

**UNIVERSIDADE ESTADUAL PAULISTA “JÚLIO DE MESQUITA FILHO”
FACULDADE DE ARQUITETURA, ARTES E COMUNICAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM MÍDIA E TECNOLOGIA**

PATRÍCIA ESTEVES TRINDADE

**O PROGRESSIVE WEB APPS – PWA – COMO FERRAMENTA PARA A
PRODUÇÃO AUDIOVISUAL**

Bauru
2020

PATRÍCIA ESTEVES TRINDADE

**O PROGRESSIVE WEB APPS – PWA – COMO FERRAMENTA PARA A
PRODUÇÃO AUDIOVISUAL**

Relatório Técnico apresentado ao Programa de Pós-Graduação em Mídia e Tecnologia, da Faculdade de Artes, Arquitetura e Comunicação da Universidade Estadual Paulista “Júlio de Mesquita Filho”, para a obtenção do título de Mestre em Mídia e Tecnologia, sob a orientação da Prof^a. Dr^a. Letícia Passos Affini.

Bauru
2020

T833p

Trindade, Patrícia Esteves

O Progressive Web Apps - PWA : como ferramenta para a produção audiovisual / Patrícia Esteves Trindade. -- Bauru, 2020

64 p.

Dissertação (mestrado profissional) - Universidade Estadual Paulista (Unesp), Faculdade de Arquitetura, Artes e Comunicação, Bauru

Orientadora: Letícia Passos Affini

1. Audiovisual. 2. Progressive Web Apps. 3. Inovação. 4. Tecnologia. 5. Aplicativos móveis. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca da Faculdade de Arquitetura, Artes e Comunicação, Bauru. Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.

ATA DA DEFESA PÚBLICA DA DISSERTAÇÃO DE MESTRADO DE PATRÍCIA ESTEVES TRINDADE



UNIVERSIDADE ESTADUAL PAULISTA

Câmpus de Bauru



ATA DA DEFESA PÚBLICA DA DISSERTAÇÃO DE MESTRADO DE PATRÍCIA ESTEVES TRINDADE, DISCENTE DO PROGRAMA DE PÓS-GRADUAÇÃO EM MÍDIA E TECNOLOGIA, DA FACULDADE DE ARQUITETURA, ARTES E COMUNICAÇÃO - CÂMPUS DE BAURU.

Aos 27 dias do mês de março do ano de 2020, às 10:30 horas, no(a) via sistemas de videoconferência e outras ferramentas para comunicação a distância, reuniu-se a Comissão Examinadora da Defesa Pública, composta pelos seguintes membros: Professora Assistente Doutora LETÍCIA PASSOS AFFINI - Orientador(a) do(a) Departamento de Comunicação Social da Faculdade de Arquitetura, Artes e Comunicação da Unesp - câmpus de Bauru / Universidade Estadual Paulista, Professor Doutor GLAUCO MADEIRA DE TOLEDO do(a) Faculdade de Comunicação e Informática / Universidade Metodista de Piracicaba, Professora Associada MARIA CRISTINA GOBBI do(a) Departamento de Comunicação Social da Faculdade de Arquitetura, Artes e Comunicação da Unesp - câmpus de Bauru / Universidade Estadual Paulista, sob a presidência do primeiro, a fim de proceder a arguição pública da DISSERTAÇÃO DE MESTRADO de PATRÍCIA ESTEVES TRINDADE, intitulada **O Progressive Web Apps como ferramenta para a produção audiovisual**. Após a exposição, a discente foi arguida oralmente pelos membros da Comissão Examinadora, tendo recebido o conceito final: *aprovada*. Nada mais havendo, foi lavrada a presente ata, que após lida e aprovada, foi assinada pelos membros da Comissão Examinadora.

Professora Assistente Doutora LETÍCIA PASSOS AFFINI

Professor Doutor GLAUCO MADEIRA DE TOLEDO

Professora Associada MARIA CRISTINA GOBBI

Letícia Affini
realizado via videoconferência
por parecer circunstanciado

PATRÍCIA ESTEVES TRINDADE

**O PROGRESSIVE WEB APPS – PWA – COMO FERRAMENTA PARA A
PRODUÇÃO AUDIOVISUAL**

Área de Concentração: Ambientes Midiáticos e Tecnológicos
Linha de Pesquisa: Gestão Midiática e Tecnológica

Banca Examinadora:

Prof^ª. Dr^ª. Letícia Passos Affini
Presidente/Orientadora / Universidade Estadual Paulista “Júlio de Mesquita Filho”

Prof^ª. Dr^ª Maria Cristina Gobbi
Docente/ Universidade Estadual Paulista “Júlio de Mesquita Filho”

Prof. Dr. Glauco Madeira de Toledo
Docente/ Universidade Metodista de Piracicaba

Resultado: _____

Bauru, _____/_____/_____

AGRADECIMENTOS

A Deus, por sempre colocar pessoas maravilhosas em meu caminho, as quais me fazem acreditar em um mundo melhor e me encorajam a prosseguir. Obrigada por nunca soltar a minha mão e me guiar em todos os momentos.

