps* – PWA (CASALS; SILVA, 2016; GRØNLI *et al.*, 2019). A partir do desenvolvimento e uso de uma aplicação PWA, é possível avaliar o quanto a execução do *offloading* de tarefas em MCC, através desse tipo de aplicação, pode ser considerado eficiente para o dispositivo móvel. Para isso, a avaliação de uma aplicação poderá ser feita em um ambiente de MCC com diferentes dispositivos móveis rodando o PWA para fazer a execução de tarefas nos próprios dispositivos “execução local” e em um servidor remoto ou nuvem computacional “execução remota na nuvem” (DENG *et al.*, 2015).

1.1. Delimitação do Tema

Este trabalho está delimitado quanto ao uso de *Progressive Web Apps* no ambiente da Computação Móvel em Nuvem. Assim, não serão aprofundados neste estudo aspectos mais amplos envolvendo outros tipos de aplicações utilizadas no ambiente de MCC. Neste sentido, não serão comparados resultados de outros tipos de aplicações com o uso do PWA desenvolvido nesse trabalho. Vale destacar ainda que o estudo em tela é focado apenas em aspectos específicos de como uma Aplicação Progressiva da Web pode ser utilizada em um ambiente de Computação Móvel em Nuvem, através de um cenário real, envolvendo dispositivos móveis, redes sem fio e servidores rodando em contêineres Docker em um host local.

1.2. Motivação e Justificativa

Este trabalho traz como principal justificativa o crescente aumento no uso de dispositivos móveis nos últimos anos. Isso pode ser visto através dos números da pesquisa realizada pela FVG (MEIRELLES, 2019), a qual aponta que no Brasil já existem em uso mais de 300 milhões de dispositivos móveis, como *smartphones* e outros computadores portáteis, como *tablets* e *notebooks*. De acordo com a pesquisa, existem em média no país mais de um *smartphone* por habitante, demonstrando a importância que esses dispositivos representam na vida da população. Isso também é graças ao uso crescente de aplicações acessíveis através desses dispositivos, fazendo deles verdadeiros canivetes suíços da era moderna (GOMES *et al.*, 2016).

Assim, o uso constante de aplicações, como redes sociais, mensagens, jogos e outras, torna o *smartphone* quase que indispensável na vida da maioria das pessoas, graças à comodidade e à conveniência de utilizá-lo em qualquer lugar. Contudo, mesmo com a evidência da importância desses dispositivos na vida de seus usuários, muitas vezes o seu uso acaba sendo considerado até um vício por conta da dependência causada em alguns indivíduos (CAPPELLOZZA *et al.*, 2017). O fato é que, através de um *smartphone*, é possível realizar uma infinidade de atividades, visto que a cada dia inúmeros aplicativos são desenvolvidos para os mais diversos fins. Ocorre que o uso constante de aplicações nesses dispositivos implica mais processamento e assim maior consumo energético (XU *et al.*, 2015).

Deste modo, mesmo considerando as novas tecnologias de carregamento rápido de bateria ou uso de carregadores portáteis, a questão energética ainda é um problema que não tem uma solução definitiva dentro da computação móvel (LI, Y. *et al.*, 2017). Com isso, é importante que os estudos dentro da MCC busquem atenuar esse problema, permitindo que novas propostas sejam desenvolvidas para que os dispositivos possam tornar-se cada vez mais eficientes quanto ao uso dos seus limitados recursos computacionais e energéticos. Diante disso, a apresentação de propostas para o desenvolvimento de novos tipos de aplicações como o PWA dentro da Computação Móvel em Nuvem é considerada relevante (CASALS; SILVA, 2016).

Diante disso, é possível ampliar as opções envolvendo o uso de aplicações na MCC, visto que novos tipos de aplicações podem trazer melhorias ou alternativas para o desenvolvimento de novas abordagens na Computação Móvel em Nuvem. Daí a importância de explorar novos tipos de aplicações na MCC, a exemplo do *Progressive Web Apps*, visto as possibilidades existentes em torno do desenvolvimento desse tipo de aplicação em trabalhos desenvolvidos dentro da MCC (CASALS; SILVA, 2016). Assim, é possível que novos tipos de aplicações como o uso de PWA possam trazer resultados relevantes, como melhorar a eficiência dos dispositivos móveis ao diminuir o uso de seus limitados recursos de CPU, RAM, SSD e bateria, através de uma aplicação multiplataforma capaz de fazer o *offloading* de tarefas na MCC.

1.3. Objetivos e Contribuições

Dada a relevância em torno do desenvolvimento de novos tipos de aplicações dentro da Computação Móvel em Nuvem que permitam tornar os dispositivos móveis mais eficientes, serão apresentados a seguir o objetivo geral e os objetivos específicos que norteiam o desenvolvimento desse trabalho. Além disso, também serão apresentadas as contribuições que são esperadas a partir do desenvolvimento desta proposta de pesquisa para a área de Computação Móvel em Nuvem.

1.3.1. Objetivo Geral

O principal objetivo deste trabalho é propor uma metodologia para o uso de aplicações *Progressive Web Apps* num ambiente de Computação Móvel em Nuvem, avaliando a utilização desse tipo de aplicação a partir da execução de tarefas em dispositivos móveis e em servidores remotos executados em contêineres Docker, através do *offloading* de diferentes tarefas sob as conexões de redes WiFi, 3G e 4G.

1.3.2. Objetivos Específicos

Buscando alcançar o objetivo geral deste trabalho, foram definidos os seguintes objetivos específicos:

- Fazer um levantamento bibliográfico da Computação Móvel em Nuvem que envolva o uso de *Progressive Web Apps* e de *Computation offloading*.
- Realizar um estudo sobre o desenvolvimento de *Progressive Web Apps* e quais as vantagens e desvantagens de sua utilização.
- Implementar estudos de casos, utilizando a metodologia apresentada para avaliar o PWA a partir da execução de tarefas em um ambiente de MCC.

1.3.3. Contribuições Esperadas

Entre as contribuições esperadas com este trabalho, está na possibilidade de desenvolvedores utilizarem a metodologia proposta para desenvolver *Progressive Web Apps* em ambientes da Computação Móvel em Nuvem. Isso servirá como mais

uma alternativa no desenvolvimento de aplicações multiplataformas dentro da MCC. Ainda é esperado com os resultados da avaliação demonstrar o quanto o uso de PWA pode ser considerado relevante no desenvolvimento de aplicações na Computação Móvel em Nuvem. Espera-se também que os resultados deste trabalho possam contribuir para que desenvolvedores e especialistas saibam quando a escolha desse tipo de aplicação é uma alternativa melhor que outros tipos de aplicações já usadas na MCC.

1.4. Estrutura do Trabalho

Para uma melhor compreensão deste trabalho, ele está dividido da forma a seguir. No Capítulo 2, está o referencial teórico contendo informações importantes sobre os principais temas e estudos relacionados com esta pesquisa, como Computação Móvel, Computação em Nuvem, Computação Móvel em Nuvem e o uso de *offloading*. Ainda são abordados o uso de aplicações móveis, como as Aplicações Web Progressivas e os principais trabalhos relacionados com esse tema.

O Capítulo 3 apresenta a metodologia proposta neste trabalho para o uso de Aplicações Web Progressivas PWA em ambientes da Computação Móvel em Nuvem. A metodologia encontra-se dividida em fases para facilitar o entendimento de sua aplicação para desenvolver e avaliar *Progressive Web Apps* em ambientes da Computação Móvel em Nuvem, através do *offloading* de tarefas.

Na sequência, o Capítulo 4 apresenta a avaliação dos dois estudos de casos desenvolvidos a partir da proposta neste trabalho, conforme as etapas descritas dentro da metodologia apresentada.

No Capítulo 5, estão os resultados obtidos a partir da avaliação realizada através dos estudos de caso desenvolvidos neste trabalho.

Por fim, o Capítulo 6 apresenta discussões finais sobre o desenvolvimento desta pesquisa, incluindo suas limitações e possíveis trabalhos futuros.

2. REFERENCIAL TEÓRICO E TRABALHOS RELACIONADOS

Neste capítulo, serão abordados os principais conceitos referentes ao tema deste trabalho, envolvendo o uso de PWA na Computação Móvel em Nuvem. No entanto, antes de abordar a MCC, é importante apresentar os conceitos básicos sobre Computação Móvel e a Computação em Nuvem. Em seguida, a Computação Móvel em Nuvem será explicada a partir da abordagem de *offloading*. Após, serão apresentados os conceitos sobre o desenvolvimento de *Progressive Web Apps* e como o uso desse tipo aplicação pode ser interessante dentro da MCC. Após isso, serão abordados os principais trabalhos de MCC envolvendo aplicações como o PWA.

2.1. Computação Móvel

Em termos históricos, o ábaco³ pode ser considerado o primeiro computador móvel utilizado pelo homem, dada a sua aplicação para resolver problemas de cunho matemáticos, como operações envolvendo números inteiros de base decimal (GUGIK, 2009). A calculadora digital é outro exemplo de computador móvel, uma vez que pode ser transportada para qualquer lugar e funcionar através do uso de baterias. No entanto, a computação atual é baseada nos princípios da *Máquina de Turing Universal*, que estabelece um modelo abstrato de um computador, onde a lógica consiste de (memórias, estados e transições), sendo a base dos computadores atuais, inclusive da Computação Móvel, como os *smartphones* (MARTINS, 2015).

Nesse sentido, a Computação Móvel, segundo Figueiredo e Nakamura (2003), consiste em um novo paradigma computacional, que permite aos seus usuários acessarem os mais variados serviços, independentemente de sua localização geográfica ou se estão ou não em movimento. Assim, dispositivos da Computação Móvel têm como característica principal oferecer aos seus usuários a mobilidade computacional de processamento de informações e de comunicação, sem a necessidade de estarem conectados por nenhum meio físico (FERNANDO *et al.*, 2013). Isso significa que a Computação Móvel não necessita da utilização de cabos

³ <https://pt.wikipedia.org/wiki/Abaco>

de redes ou de alimentação de energia, como os utilizados em computadores convencionais, como *desktops*.

Diante disso, os avanços da Computação Móvel têm representado uma verdadeira revolução no mundo da tecnologia, principalmente a partir do início do século XXI, com a popularização de dispositivos, como *smartphones*, *tabletes* e *notebooks* (KHAN, S. *et al.*, 2014). No entanto, os primeiros passos da Computação Móvel foram vistos a partir da década de 1980, com o surgimento do primeiro Assistente Pessoal Digital ou *Personal Digital Assisistent* (PDA), que teve como principal fabricante a empresa *BlackBerry*⁴. Esse tipo de dispositivo é considerado o precursor do *smartphone*, visto que na época já apresentava características, como telas *touchscreen* e uso de e-mails (FIGUEIREDO; NAKAMURA, 2003).

Nesse sentido, a Computação Móvel teve como marco o lançamento do *Iphone* em 2007, *smartphone* produzido pela fabricante de computadores *Apple*, que rapidamente ganhou destaque por suas diversas funcionalidades e principalmente pelo sucesso dos aplicativos desenvolvidos para esse dispositivo (ALIZADEH; HASSAN, 2013). Isso deu início a uma nova era, onde a mobilidade computacional rapidamente ganhou destaque, principalmente, com o surgimento do sistema operacional móvel *Android*, que por ser de uso gratuito fez com que fabricantes, como a *Samsung*, desenvolvessem dispositivos móveis mais acessíveis em termos financeiros por não ter atrelado ao preço de venda do dispositivo o valor do sistema operacional utilizado por eles (SARWAR; SOOMRO, 2013).

Assim, o desenvolvimento dos dispositivos da Computação Móvel partiu da necessidade de disponibilizar um computador capaz de processar e trocar informações via redes sem fio e com tamanho reduzido para facilitar ser transportado por seus usuários para qualquer lugar que assim desejarem (FIGUEIREDO; NAKAMURA, 2003). Com isso, a partir do desenvolvimento da computação móvel, foi possível acabar com a necessidade do usuário ter que ficar em frente a um computador de mesa, ou seja, estar em um lugar fixo para, assim, poder utilizar algum tipo de ferramenta ou aplicação específica, como mandar e-mails, assistir vídeos e

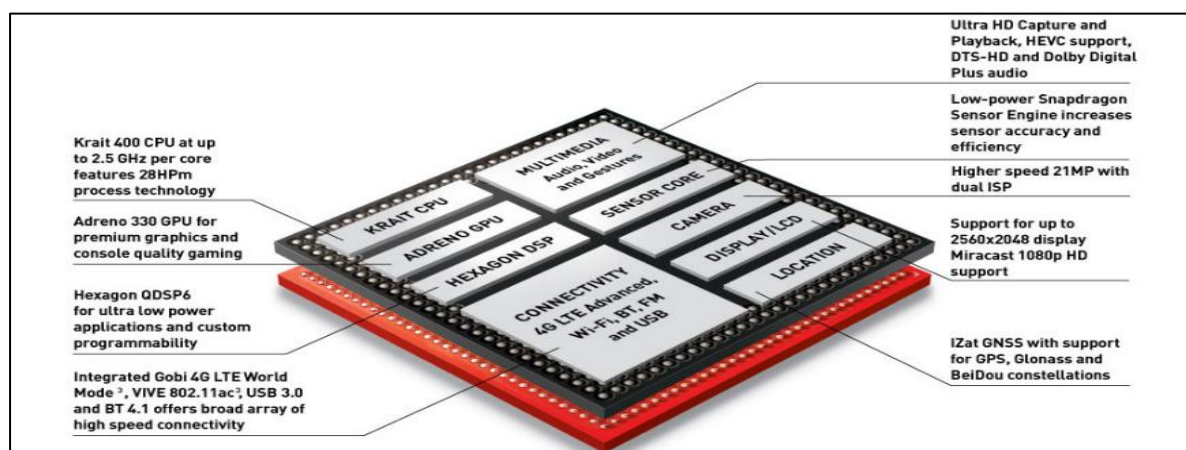
⁴ <https://pt.wikipedia.org/wiki/BlackBerry>

jogar (FERNANDO *et al.*, 2013). Entre as principais características que definem a Computação Móvel e podem ser citadas estão:

- **O processamento** – é a capacidade que o computador tem de fazer cálculos binários convertidos em saídas, como um cálculo matemático ou a conversão de uma imagem em pixel na tela do usuário.
- **A mobilidade** – é uma característica que permite que um dispositivo móvel seja utilizado em qualquer lugar, sem depender de conexões por meios físicos, como uma fonte de energia ou cabos de redes.
- **A comunicação** – é a capacidade de um dispositivo móvel se comunicar com outros dispositivos, utilizando uma conexão sem fio, como WiFi, 3G, 4G, 5G, Bluetooth, sendo essa uma premissa para a mobilidade.

Vale ressaltar que o desenvolvimento da Computação Móvel só foi possível graças às novas arquiteturas de hardware, que permitiram diminuir o tamanho de diversos componentes computacionais, dando origem aos chamados Sistemas em um Chip ou *System on a Chip* (SoC) (LONGO *et al.*, 2015). Os avanços no desenvolvimento desses chips estão diretamente associados com a evolução da mobilidade computacional de diferentes dispositivos móveis, visto que um SoC é capaz de integrar diferentes componentes em único chip, além de apresentar baixo consumo energético (KITANOV *et al.*, 2016). A Figura 1 apresenta o exemplo de um SoC e dos diversos componentes utilizados para compô-lo.

Figura 1 – Exemplo de um SoC e seus componentes



Fonte: (SMITH, 2014)

Além do SoC, é preciso considerar que o uso de baterias também foi fundamental para o desenvolvimento da Computação Móvel, tendo ainda destaque as indispensáveis redes de comunicação sem fio, (*wireless*), utilizadas por dispositivos móveis, como *tablets* e *smartphones*, como as redes *Bluetooth*⁵, WiFi, 3G, 4G e 5G. O uso dessas conexões permite aos dispositivos móveis comunicarem-se com outros computadores remotamente, a partir de qualquer lugar que ofereça cobertura para o tipo de conexão utilizada por eles (ALI *et al.*, 2016). Alguns detalhes das redes sem fio usadas na computação móvel podem ser vistos na Tabela 1.

Tabela 1 – Relação dos tipos de infraestrutura de Nuvens

Tipo de rede	Nome da rede	Ano que surgiu	Velocidade Máxima	Faixa de Frequência	Características de utilização
1G	GPRS	1980	~ 2,4 Kb/s	-	Voz Analógica
2G	GSM, EDGE	1990	~ 64 Kb/s	-	Voz Digital, SMS
3G	HSPA, WiMax	2003	~ 2 Mb/s	-	Voz e dados multimídia
4G	LTE, LTE-A	2009	~ 100 Mb/s	-	Vídeos de alta qualidade
5G	5ª Geração	2019	~ 5 Gb/s	-	Latência menor que 5 ms
802.11	WiFi 1	1997	~ 2 Mb/s	2.4 GHz	Transferência limitada
802.11b	WiFi 2	1999	~ 11 Mb/s	5 GHz	Melhor taxa transferência
802.11g	WiFi 3	2003	~ 54 Mb/s	2.4 GHz	Transmissão de vídeos
802.11n	WiFi 4	2008	~ 600 Mb/s	2.4 e 5 GHz	Vídeos em alta resolução
802.11ac	WiFi 5	2012	~ 7 Gb/s	5 GHz	Menor latência da rede
802.11ax	WiFi 5	2019	~ 10 Gb/s	2.4 e 5 GHz	Alta taxa de transmissão

Fonte: (KUMAR *et al.*, 2013; SHARMA, 2013)

No entanto, se comparados com os computadores convencionais, dispositivos, como os *smartphones* e *tablets*, são considerados limitados quanto aos seus recursos computacionais, de comunicação e principalmente de energia (URRIZA *et al.*, 2004). Entre as principais limitações encontradas em um dispositivo móvel que podem ser citadas estão as seguintes:

- **O Tamanho da tela** – A tela é menor que notebooks e outros computadores para permitir um melhor manuseamento e usabilidade do dispositivo.

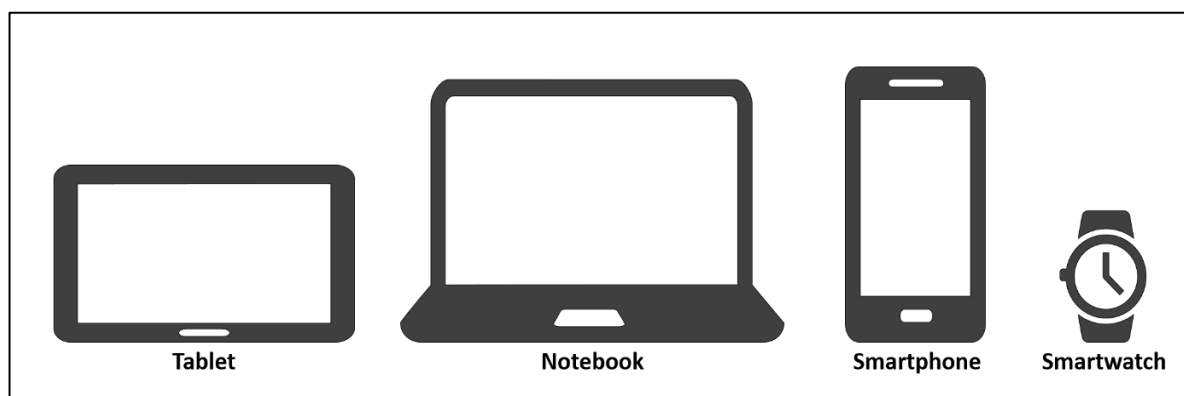
⁵ <https://pt.wikipedia.org/wiki/Bluetooth>

- **O Número de CPUs** – A CPU é fixa e não pode ser trocada por uma mais avançada como nos computadores convencionais.
- **O Tamanho da memória RAM** – Assim como a CPU, a memória RAM também é fixa e não pode ter seu tamanho ampliado.
- **A Capacidade de armazenamento SSD** – Apesar do uso de cartões que ampliam o armazenamento, o tamanho interno ainda é limitado.
- **A Conexão de Rede** – Apesar das novas tecnologias de conexões sem fio, a limitação está no uso de redes móveis pagas que limitam o uso de dados.
- **O Uso de Bateria** – A necessidade do uso de baterias impacta na utilização do dispositivo por longos períodos, devido a sua limitada capacidade energética.