Agradeço ao meu marido, Éder Carlos, que esteve ao meu lado todo o tempo e não mediu esforços para eu chegar até aqui. Aos meus filhos, Léo e Pedro, que entenderam meus momentos de ausência. A eles minha eterna gratidão e todo o meu amor; não sou nada sem vocês.

Agradeço aos meus pais, Rubens e Altair, que sempre me apoiaram e incentivaram a ir atrás dos meus sonhos. Obrigada pelo amor incondicional.

Agradeço à minha orientadora, Prof^a. Dr^a. Letícia Passos Affini, pela oportunidade de realizar este trabalho. Obrigada pela confiança e por me atender com paciência todas as vezes que bati em sua porta. Agradeço por todos os ensinamentos compartilhados, de forma admirável, e por me guiar nos primeiros passos da pós-graduação. Muito obrigada por tudo.

Aos docentes que contribuíram com a minha formação e com este trabalho, meu muito obrigado.

RESUMO

Com o avanço das tecnologias móveis, alicerçado pelas funcionalidades de mobilidade e flexibilidade no processo de comunicação dos *smartphones*, alavancou-se uma nova modalidade de desenvolvimento *mobile*. Pode-se dizer que o desenvolvimento de aplicações móveis é caracterizado pela mobilidade e conectividade do usuário, independentemente do tipo de dispositivo móvel (*smartphone*, *tablet*, *palm-top*, *notebook*), bem como pelo ambiente computacional (*software*) disponível. Para tanto, este trabalho apresenta os resultados de um estudo exploratório sobre a tecnologia *Progressive Web Apps*, por meio da revisão bibliográfica junto aos autores Tal Alter (2017), Sérgio Lopes (2016), Reinaldo Ferraz (2018), Murarolli e Girotti (2015), Miletto e Bertagnolli (2014), Kurose e Ross (2013) e de fontes não tradicionais, como o *site* da *Google Developers*. Tal conceito surgiu em 2015, tendo a proposta de um *web app* com a interface de um aplicativo e a praticidade de um *site*. Com a evolução da telefonia móvel e da oferta de aplicativos, analisou-se, a partir de coleta de dados, *Comscore* e *MobileTime*, que: o *smartphone* é o principal meio de acesso à internet; queda no número de *downloads* de aplicativos nas lojas *Apple* e *Play Store*; e instalação de poucos aplicativos, com destaque para os de mensagens e redes sociais. Nesse contexto, o *Progressive Web Apps* apresenta-se como conceito emergente, que pode ser entendido como uma tecnologia que agrega os recursos disponíveis nos aplicativos nativos com os recursos da web, incluindo: conectividade progressiva (para qualquer usuário, independentemente do browser e do tipo de conexão); responsivo (elaborado para qualquer dispositivo, *desktop* ou *mobile*); *app-like*, semelhante a um aplicativo – (o usuário se sente em um aplicativo nativo); atual (não há necessidade de fazer atualizações, como está tudo na web, quando abrir o *app* a nova versão já será carregada); descobrível - *SEO Friendly* – (os mecanismos de busca conseguem encontrar o conteúdo dos aplicativos); reengajável (através de *push notifications*, o usuário pode ser constantemente engajável); instalável (podem ser adicionados à *homescreen* do dispositivo móvel); e linkável (mais fácil de compartilhar conteúdo ao enviar o *link* para alguém). Salienta-se que a tecnologia PWA tem sido utilizada para *e-commerce* e redes sociais. Nesse sentido, o objetivo geral da pesquisa é observar os contributos de uma aplicação *Progressive Web Apps* no processo de produção audiovisual. Determinou-se como pergunta de pesquisa: quais os contributos da

tecnologia PWA no desenvolvimento de aplicações móveis para a produção audiovisual? Para responder a esta pergunta, será elaborado um protótipo, como Prova de Conceito (PoC), abordando as possíveis inovações tecnológicas e recursos disponíveis da web. Para o desenvolvimento da aplicação serão utilizadas as linguagens padrão da web: HTML, CSS e *JavaScript*, em conjunto com os *frameworks* *Vue.js*, *Quasar*, *Spring Boot* e o banco de dados *SQLServer*. Como resultado, espera-se desenvolver o protótipo da aplicação proposta, apresentando os principais recursos da tecnologia *Progressive Web Apps*, como a implementação do *service worker*, *app manifest*, notificações *push* e atalho na *homescreen* do dispositivo. Desse modo, melhorar a experiência do usuário, proporcionando conectividade *offline*, carregamento rápido e aparência nativa.

Palavras-chave: Audiovisual. *Progressive web apps*. Inovação. Tecnologia. Aplicações móveis.