Essas limitações representam um desafio para os avanços da computação móvel, visto que, ao aumentar o poder computacional de um dispositivo móvel, como CPU e memória RAM para torná-lo mais rápido, também será preciso levar em consideração o consumo de mais energia, devido a sua maior capacidade de processamento (CHAMAS, 2018). Além disso, os componentes que formam um dispositivo móvel são caros, e qualquer melhoria em termos computacionais, como aumento de memória RAM e CPU, terá impacto no valor do dispositivo fazendo com que fique mais caro para os usuários que desejarem adquiri-lo. Assim, é importante o desenvolvimento de alternativas que permitam minimizar tais limitações, sem a necessidade de grandes investimentos financeiros para aumentar o poder computacional desses dispositivos móveis (GOUDARZI *et al.*, 2016).

O uso crescente desses dispositivos está associado com a praticidade que eles fornecem aos usuários para realização de atividades diárias, por meio de aplicações e serviços que são acessíveis através desses dispositivos, como planilhas eletrônicas, editores de texto, players de músicas, vídeos, entre outros (COSTA, P. B. *et al.*, 2014). A Figura 2 apresenta alguns dos principais dispositivos desenvolvidos graças aos avanços tecnológicos da Computação Móvel.

Figura 2 – Principais Dispositivos da Computação Móvel



Fonte: Elaboração própria

Assim, considerando os conceitos básicos da computação móvel, é possível verificar que uma das características principais desses dispositivos está na sua capacidade de se comunicar com outros dispositivos e acessar recursos ou computadores remotos, como servidores ou nuvem computacionais (YU *et al.*, 2017). Com isso, surge a importância das tecnologias de redes de comunicação sem fio, que permitem que esses dispositivos possam se conectar através de diferentes tipos de conexão para acessar e utilizar os recursos de servidores remotos ou de nuvens computacionais (KHODA *et al.*, 2016).

2.2. Computação em Nuvem

A computação em nuvem é um modelo para habilitar o acesso por rede ubíquo, conveniente e sob demanda a um conjunto compartilhado de recursos de computação, como redes, servidores, armazenamento, aplicações e serviços que possam ser rapidamente provisionados e liberados com o mínimo de esforço, de gerenciamento ou interação com o provedor de serviços em nuvem (MELL; GRANCE, 2011). A computação em nuvem permite armazenar, compartilhar e disponibilizar dados, aplicações, serviços e softwares por meio da internet.

O *National Institute of Standards and Technology* (NIST) publicou algumas características essenciais da Computação em Nuvem (SANTOS, 2018). São elas:

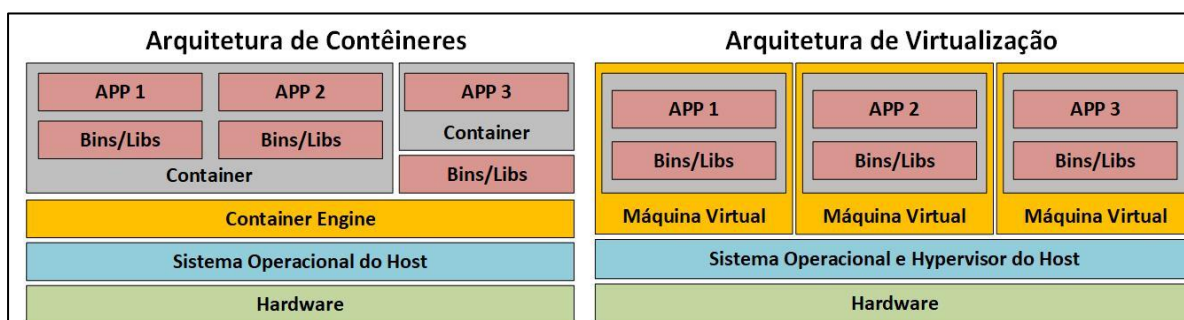
- **Autosserviço sob demanda:** O consumo é feito de forma automática. O usuário o pode provisionar por conta própria recursos de computação, como

tempo de servidor e armazenamento em rede, sem depender de intervenção humana dos provedores de serviços.

- **Amplio acesso por rede:** Os recursos estão disponíveis via internet e de maneira padronizada, promovendo o uso por meio dos mais diversos tipos de dispositivos, como smartphones, tablets, laptops ou desktops.
- **Agrupamento de recursos:** Os provedores de computação em nuvem agrupam recursos para atender a múltiplos consumidores em modalidade multi-inquilinos, com recursos físicos e virtuais diferentes, dinamicamente atribuídos e reatribuídos, conforme a demanda dos consumidores.
- **Rápida elasticidade:** Os recursos podem ser provisionados e liberados elasticamente, em alguns casos automaticamente, para rapidamente aumentar ou diminuir de acordo com a demanda.
- **Serviços mensuráveis:** A utilização de recursos pode ser monitorada, controlada e informada, gerando transparência, tanto para o fornecedor, como para o consumidor do serviço utilizado. Em geral, esses serviços são adquiridos na forma de “*pay per use*” - “pague por uso”.

Isso permite que sejam desenvolvidos e utilizados os mais diferentes tipos de soluções em nuvem, sendo muitas delas através da virtualização de recursos computacionais usando máquinas virtuais (*Virtual Machine* – VM). Ainda é possível oferecer serviços em nuvem através do uso de contêineres que é uma espécie de virtualização mais leve se comparada com a virtualização tradicional usando máquinas virtuais, como pode ser visto na Figura 2.

Figura 3 – Comparação entre a arquitetura de contêineres e virtualizada



Fonte: Elaboração própria

A arquitetura de Nuvem pode ser dividida em quatro camadas: centro de dados (hardware), infraestrutura, plataforma e aplicação. (ZHANG, J. *et al.*, 2018) Cada uma delas pode ser vista como um serviço para a camada superior e como um cliente para a camada inferior. Essa arquitetura alcançou popularidade pela oferta da infraestrutura da Nuvem em três principais modelos de serviço: Software como Serviço (Software as a Service, SaaS), Plataforma como Serviço (Platform as a Service, PaaS), Infraestrutura como Serviço (*Infrastructure as a Service - IaaS*). A Figura 4 apresenta a concepção da computação em nuvem a partir dos seus diferentes tipos de serviços.

Figura 4 – Composição da Computação em Nuvem



Fonte: Elaboração própria

Já as formas de implementação da computação em nuvem podem ser através de nuvens públicas, privadas, híbridas e comunitárias. Essa divisão entre tipos diferentes de nuvem está associada com as várias necessidades envolvendo organizações públicas e privadas, além de instituições como universidades. Diante disso, para estabelecer qual o melhor tipo de infraestrutura em nuvem, é preciso antes saber qual a finalidade de cada uma das diferentes arquiteturas oferecidas pela computação em nuvem.

Considerando os diferentes modelos de arquitetura em nuvem, são apresentadas na Tabela 2 as principais infraestruturas envolvendo nuvens computacionais denominadas como públicas, privadas, híbridas e comunitárias. Essas nuvens são classificadas e descritas a seguir conforme o tipo de finalidade para o qual elas podem ser utilizadas, como os principais detalhes de implementação.

Tabela 2 – Relação dos tipos de infraestrutura de Nuvens

Tipo de Nuvem	Descrição do Tipo de Nuvem	Detalhes de Implementação
Nuvem Pública	Através da nuvem pública, os serviços são entregues aos clientes por meio de uma rede aberta para uso público, ou seja, seu acesso é através da internet.	O controle de acesso desse tipo de nuvem é mais difícil de ser avaliado e gerenciado por ser mantido por terceiros.
Nuvem Privada	Com a nuvem privada, a segurança é considerada maior, visto o controle em torno dos serviços que são mantidos em uma rede privada protegida por firewall.	O controle de acesso é considerado mais simples de ser gerenciado, devido à nuvem ser privada.
Nuvem Híbrida	Já a nuvem híbrida é formada pela composição de modelos de nuvens privadas e públicas, oferecendo, desta forma, maior diversidade no seu uso.	O controle de acesso e gerenciamento vai depender do escopo que a nuvem será aplicada.
Nuvem Comunitária	A infraestrutura de uma nuvem comunitária é compartilhada por várias organizações que partilham interesses, como a missão, requisitos de segurança, políticas, entre outros.	O controle de acesso e o gerenciamento podem ser de uma ou mais organizações, dentro ou fora das instalações das instituições.

Fonte: (DE SANTANA *et al.*, 2016; OLIVEIRA, 2014)

A computação em nuvem proporciona às organizações a migração de quase toda infraestrutura e os dados disponíveis para a internet. Ela permite o uso da computação simultânea com ilimitados usuários e empresas. Uma das principais vantagens está na possibilidade de utilização mais eficiente de hardware e software. Com a utilização de recursos da nuvem, é possível reduzir a capacidade ociosa em armazenamento e processamento de dados de forma escalável e conforme as necessidades reduzindo custos de infraestrutura ao pagar apenas pelo consumido.

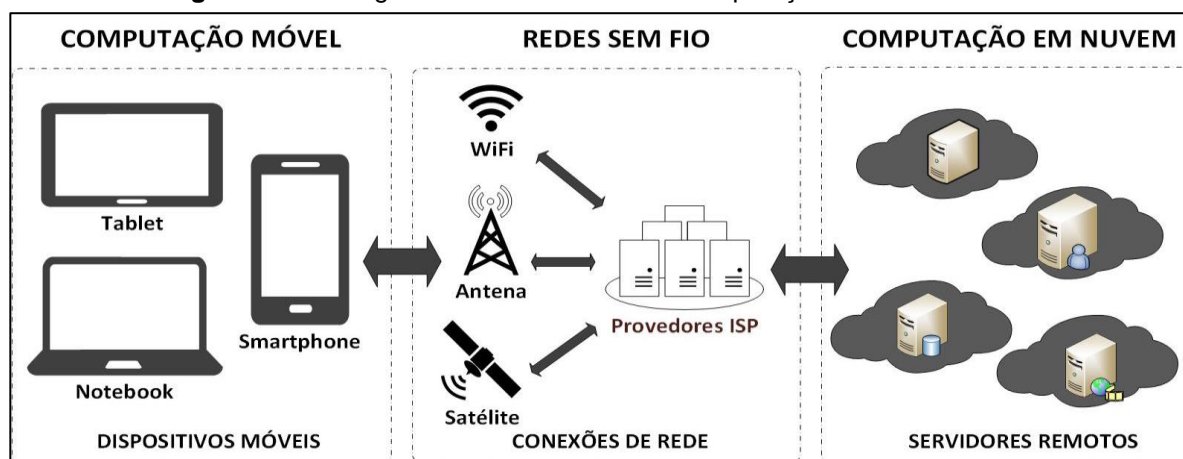
2.3. Computação Móvel em Nuvem

A Computação Móvel em Nuvem ou *Mobile Cloud Computing* (MCC) é um paradigma computacional envolvendo a união da Computação Móvel, da Computação em Nuvem e de conexões de redes sem fio (COSTA, P. B. *et al.*, 2014; DOS SANTOS *et al.*, 2017). Com isso, os trabalhos desenvolvidos nessa área de pesquisa visam buscar alternativas que possam diminuir as limitações computacionais e de consumo de energia de dispositivos móveis, como os *smartphones* (NAKAHARA, 2018). No que

concerne a tais limitações, estão aspectos do dispositivo móvel, como o número de processadores (CPUs), o tamanho da memória (RAM), a capacidade de armazenamento (SSD) e o uso constante de baterias (DOS SANTOS *et al.*, 2017).

Dessa forma, a MCC pode ser vista como uma infraestrutura utilizada para tornar os dispositivos móveis mais eficientes, ao permitir que esses dispositivos possam fazer o armazenamento e o processamento de dados através de servidores ou nuvem computacionais, diminuindo o uso dos seu limitados recursos de CPU, memória RAM, armazenamento e, conseqüentemente, bateria para ampliar o poder de processamento e armazenamento de dados nesses dispositivos (AKHERFI *et al.*, 2018). (JARARWEH *et al.*, 2013). Contudo, melhorar a eficiência dos dispositivos da Computação Móvel através da MCC não é uma questão trivial, dada a dinamicidade do cenário envolvendo diferentes dispositivos móveis, redes sem fio e nuvens computacionais (KAUSHIK; GAURAV, 2014). Uma visão geral de um ambiente de MCC é visto na Figura 5.

Figura 5 – Visão geral de um ambiente de Computação Móvel em Nuvem



Fonte: Elaboração própria

A partir deste cenário, é possível entender a relação dentro da MCC, como explanado anteriormente, entre a Computação Móvel, as Redes sem Fio e a Computação em Nuvem, visto que são a base do paradigma da Computação Móvel em Nuvem (RAHIMI *et al.*, 2014). A técnica de *offloading* é uma operação que visa transferir o processamento e os dados de um dispositivo móvel para computadores com maior poder computacional, com o intuito de aumentar o desempenho da aplicação e economizar energia do dispositivo móvel (GANI *et al.*, 2014). Devido à

exigência de uma maior capacidade de processamento em dispositivos móveis, a técnica de descarregamento mostra-se bastante eficiente. Para isso, é imprescindível o uso de uma conexão de rede capaz de atender as necessidades de transmissão.

Dispositivos móveis possuem menos poder computacional em comparação a computadores servidores, visto que o processamento de tarefas complexas representa maior impacto no consumo de energia da bateria nesses dispositivos. Uma forma de contornar o problema é por meio de infraestruturas de nuvem (GUO *et al.*, 2016).. Um exemplo é a estrutura de descarregamento de computação do ASM (CHUN *et al.*, 2011), a qual mostrou que o descarregamento de computação ajudou a reduzir o custo de consumo de energia em dispositivos móveis em 33%, e o tempo de resposta do aplicativo em 45% (SHIRAZ *et al.*, 2014).

O descarregamento de computação refere-se à transferência de tarefas computacionais que exigem muitos recursos para uma plataforma externa, como um cluster, uma grade ou uma nuvem. A técnica diminui a carga de processamento sobre um dispositivo sobrecarregado, por meio da migração da aplicação para uma máquina remota com recursos computacionais disponíveis (KUMAR *et al.*, 2013). Assim, as técnicas de descarregamento visam à melhoria das capacidades de dispositivos móveis, economizando energia (GUO *et al.*, 2016), melhorando o desempenho de uma aplicação, alcançado a partir da divisão dessa aplicação em vários subprogramas em que, para cada um desses subprogramas, será atribuído um diferente processador para sua execução (KHAN, M. A., 2015).

Porém, um dispositivo móvel por si só não é capaz de fazer muita coisa, pois de nada adianta um bom hardware sem um bom software que o gerencie. Para isso, diferentes sistemas operacionais foram desenvolvidos ao longo dos anos voltados para atender as necessidades apresentadas por esses dispositivos. Em comparação com computadores convencionais, os dispositivos móveis apresentam limitações de tamanho de tela, número de processadores, tamanho de memórias primárias e secundárias, além do uso de baterias que precisam sempre serem recarregadas.

Logo, o uso crescente de dispositivos móveis como os *smartphones* está diretamente relacionado com as inúmeras aplicações que podem ser utilizadas a partir

desses dispositivos. Essas aplicações são normalmente desenvolvidas de acordo com a plataforma móvel do dispositivo e requerem o uso de ferramentas e conhecimentos específicos de linguagens de programação, como será visto a seguir.

2.4. Progressive Web Apps

A partir do surgimento de dispositivos como *smartphones* e *tablets*, foi necessário o desenvolvimento de novas aplicações que pudessem ser executadas a partir de diferentes dispositivos e plataformas de sistemas operacionais móveis. Isso fez surgir novos conceitos de desenvolvimento de software, visto os requisitos em torno da nova forma de interação entre usuários e *smartphones*, por exemplo. No cenário móvel atual, existem diversas plataformas de sistemas operacionais móveis, sendo as principais no mercado o Android do Google, o IOS da Apple, o Windows Phone da Microsoft e o BlackBerry OS da fabricante BlackBerry.

Cada uma das plataformas móveis do mercado desenvolve suas próprias tecnologias, a partir de novos hardwares e softwares, aperfeiçoando, por exemplo, aspectos importantes, como a câmera fotográfica e outros recursos do dispositivo. Logo, é possível estabelecer que o desenvolvimento de aplicações móveis requer um conhecimento da plataforma para qual a aplicação será desenvolvida, como por exemplo, ferramentas de desenvolvimento ou SDK (*Software Development Kit*) que fornecem todo um arcabouço necessário para o desenvolvimento de aplicações para uma determinada plataforma móvel, a partir do seu kit de desenvolvimento oferecido.

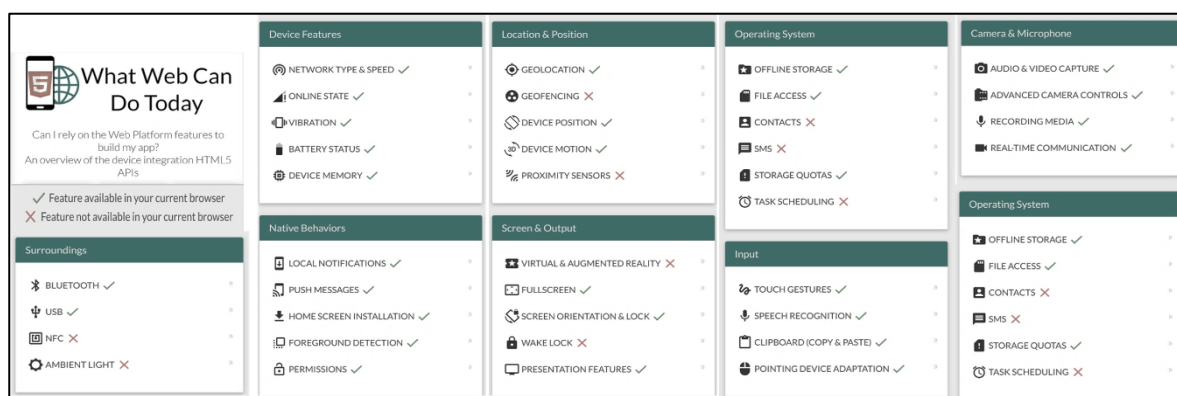
Porém, além do SDK, é preciso saber qual linguagem utilizar para o desenvolvimento de aplicações a serem utilizadas por uma determinada plataforma móvel. Essa diferença entre linguagens de programação também é vista nos computadores convencionais, como desktops, onde uma aplicação deve ser desenvolvida para os diferentes sistemas operacionais, como Linux, Windows e MacOS. Assim, no mundo dos dispositivos móveis, uma aplicação desenvolvida para uma plataforma específica não será compatível com outra plataforma. As diferenças entre os tipos de aplicações móveis disponíveis podem ser vistas na Tabela 3.

Tabela 3 – Relação das plataformas e aplicações móveis

Plataformas de Sistemas Móveis	Linguagem de Programação	Ambiente de Programação	Tipo de Aplicação Móvel	Onde Obter Aplicação
Android/Google	Java, Kotlin	Android Studio, Android SDK	Nativa	Play Store
iOS/Apple	Objective C, Swift	XCode	Nativa	iTunes
Windows Phone/MS	C#, .NET	Visual Studio	Nativa	Microsoft Store
BlackBerry/RIM	Java	BlackBerry plugin para Eclipse	Nativa	BlackBerry Apps World
Todas	JavaScript, CSS, HTML5	Visual Studio Code	WebApp	URL
Todas	JavaScript, CSS, HTML5, Flutter	Xamarin, PhoneGap, Ionic	Híbrida	Todas as lojas

O surgimento de novas tecnologias para o desenvolvimento de aplicações vem juntamente com o aumento expressivo no uso de dispositivos, como smartphones, tablets e dispositivos vestíveis, como os relógios inteligentes. Assim, o desenvolvimento de aplicações nativas, híbridas ou Web permite que esses diferentes tipos de aplicações possam alcançar o máximo de usuários dos dispositivos móveis no mercado. As funcionalidades que podem ser acessadas a partir das APIs disponíveis em navegadores podem ser vistas na Figura 6.

Figura 6 – Funcionalidades acessíveis a partir de um PWA



Fonte: (CECONI, 2019)

O foco em torno dos PWAs é oferecer uma experiência ao usuário através de algumas características que fazem desse tipo de aplicação uma alternativa interessante ao desenvolvimento de aplicações Web para as diferentes plataformas e sistemas operacionais do mercado. Essas características são essenciais para que uma aplicação PWA seja capaz de fornecer ao usuário a qualidade e funcionalidades

que tornam esse tipo de aplicação relevante no cenário dos aplicativos móveis, como podem ser vistas na Tabela 4.

Tabela 4 – Características que definem uma aplicação PWA

Característica	Descrição da característica
Progressivos	É o conceito de <i>progressive enhancement</i> (aprimoramento progressivo) que está relacionado com a prática de desenvolvimento que considera o aprimoramento progressivo dos navegadores Web para garantir a qualidade e o funcionamento da aplicação PWA para os usuários.
Responsivos	É o conceito de <i>Responsive Web Design</i> (RWD) que permite que as aplicações Web sejam adaptáveis aos diferentes tamanhos de tela dos dispositivos onde a aplicação está sendo acessada, fazendo com que o conteúdo da aplicação seja adaptado à tela do dispositivo, sem afetar a experiência do usuário.
Independentes de conexão	Uma aplicação Web convencional não é capaz de funcionar offline. Assim, esse conceito trata do <i>offline first</i> , ou seja, a aplicação será utilizada mesmo sem uma conexão com a internet, graças à existência de um cache dos dados da aplicação gerenciado pelo <i>Service Worker</i> , um script que roda no navegador e é responsável por gerenciar as páginas da aplicação entre o cliente e o servidor.
Semelhantes com aplicativos nativos	Para prover uma experiência do usuário semelhante de aplicativos nativos, o PWA conta com os recursos do <i>WebView</i> através de um <i>shell Browser</i> , ou seja, uma janela do navegador sem a barra e botões de navegações que oferecem ao PWA uma visão igual aos aplicativos nativos e híbridos, sendo que o conteúdo desse <i>shell</i> é carregado a partir dos dados no cache do <i>Service Worker</i> .
Sempre atualizados	Por ser uma aplicação Web, o conteúdo pode ser atualizado a partir dos novos acessos ou simplesmente ao fechar e abrir novamente a aplicação. Para isso, o <i>Service Worker</i> irá gerenciar as novas versões de cache para atualizar a versão existente no usuário quando a aplicação for fechada e aberta novamente.
Seguros	Os <i>Progressives Web Apps</i> devem ser servidos a partir do protocolo HTTPS que fornece um certificado capaz de garantir a segurança das informações trafegadas entre o cliente e o servidor e, assim, garantir a integridade dos dados armazenados pelo cache do <i>Service Worker</i> .
Descobríveis e Instaláveis	As duas características estão associadas à criação do <i>Web App Manifest</i> que consiste em um arquivo no formato JSON, em que são definidas algumas informações sobre a aplicação PWA, como cor de fundo da tela inicial, ícones da aplicação e cor da barra de notificação do dispositivo móvel.
Engajáveis	A forma de engajamento dos PWAs está no uso de notificação (<i>push notifications</i>) para que o usuário permaneça usando a aplicação PWA. Essas notificações podem ser geradas pela aplicação local ou enviadas pelo servidor.
Acessível por links	Assim, como sites convencionais, o PWA pode ser acessado e executado a partir de uma URL e, desta forma, ser compartilhado por link para outros usuários.

O desenvolvimento de PWA é possível por meios de diferentes tecnologias, como IONIC, Polymer, entre outros. No entanto, os dois pontos mais importantes a serem considerados em um PWA está na definição do arquivo do *Service Worker* e no arquivo do *Web App Manifest*. Ambos arquivos serão explicados com mais detalhes no Capítulo 3 referentes à metodologia proposta neste trabalho de pesquisa.

2.5. Avaliação de Aplicações na Computação Móvel em Nuvem

O desenvolvimento de sistemas envolve diversas atividades entre elas a de avaliar o comportamento desse sistema em uma situação real. Isso é importante para entender quais melhorias que podem ser aplicadas no sistema para torná-lo mais eficientes em termos de usabilidade e execução (LEAL, 2016). Assim, em ambientes de MCC, a avaliação de sistemas consiste em mensurar o desempenho do mesmo, através de técnicas que possibilitam obter diferentes métricas, como o tempo de execução de uma tarefa ou até o tempo de resposta da rede ou sistema avaliado.

Para que um sistema possa ser avaliado na MCC, é preciso que haja inicialmente um entendimento desse sistema para que dessa forma seja possível determinar um objetivo em relação ao que será avaliado (COSTA, I. O., 2015). Entre os objetivos que podem determinar o que será avaliado em um sistema estão os seguintes:

- **Comparar alternativas** – Consiste em definir entre as alternativas analisadas, aquela que mais se adequa com a necessidade inicialmente definida. Em termos de hardware, um exemplo seria comparar servidores para determinar o mais rápido para operações de escrita e leitura em disco.
- **Identificar gargalos e/ou de desperdício de sistemas** – Permite determinar o gargalo no sistema e após identificado, buscar uma melhoria no desempenho capaz de mitigar o problema encontrado. Exemplos de sistemas que possuem gargalos são as aplicações Web que dependem da capacidade do servidor.
- **Determinar configuração ótima dos parâmetros de um sistema** – Permite ampliar o desempenho de um sistema otimizado os seus recursos. Por exemplo, em um banco de dados é necessário verificar se as configurações de hardware (CPU, memória RAM e armazenamento) serão suficientes para atender o fluxo de armazenamento e busca de dados.
- **Fazer previsão de desempenho em função do crescimento da carga** – É usado em sistemas que podem sofrer picos de usuários e ficar fora do ar.

Para realizar a avaliação de sistemas, são utilizadas diferentes técnicas sendo que basicamente a avaliação de desempenho de sistemas está dividida em três técnicas (JAIN, 1991). São elas:

- **Medição.** Consiste na realização de medições utilizando um sistema real ou um protótipo. Para isso, é possível usar ferramentas que permitem observar o desempenho do sistema, coletar estatísticas e analisar os resultados coletados.
- **Simulação.** A simulação é definida pela construção de modelos estruturais do sistema a ser estudado para que sejam feitas experimentações desses modelos. Na simulação, o hardware, o software, os outros mecanismos internos do sistema devem ser previamente especificados no modelo.
- **Modelagem analítica.** Consiste de modelos e equações matemáticas que reproduzem o comportamento de um sistema. A modelagem analítica pode ser usada quando o sistema real ainda não existe e é feita através de formalismos matemáticos, como: cadeias de Markov e Redes de Petri.

2.6. Trabalhos Relacionados

A Computação Móvel em Nuvem ou *Mobile Cloud Computing* – MCC é um paradigma que surgiu por volta de 2007 e envolve a Computação Móvel, redes de comunicação e da Computação em Nuvem. Já as pesquisas e trabalhos da área de MCC têm como objetivo principal tornar mais eficiente o uso dos recursos dos dispositivos móveis através de abordagem como o *computing offloading*. Além disso, muitos trabalhos de MCC buscam também apresentar propostas que viabilizem a redução do consumo de energia nos dispositivos móveis, tornando o uso da bateria nesses dispositivos mais eficiente através de diferentes abordagens.

Na técnica de *offloading*, existem diferentes frameworks, tais como: *CloneCloud*, que propôs um sistema de particionamento do dispositivo para um clone virtualizado, através da combinação de análises estatísticas do ambiente de MCC. O estudo usou as redes 3G e WiFi e apresentou melhorias de tempo de execução e consumo de energia ao usar um particionamento de granularidade fina. Sua limitação é quanto à migração completa do dispositivo móvel para a sua versão virtual na nuvem, sendo necessário o envio de dados repetidos toda vez que uma nova ação for realizada (CHUN *et al.*, 2011).

Outro trabalho citado na literatura é o *MAUI*, que utilizou chamada de procedimento remoto (RPC) para particionar a execução de métodos de aplicações entre o dispositivo móvel e uma Máquina Virtual (VM) em nuvem. O estudo avaliou conexões 3G e WiFi, apresentando redução no consumo de energia e no tempo de execução, através de três estudos de caso diferentes. A limitação do trabalho está na necessidade de um programador intervir para determinar quais métodos devem ser particionados, o que abre oportunidades para uma abordagem mais inteligente e com menos intervenção do programador (CUERVO *et al.*, 2010).

Já o *ThinkAir* é um framework que utiliza virtualização para fazer *offloading* de processamento do dispositivo móvel em instâncias semelhantes do próprio dispositivo na nuvem. Este trabalho apresentou melhorias no tempo de execução e na economia de energia ao utilizar várias VMs para distribuir a execução dos métodos de forma dinâmica e de acordo com a demanda do dispositivo móvel. Neste trabalho foram comparadas execuções completas no dispositivo com execuções parciais na nuvem, usando redes 3G e WiFi com diferentes benchmarks. A limitação da proposta está na distribuição da execução de métodos na nuvem de forma contínua o que afeta o consumo de dados em redes pagas e limitadas (KOSTA *et al.*, 2012).

JSCloud (LI, W. Y. S. *et al.*, 2012) propõe o uso de algoritmos para estimar dinamicamente o tempo de migração de códigos Java Script (JS) local e remotamente, tendo como objetivo o aprimoramento da execução de aplicações web. Sendo inspirado no CloneCloud (CHUN *et al.*, 2011), a principal limitação a depender do tamanho da tarefa é o custo computacional da execução do algoritmo que determina como ocorrerá a execução e migração do código JavaScript para um servidor remoto ou nuvem computacional.

MOJA (XU *et al.*, 2014) propõe o *offloading* para *Web Applications (WebApps)* utilizando *WebSocket* e *NodeJS*. A solução baseia-se no *offloading* de funções JavaScript entre o dispositivo e a nuvem por meio de uma conexão por *Web socket* que, após a execução da função, transmite de volta o resultado para o dispositivo. O trabalho apresenta melhoria na experiência do usuário por diminuir o tempo de execução de funções através do uso de nuvens computacionais, e ainda menciona a

importância de uma forma inteligente para realizar o *offloading* a partir de informações referentes ao contexto do dispositivo móvel.

O projeto mCloud (ZHOU *et al.*, 2015) propõe um algoritmo para avaliar diferentes contextos, como velocidade da rede e disponibilidade de serviços, envolvendo o dispositivo móvel, *cloudlets* próximos e serviços em nuvem (ROUTAIB *et al.*, 2014). O trabalho buscou melhorar o desempenho do dispositivo fornecendo decisões em tempo de execução para a seleção de redes sem fio e recursos de nuvem computacionais mais apropriados para o *offloading*. Ele apresentou diminuição do consumo de energia e tempo de execução no dispositivo móvel. Sua limitação está na necessidade de elaborar novas estimativas de custos sempre que ocorrem mudanças de contexto na infraestrutura de MCC.

WWOF (ZHANG, J. *et al.*, 2018) *Webpages Offloading Framework* utiliza um recurso do HTML5 conhecido por *WebWorker*, que permite gerar segmentos de execução de *threads* fora do escopo do navegador e, com isso, fazer o *offloading* da execução utilizando *WebSockets* como meio de troca de mensagens entre o cliente e o servidor. No trabalho foram utilizados benchmarks em PCs, *tablets*, *smartphones* e *smartTVs*, com resultados positivos de economia energética e tempo de execução para realização de grandes cargas de trabalho. A limitação está em execuções paralelas, visto que o *WebWorker* só é capaz de executar um thread por vez (single thread). O trabalho menciona a importância de se ter uma inteligência capaz de decidir sobre o *offloading* em MCC a partir de uma análise do contexto do dispositivo móvel.

3. DESENVOLVIMENTO E AVALIAÇÃO DE PWA NA MCC

O desenvolvimento de aplicações móveis tem sido cada vez mais ampliado, graças às tecnologias envolvendo novas linguagens de programação e também conceitos inovadores de como essas aplicações podem ser desenvolvidas. Dentro da Computação Móvel em Nuvem, são limitados os trabalhos que fazem referência ao uso de *Progressive Web Apps* como forma de implementar aplicações. Desta forma, é relevante que o conceito de PWA seja explorado na MCC, através de estratégias envolvendo o *offloading* de execuções de tarefas, utilizando esse tipo de aplicação.

Assim, neste capítulo, será apresentada uma metodologia que permite o desenvolvimento de aplicações na Computação Móvel em Nuvem, através do uso de *Progressive Web Apps*. Com isso, é possível, através da metodologia deste trabalho, desenvolver Aplicações Web Progressivas (PWA) em ambientes de MCC para fazer a execução de tarefas tanto no dispositivo móvel como em um servidor remoto ou nuvem computacional. A metodologia apresentada considerou aspectos importantes no desenvolvimento de *Progressive Web Apps* em um ambiente da Computação móvel em Nuvem, além de como esse tipo de aplicação pode ser avaliada.

Para facilitar o entendimento da metodologia proposta neste trabalho, ela foi dividida em cinco fases, que vão desde a definição da aplicação a ser desenvolvida, até a avaliação dessa aplicação em um ambiente de MCC. Essa divisão da metodologia em várias etapas pode ser vista na Figura 7.

Figura 7 – Metodologia para uso de PWA em ambientes de MCC



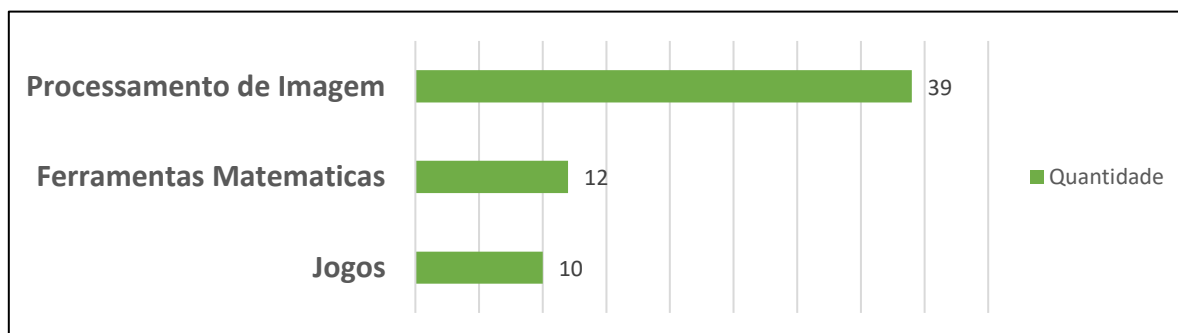
Fonte: Elaboração própria

Desta forma, é importante que a metodologia seja explicada a partir das fases que a compõe, e como cada uma dessas etapas pode ser utilizada no desenvolvimento de uma Aplicação Web Progressiva na MCC. Assim, a seguir serão descritas, respectivamente, nas seções 3.1, 3.2, 3.4, 3.5 e 3.5, as cinco etapas da metodologia e o que deve ser feito em cada uma delas para que seja possível desenvolver e avaliar uma Aplicação PWA na Computação Móvel em Nuvem.

3.1. Planejamento – Definir objetivos da aplicação PWA na MCC

Essa primeira fase da metodologia consiste em definir qual será o objetivo da aplicação PWA a ser desenvolvida no ambiente de MCC, ou seja, qual a finalidade dessa aplicação e o que ela é capaz de fazer. Com isso, o desenvolvimento e uso de uma aplicação em MCC terá um fim específico, como fazer o processamento de imagens ou cálculos matemáticos (SILVA, F. A. *et al.*, 2016). Na Figura 8, é possível ver as principais aplicações utilizadas na Computação Móvel em Nuvem.

Figura 8 – Principais tipos de aplicações da Computação Móvel em Nuvem



Fonte: Adaptado de (SILVA, F. A. *et al.*, 2016)

Porém, é importante que uma aplicação desenvolvida em MCC tenha a capacidade de fazer a transferência de métodos ou tarefas para um servidor remoto ou nuvem computacional, através da descarga computacional também chamada de *offloading* (AKHERFI *et al.*, 2018). Com isso, a partir da definição do objetivo da aplicação PWA em MCC, é possível determinar o que será processado por essa aplicação no dispositivo móvel ou o que será transferido do dispositivo para ser processado em um computador com melhores recursos computacionais. Essa descarga computacional do dispositivo é feita através de redes sem fio e busca diminuir o impacto que esse processamento poderia representar, caso realizado no próprio dispositivo móvel (ALI *et al.*, 2016).

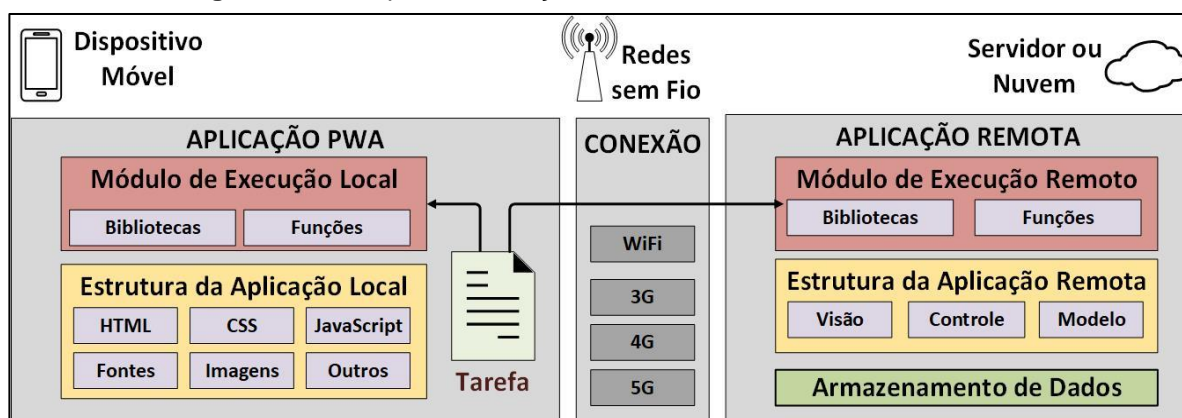
Assim, considerando que essa primeira fase da metodologia busca definir o objetivo de uma aplicação PWA na MCC, um exemplo dessa definição seria uma aplicação para fazer a detecção de faces em imagens ou talvez para realizar cálculos matemáticos, a partir do próprio dispositivo móvel ou em um servidor remoto. Para isso, é preciso utilizar módulos capazes de, por exemplo, processar uma determinada imagem e fazer a detecção de faces tanto no dispositivo quanto fora dele. Desta forma, a definição dos módulos de execução é uma fase importante no desenvolvimento de uma aplicação PWA na MCC, como apresentado a seguir.

3.2. Modularização – Selecionar módulos de execução local e remoto

Considerando que uma aplicação em MCC irá fazer a execução de métodos ou tarefas, é preciso que sejam definidos os módulos a serem utilizados pela aplicação tanto no dispositivo móvel (execução local) quanto no servidor ou nuvem (execução remota ou *offloading*) (AKHERFI *et al.*, 2018). Assim, esses módulos serão determinados a partir da definição da aplicação PWA na MCC, ou seja, o que será realizado utilizando essa aplicação, como exemplo, fazer a detecção de faces em imagens ou cálculos matemáticos de números inteiros.

Esses módulos de execução permitem que uma aplicação PWA seja capaz de fazer a execução de tarefas referentes a essa aplicação, a partir do próprio dispositivo móvel (execução local) quanto através do *offloading*, em uma máquina com maiores recursos computacionais, como um servidor ou nuvem (execução remota). Para exemplificar melhor esses módulos e como eles são utilizados em um ambiente MCC, a Figura 9 apresenta um modelo de como uma determinada tarefa será executada em uma Aplicação Web Progressiva (execução local) e na Computação Móvel em Nuvem através do *offloading* ou execução remota.

Figura 9 – Exemplo da execução de tarefas em um ambiente de MCC



Fonte: Elaboração própria

Esses módulos de execução local e remoto serão compostos por bibliotecas ou funções capazes de fazer um tipo de processamento específico, de acordo com a definição da aplicação PWA na MCC. Assim, considerando que o desenvolvimento de uma Aplicação Web Progressiva é feito, principalmente, através da linguagem *JavaScript*, é possível que os módulos de execução local e remoto sejam determinados através das inúmeras bibliotecas e funções existentes e compatíveis com o uso dessa linguagem (JEONG; MOON, 2015). Esses módulos podem ser baixados através de gerenciadores de pacotes da linguagem *JavaScript* para implementar o servidor *NodeJS*⁶, como o *NPM*⁷ (*Node Package Manager*) e o *Yarn*⁸.

No entanto, caso não seja possível determinar um módulo de execução local ou remoto, devido à ausência de bibliotecas ou funções específicas para o funcionamento do módulo em questão, será preciso implementá-lo usando a linguagem *JavaScript* possibilitando, dessa forma, que a aplicação desenvolvida seja capaz de realizar a execução da tarefa especificada (XU *et al.*, 2014).