ABSTRACT

With the advancement of mobile technologies, based on the mobility and flexibility features in the smartphone communication process, a new form of mobile development has been leveraged. It can be said that the development of mobile applications is characterized by the mobility and connectivity of the user, regardless of the type of mobile device (smartphone, tablet, palm-top, notebook), as well as the computational environment (software) available. To this end, this work presents the results of an exploratory study on the Progressive Web Apps technology, through a bibliographic review with the authors Tal Alter (2017), Sérgio Lopes (2016), Reinaldo Ferraz (2018), Murarolli and Girotti (2015), Miletto and Bertagnolli (2014), Kurose and Ross (2013) and from non-traditional sources, such as the Google Developers website. This concept emerged in 2015, with the proposal of a web app with the interface of an application and the practicality of a website. With the evolution of mobile telephony and the offer of applications, it was analyzed, from data collection, Comscore and MobileTime, that: the smartphone is the main means of accessing the internet; drop in the number of app downloads at Apple and Play Store stores; and installation of few applications, with emphasis on messaging and social networks. In this context, Progressive Web Apps presents itself as an emerging concept, which can be understood as a technology that aggregates the resources available in native applications with web resources, including: progressive connectivity (for any user, regardless of browser and type) of connection); responsive (designed for any device, desktop or mobile); app-like, similar to an app - (the user feels like a native app); current (there is no need to make updates, as everything is on the web, when you open the app the new version will already be loaded); discoverable - SEO Friendly - (search engines can find application content); re-engaging (through push notifications, the user can be constantly engaged); installable (can be added to the mobile device's homescreen); and linkable (easier to share content when sending the link to someone). It should be noted that PWA technology has been used for e-commerce and social networks. In this sense, the general objective of the research is to observe the contributions of a Progressive Web Apps application in the audiovisual production process. It was determined as a research question: What are the contributions of PWA technology in the development of mobile applications for audiovisual production? To answer this question, a prototype, such as Proof of

Concept (PoC), will be elaborated, addressing the possible technological innovations and resources available on the web. For the development of the application the standard web languages will be used: HTML, CSS and JavaScript, together with the frameworks Vue.js, Quasar, Spring Boot and the SQLServer database. As a result, it is expected to develop the prototype of the proposed application, presenting the main features of the Progressive Web Apps technology, such as the implementation of the service worker, app manifest, push notifications and shortcut on the device's homescreen. Thus, improving the user experience, providing offline connectivity, fast loading and native appearance.

Keywords: Audiovisual. Progressive web apps. Innovation. Technology; Mobile applications.

LISTA DE ILUSTRAÇÕES

Figura 1 - Evolução da Web.....	27
Figura 2 - Funcionamento do Service Worker	33
Figura 3 - Arquitetura em 3 camadas	42
Figura 4 - Trecho de código Spring Boot.....	43
Figura 5 - Código Repositório de Classe	44
Figura 6 - Diagrama de entidade e Relacionamento	45
Quadro 1 – Requisitos funcionais do projeto.....	49
Figura 7 - Diagrama de Caso de Uso do projeto	50
Figura 8 - Fluxograma da Aplicação.....	51
Figura 9 - Responsividade da Aplicação.....	55
Figura 10 - Interface no modo <i>offline</i>	55
Figura 11 - Ícone instalado, <i>spashscreen</i> , arquivo <i>manifest.json</i>	56

LISTA ABREVIATURAS E SIGLAS

HTML	HyperText Markup Language
APP	Aplicativo
PWA	Progressive Web App
CSS	Cascading Style Sheets
URL	Uniform Resource Locator
API	Application Programming Interface
GPS	Global Positioning System
HTTPS	Hyper Text Transfer Protocol Secure
JSON	JavaScript Object Notation
SEO	Search Engine Optimization
UX	User Experience
UC	User Case
Java EE	Java Enterprise Edition
Java ME	Java Micro Edition
Java SE	Java Standard Edition
SPAs	Aplicativo de página única
CLI	Interface de linha de comando
SSL	Security Sockets Layer – Certificado de Segurança Digital
TSL	Transport Layer Secutiry – Transporte de Dados Seguro
IDE	Ambiente de Desenvolvimento Integrado
SQL	Structured Query Language – Linguagem de Consulta Estruturada
SGBD	Sistema Gerenciador de Banco de Dados
GUI	Graphical User Interface – Interface Gráfica do Usuário

SUMÁRIO

I INTRODUÇÃO	15
II RELATÓRIO TÉCNICO-CIENTÍFICO.....	18
Dados Gerais do Projeto	18
III DESCRIÇÃO DO PROJETO	19
3.1 Tema.....	19
3.2 Objeto/Problema	20
3.3 Objetivo Geral.....	20
3.4 Objetivos Específicos.....	21
IV JUSTIFICATIVA	22
V REFERENCIAL TEÓRICO	25
5.1 A web e sua evolução.....	25
5.2 Desenvolvimento de aplicações móveis.....	27
5.3 Tipos de desenvolvimento de aplicações móveis	28
5.3.1 Aplicações nativas.....	28
5.3.2 Aplicações híbridas	29
5.3.3 Aplicações web apps.....	30
5.4 Progressive Web Apps	31
5.4.1 Definição.....	31
5.4.2 Service Workers.....	33
5.4.3 App Manifest	35
5.4.3.1 JSON	35
5.4.4 HTTPS	36
5.4.5 HTML.....	36
5.4.6 CSS	37
5.4.7 Linguagem Javascript	38
5.4.8 Linguagem de Programação Java.....	39
5.4.9 Vue.js	40
5.4.10 Quasar	40
5.4.11 SQL Server	40
5.4.12 Spring Boot	41