3.3. Implementação – Desenvolver arquivos da aplicação PWA na MCC

Para desenvolver um PWA, considerando seus arquivos base, como HTML, CSS e *JavaScript*, é possível utilizar diferentes tecnologias, como ionic, angular, react, entre outros frameworks de desenvolvimento Web (BIØRN-HANSEN *et al.*, 2017). No

⁶ <https://nodejs.org/pt-br/>

⁷ <http://nodebr.com/o-que-e-a-npm-do-nodejs/>

⁸ <https://yarnpkg.com/lang/pt-br/docs/install/#windows-stable>

entanto, dois arquivos são considerados essenciais no desenvolvimento de um PWA: o “*sw.js*” e o “*manifest.json*”. A definição desses arquivos permite que o PWA funcione sem internet “*offline*” e ainda seja instalado como uma aplicação nativa no dispositivo móvel, oferecendo uma experiência de usuário similar aos dos aplicativos nativos da plataforma móvel em que esteja sendo executado. Além disso, existem inúmeros *templates* de Aplicações Web Progressivas disponíveis no *Github* e de fácil utilização. Desta forma, o foco principal para implementar um PWA a ser utilizado na MCC estará nas definições dos arquivos do *Service Worker* e do *Web Manifest*.

Considerando que os módulos de execução da tarefa estejam tanto na aplicação do dispositivo móvel quanto no servidor remoto utilizado pela aplicação, é preciso que os arquivos do módulo de execução local sejam inseridos no cache do *Service Worker* junto com os arquivos de HTML, CSS e *JavaScript*, que permitem o funcionamento da aplicação *offline* (LIMA, 2017). Como visto na Seção 2.4, o cache do *Service Worker* é formado pelos arquivos que permitem o funcionamento *offline* da aplicação, sendo apontado o caminho onde esses arquivos estão em sua estrutura. Com isso, o módulo de execução local poderá ser inserido no cache do *Service Worker* de um PWA, como pode ser visto na Figura 10.

Figura 10 – Exemplo do Módulo de Execução no cache do *Service Worker*

```
1 self.addEventListener('install', function (event) {
2     event.waitUntil(
3         caches.open(CACHE_NAME).then(function (cache) {
4             return cache.addAll([
5                 '/',
6                 '/html/',
7                 '/css/',
8                 '/javascript/...',
9                 '/manifest.js',
10                '/fonts/',
11                '/imagens/...',
12                '/mcc/...'
13                '/outros/...',
14            ]);
15        })
16    )
17 });
```

Após isso, é preciso criar o *Web Manifest* contendo as informações que darão ao PWA a capacidade de ser instalável e ter um ícone de acesso no dispositivo, além da tela de abertura semelhante aos aplicativos móveis convencionais. A Figura 11 apresenta um exemplo de como seriam as definições de um arquivo *Web Manifest* utilizado para desenvolver uma aplicação PWA na MCC.

Figura 11 – Exemplo das definições de um arquivo *Web Manifest* de um PWA

```
1  {
2    "name": "AgilePoker",
3    "short_name": "AgilePoker",
4    "theme_color": "#2a0044",
5    "background_color": "#2a0044",
6    "display": "standalone",
7    "start_url": "/",
8    "icons": [
9      {
10     "src": "images/icon-72x72.png",
11     "sizes": "72x72",
12     "type": "image/png"
13   }
14 ]
15 }
```

Fonte: Adaptado de (LIMA, 2017)

Já do lado do servidor, o módulo de execução remoto pode ser implementado utilizando o *NodeJS*, que permite o desenvolvimento de aplicações que rodam diretamente no servidor utilizando *JavaScript*. Assim, através do NPM, o gerenciador de pacotes que vem com o *NodeJS*, é possível instalar diversos complementos que permitem facilitar a criação de aplicações executadas no servidor, como o framework *ExpressJS*, que ajuda na criação de rotas da aplicação e no uso dos métodos HTTP, como Get (pegar) e Post (enviar) (CECONI, 2019). O NPM também permite baixar bibliotecas para diversas finalidades, como por exemplo, para implementar tanto o módulo de execução remoto utilizado para fazer o processamento da tarefa, através da descarga computacional, ou módulo local usado pela aplicação PWA para fazer a execução de tarefas localmente no próprio dispositivo móvel.

Após desenvolvimento do PWA, é importante que esse seja analisado permitindo obter informações sobre seu desempenho e possíveis melhorias que podem ser realizadas para melhorar a performance e qualidade da Aplicação PWA.

3.4. Inspeção – Analisar aspectos da aplicação PWA na MCC

Para que seja possível analisar a aplicação PWA desenvolvida, é preciso que essa aplicação esteja sendo disponibilizada a partir de um servidor remoto ou nuvem, através de um localizador padrão de recursos na Web ou *Uniform Resource Locator* (URL) (CECONI, 2019). Além disso, é importante que o PWA seja acessível através do protocolo HTTPS que fornece segurança quanto aos dados que trafegam entre a aplicação e o servidor remoto. Assim, o servidor utilizado para disponibilizar o PWA terá que ter suporte ao protocolo de segurança do HTTPS (o S é de segurança). Já no caso de utilizar um servidor rodando numa máquina local (*localhost*), a aplicação poderá ser testada sem a necessidade de um servidor com suporte HTTPS.

Com o servidor rodando e provendo a aplicação, seja esse servidor remoto ou local, o próximo passo para fazer a análise do PWA é utilizar o navegador Google Chrome e a extensão que permite instalar a ferramenta para analisar Aplicações Web Progressivas, chamada de *LightHouse*⁹. Após instalar o *LightHouse* no navegador Chrome, é preciso abrir a aplicação PWA, a partir do caminho fornecido pelo servidor remoto ou local. Dessa forma, com a aplicação já aberta no navegador, basta clicar na ferramenta *LightHouse* e escolher o que deverá ser analisado no PWA por ela.

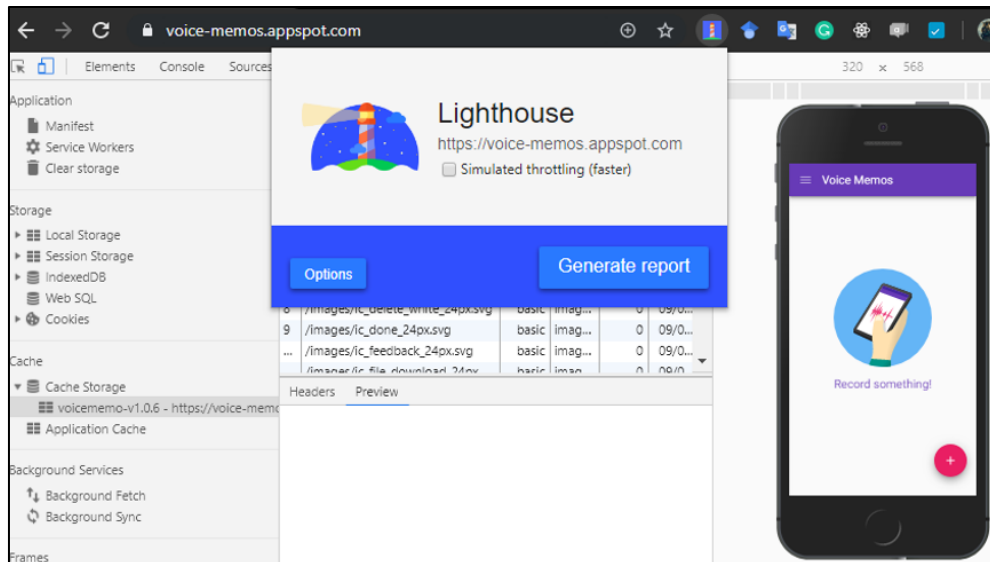
Vale destacar que as versões mais atuais do navegador Chrome já possuem nas ferramentas do desenvolvedor o *LightHouse* por padrão, não sendo necessário instalar a extensão no navegador. Assim, para utilizar o *LightHouse* sem a necessidade de instalação da extensão, será preciso entrar no menu do navegador e acessar as ferramentas do desenvolvedor e a guia de auditoria. A partir das ferramentas do desenvolvedor, também é possível ver diversas informações da aplicação PWA, como os arquivos de cache do *service worker*, entre outros.

A partir do uso da ferramenta *LightHouse*, é possível inspecionar diversos aspectos do PWA, como performance e acessibilidade, além de outras categorias que

⁹ <https://developers.google.com/web/tools/lighthouse/>

podem ser escolhidas no menu de opções antes de iniciar a análise da aplicação. Um exemplo de como essa ferramenta é utilizada poder ser visto na Figura 12.

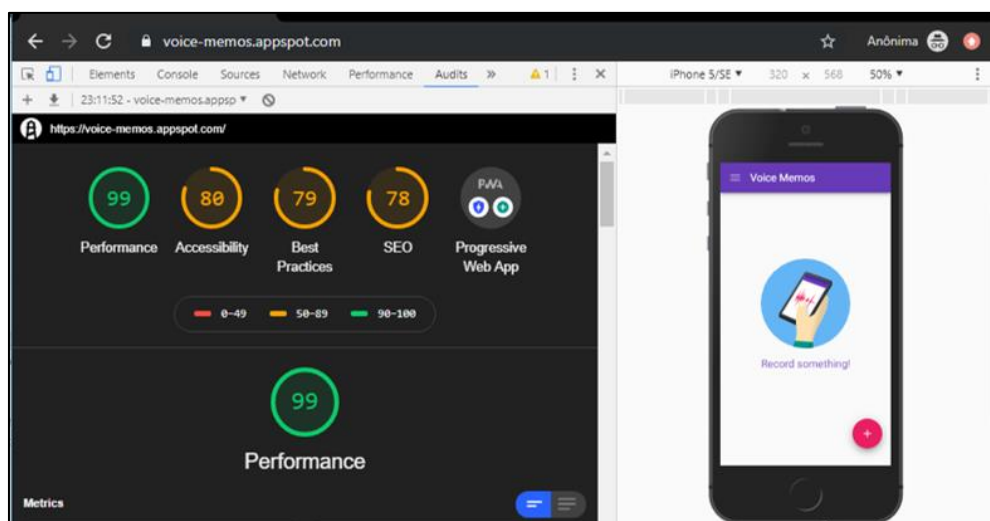
Figura 12 – Exemplo da ferramenta de análise de qualidade de PWA



Fonte: Elaboração própria

Após selecionar o que será analisado nas opções da ferramenta, é necessário clicar em gerar relatório para que a ferramenta faça uma análise completa da aplicação PWA. Ao final da análise, a ferramenta irá fornecer diversas informações, como visto na Figura 13.

Figura 13 – Resultados da análise do PWA com a ferramenta *Lighthouse*



Fonte: Elaboração própria

3.5. Avaliação – Avaliar o desempenho da aplicação PWA na MCC

Após analisar a Aplicação Web Progressiva com a ferramenta *LightHouse* e obter os resultados referentes à performance e à qualidade da aplicação, a última fase da metodologia consistirá na avaliação de desempenho da aplicação em um ambiente de Computação Móvel em Nuvem. Para isso, a aplicação deverá ser disponibilizada em um servidor remoto ou nuvem para que seja possível avaliar a execução de tarefas tanto no dispositivo móvel quanto fora dele. Assim, é preciso definir alguns pontos importantes que irão permitir a avaliação do PWA no ambiente de MCC.

Na avaliação de desempenho de sistemas (ADS), existem diversas técnicas que ajudam a definir como uma avaliação poderá ser realizada e quais os resultados que poderão ser obtidos do sistema a ser avaliado (LEAL, 2016). Assim, para avaliar uma aplicação PWA em um ambiente de MCC, é preciso definir um planejamento básico para conduzir a avaliação. O planejamento para fazer a avaliação do PWA na Computação Móvel poderá ser definido da seguinte maneira:

- **A carga de Trabalho** – Será formada por tarefas referentes à aplicação PWA e será executada no dispositivo móvel e no servidor remoto.
- **A métrica a ser coletada** – É a métrica utilizada para comparar as execuções das tarefas realizadas no dispositivo e no servidor remoto pelo PWA.
- **Os dispositivos da coleta** – São os dispositivos em que a aplicação PWA será utilizada para fazer a execução das tarefas local e remotamente.
- **As redes sem fio** – Conexões de rede utilizadas para realizar o *offloading* das tarefas do dispositivo móvel para serem executadas no servidor remoto.
- **O número de execuções** – Quantidade de vezes em que as execuções serão realizadas tanto nos dispositivos móveis quanto no servidor remoto.

Desta forma, após definir os pontos do plano da avaliação, será preciso executar o planejamento para que seja possível coletar os dados referentes à execução da carga de trabalho nos diferentes dispositivos móveis e nas diferentes redes em fio. Com isso, é possível obter dados contendo métricas de desempenho acerca das tarefas contidas na carga de trabalho utilizada na avaliação. Para que essa medição e comparação entre as tarefas executadas seja possível, será preciso usar uma métrica, como o tempo de execução por tarefa, que poderá determinar onde a

tarefa foi executada com maior eficiência, se local ou remotamente. A Tabela 5 apresenta um exemplo das possíveis definições de um plano de avaliação.

Tabela 5 – Exemplo de um plano de avaliação

Carga de Trabalho	Métrica a ser coletada	Dispositivos da coleta	Redes sem fio	Número de execuções
Imagens	Tempo de Execução	Dispositivos Móveis	3G, 4G, 5G e WiFi	'N' Execuções no dispositivo móvel
Números	Consumo de Energia	Servidor Remoto ou nuvem		'N' Execuções na nuvem ou servidor

Após realizar a avaliação, é preciso fazer uma análise dos dados coletados permitindo obter resultados, como a média da execução das tarefas tanto no dispositivo móvel quanto no servidor remoto utilizado para fazer o *offloading* em MCC. Essa análise poderá ser feita através de ferramentas de análises estatísticas de dados, como o MatLab ou Weka, que permitem realizar análises completas em diferentes tipos de dados. Já nos casos em que uma análise mais simples poderá ser considerada suficiente, é possível utilizar a ferramenta Microsoft Excel.

Com isso, os resultados da avaliação poderão ser apresentados através de gráficos ou tabelas, por meio das relações entre as tarefas utilizadas, como carga de trabalho, os dispositivos da coleta e as redes sem fio utilizadas por eles (SILVA, J. S. *et al.*, 2017). É importante que a métrica definida para realizar a avaliação seja associada com essas tarefas e com os demais aspectos que integram o plano de avaliação, tornando possível uma avaliação mais ampla de um PWA na MCC. Desta forma, os resultados da avaliação irão apresentar o quanto uma tarefa executada através do *offloading* utilizando um PWA é mais eficiente que uma execução local no próprio dispositivo móvel.

Assim, chega ao fim a apresentação da metodologia para o desenvolvimento e avaliação de uma Aplicação Web Progressiva na Computação Móvel em Nuvem. No entanto, é importante que a metodologia apresentada seja validada a partir do desenvolvimento de estudos de caso, como será visto no capítulo a seguir.

4. AVALIAÇÃO DA METODOLOGIA

Neste capítulo, a metodologia apresentada no Capítulo 3 será validada através do desenvolvimento e avaliação de um *Progressive Web Apps* no ambiente da Computação Móvel em Nuvem. Assim, para uma melhor apresentação do PWA desenvolvido como estudo de caso deste trabalho, foi adicionada a esta aplicação a capacidade de realizar a execução de duas tarefas distintas, envolvendo a detecção de faces em imagens e o cálculo de números da sequência Fibonacci. A escolha por fazer o desenvolvimento de uma única aplicação PWA, unindo os dois estudos de caso, é para evitar a repetição das etapas que compõem a metodologia.

Assim, a estrutura da aplicação PWA de ambos os estudos de caso, em geral, são semelhantes, exceto pelos arquivos que formam os módulos de execução de cada tipo de tarefa da aplicação, ou seja, de detecção de faces e de cálculos da sequência Fibonacci. Desta forma, a apresentação da Aplicação Web Progressiva utilizada como estudo de caso irá seguir as etapas descritas na metodologia apresentada neste trabalho para fazer o desenvolvimento e avaliação de uma Aplicação PWA na MCC.

Os estudos de caso desenvolvidos neste trabalho referem-se ao desenvolvimento de uma Aplicação Web Progressiva capaz de fazer a detecção de faces em imagens e o cálculo de números da sequência Fibonacci na Computação Móvel em Nuvem. Para isso, o desenvolvimento da aplicação PWA, contendo os dois estudos, será dividida, conforme as cinco etapas da metodologia deste trabalho.

4.1. Planejamento – Definir objetivos da aplicação PWA na MCC

Considerando a primeira etapa da metodologia, o objetivo definido para a aplicação PWA na MCC a ser desenvolvida foi de fazer a detecção de faces em diferentes imagens e o de calcular números da sequência Fibonacci. Assim, a aplicação PWA executada no dispositivo móvel será capaz de fazer o processamento de detecção de faces em imagens carregadas pelo usuário, apresentando como saída o número de faces encontradas. Ela ainda deverá fazer o cálculo da sequência Fibonacci a partir de um número selecionado pelo usuário. A aplicação PWA também será capaz de fazer a execução dessas tarefas de detecção de faces e cálculo

Fibonacci, através do envio para um servidor remoto, utilizando a descarga computacional ou *offloading* sob as redes 3G, 4G e WiFi.

Assim, ao fazer o *offloading*, o processamento das tarefas de detecção de faces em imagens e do cálculo da sequência Fibonacci serão feitos fora do dispositivo móvel em um servidor remoto ou nuvem computacional. Com isso, logo após fazer a execução dessas tarefas, o servidor irá retornar para o dispositivo móvel o resultado desse processamento. Desta forma, a aplicação PWA executada no dispositivo móvel será capaz de fazer o processamento das tarefas desses estudos de caso de forma local e também através do envio para execução remota ou na nuvem.

Com a definição dos objetivos da aplicação, é preciso determinar os módulos de execução que serão utilizados pelo PWA para fazer o processamento local da tarefa no dispositivo móvel e os módulos que serão utilizados no servidor da aplicação para fazer a execução remota dessas tarefas.

4.2. Modularização – Selecionar módulos de execução local e remoto

Os módulos de execução escolhidos para processamento da detecção de faces local e remotamente foram, respectivamente, o *Tracking.js* e o *OpenCV.js*. Considerada uma das principais bibliotecas disponíveis para processamento de imagens, a *OpenCV (Open Source Computer Vision Library)* fornece diversas funcionalidades de Visão Computacional, como a detecção de faces e de objetos. Seu uso pode ser feito em diferentes plataformas, como Windows, Linux, Mac OS e *Android*. Já sua implementação pode ser realizada através de diferentes linguagens de programação, como C++, Python e *JavaScript*, observando que a *OpenCV* é uma biblioteca de código aberto.

Assim, considerando que a *OpenCV* pode ser implementada com *JavaScript*, essa biblioteca foi utilizada como um dos módulos de execução de tarefas que irão compor a aplicação PWA no ambiente de MCC. No entanto, a *OpenCV* apresenta algumas desvantagens quanto ao seu uso em uma aplicação PWA. Uma delas é que a *OpenCV* precisa ser compilada antes de ser utilizada na aplicação gerando impactos negativos quanto ao processamento de imagens em dispositivos móveis com recursos

computacionais limitados. Além disso, o tamanho dos arquivos de implementação da *OpenCV.JS* é considerado grande para uma aplicação PWA, visto que uma das características de um Progressive Web App está em ser leve, em termos de armazenamento no dispositivo móvel facilitando o carregamento dessa aplicação.

Por outro lado, no servidor remoto ou na nuvem, o uso da *OpenCV* mostra-se relevante, visto que um servidor ou nuvem é bem mais rico em recursos computacionais que um dispositivo móvel. Assim, utilizando o NPM, essa biblioteca foi obtida para ser utilizada no servidor *NodeJS* da aplicação PWA. Ainda usando o NPM, o módulo de execução local no dispositivo móvel foi definido com o download da biblioteca *Tracking.JS*, que é bem mais leve que a biblioteca *OpenCV* utilizada no servidor remoto responsável pelo *offloading* da aplicação. A biblioteca *TrackingJS* também permite fazer a detecção de faces em imagens e foi desenvolvida exclusivamente para ser utilizada do lado cliente, ou seja, no navegador web, sendo seu uso bem mais interessante em um PWA por ser mais leve que a *OpenCV*.

Já os módulos de execução local e remoto utilizados para fazer o cálculo de números da sequência Fibonacci foram determinados utilizando o algoritmo recursivo, visto que o uso de algoritmos de interação e memorização realizam o processamento de forma mais rápida, acabando por impactar diretamente na coleta de métricas, como o tempo de execução da tarefa referente ao cálculo de um número da sequência Fibonacci. Assim, o uso do algoritmo recursivo para fazer o cálculo da sequência Fibonacci foi o mais interessante para este estudo de caso, pois sua execução é realizada através da chamada recursiva, ou seja, a função chamando a própria função, o que acaba gerando maior tempo durante o seu processamento.

Um exemplo do algoritmo recursivo utilizado para calcular a sequência Fibonacci na aplicação PWA, desenvolvido e implementado como uma função utilizando a linguagem *JavaScript*, tanto no módulo local quanto no servidor remoto, pode ser visto na Figura 14.

Figura 14 – Algoritmo recursivo para calcular números do Fibonacci

```
1 | function fiboRec(num) {  
2 |     if (num <= 1) return 1;  
3 |     return  
4 |         fiboRec(num - 1) + fiboRec(num - 2);  
5 | }
```

Fonte: Elaboração própria

Assim, após a definição das bibliotecas e funções a serem utilizadas pelos módulos de execução local e remoto da aplicação PWA na MCC, é possível desenvolver então a Aplicação Web Progressiva a ser utilizada em um ambiente de MCC para fazer a detecção de faces em imagens e o cálculo da sequência Fibonacci.

4.3. Implementação – Desenvolver arquivos da aplicação PWA na MCC

Para desenvolver a aplicação PWA, considerando que essa será executada em um ambiente de MCC, foi escolhido primeiramente o servidor que será responsável por disponibilizar a aplicação PWA e também por fazer a execução das tarefas dos estudos de caso, através do *offloading*. Assim, considerando que a linguagem principal no desenvolvimento de *Progressive Web Apps* é o *JavaScript*, o servidor escolhido para disponibilizar o PWA foi o *NodeJS*, por ser implementado utilizando essa mesma linguagem. Com o *NodeJS*, é possível utilizar o gerenciador de pacotes NPM (*Node Package Manage*) que dá acesso ao download de diferentes bibliotecas, como por exemplo, a *OpenCV* e a *Tracking.JS*.

O NPM também permite baixar diversos exemplos prontos de uma aplicação PWA, tornando mais simples a implementação de toda a estrutura de diretórios e arquivos que compõem uma Aplicação Web Progressiva. Esses diretórios são compostos de arquivos que formam a base de uma aplicação PWA, como HTML, CSS, *JavaScript*, fontes, imagens e outros. No entanto, dois arquivos são fundamentais para que seja possível que uma aplicação PWA possa ser instalada no dispositivo e funcionar, mesmo sem uma conexão com a internet. São eles: o “*sw.js*” (*Service Worker*) e o “*manifest.json*” (*Web Manifest*). A definição desses arquivos é

essencial para que o PWA tenha as mesmas semelhanças que uma aplicação nativa do dispositivo, como tela de abertura da aplicação, ícone de acesso na tela do dispositivo e ajuste de cores da barra de notificação do dispositivo móvel.

Desta forma, o primeiro passo, após estabelecer a estrutura da aplicação PWA, é adicionar ao cache da implementação do *Service Worker* os arquivos indispensáveis ao funcionamento da aplicação quando não tiver uma conexão com a internet, permitindo seu funcionamento *offline*. Porém, para que um PWA seja utilizado na MCC, ele deverá ter, além dos arquivos básicos de funcionamento da aplicação HTML, CSS e *JavaScript*, os arquivos necessários para que o módulo de execução local possa fazer o processamento da tarefa da aplicação no próprio dispositivo móvel.

Desta forma, os arquivos definidos no cache do *Service Worker* utilizado no desenvolvimento da aplicação PWA deste estudo de caso incluíram as bibliotecas e funções para que o módulo de execução local da aplicação seja capaz de fazer o processamento das tarefas de detecção de faces e cálculos da sequência Fibonacci, como pode ser visto na Figura 15.

Figura 15 – Arquivos de cache do *Service Worker* da aplicação PWA

```
1 this.addEventListener('install', function(event) {
2   event.waitUntil(
3     caches.open(CACHE_NAME).then(function(cache) {
4       return cache.addAll([
5         //Rotas da Aplicação HTML
6         './',
7         './face',
8         './fibo',
9         //Estilos da Aplicação CSS
10        './css/materialize.css',
11        './css/custom.css',
12        //Bibliotecas do Módulo de Execução Local
13        './js/lib/face-min.js',
14        './js/lib/tracking-min.js',
15        //Arquivos JavaScript da Aplicação
16        './js/jquery-3.2.1.min.js',
17        './js/materialize.min.js',
18        //Imagens da Aplicação
19        './assets/img/iconapp.png',
20        './assets/img/brasao.png',
21        './assets/img/favicons.ico',
22        //Arquivo Web Manifest
23        './manifest.json',
24      ]);
25    });
26  });
27 });
```

Fonte: Elaboração própria

Com a definição do cache do *Service Worker*, a aplicação PWA poderá ser utilizada sem a necessidade de uma conexão com a internet, visto que os arquivos necessários para o funcionamento da aplicação serão baixados para o dispositivo móvel quando a aplicação PWA for acessada pela primeira vez pelo usuário. Para que o PWA tenha funcionamento semelhante a um aplicativo nativo, sendo instalado e acessível, através de um ícone no dispositivo móvel, foi necessário definir o arquivo *Web Manifest* ou *manifest.json*, como pode ser visto na Figura 16.

Figura 16 – Exemplo do arquivo *Web Manifest* da aplicação PWA

```
1  "name": "mcc-pwa",
2  "short_name": "mcc-pwa",
3  "theme_color": "#028c00",
4  "background_color": "#44b4e7",
5  "display": "standalone",
6  "start_url": "./",
7  "icons": [
8    {
9      "sizes": "96x96",
10     "type": "image/png"
11     }, {"src": "../assets/img/iconapp.png",
12     "sizes": "76x76",
13     "type": "image/png"
14     }, {"src": "../assets/img/iconapp.png",
15     "sizes": "36x36",
16     "type": "image/png"
17   ]
```

Fonte: Elaboração própria

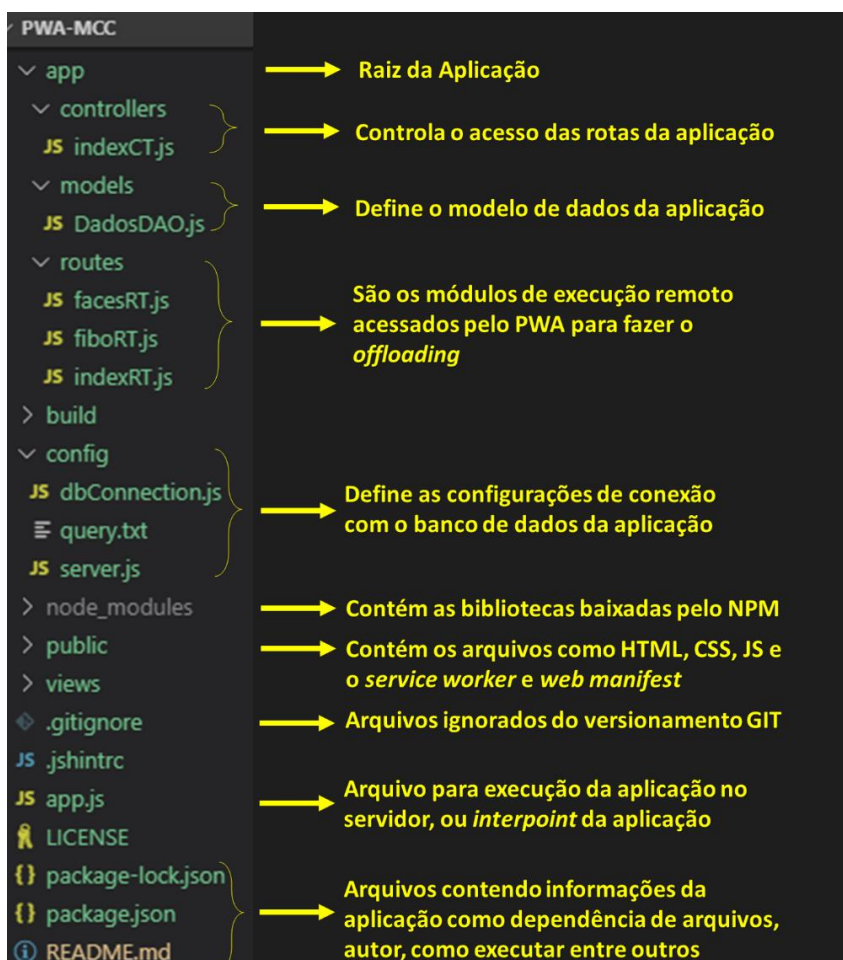
Assim, com a definição da aplicação PWA, é preciso agora estabelecer o módulo de execução remoto utilizado pelo servidor ou nuvem para fazer o processamento do *offloading* da tarefa enviada pela aplicação PWA executada no dispositivo móvel. Essa implementação pode ser feita considerando a estrutura da aplicação desenvolvida no servidor remoto, sendo um dos principais padrões dessa estrutura o padrão de desenvolvimento MVC (*Model, View, Controller*) ou modelo, visão e controle.

Esse padrão consiste na definição de camadas que permitem que a aplicação seja desenvolvida, considerando as interações entre cada uma dessas camadas

sendo que, a camada de controle irá tratar as requisições feitas pelo usuário da aplicação, como acessar um determinada rota que leve para uma página específica, ou seja, a visão a que o usuário terá acesso. Esse controle também se comunica com o modelo que está relacionado aos dados da aplicação, como arquivos de um banco ou outros específicos, como imagens, por exemplo. Esse modelo foi definido como estrutura da aplicação utilizada no servidor remoto da aplicação PWA desenvolvida.

Ao utilizar o NPM, a aplicação irá adicionar um arquivo chamado *package.json* que irá conter todas as informações da aplicação, como o nome, dados do autor e as dependências utilizadas por ela. Assim, também é gerada uma pasta na aplicação chamada *node_modules* com todas as bibliotecas que foram baixadas utilizando o gerenciador de pacotes do *NodeJS*, o NPM. A Figura 17 apresenta como a estrutura de diretório da aplicação remota foi implementada.

Figura 17 – Estrutura da aplicação remota para execução do *offloading*

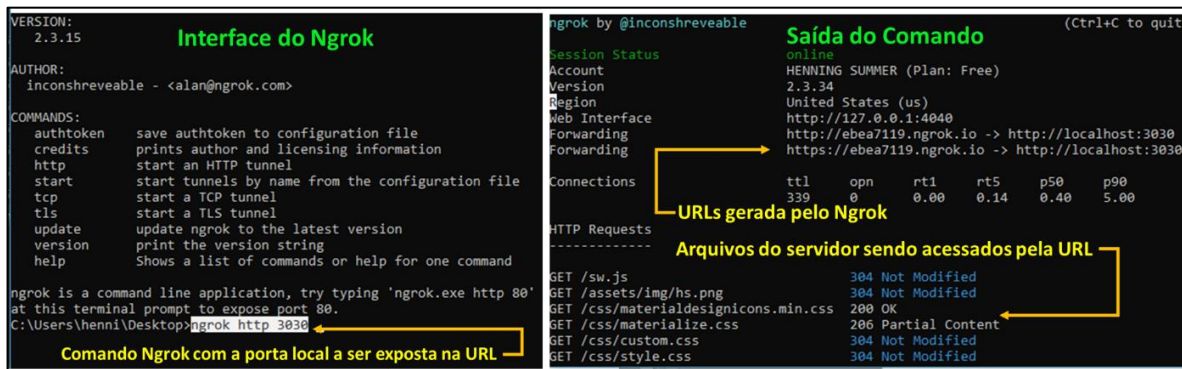


Ao finalizar o desenvolvimento da aplicação PWA, é importante que seja feita uma análise utilizando uma ferramenta capaz de fornecer informações tanto da performance quanto da qualidade da aplicação PWA que foi desenvolvida.

4.4. Inspeção – Analisar aspectos da aplicação PWA na MCC

A análise feita da aplicação PWA, através da ferramenta *LightHouse*, possibilitou a obtenção de informações importantes sobre alguns aspectos da aplicação PWA que foi desenvolvida para um ambiente de MCC. Assim, para realizar a análise da aplicação PWA, primeiro foi preciso disponibilizar essa aplicação em um servidor, tornando-a acessível através de um endereço na internet. Para isso, a aplicação foi disponibilizada em um servidor local e seu acesso foi feito através de uma URL utilizando uma ferramenta chamada *Ngrok*, que permite gerar URLs para acesso de servidores rodando em uma máquina local ou *localhost*. Assim, a partir do endereço local da aplicação e da porta utilizada por ela, <http://localhost:3030>, foi gerada a URL externa da aplicação utilizando o *Ngrok*, tornando o acesso desse caminho, o *localhost* da aplicação local, acessível a partir da internet, através de uma nova URL, como pode ser visto na Figura 18.

Figura 18 – Utilização da ferramenta Ngrok para gerar URLs externas



```
VERSION:
 2.3.15
Interface do Ngrok
AUTHOR:
inconshreveable - <alan@ngrok.com>
COMMANDS:
authtoken save authtoken to configuration file
credits prints author and licensing information
http start an HTTP tunnel
start start tunnels by name from the configuration file
tcp start a TCP tunnel
tls start a TLS tunnel
update update ngrok to the latest version
version print the version string
help Shows a list of commands or help for one command

ngrok is a command line application, try typing 'ngrok.exe http 80'
at this terminal prompt to expose port 80.
C:\Users\henri\Desktop>ngrok http 3030
Comando Ngrok com a porta local a ser exposta na URL

ngrok by @inconshreveable
Session Status
Account HENNING SUMMER (Plan: Free)
Version 2.3.34
Region United States (us)
Web Interface http://127.0.0.1:4040
Forwarding http://ebee7119.ngrok.io -> http://localhost:3030
Forwarding https://ebee7119.ngrok.io -> http://localhost:3030
Connections
-----
t1l opn rt1 rt5 p50 p90
339 0 0.00 0.14 0.40 5.00
URLs gerada pelo Ngrok

HTTP Requests
-----
GET /sw.js 304 Not Modified
GET /assets/img/hs.png 304 Not Modified
GET /css/materialdesignicons.min.css 200 OK
Arquivos do servidor sendo acessados pela URL
GET /css/materialize.css 206 Partial Content
GET /css/custom.css 304 Not Modified
GET /css/style.css 304 Not Modified
```

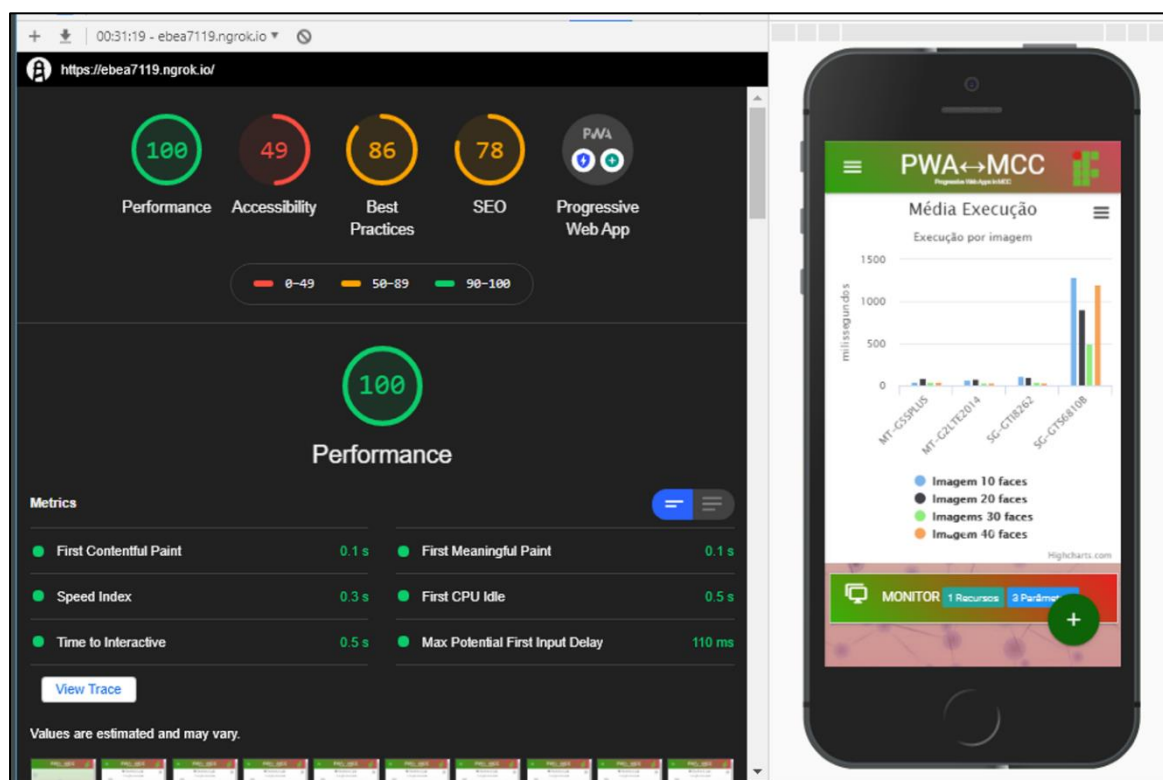
Fonte: Elaboração própria

Assim, o uso do *Ngrok* permite gerar URLs já com o protocolo de segurança exigido pelas Aplicações Web Progressivas, o HTTPS. O uso de HTTPS permite que o PWA tenha mais segurança quanto às informações trafegadas entre um servidor remoto e esse tipo de aplicação, garantido ainda que os arquivos acessados e guardados em cache para funcionamento da aplicação sejam realmente os que estão

armazenados no servidor dela, impedindo que sofram algum tipo de alteração indevida. Com isso, a partir da URL gerada pelo *Ngrok*, foi possível realizar a análise da qualidade da aplicação PWA utilizando a ferramenta de inspeção *LightHouse*.

A análise do *LightHouse* permitiu obter diferentes informações da aplicação PWA na MCC. Essas informações foram definidas antes de iniciar a inspeção da aplicação pela ferramenta, incluindo a seleção das categorias a serem analisadas e também outros parâmetros que permitem obter uma melhor análise do PWA, como simular diferentes redes móveis durante a avaliação realizada pela ferramenta. Um exemplo dos parâmetros que foram definidos e dos resultados obtidos com a análise feita com o uso do *LightHouse* pode ser visto na Figura 19.

Figura 19 – Resultados da análise da aplicação PWA com o *LightHouse*



Fonte: Elaboração própria

Os resultados da análise feita com a ferramenta *LightHouse* na aplicação PWA desenvolvida para ser usada na MCC apresentaram informações importantes sobre a qualidade dessa aplicação. Esses resultados permitem entender o comportamento da aplicação desenvolvida, pois, além de trazer resultados com notas sobre algumas categorias específicas, também apontam observações sobre como a aplicação pode

ser melhorada em cada um dos pontos categorizados pela análise realizada pela ferramenta *LightHouse*, como performance, acessibilidade, melhores práticas de desenvolvimento e, ainda, a otimização para mecanismos de busca da aplicação ou (*Search Engine Optimization* – SEO). Os resultados obtidos e a explicação das categorias analisadas pelo *LightHouse* podem ser vistos na Tabela 6.

Tabela 6 – Resultados da análise da aplicação pela ferramenta *LightHouse*

Categoria Analisada	Descrição da Categoria da Análise	Resultado
Performance da Aplicação	Mede a performance de execução e carregamento da aplicação, simulando diferentes condições de redes	Nota 100
Acessibilidade da Aplicação	Analisa o quanto a aplicação é acessível para usuários especiais, apresentando possíveis melhorias	Nota 49
Melhores Práticas de Desenvolvimento	Analisa a qualidade das práticas utilizadas no desenvolvimento da aplicação como minificar códigos	Nota 86
Otimização de Busca da Aplicação	Analisa o quanto a aplicação encontra-se otimizada para ser encontrada por mecanismos de busca	Nota 78

A média geral dos resultados obtidos com a análise da aplicação PWA desenvolvida, através da ferramenta *LightHouse*, foi de 78,25, sendo considerado um resultado positivo, visto que o objetivo no desenvolvimento desta aplicação não esteve voltado para torná-la mais acessível ou otimizada para mecanismos de buscas. Assim, o resultado mais importante para uma aplicação PWA a ser utilizada na MCC é sua performance, sendo que a ferramenta *LightHouse* apresentou a nota máxima 100 para essa categoria referente à aplicação PWA que foi desenvolvida.