VI MATERIAIS E MÉTODOS	42
6.1 Tecnologias e Recursos utilizados	42
6.1.1 Arquitetura e Tecnologias	42
6.1.1.1 Camada de Apresentação	43
6.1.1.2 Camada de Negócio	44
6.1.1.3 Camada de Persistência	44
6.2 Descrição da Prova de Conceito (PoC)	46
6.3 Aplicação de <i>Software</i>	46
6.3.1 Proposta de implementação do PWA <i>Pau a Pixel Design Movies</i>	48
6.4 Metodologia utilizada	52
6.4.1 Etapas do processo de pesquisa e desenvolvimento	53
VII RESULTADOS OBTIDOS	55
VIII PARCERIA INSTITUCIONAL	58
IX IMPACTOS E BENEFÍCIOS	59
X DIFICULDADES	60
XI COMENTÁRIOS GERAIS E PERSPECTIVAS	61
REFERÊNCIAS	63

I INTRODUÇÃO

A complexidade do processo de produção audiovisual é inimaginável pelo espectador. Envolve captação de recursos, negociação de direitos autorais, coordenação de equipe técnica, pedidos de autorização, cálculos, entre muitos outros aspectos despercebidos por trás da lente. Pode-se dizer que o crescente número de produtos audiovisuais disponibilizados na internet está relacionado à transformação tecnológica dos últimos anos, aliada à onipresença do *smartphone*.

Segundo dados da Digital 2019, da We Are Social e da Hootsuite¹, a população do país cresceu 0,7%, totalizando 211,6 milhões de habitantes. O número de brasileiros com acesso à internet aumentou em 10 milhões, um crescimento de 7,2% em relação a 2018. A penetração da internet atingiu 70% no Brasil, acima da média global, de 57%, onde mais de 149 milhões, dos 212 milhões de habitantes do país, são usuários e 85% dos internautas navegam na web todos os dias, sendo que 66% do acesso é por meio da internet móvel. Existem 215,2 milhões de conexões móveis no Brasil, o que significa que, para cada habitante, há 1,2 conexão por meio de algum tipo de *smartphone* (dados compilados até janeiro de 2019).

O Final Draft Mobile, *Untitled Screenwriting Notes*, *Artemis Pro* e *Movie Salate* 8 são exemplos de aplicativos *mobile* para a produção audiovisual existentes no mercado (outubro/2019). São *softwares* com licença de uso, utilizam o sistema de versões e há a necessidade de pagamento para novas funcionalidades. *O Weekend Read* e *Paper* se destacam por serem gratuitos, porém fazem uso da licença *shareware*². Muitos dos aplicativos existentes para a produção audiovisual são encontrados no formato *desktop* e *mobile*; entretanto, as funcionalidades oferecidas no ambiente *mobile* são restritas e com direcionamento para o *desktop*, sem contar a exigência de *hardware* e de *software*, solicitados para a instalação e a utilização do aplicativo.

Os diferenciais do uso dos dispositivos móveis residem em sua flexibilidade, liberdade e facilidade e, com esse tipo de tecnologia evoluindo rapidamente, a demanda por aplicativos cresce de forma exponencial, gerando mudanças na vida

¹ Fonte: Datareportal.

² Geralmente possuem funcionalidades limitadas e/ou tempo de uso gratuito do software limitado, após o fim do qual o usuário é requisitado a pagar para acessar a funcionalidade completa ou poder continuar utilizando o programa. Fonte: Wikipedia.

peçoal, empresarial, escolar, do transporte público, industrial etc. Com o crescimento das plataformas móveis, aliado à web 4.0, o *mobile-first* entrou em cena e sinalizou uma mudança na forma de analisar e desenvolver aplicativos para a *web*, o que permitiu o surgimento de novos conceitos, ideias, e a junção de diversas tecnologias.

O termo *mobile-first* é um conceito de desenvolvimento de software no qual o foco são os dispositivos *mobile*. Ou seja, ao se elaborar um site, cria-se toda a organização das páginas e a exibição dos dados com foco nos usuários de dispositivos *mobile*, mas sem se esquecer dos usuários de microcomputador. Como veremos em breve, atualmente existem muitos frameworks CSS para este tipo de desenvolvimento (PONTES, 2018, p.02).

Com isto, o desenvolvimento *web* mudou consideravelmente, os navegadores passaram a renderizar as páginas com maior precisão e os *sites* se adaptaram ao tamanho da tela dos dispositivos (*design* responsivo). Esses recursos têm permitido o desenvolvimento de novas interfaces, que estão mudando a forma como as organizações apresentam seus conteúdos, aumentando a eficiência e promovendo a inovação, trazendo padronização e mais possibilidades para as aplicações *web* em dispositivos móveis, com acesso a recursos como GPS, acelerômetro e câmera, por exemplo.