Assim, depois da análise do PWA e dos resultados obtidos com a ferramenta *LightHouse*, é importante que essa aplicação seja avaliada a partir do seu desempenho em um ambiente de MCC, permitindo obter resultados referentes ao uso dessa aplicação para realizar a execução de tarefas específicas em diferentes dispositivos móveis. A utilização desta Aplicação PWA na MCC também possibilita o envio de tarefas para serem processadas em um servidor remoto através do *offloading*.

4.5. Avaliação – Avaliar a aplicação PWA na MCC

Para avaliar a aplicação PWA na MCC desenvolvida, é preciso considerar que na mesma aplicação encontram-se dois estudos de caso diferentes, envolvendo a execução de tarefas referentes à detecção de faces em imagens e ao cálculo de números da sequência Fibonacci. Assim, para avaliar a aplicação desenvolvida em um ambiente MCC, foram definidos no planejamento os seguintes pontos:

- cargas de trabalho formadas por tarefas de detecção de faces e cálculos da sequência Fibonacci;
- a métrica de tempo execução por tarefa executada na aplicação PWA e no servidor remoto;
- o uso de diferentes dispositivos móveis e um servidor remoto para fazer a execução das tarefas local através do *offloading*;
- o uso das redes sem fio WiFi, 3G e 4G para fazer a execução das tarefas da aplicação PWA na MCC no servidor remoto;
- o número de execução de 30 vezes para cada uma das tarefas das cargas de trabalho de detecção de faces e cálculo da sequência Fibonacci da aplicação.

Com isso, considerando o primeiro ponto do planejamento, foram definidas inicialmente as tarefas a serem utilizadas nas cargas de trabalhos para fazer a detecção de faces e o cálculo da sequência Fibonacci. Para cada carga de trabalho dos estudos de caso, foram definidas 15 tarefas referentes à detecção de faces em imagens e cálculos de números da sequência Fibonacci da aplicação PWA na MCC. Assim, para as tarefas de detecção de faces, foram definidas imagens contendo 10, 20, 30, 40 e 50 faces, sendo que cada imagem contendo essas faces foi variada em três diferentes dimensões, totalizando 15 imagens. Já os números escolhidos como tarefa para cálculo da sequência Fibonacci foram o 20, 22, 24, 26, 28, 30, 31, 32, 33, 34, 35, 36, 37, 38 e 40, conforme visto na Tabela 7.

Tabela 7 – Tarefas das cargas de trabalhos da avaliação

IMAGENS DA CARGA DE TRABALHO DE DETECÇÃO DE FACES														
Nome da Tarefa			Imagens Dimensão 1		Imagem Dimensão 2		Imagem Dimensão 3							
Faces-Dimensão			Tamanho	Resolução	Tamanho	Resolução	Tamanho	Resolução						
F10-D1	F10-D2	F10-D3	968 KB	4094x2091	185 KB	1446x713	16 KB	324x163						
F20-D1	F20-D2	F20-D3	1667 KB	4032x3966	366 KB	1453x1430	52 KB	327x323						
F30-D1	F30-D2	F30-D3	1803 KB	3963x4035	397 KB	1467x1493	58 KB	333x340						
F40-D1	F40-D2	F40-D3	1496 KB	4020x3104	324 KB	1475x1139	35KB	318x245						
F50-D1	F50-D2	F50-D3	1451 KB	4007x2487	272 KB	1434x891	24 KB	325x201						

NÚMEROS DA CARGA DE TRABALHO DA SEQUÊNCIA FIBONACCI														
Fibo	Fibo	Fibo	Fibo	Fibo	Fibo	Fibo	Fibo	Fibo	Fibo	Fibo	Fibo	Fibo	Fibo	Fibo
20	22	24	26	28	30	31	32	33	34	35	36	37	38	40

Para coletar as métricas de tempo de execução das tarefas desta avaliação, foram definidos os dispositivos móveis em que a aplicação PWA será executada e o servidor remoto que irá fazer o processamento do *offloading* das tarefas, como pode ser visto na Tabela 8.

Tabela 8 – Dispositivos móveis e servidor remoto da avaliação

Informações dos Recursos	Dispositivos Móveis				Servidor Remoto
	Moto G5S PLUS	Moto G2 2014 LTE	Samsung GT-I8262	Samsung GT-S6810B	Containers Docker no Windows
CPU	8 cores	4 cores	2 cores	1 core	4 cores
RAM	2857 MB	869 MB	842 MB	445 MB	4 GB
SSD	23,9 GB	12,2 GB	4,7 GB	2 GB	60 GB
Tela	1920x1080	1280x720	800x480	480x320	1366x768
Bateria	3000 mAh	2400 mAh	2100 mAh	1500 mAh	Ilimitado
Rede	WiFi,3G,4G	WiFi,3G,4G	WiFi,3G	WiFi,3G	Ethernet
Sistema	Android 8.1	Android 7.1	Android 6.0	Android 4.1	Windows 10/Linux

Assim, para realizar a avaliação em cada um dos quatro dispositivos móveis, o servidor da aplicação PWA foi executado dentro de quatro *Containers Docker* em uma

máquina rodando o sistema operacional Windows 10. Esse isolamento do servidor da aplicação foi feito para evitar interferência durante o processamento das tarefas enviadas pelos dispositivos móveis através do *offloading*, durante a coleta dos resultados da avaliação. Assim, em cada container Docker, foi executado um servidor *localhost*, sendo esse caminho exposto por um URL utilizando o *Ngrok*, como foi mencionado na fase 4 da Seção 3.4, durante a análise da qualidade da aplicação PWA. Desta forma, cada um dos dispositivos acessou um servidor específico para obter acesso à aplicação PWA e fazer o processamento das tarefas da aplicação.

Considerando ainda a interferência quanto aos resultados da avaliação, cada dispositivo foi configurado de modo a manter um padrão durante a coleta dos resultados, sendo consideradas as seguintes definições:

- Número mínimo de aplicativos executando em segundo plano
- Nível de luminosidade da tela em 50%
- Carga completa da bateria de todos dispositivos

Esses ajustes ajudam a diminuir as possíveis interferências nos resultados da avaliação feita com esses dispositivos, visto que um dispositivo com maior brilho de tela ou com outras aplicações sendo executadas paralelamente poderia causar interferência nas métricas de tempo de execução por tarefas coletadas (JUNIOR *et al.*, 2019). Com isso, foi definido o número de 30 execuções de cada uma das tarefas das cargas de trabalhos em cada dispositivo móvel e no servidor remoto, sob cada conexão de rede utilizada por eles, como resumido na seguinte equação:

$$DC = ((CT \times QD \times QR) \times NR) \times 2 \quad (1)$$

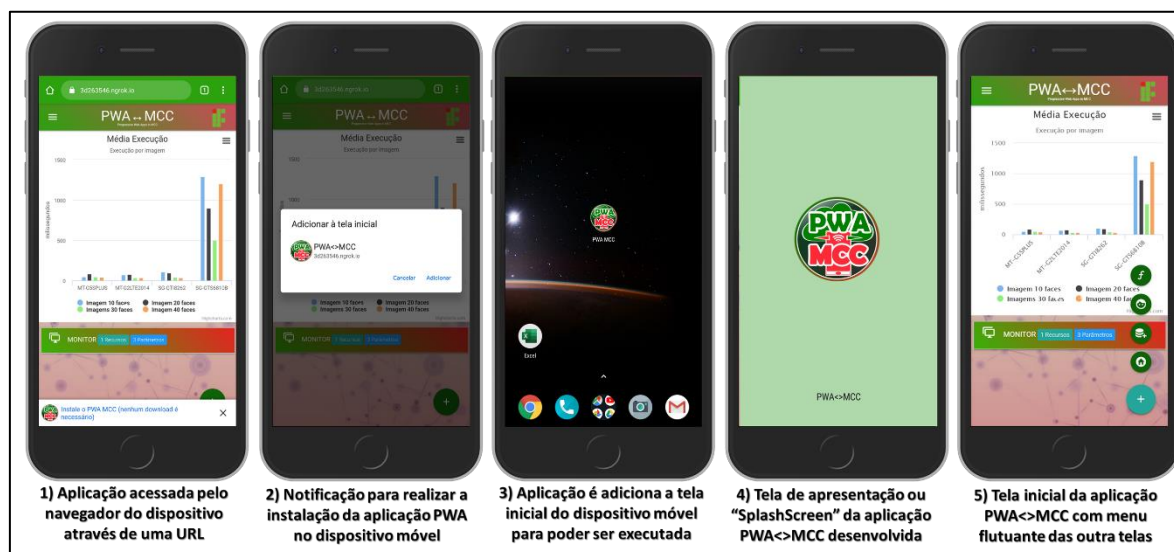
Onde **DC** é quantidade de dados coletados; **CT** a quantidade de tarefas das cargas de trabalhos; **QD** quantidade de dispositivos; **QR** quantidade de redes sem fio; e **NR** número de repetições das execuções; **2** é a constante referente à execução local e na nuvem para cada repetição. Assim, através das execuções das tarefas das cargas de trabalhos nos dispositivos móveis e no servidor remoto, sob as redes sem fio, foram obtidos os resultados desta avaliação, como serão apresentados a seguir.

A avaliação deste trabalho referente ao desenvolvimento de uma Aplicação Web Progressiva (PWA) na Computação Móvel em Nuvem foi possível através da execução de tarefas referentes à detecção faces e cálculos da sequência Fibonacci, em diferentes dispositivos móveis e em servidores utilizando containerizações Docker, como será apresentada na Seção 5, referente aos resultados dos estudos de caso desta pesquisa. Desta forma, através dos dispositivos móveis definidos no planejamento da avaliação, a aplicação PWA na MCC foi acessada utilizando a versão padrão do navegador Chrome instalado nos dispositivos móveis e, desta forma, acessando as *URLs* geradas pela *Ngrok* para acesso das aplicações executadas nos contêineres Docker rodando em uma máquina física local.

Com isso, foram criados quatro containers executando em cada um deles um servidor da aplicação PWA, sendo que para cada um desses servidores, foi gerado uma URL específica acessada em cada um dos quatro dispositivos móveis utilizados na avaliação. Com isso, no momento em que a URL é acessada no navegador dos dispositivos móveis pela primeira vez, o próprio navegador já reconhece que se trata de uma Aplicação Web Progressiva, devido à existência dos arquivos do *Service Worker* e *Web Manifest* e emite uma notificação para que a aplicação seja instalada no dispositivo móvel, tornando-a acessível através de um ícone instalado na tela inicial dos dispositivos móveis utilizados na avaliação

É interessante ressaltar que uma aplicação PWA precisa ser acessada através do protocolo de segurança HTTPS para garantir a segurança dos dados trafegados entre a aplicação PWA e o servidor que a disponibiliza. Assim, através do *Ngrok*, é possível gerar links pelo protocolo HTTPS tornando a aplicação PWA instalável como uma aplicação nativa convencional. Com aplicação PWA instalada em cada um dos dispositivos móveis, foi possível acessar a tela referente à execução de umas tarefas relacionadas aos estudos de caso de detecção e faces em imagens e cálculos da sequência Fibonacci. A Figura 20 apresenta como a aplicação PWA foi acessada em cada um dos dispositivos móveis utilizando um navegador web e como ocorreu seu processo de instalação em cada um dos dispositivos móveis da avaliação.

Figura 20 – Aplicação PWA sendo acessada e instalada nos dispositivos



Fonte: Elaboração própria

Utilizando o menu flutuante da aplicação, as telas referentes a cada tarefa dos estudos de caso foram acessadas para então realizar a execução das tarefas das cargas de trabalho de cada estudo de caso envolvendo a avaliação da aplicação PWA na MCC. Antes de iniciar a execução das tarefas referentes à avaliação dos estudos de caso, primeiro foram definidos os aspectos referentes ao ambiente de execução das tarefas relacionadas aos estudos de casos. Foram definidas as seguintes ações:

- Carregar completamente a bateria de todos os dispositivos móveis, definindo ainda o brilho da tela de todos em 50% e deixando executando em segundo plano apenas aplicações essenciais para o funcionamento do dispositivo.
- Executar o servidor da aplicação em contêineres Docker, disponibilizando a aplicação através de *URLs* sobre o protocolo HTTPS, e assim acessar essas *URLs* em cada um dos dispositivos móveis instalando a aplicação PWA.
- Colocar as imagens referentes à carga de trabalho de detecção de faces nos dispositivos móveis para serem processadas pela aplicação PWA nos próprios dispositivos e remotamente através do *offloading* no servidor remoto.
- Definir em cada execução das cargas de trabalhos as redes suportadas pelos dispositivos, sendo a ordem das definições as redes WiFi, 3G e por último 4G, visto que só apenas dois dispositivos da avaliação suportam essa rede.

Com isso, todas as tarefas da carga de trabalho de detecção de faces foram executadas local e na nuvem utilizando cada um as redes WiFi, 3G e 4G, sendo que primeiro foi selecionado o mesmo tipo de conexão de rede em todos os dispositivos móveis para, em seguida, executar todas as tarefas da carga de trabalho utilizando essa rede. Esse processo foi feito para cada uma das conexões de rede suportada pelos dispositivos móveis, como o WiFi, 3G e o 4G. Desta forma, foi possível coletar as informações referentes à execução das tarefas, sendo esses dados coletados e armazenados em um banco de dados *MongoDB*, que permite o armazenamento de dados usando o formato JSON (*JavaScript Object Notation*).

Assim, considerando a equação (1) $DC = ((CT \times QD \times QR) \times NR) \times 2$ apresentada na Seção 4.5 referente ao planejamento da avaliação da aplicação PWA na MCC, essa ficou definida conforme apresentado na Tabela 9.

Tabela 9 – Aplicação da equação (1) no planejamento da avaliação

Termo	Descrição dos Termos da Equação	Referência
2	Constante referente à execução de cada tarefa local em cada um dos dispositivos móveis e remotamente no servidor.	2
NR	Quantidade de vezes em que as cargas de trabalho serão executadas em cada um dos dispositivos sob cada uma das redes móveis suportadas	15
QR	Quantidade de redes móveis utilizadas para fazer a conexão entre os dispositivos móveis e o servidor remoto, sendo que dois dispositivos não suportam a conexão de rede 4G.	3
QD	Quantidade de dispositivos móveis utilizados na avaliação sendo desconsiderado o servidor remoto da constante da equação.	4
CT	Tarefas das cargas de trabalhos referentes aos estudos de caso da aplicação PWA de detecção e faces e cálculos da sequência Fibonacci	30
DC	Dados coletados com as execuções referentes aos dispositivos móveis, redes e tarefas das cargas de trabalho.	10800

Assim, primeiro foram feitas as execuções das tarefas da carga de trabalho referentes ao estudo de caso de detecção de faces, como é visto na Figura 21.

Figura 21 – Execução da tarefa de detecção de face na aplicação PWA



Fonte: Elaboração própria

Em seguida, foram feitas as execuções referentes às tarefas da carga de trabalho da sequência Fibonacci, como pode ser visto na Figura 22.

Figura 22 – Execução da tarefa de cálculo Fibonacci na aplicação PWA



Fonte: Elaboração própria

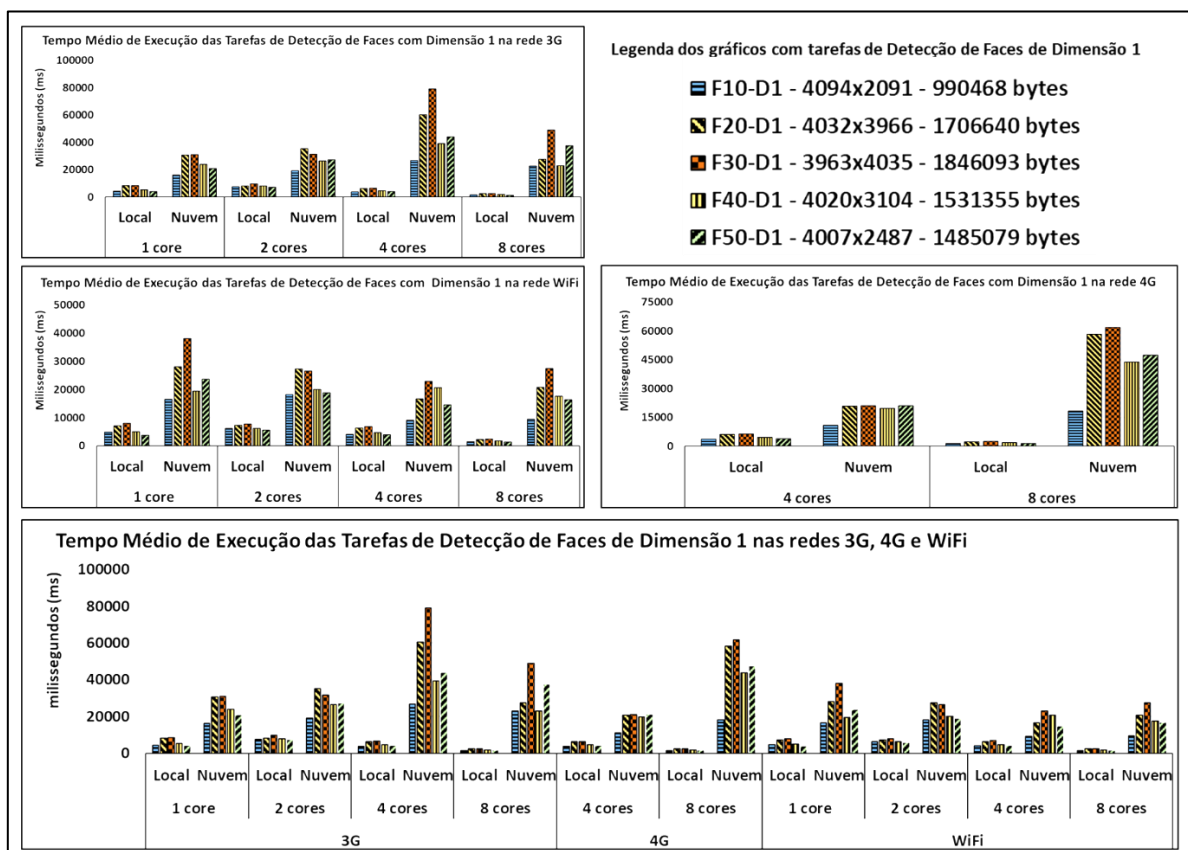
5. Resultados

Neste capítulo, serão apresentados os resultados da avaliação realizada através dos estudos de caso utilizando a aplicação PWA desenvolvida, através da metodologia proposta neste trabalho.

5.1. Resultados Detecção de Faces

Os resultados apresentados a seguir são referentes às execuções das 15 tarefas presentes na carga de trabalho utilizada para avaliar o estudo de caso de da aplicação PWA de detecção de faces no ambiente de MCC. Os resultados são apresentados a partir das imagens de Dimensão 1 (grandes), Dimensão 2 (médias) e Dimensão 3 (pequenas). Os resultados, apresentam, respectivamente, o tempo médio de execução local e na nuvem das tarefas de Dimensão 1, sob as conexões de rede 3G, 4G e WiFi em todos os dispositivos móveis, como podem ser vistos na Figura 23.

Figura 23 – Tempo de execução imagens de Dimensão 1 na rede 3G, 4G e WiFi

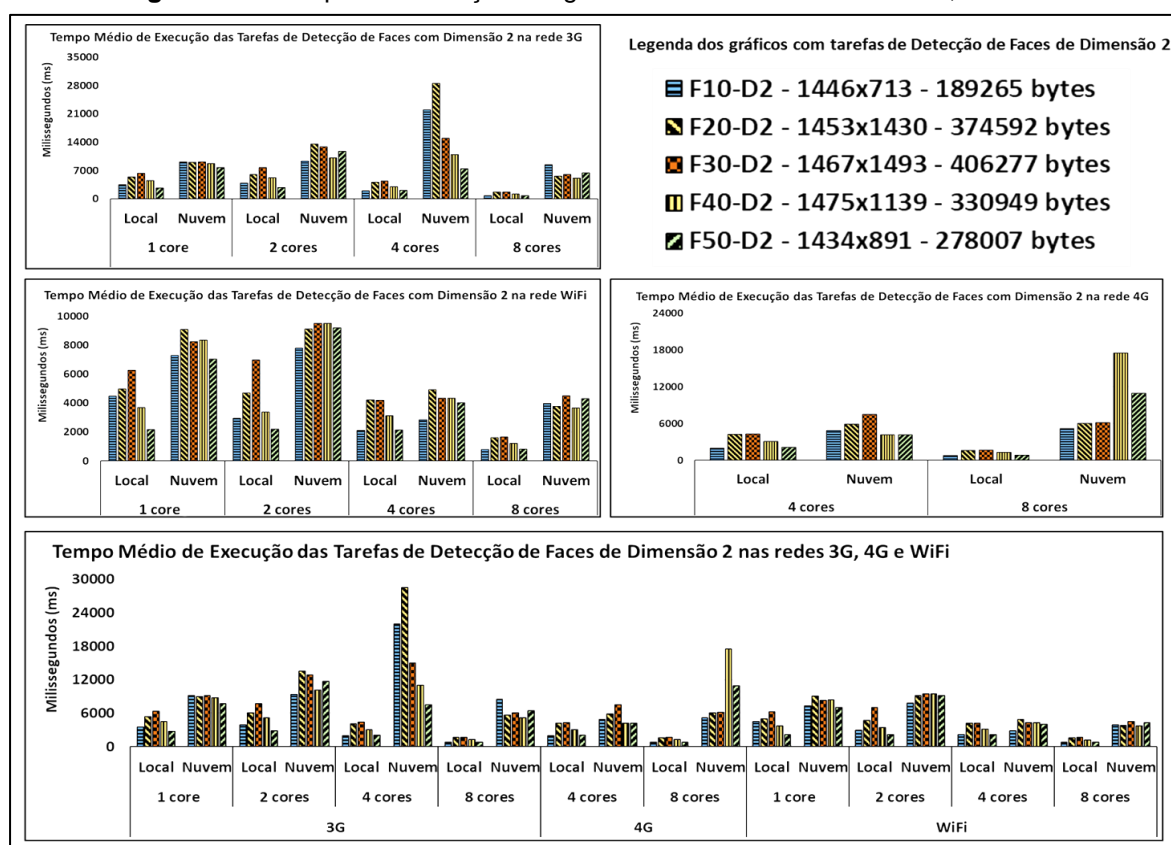


Fonte: Elaboração própria

As tarefas de detecção de faces executadas pela aplicação PWA na MCC através do *offloading* no servidor remoto, utilizando as redes 3G, 4G e WiFi, apresentaram resultados que apontam para uma melhor execução das imagens de Dimensão 1 no próprio dispositivo móvel. É interessante ainda observar que o dispositivo com 4 cores de CPU apresentou um tempo médio de execução local das tarefas bem próximo ao de dispositivos inferiores com apenas 1 ou 2 cores de CPU.

Com isso, é possível verificar que o tamanho em bytes da imagem acaba sendo um dos principais causadores na demora envolvendo o processamento dessas imagens no servidor remoto a partir das conexões de redes avaliadas. Isso é evidenciado analisando o tempo médio de execução dessas tarefas no servidor remoto, que é cinco vezes maior que o tempo médio da execução dessas tarefas nos dispositivos móveis, como pode ser visto na Figura 23 em todas as redes avaliadas. Já os resultados referentes às tarefas com Dimensão 2 sob as redes 3G, 4G e WiFi, nos dispositivos móveis usados na avaliação, podem ser vistos na Figura 24 a seguir.

Figura 24 – Tempo de execução imagens de Dimensão 2 na rede 3G, 4G e WiFi

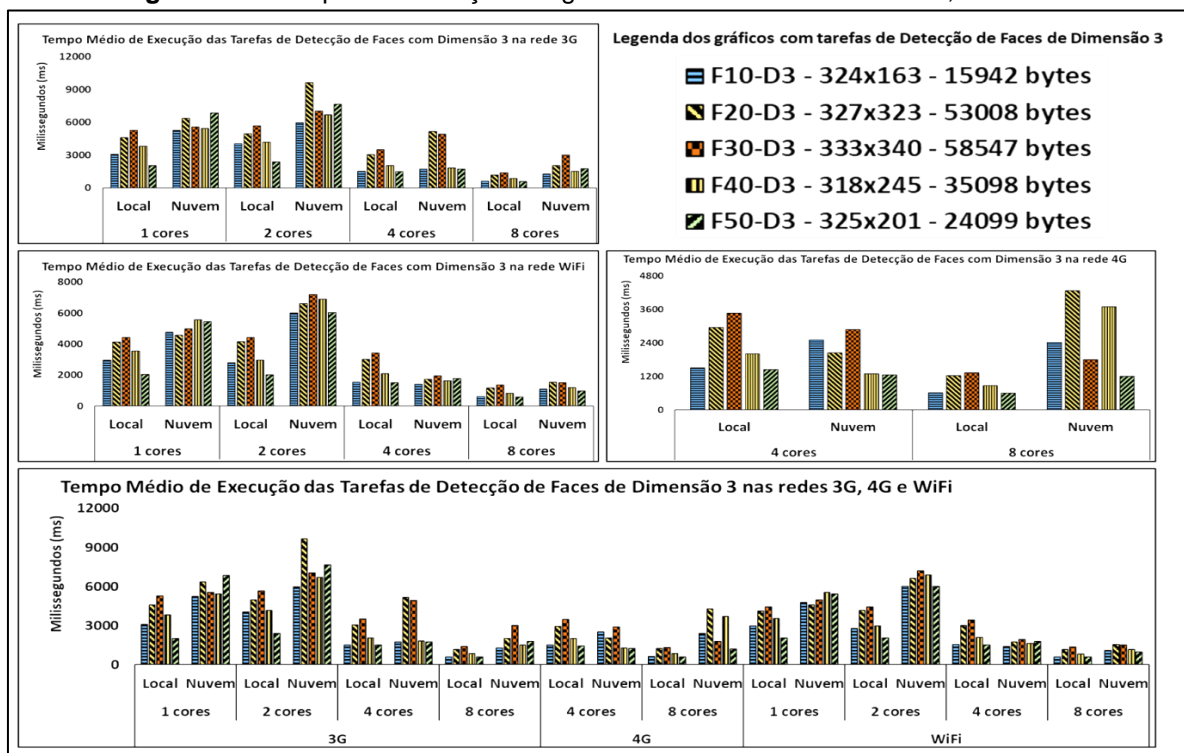


Fonte: Elaboração própria

O tempo médio de execução das tarefas de detecção de faces de Dimensão 2, quando executadas localmente e na nuvem através da rede 3G, apresentou nos dispositivos de 1 e 2 cores resultados semelhantes com as execuções sob a rede WiFi. Contudo, é possível observar uma diferença no tempo de execução das imagens F20-D2, F30-D2 e F40-D2 nas redes 3G e WiFi nos dispositivos de 1 e 2 cores. Já o tempo médio de execução das imagens de Dimensão 2 na nuvem, utilizando a rede 3G, foi maior que a utilizando a rede WiFi nos dispositivos móveis com 4 e 8 cores possivelmente por possuírem maior compatibilidade com os novos padrões de WiFi.

Importante destacar o tempo de execução das tarefas de Dimensão 2 na nuvem sob a rede 3G, a partir do dispositivo de 4 cores, que apresentou um tempo de execução acima da média do tempo apresentado nos demais dispositivos da avaliação. Os maiores tempos de execução das tarefas com Dimensão 2 na nuvem deram-se no dispositivo de 4 cores, sob a rede 3G, e no de 8 cores, sob a rede 4G. Essa diferença pode estar associada com a mudança repentina na conexão de rede móvel, onde uma rede 4G passa para uma rede inferior. Os próximos resultados são referentes às tarefas de Dimensão 3, como pode ser visto na Figura 25.

Figura 25 – Tempo de execução imagens de Dimensão 3 na rede 3G, 4G e WiFi



Fonte: Elaboração própria

Os resultados referentes ao tempo médio de execução das tarefas de detecção de faces local e na nuvem de Dimensão 3, utilizando a rede 4G, nos dispositivos de 4 e 8 cores, são melhores se comparados com as execuções realizadas utilizando a rede 3G. No entanto, não são melhores que o tempo médio das execuções utilizando a rede WiFi. É importante também observar que as imagens de Dimensão 3, que possuem em média menos de 100 Kb de tamanho, são as mais rápidas de serem enviadas pelo *offloading*, sob a rede 3G. Também vale mencionar que o tempo de execução das imagens de Dimensão 3 na nuvem utilizando a rede 3G foi maior que na rede WiFi, a partir dos dispositivos móveis de 4 e 8 cores, como visto na Figura 25.

Os resultados da avaliação do estudo de caso envolvendo a execução de tarefas de detecção de faces local e na nuvem, através das redes WiFi, 3G e 4G, apresentam médias referentes às execuções de diferentes tarefas compostas por imagens categorizadas, como Dimensão 1, 2 e 3. Os resultados apresentados demonstram a relação entre as 15 tarefas da carga de trabalho de detecção de faces em relação aos 4 dispositivos móveis utilizados para executar a aplicação PWA desenvolvida e, assim, fazer a execução local das tarefas no próprio dispositivo e também através do *offloading* no servidor remoto.

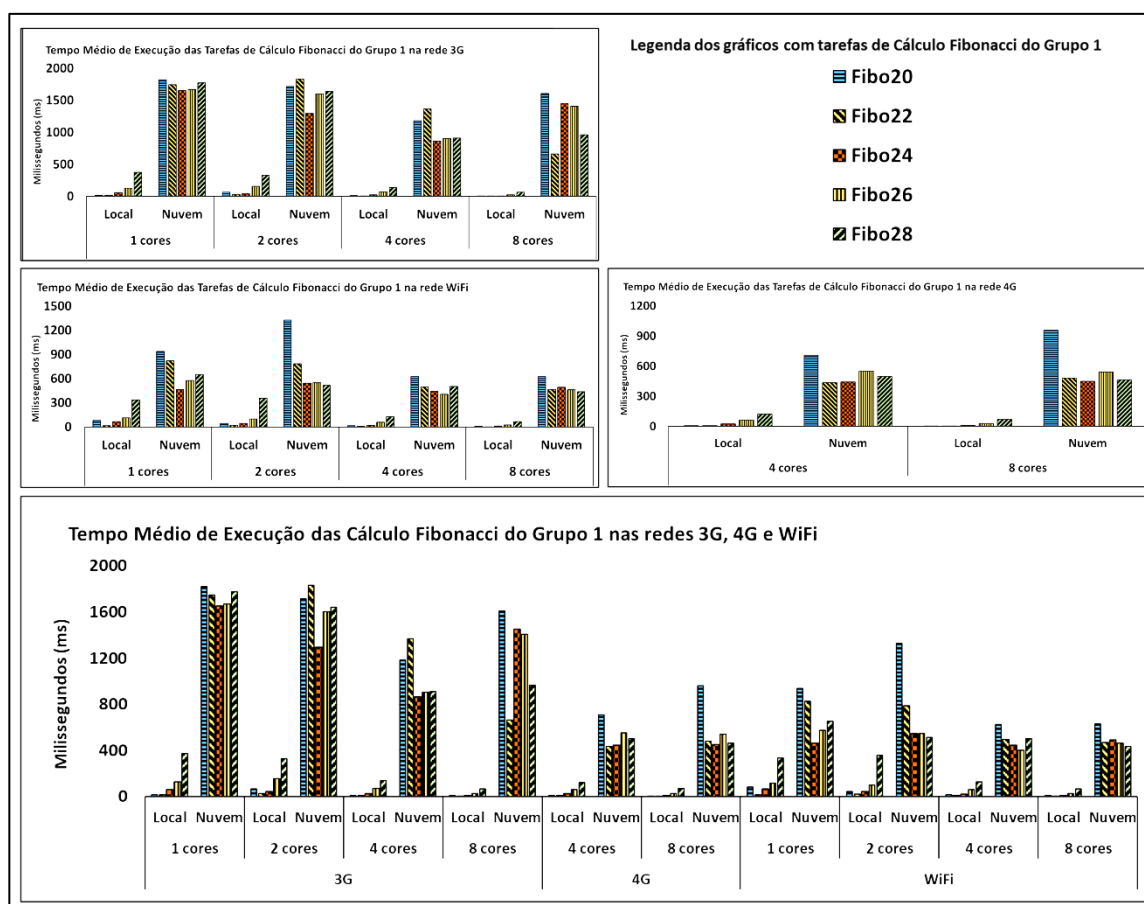
Assim, diante dos resultados apresentados, é possível observar que o *offloading* não foi tão benéfico em relação às tarefas de detecção de faces, principalmente, as tarefas de Dimensão 1, que possuem imagens com maiores tamanhos (bytes) e dimensões (pixels) em relação às imagens de Dimensão 2 e 3. No entanto, em alguns casos, a execução do *offloading* foi considerada positiva para as imagens de Dimensão 3, por essas possuírem menor tamanho em bytes, facilitando que sejam transferidas e processadas mais rapidamente pelo servidor.

Contudo, mesmo para as imagens com menor tamanho, o *offloading* só foi positivo em pouquíssimos casos, principalmente quando a rede utilizada foi a WiFi. Assim, as redes 3G e 4G apresentaram um tempo médio de execução das tarefas acima do tempo médio das execuções feitas sob a rede WiFi. Nas execuções na nuvem das tarefas do estudo de caso de detecção de faces, os melhores resultados foram dos dispositivos de 4 e 8 núcleos, a partir da rede WiFi. Já nas execuções locais, o melhor desempenho foi do dispositivo com 8 cores, por seu poder computacional.

5.2. Resultados Cálculo do Fibonacci

Os resultados apresentados a seguir são referentes às execuções das 15 tarefas presentes na carga de trabalho utilizada para avaliar o estudo de caso de cálculos da sequência Fibonacci, a partir da Aplicação PWA na MCC. Os resultados são apresentados a partir de três grupos, sendo primeiramente exibidos os resultados com o tempo médio de execução das tarefas do Grupo 1 referente aos números 20, 22, 24, 26 e 28. Depois do grupo 2, dos números 30, 31, 32, 33 e 34 e, por último, as do grupo 3, com os números 35, 36, 37, 38 e 40. Os resultados da avaliação do grupo 1 podem ser vistos na Figura 26, sob as redes 3G, 4G e WiFi.

Figura 26 – Tempo de execução números do Grupo 1 nas redes 3G, 4G e WiFi



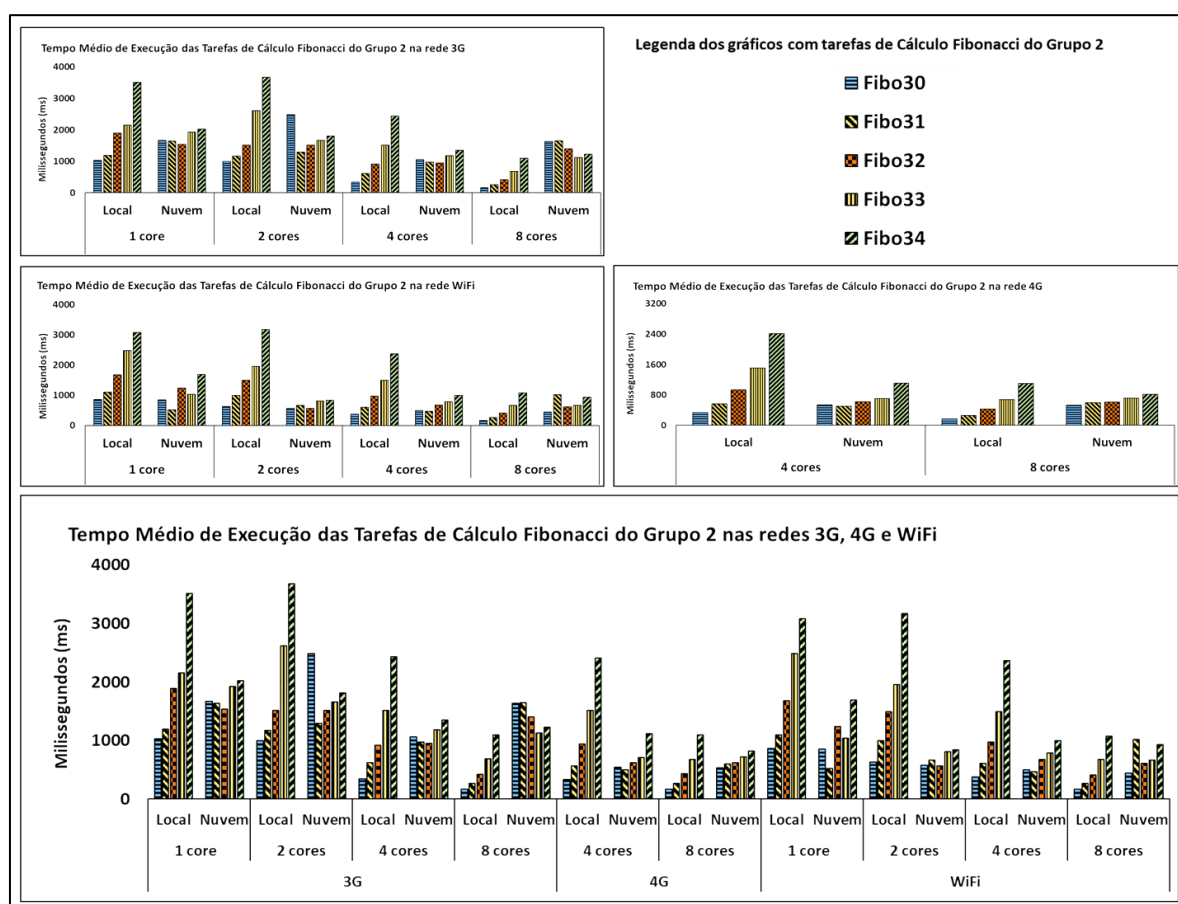
Fonte: Elaboração própria

O resultado da execução das tarefas do Grupo 1 do estudo de caso do cálculo Fibonacci apresenta uma melhor execução de todas as tarefas localmente, em todos os dispositivos usados na avaliação, visto que são números com pouco grau de

complexidade de processamento. Deste modo, a execução na nuvem acaba sendo mais demorada que uma execução local, uma vez que a tarefa precisa ser enviada para a nuvem para que, após seu processamento através do *offloading*, seja retornada para o dispositivo móvel já com o resultado do cálculo realizado. Com isso, só o tempo de envio dessas tarefas acaba sendo maior que o processamento delas nos próprios dispositivos da avaliação, até mesmo o de apenas 1 core de processamento.

Já as tarefas do Grupo 2 apresentam resultados diferentes do Grupo 1, por possuírem números com maior grau de complexidade de processamento do cálculo Fibonacci, como pode ser visto na Figura 27.