O crescimento de aplicações *mobile* na educação muito se dá pela popularização das tecnologias que utilizam esse tipo de serviço, o que permite o acesso desse tipo de recurso em dispositivos como *smartphones* e *tablets*. Com o advento do conceito *mobile learning* (*m-learning*), que integra o uso de tecnologias móveis no contexto educativo, o que introduz o processo educacional no cenário da “computação ubíqua” ou “computação pervasiva”, pautado pela mobilidade do usuário, conectividade, independência de dispositivo e ambiente computacional disponível em qualquer lugar, a qualquer tempo. A partir disso, e com o objetivo de investigar tais inovações, foi proposto o desenvolvimento de uma aplicação móvel, em caráter experimental, apoiada nos princípios da tecnologia *Progressive Web Apps*.

A partir desse cenário, este trabalho aborda o desenvolvimento de um aplicativo *mobile*, como prova de conceito (PoC) destinado à produção audiovisual universitária, utilizando os conceitos da tecnologia *Progressive Web Apps*. Serão

observadas quais as contribuições que o PWA pode oferecer para o processo de produção audiovisual, nas fases da pré-produção e produção.

A decisão pelo desenvolvimento utilizando PWA deve-se, principalmente, ao baixo custo e menor tempo dedicado à sua produção. Aplicações progressivas são *sites* tecnicamente regulares e, quando acessados em dispositivos móveis, possuem a aparência de um aplicativo nativo, e que, com o desenvolvimento de apenas um código, pode ser acessado por diversos tipos de plataformas e sistemas operacionais; além disso, a aplicação buscará oferecer diferentes níveis de envolvimento e novas possibilidades de aprendizagem, considerando que o aluno não está em um ponto fixo, mas sim fora da sala de aula tradicional, fazendo uso dos recursos oferecidos pelas tecnologias inerentes aos dispositivos móveis.

II RELATÓRIO TÉCNICO-CIENTÍFICO

Dados Gerais do Projeto

Mestranda: Patrícia Esteves Trindade

Área de Concentração – Linha de Pesquisa: Ambientes Midiáticos e Tecnológicos – Gestão Midiática e Tecnológica

Orientadora: Prof^a. Dr^a. Letícia Passos Affini

Título do Projeto: O *Progressive Web Apps* como ferramenta para a produção audiovisual

Período de Execução Física: 6 meses

Valor total do projeto: Este projeto não recebeu nenhum tipo de investimento financeiro.

Bolsas – Financiamento – Convênios e Parcerias: (Não houve)

Instituições participantes: UNESP Bauru

Caracterização da Pesquisa: Pesquisa Técnico-Científica de Base Tecnológica

Caracterização da pesquisa com uma breve justificativa para o enquadramento:

Diante do crescimento do mercado de aplicativos mobiles apoiada pelo desenvolvimento acelerado dos dispositivos móveis, proporcionou aos desenvolvedores web novas possibilidades em torno do seu uso. Objetivando explorar os avanços das aplicações web, detectou-se a necessidade de pesquisas que auxiliem no desenvolvimento de aplicativos voltados para a web bem como sua viabilidade.

III DESCRIÇÃO DO PROJETO

3.1 Tema

Na sociedade contemporânea, a tecnologia é parte essencial da vida do ser humano. Questões como conectividade, mobilidade e ubiquidade, apoiadas pelo crescimento acelerado dos *smartphones*, *tablets* e computadores, mudaram os hábitos da sociedade em um curto espaço de tempo. Imaginar a vida sem o uso, principalmente, da Internet pode parecer algo quase impossível, já que, desde a década de noventa, os indivíduos passaram a usufruir desta tecnologia de comunicação eletrônica, que transformou a maneira como se conversa, estabelecendo os relacionamentos interpessoais, realizando compras, aprendendo, adquirindo informações, etc.

Com a criação da *World Wide Web*, em 1990, por Tim Berners-Lee, foi possível levar a internet para milhões de pessoas no mundo inteiro. A evolução da web é marcada por estágios e a primeira implementação da web pode ser definida como a *web 1.0*, por ser configurada de forma estática, em um modelo de comunicação unilateral, ou seja, somente leitura. Com a *web 2.0*, em 2004, emergem os sites de relacionamento e a possibilidade da interação dos usuários com os conteúdos. Combinando inteligência humana e inovação tecnológica, como processos de cooperação que atribuem significado às informações e conteúdos disponíveis à web, chegamos a era da *web 3.0*, ou Web Semântica, tornando os resultados mais precisos e adequado ao perfil dos usuários. A *web 4.0* conecta pessoas e objetos, em qualquer lugar e a qualquer momento, com destaque para a mobilidade e a ubiquidade, proporcionadas pelos dispositivos móveis.

A onipresença cada vez maior das redes wi-fi públicas de alta velocidade (54 Mbits/s e mais altas) e o acesso à Internet com velocidade média (até alguns Mbits/s) por redes de telefonia celular 3G e 4G não apenas está possibilitando permanecer constantemente conectados enquanto se desloca, mas também permite novas aplicações específicas à localização. [...] Esse acesso sem fio em alta velocidade preparou a cena para o rápido surgimento de computadores portáteis (iPhones, Androids, iPads etc.), que possuem acesso constante e livre à Internet (KUROSE; ROSS, 2013, p.48).

Pode-se dizer que o mercado de celulares iniciou uma nova era após o lançamento do primeiro *smartphone* da *Apple*, o *iPhone*, em 2007, criando

tendências de formato e aplicações para estes tipos de aparelhos. Com esse tipo de tecnologia evoluindo rapidamente, a demanda por aplicativos cresce de forma exponencial, gerando mudanças na vida das pessoas, empresas, escolas, transportes públicos, indústrias etc. Com o crescimento das plataformas móveis, aliado à web 4.0, o *mobile-first* entrou em cena e sinalizou uma mudança na forma de analisar e desenvolver aplicativos para a web, o que permitiu novos conceitos, ideias e a junção de diversas tecnologias. O *Mobile-first* é conceituado por Tal Ater como “(...) sempre oferecer ao usuário a melhor experiência possível para o seu dispositivo” (ATER, 2017, p.80, tradução da autora). As mudanças ocorridas na plataforma web, o interesse em interligar interfaces e em agilizar o desenvolvimento de aplicações, já estão acontecendo e este evento é nomeado de *Progressive Web Apps*.

3.2 Objeto/Problema

Hoje, o *smartphone* é o principal meio de acesso à internet e isto se dá pela expansão das redes de comunicação, do investimento em tecnologia móvel e da ubiquidade. Neste contexto, aplicativos desenvolvidos com tecnologia *Progressive Web Apps* – PWA, são capazes de prover aos usuários *mobile* um ambiente confiável e rápido, com o uso de recursos tecnológicos eficientes e uma experiência sólida e fluida, proporcionada pela combinação das funcionalidades de um *app* móvel com recursos da *web*, permitindo uma navegação veloz, responsiva, imersiva, *offline* e independente de plataforma. Sendo assim, o projeto de pesquisa buscará responder a seguinte pergunta: quais os contributos da tecnologia PWA no desenvolvimento de aplicações móveis para a produção audiovisual?

3.3 Objetivo Geral

Observar os contributos de uma aplicação *Progressive Web Apps* no processo de produção audiovisual.

3.4 Objetivos Específicos

- Explorar as funcionalidades da tecnologia *Progressive Web Apps* - PWA.
- Identificar as técnicas essenciais para o desenvolvimento de uma aplicação PWA, como: design responsivo, armazenamento *offline* (*service worker*), e instalação na *home screen* (*app manifest*), durante a elaboração do protótipo.
- Estudar e Analisar as aplicações na área da produção audiovisual a fim de projetar um aplicativo voltado para esta área.
- Identificar quais as principais ferramentas utilizadas no processo de pré-produção e produção audiovisual.
- Desenvolver o protótipo de uma aplicação *progressive web apps* capaz de agregar as principais ferramentas de produção audiovisual.

IV JUSTIFICATIVA

No Brasil há cerca de cento e vinte milhões de usuários de Internet, e grande parte dos acessos é realizada por *smartphones* (CETIC.BR, 2018). O avanço na tecnologia móvel fez com que os *smartphones* obtivessem prioridade entre os usuários, pois os mesmos proporcionam troca de mensagens, consumo de conteúdo audiovisual e conversas por chamadas de voz ou vídeo. A preferência pelo *smartphone* como meio de comunicação e interatividade foi impulsionada pela utilização de aplicativos que ajudam os usuários em atividades corriqueiras do dia a dia, de forma rápida e fácil.

Os aplicativos móveis (*Apps*) são um *software* projetado para ser executado especificamente em um dispositivo móvel, como um *smartphone* ou um *tablet*. Os *apps* podem ser instalados por meio de lojas *on-line* específicas, como é o caso da *App Store*, *Play Store*, *Microsoft Store*, entre outras diversas lojas. Há uma infinidade de fornecedores que disponibilizam aplicativos para *download* nas lojas virtuais, para todo tipo e gênero de aplicação. Pelo fato do mercado de dispositivos móveis ser ramificado, há uma gama de plataformas de desenvolvimento de sistemas operacionais móveis, gerando uma variedade de aplicativos específicos para cada plataforma e que podem ser executados sob uma arquitetura específica. Entre os sistemas operacionais *mobile* mais utilizados, o que mais se destaca é o *Android*, da *Google*, que domina 89% (oitenta e nove por cento) do mercado brasileiro de *smartphones*, seguido pelo *IOS*, da *Apple*, com 7% (sete por cento) (PAIVA, 2018). Existem outras plataformas de sistemas operacionais, porém a porcentagem de utilização é tão irrisória que não serão abordadas no presente trabalho. Em resumo, isto significa que uma aplicação desenvolvida para o sistema *Android* não funciona nos sistemas da *Apple* ou do *Windows*, e vice e versa; para cada sistema operacional é necessária uma nova aplicação. Essa multiplicidade torna imprescindível a necessidade de se reescrever o mesmo aplicativo diversas vezes, codificado de maneira diferente, para que possa ser executado sob a sua arquitetura específica. Isto dificulta a tarefa de desenvolver um aplicativo que atenda a uma grande parcela de usuários e, conseqüentemente, ao mercado, pois, como cada sistema operacional tem a sua própria linguagem para desenvolvimento, a capacidade de portabilidade de código se torna baixa, aumentando o prazo e o custo

para o desenvolvimento de um único aplicativo para diferentes plataformas (FERNANDES, 2017).