Figura 27 – Tempo de execução números do Grupo 2 nas redes 3G, 4G e WiFi



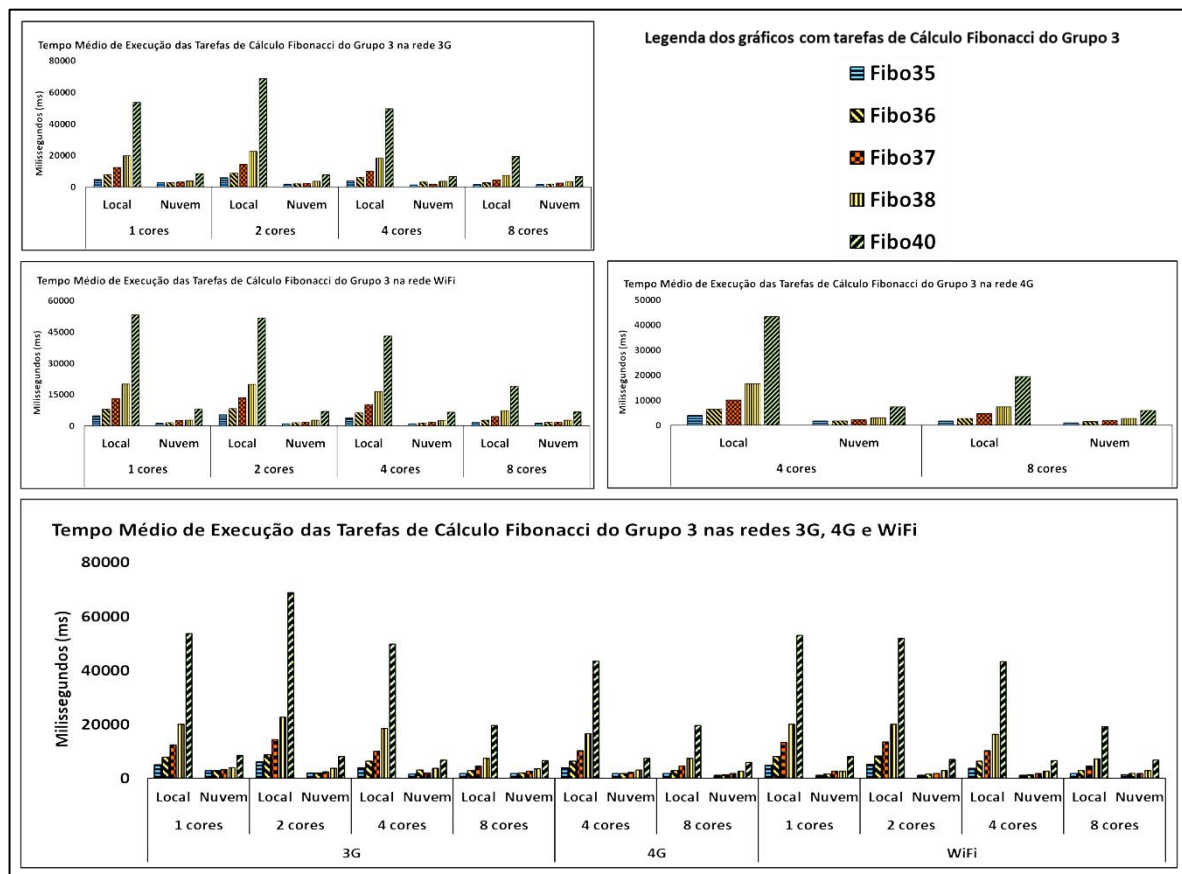
Fonte: Elaboração própria

As tarefas representadas pelos números 30 e 31 tiveram um tempo de execução menor quando executadas localmente nos dispositivos com 1, 2, 4 e 8 cores, sob a rede 3G. Já as tarefas referentes aos números 32, 33 e 34 tiveram um

tempo de execução menor quando processadas na nuvem, sob a rede 3G. Já sob a rede WiFi, o tempo de execução foi menor na nuvem para todas as tarefas do Grupo 2, nos dispositivos móveis utilizados na avaliação, com exceção para as tarefas 30, 31 e 32, quando executadas no dispositivo com 8 cores, sendo o tempo de execução dessas tarefas menor quando executadas localmente nesse dispositivo.

Em média, a execução das tarefas do Grupo 2 foi melhor em termos de tempo de execução quando processadas através do *offloading* em todos dispositivos móveis da avaliação. Os resultados das tarefas do Grupo 3 podem ser vistos na Figura 28.

Figura 28 – Tempo de execução números do Grupo 3 nas redes 3G, 4G e WiFi



Fonte: Elaboração própria

Os resultados referentes ao Grupo 3 apresentam melhores resultados quando processadas na nuvem em todos os dispositivos móveis da avaliação. Isso é devido à complexidade envolvendo o processamento desses números nos dispositivos móveis, visto que são números maiores que os das tarefas do Grupo 1 e 2 da carga de trabalho do estudo de caso do cálculo Fibonacci.

6. CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Nesse último capítulo, serão apresentadas as considerações finais sobre o uso de *Progressive Web Apps* na Computação Móvel em Nuvem, incluindo uma análise de como esse tipo de aplicação pode ser considerada uma opção no desenvolvimento de soluções dentro da MCC. Também serão considerados pontos importantes sobre o que foi desenvolvido e apresentado e que podem ser revistos, melhorados e explorados em trabalhos futuros de aplicações PWA na MCC.

6.1. Considerações Finais

Neste trabalho, foi visto como o uso de aplicações baseadas em *Progressive Web Apps* pode ser considerado relevante em ambientes da Computação Móvel em Nuvem. Para isso, foram apresentados os conceitos envolvendo a Computação Móvel em Nuvem, como a Computação Móvel, a Computação em Nuvem e as tecnologias de redes sem fio. Também foi apresentado o conceito de descarga computacional ou *computing offloading*, que é utilizado pela MCC para realizar a transferência da execução parcial ou total de métodos e tarefas do dispositivo móvel para a nuvem.

A partir desses conceitos, foi proposta uma metodologia para auxiliar no uso desse tipo de aplicação, além de permitir avaliar seu desempenho utilizando métricas importantes de avaliação da MCC. Com isso, foi possível observar o quanto aplicações baseadas em *Progressive Web Apps* podem ser consideradas uma alternativa interessante dentro da Computação Móvel em Nuvem, em abordagem envolvendo a descarga computacional, conhecida como *offloading*.

Assim, através da avaliação de desempenho realizada na aplicação PWA, a partir dos estudos de caso de detecção de faces e cálculo Fibonacci, foi possível demonstrar como o uso de PWA pode ser considerado uma alternativa relevante para desenvolver aplicações envolvendo a execução de diferentes tarefas em ambientes de MCC. Logo, esse tipo de aplicação surge como mais uma alternativa relevante entre outros modelos de aplicações já utilizadas por desenvolvedores ou projetistas, sendo o PWA mais uma alternativa a ser utilizada em projetos voltados para ambientes de MCC que usem estratégias de *offloading*.

6.2. Limitações

A proposta deste trabalho é considerada limitada, visto a abrangência do tema envolvendo a temática de Computação Móvel em Nuvem e uso de alternativas de descarga computacional “*offloading*” que podem ser desenvolvidas nessa área de pesquisa. Como destaque, está a importância de desenvolver opções de *offloading* parciais, visto que neste trabalho a proposta do *offloading* foi através da execução completa da tarefa no dispositivo móvel e no computador remoto.

Outra limitação está na métrica utilizada na avaliação deste trabalho que foi a de tempo de execução. Assim, é interessante também avaliar essas aplicações, a partir da métrica de consumo de energia, para determinar o impacto do consumo energético de tarefas de um PWA quando processadas nos dispositivos e na nuvem.

6.3. Trabalhos Futuros

Alguns pontos importantes podem ser trabalhados em propostas futuras, tornando possível o aperfeiçoamento do uso de aplicações PWA em ambientes de MCC, como desenvolver uma aplicação PWA que realize a execução de tarefas ou métodos em MCC de forma parcial. Esse tipo de abordagem de *offloading* parcial pode ser feito utilizando um recurso de troca de mensagens, chamado *WebSocket*, para distribuir, de forma assíncrona, execuções parciais entre diferentes dispositivos móveis e também em uma nuvem computacional.

Outro ponto que pode ser considerado em trabalhos futuros está na avaliação de outros estudos de caso que sejam diferentes do apresentado nesse trabalho. Isso consiste em utilizar estudos de caso já aplicados em outros trabalhos de MCC para desenvolver uma aplicação, através do conceito de *Progressive Web Apps*, usando a metodologia deste trabalho. Assim, é possível avaliar o quanto uma aplicação PWA pode ser mais relevante que aplicações nativas ou híbridas já utilizadas em outros trabalhos dentro da MCC. Desta forma, a metodologia deste trabalho consistiu no desenvolvimento e na avaliação de aplicações PWA dentro da MCC, através de dois estudos de casos para destacar a relevância que o uso de *Progressive Web Apps* pode ter em ambientes da Computação Móvel em Nuvem.

REFERÊNCIAS

- AKHERFI, K.; GERNDT, M.; HARROUD, H. Mobile cloud computing for computation offloading: Issues and challenges. **Applied Computing and Informatics**, 14, n. 1, p. 1-16, 2018.
- ALI, F. A. *et al.* Mobile device power models for energy efficient dynamic offloading at runtime. **Journal of Systems and Software**, v. 113, p. 173-187, 2016.
- ALIZADEH, M.; HASSAN, W. H. **Challenges and opportunities of mobile cloud computing**. *ieee*. 660-666, 2013.
- BIØRN-HANSEN, A.; MAJCHRZAK, T. A.; GRØNLI, T-M. Progressive Web Apps: The Possible Web-native Unifier for Mobile Development. In: **WEBIST**. p. 344-351, 2017.
- CAPPELLOZZA, A.; DE MORAES, G. H. S. M.; MUNIZ, L. M. Uso pessoal das tecnologias no trabalho: motivadores e efeitos à distração profissional. **RAC-Revista de Administração Contemporânea**, 21, n. 5, p. 605-626, 2017.
- CASALS, A.; SILVA, V. N. Computação na Nuvem em aplicações móveis. **USP**, 2016. Disponível em: https://www.ime.usp.br/~diogojp/computacao-movel-2017/seminar/ArthurCasals_ViniciusNascimento_CloudComputingAplicacoesMoveis_v2.pdf. Acesso em: 20/04/2018.
- CECONI, L. Experiência do usuário em Progressive Web Apps. **UCS**, 2019. Disponível em: <https://repositorio.ucs.br/xmlui/handle/11338/4780>. Acesso em: 20/08/2018
- CHAMAS, C. L. Consumo de energia em dispositivos móveis Android: análise das estratégias de comunicação utilizadas em Computation Offloading. Tese de Doutorado. **Universidade de São Paulo**. 2017.
- CHUN, B-G. *et al.* Clonecloud: elastic execution between mobile device and cloud. **ACM**. 301-314, 2011.
- COSTA, P. B. *et al.* An Analysis of the Impact of the Quality of Mobile Networks on the Use of Cloudlets. **2014 Brazilian Symposium on Computer Networks and Distributed Systems**. **IEEE**, p. 113-121, 2014.
- CUERVO, E. *et al.* MAUI: making smartphones last longer with code offload. **Proceedings of the 8th international conference on Mobile systems, applications, and services**. p. 49-62, 2010.
- DENG, S.; HUANG, L.; TAHERI, J.; ZOMAYA, A. Y. Computation offloading for service workflow in mobile cloud computing. **IEEE Transactions on Parallel and Distributed Systems**, 26, n. 12, p. 3317-3329, 2015.

DOS SANTOS, G. *et al.* CAOS D2D: Uma solução para offloading de métodos entre dispositivos móveis. **Anais Estendidos do XXIII Simpósio Brasileiro de Sistemas Multimídia e Web**. SBC, p. 164-168, 2017.

EWALDSSON, U. Ericsson mobility report 2016. Stockholm, Sweden, **Nov**, 2016.

FERNANDO, N.; LOKE, S. W.; RAHAYU, W. Mobile cloud computing: A survey. **Future Generation Computer Systems**, 29, n. 1, p. 84-106, 2013.

FIGUEIREDO, C. M.; NAKAMURA, E. Computação móvel: Novas oportunidades e novos desafios. **T&C Amazônia**, 1, n. 2, p. 21, 2003.

GAD-ELRAB, A. A.; ZAKY, E. H.; GHALI, N. I. an Adaptive Multi-Replica Data Offloading Scheme in Mobile Cloud Computing. **International Journal of Computer Science and Information Security**, 15, n. 1, p. 558, 2017.

GANI, A. *et al.* A review on interworking and mobility techniques for seamless connectivity in mobile cloud computing. **Journal of Network and Computer Applications**, 43, p. 84-102, 2014.

GOMES, F. A. A. *et al.* A contextual data offloading service with privacy support. **Proceedings of the 22nd Brazilian Symposium on Multimedia and the Web**. p. 23-30, 2016.

GOUDARZI, M.; MOVAHEDI, Z.; NAZARI, M. Mobile cloud computing: A multisite computation offloading. **2016 8th International Symposium on Telecommunications (IST)**. IEEE, p. 660-665, 2016.

GRØNLI, T.; BIØRN-HANSEN, A.; MAJCHRZAK, T. A. Software development for mobile computing, the internet of things and wearable devices: Inspecting the past to understand the future. **Proceedings of the 52nd Hawaii International Conference on System Sciences**. 2019.

GUGIK, Gabriel. A história dos computadores e da computação. **Tecmundo**, v. 6, 2009. Disponível em: <https://www.tecmundo.com.br/tecnologia-da-informacao/1697-a-historia-dos-computadores-e-da-computacao.htm>

GUO, S *et al.* Energy-efficient dynamic offloading and resource scheduling in mobile cloud computing. **IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications**. IEEE,. p. 1-9, 2016.

JAIN, R. The Art of Computer Systems Performance Analysis (Techniques for Experimental Design, Measurement, Simulation, and Modeling). **The Computer Journal**, 35, n. 3, p. 207-207, 1991.

JARARWEH, Y *et al.* The future of mobile cloud computing: integrating cloudlets and mobile edge computing. **2016 23rd International conference on telecommunications (ICT)**. IEEE, p. 1-5, 2016.

JEONG, H.; MOON, S. Offloading of web application computations: A snapshot-based approach. **2015 IEEE 13th International Conference on Embedded and Ubiquitous Computing**. IEEE, p. 90-97, 2015.

JIAO, Lei *et al.* Cloud-based computation offloading for mobile devices: State of the art, challenges and opportunities. In: **2013 Future Network & Mobile Summit**. IEEE, p. 1-11, 2013.

JUNIOR, W. *et al.* A context-sensitive offloading system using machine-learning classification algorithms for mobile cloud environment. **Future Generation Computer Systems**, 90, p. 503-520, 2019.

KADU, C.; BHANODIYA, P.; SAMVATSAR, M. Review of Challenges in Accessing Cloud services through Mobile Devices. **International Journal of Scientific & Engineering Research**, 3, n. 11, 2012.

KAUSHIK, N.; GAURAV, J. K. A literature survey on mobile cloud computing: open issues and future directions. **International Journal of Engineering and Computer Science**, 3, n. 05, 2014.

KHAN, M. A. A survey of computation offloading strategies for performance improvement of applications running on mobile devices. **Journal of Network and Computer Applications**, 56, p. 28-40, 2015.

KHAN, S. *et al.* Forensic challenges in mobile cloud computing. **2014 International Conference on Computer, Communications, and Control Technology (I4CT)**. IEEE, p. 343-347, 2014.

KHODA, M. E. *et al.* Efficient Computation Offloading Decision in Mobile Cloud Computing over 5G Network. **Mobile Networks and Applications**, 21, n. 5, p. 777-792, 2016.

KITANOV, S.; MONTEIRO, E.; JANEVSKI, T. 5G and the Fog—Survey of related technologies and research directions. **2016 18th Mediterranean Electrotechnical Conference (MELECON)**. IEEE, p. 1-6, 2016.

KOSTA, S. *et al.*, Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading. **IEEE**. p. 945-953, 2012.

KUMAR, K. *et al.* A survey of computation offloading for mobile systems. **Mobile Networks and Applications**, v. 18, n. 1, p. 129-140, 2013.

LEAL, B. G. Avaliação de Desempenho de Sistemas. **Juazeiro, Bahia**. Disponível em: <http://www.univasf.edu.br/~brauliro.leal/ensino/ADS/ADS.pdf> 2016.

LI, W. Y. S. *et al*, JSCloud: Toward Remote Execution of JavaScript Code on Handheld Devices. **IEEE**, 240-245, 2012.

LI, Yi. *et al*. Energy optimization with dynamic task scheduling mobile cloud computing. **IEEE Systems Journal**, v. 11, n. 1, p. 96-105, 2015.

LILJA, D. J. Measuring computer performance: a practitioner's guide. **Cambridge university press**, 2005.

LIMA, M. Como criar seu primeiro Progressive Web App do Zero. **IMasters**, 2017. Disponível em: <https://imasters.com.br/desenvolvimento/como-criar-seu-primeiro-progressive-web-app-do-zero>

LO'AI, A. T. *et al*. Mobile cloud computing model and big data analysis for healthcare applications. **IEEE Access**, 4, p. 6171-6180, 2016.

LONGO, J. *et al*. SoC It to EM. ElectroMagnetic Side-Channel Attacks on a Complex System-on-Chip. : **Springer**. 9293: 620-640 p. 2015.

MARTINS, M. d. C. Quando o preconceito condena a genialidade: Alan Turing e as suas máquinas. **Correio dos Açores**, p. 11-11, 2015.

MEIRELLES, F. Pesquisa Anual do Uso de TI nas Empresas 2019. **FGVcia, FGV-EAESP, 30ª edição**, 2019.

MELL, P.; GRANCE, T. The NIST Definition of Cloud Computing. **NIST**, 2011. Disponível em: <https://csrc.nist.gov/publications/detail/sp/800-145/final>

NAKAHARA, F. A. Um modelo auto-adaptativo para apoio ao offloading dinâmico em aplicações móveis. Dissertação de Mestrado, f, 125 **Repositório UFSCAR**, 2018. Disponível em: https://repositorio.ufscar.br/bitstream/handle/ufscar/10113/NAKAHARA_FI%C3%A1vio_2018.pdf?sequence=4

OLIVEIRA, D. M. Análise de Disponibilidade e Consumo Energético em Ambientes de Mobile Cloud Computing. Dissertação de Mestrado. **Universidade Federal de Pernambuco**. 2014

RAHIMI, M. R. *et al*. Mobile cloud computing: A survey, state of art and future directions. **Mobile Networks and Applications**, 19, n. 2, p. 133-143, 2014.

RAJ, P. A Literature Study On Various Factors Affecting Computational Offloading Performance. **IJITR**, 5, n. 1, p. 5511-5515, 2017.

ROUTAIB, H. *et al*. **Modeling and evaluating a cloudlet-based architecture for mobile cloud computing**. **IEEE**. 1-7, 2014.

SANTANA, N. A.; LINS, F. A. A.; DE SOUSA, E. T. G. Performance Evaluation of Mobile Applications in Mobile Cloud Environments. **IEEE Latin America Transactions**, 14, n. 11, p. 4597-4602, 2016.

SANTOS, T. Fundamentos da computação em nuvem. **Editora Senac São Paulo**, 2018. 9788539611980.

SARWAR, M.; SOOMRO, T. R. Impact of Smartphone's on Society. **European journal of scientific research**, 98, n. 2, p. 216-226, 2013.

SHARMA, P. Evolution of mobile wireless communication networks-1G to 5G as well as future prospective of next generation communication network. **International Journal of Computer Science and Mobile Computing**, 2, n. 8, p. 47-53, 2013.

SHIRAZ, M.; AHMED, E.; GANI, A.; HAN, Q. Investigation on runtime partitioning of elastic mobile applications for mobile cloud computing. **The Journal of Supercomputing**, 67, n. 1, p. 84-103, 2014.

SILVA, Francisco Airton *et al.* Benchmark applications used in mobile cloud computing research: a systematic mapping study. **The Journal of Supercomputing**, v. 72, n. 4, p. 1431-1452, 2016.

SILVA, J. S. *et al.* Invasive technique for measuring the energy consumption of mobile devices applications in mobile cloud environments. **2017 IEEE international conference on systems, man, and cybernetics (SMC)**. IEEE, p. 2724-2729, 2017.

SMITH, C. **These are the Galaxy S5's next-gen processors**. 2014. Disponível em: <https://bgr.com/2014/02/26/galaxy-s5-processor-snapdragon-801-exynos-5422/>. Acesso em: 04/10/2019.

URRIZA, José M. *et al.* Economia de energia em dispositivos móveis. In: **VI Workshop de Comunicação sem Fio e Computação Móvel**, p. 48-56, 2004.

XU, C *et al.* Energy consumption of mobile offloading for javascript applications. **2015 26th Irish Signals and Systems Conference (ISSC)**. IEEE, p. 1-6, 2015.

XU, C. *et al.* MOJA-Mobile Offloading for JavaScript Applications. **IET Conference Proceedings**. The Institution of Engineering & Technology, 2014.

YU, S.; WANG, X.; LANGAR, R. Computation offloading for mobile edge computing: A deep learning approach. **2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)**. IEEE, p. 1-6, 2017.

ZHANG, J. *et al.* A webpage offloading framework for smart devices. **Mobile Networks and Applications**, v. 23, n. 5, p. 1350-1363, 2018.

ZHOU, B.; DASTJERDI, A. V.; CALHEIROS, R.; SRIRAMA, S. *et al.* mCloud: A context-aware offloading framework for heterogeneous mobile cloud. **IEEE Transactions on Services Computing**, 2015.