Na busca por deixar os usuários de *smartphone* sempre conectados aos assuntos de seu interesse, as aplicações móveis surgiram como uma inovação para facilitar a sua rotina. Ao atender os requisitos de conectividade e eficiência, esperados nesse tipo de experiência móvel, os aplicativos se tornaram uma das melhores maneiras para se estabelecer uma conexão entre usuários e suas demandas. Entretanto, muitos aplicativos são instalados e utilizados apenas uma vez, visto que, nos *sites* de busca, não há como consultar as ferramentas neles disponíveis, pois esses são encontrados somente em suas respectivas lojas *on-line*. Nas lojas da *Apple Store* e da *Play Store* há milhões de aplicativos, contudo o número de *downloads* de novos *apps* começa a desacelerar.

Os usuários utilizam a internet para suprir as suas necessidades, como entretenimento, serviços, relacionamentos interpessoais, pesquisas educacionais, e normalmente o fazem por meio de um *smartphone*. Como muitas empresas divulgam o seu *app* com a exigência de sua instalação, vários usuários, que estavam interessados em apenas um serviço, deixam de utilizá-lo, por vários motivos, entre os quais se destacam a insuficiência de espaço de armazenamento, a limitação da conexão de internet em banda larga, além da burocracia no preenchimento de dados. Dessa forma, observa-se um número baixo de *downloads*, o que demonstra a dificuldade de instalação e manutenção de um aplicativo pelos usuários, pois grande parte dos *smartphones* é constituída por modelos simples, de baixo custo e com recursos escassos, o que faz com que somente os aplicativos funcionais e prioritários sejam mantidos.

Dentro desse contexto, esta pesquisa se justifica pelo fato dos aplicativos desenvolvidos com a tecnologia PWA suprirem as lacunas apresentadas acima.

Aplicativos *web* progressivos são desenvolvidos com recursos da *web* agregados aos recursos das aplicações nativas e, dentre as suas principais características, não necessitam de instalação por meio de loja de aplicativos, pois são *sites* tecnicamente regulares que, progressivamente e, conforme a utilização do usuário, vão adquirindo a aparência de um aplicativo nativo. Funcionam independentemente de conectividade, pois em sua configuração possuem o *service*

work, que pré-armazena o conteúdo que já foi disponibilizado pelo usuário, estando este *offline*, *online* ou em conexão não confiável, obtendo, assim, acesso ao conteúdo de modo rápido e confiável. Por meio das notificações *push* é possível engajar novamente o usuário, assim como lembrá-lo de voltar à aplicação, sensação antes possibilitada somente pelos aplicativos nativos. É possível também, caso o usuário tenha interesse, fixar um atalho na *homescreen* do dispositivo.

Sua implementação se dá por linguagens simples como HTML, CSS e *JavaScript*, portanto funcionam independente de plataforma. Suas atualizações não precisam de permissão, acontecem diretamente no navegador, passando despercebidas pelo usuário.

V REFERENCIAL TEÓRICO

5.1 A web e sua evolução

Na sociedade contemporânea, a tecnologia é parte essencial da vida do ser humano. Questões como conectividade, mobilidade e ubiquidade, apoiadas pelo crescimento acelerado dos *smartphones*, *tablets* e computadores, mudaram os hábitos da sociedade em um curto espaço de tempo. Imaginar a vida sem o uso principalmente da Internet pode parecer algo quase impossível. Segundo Kurose e Ross (2013), a internet é o maior sistema de engenharia já criado pela humanidade, com centenas de computadores conectados; bilhões de usuários que se conectam por meio de *laptops*, *tablets* e *smartphones*; e com uma série de dispositivos como sensores, *webcams*, console de jogos, e até mesmo máquinas de lavar sendo conectadas. Para Castells (2003), a criação e o desenvolvimento da internet são a história de uma aventura humana extraordinária, com a capacidade de estabelecer processos de inauguração de um novo mundo.

Pode-se dizer que a história da internet se inicia com a Arpanet, em 1969, uma rede de computadores vinculada ao Departamento de Defesa dos Estados Unidos que tinha a finalidade de mobilizar recursos de pesquisa e alcançar superioridade tecnológica. As redes de computadores se expandiram, e com as inovações tecnológicas em constante aprimoramento, surgem as redes globais de computadores, interligando países e continentes, o que, posteriormente, foi denominado de internet.

Criada por Tim Berners-Lee, em 1990, a *World Wide Web* (*www*) possibilitou compartilhar informações pela internet, com base no conceito de hipertexto, para milhões de pessoas no mundo inteiro. O que permitiu à internet abarcar o mundo todo foi o desenvolvimento da *www* (CASTELLS, 2003, p.16); a partir da sua criação, as tecnologias da web evoluíram e se popularizaram e, em poucos anos, a mesma passou a estar presente na vida das pessoas, transformando a maneira como elas conversam, estabelecem seus relacionamentos interpessoais, realizam compras, aprendem, se informam etc. A evolução das tecnologias da web tem proporcionado diversas experiências aos usuários por meio da interação das páginas, com fotos, vídeos, conteúdos interativos interligados aos aplicativos web. Pode-se dizer que a web é um universo em crescimento e, para melhor compreendê-la, abaixo são descritas as suas fases, desde a sua concepção:

- **Web 1.0:** considerada como a era somente de leitura. Configurada de forma estática, em um modelo de comunicação unilateral, onde não há interação e produção de conteúdos por parte dos usuários, restringindo-se apenas ao papel de observador.

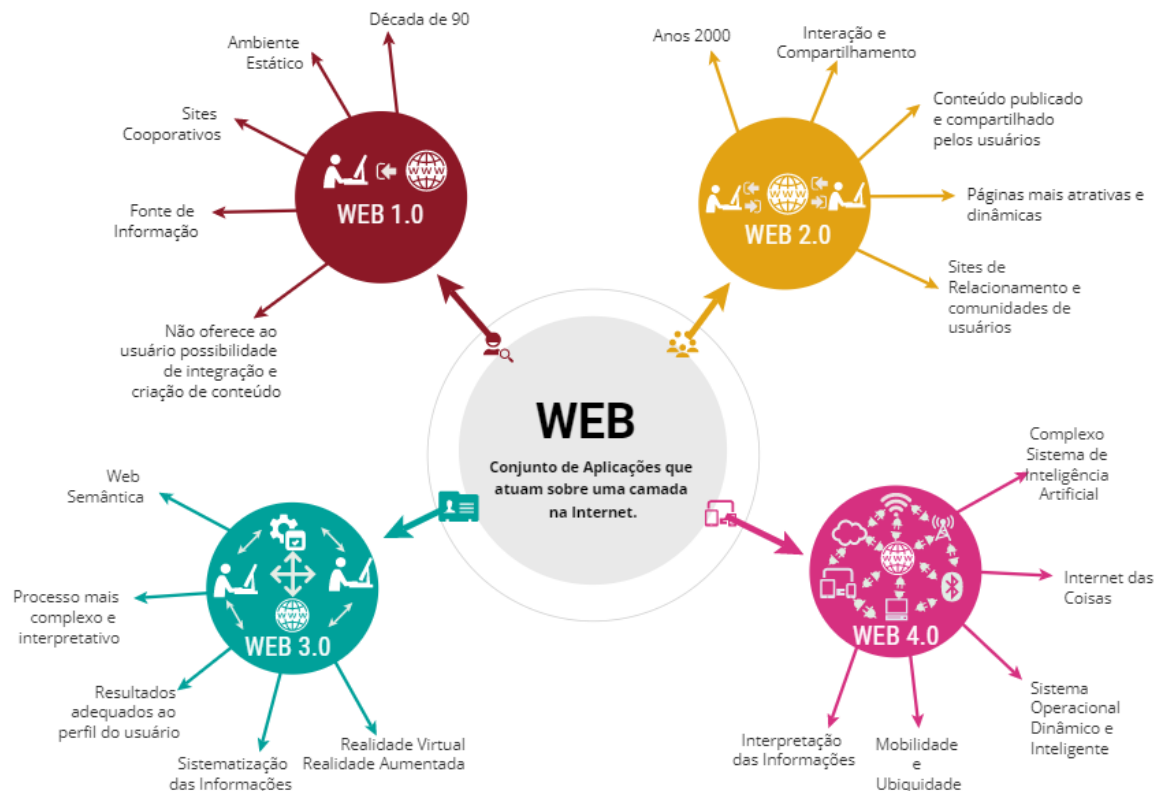
- **Web 2.0:** conhecida como a web participativa. Tendo como principal característica a possibilidade de interação dos usuários com os conteúdos, seja em sua produção, em seus comentários ou no compartilhamento. Nesta fase emergem os sites de relacionamento, *blogs*, *wikis* e mídias sociais colaborativas.

- **Web 3.0:** conhecida como a web semântica. Combina inteligência humana e inovação tecnológica, como processos de cooperação que atribuem significado às informações e conteúdos disponíveis à web, tornando os resultados mais precisos e adequados ao perfil dos usuários.

- **Web 4.0:** baseada em um complexo sistema de inteligência artificial, tende a conectar pessoas e objetos, em qualquer lugar e a qualquer momento, com destaque para a mobilidade e a ubiquidade dos dispositivos móveis. As tecnologias sem fio darão suporte à tomada de decisões e à comunicação, disponibilizando dados instantâneos quando necessário.

A Figura 1 resume as principais diferenças detectadas no processo de evolução da web.

Figura 1 - Evolução da Web



Fonte: Elaborado pela autora.

5.2 Desenvolvimento de aplicações móveis

Responsáveis por disponibilizar os conteúdos da internet, a história dos navegadores se mistura com a evolução da web. Com os avanços tecnológicos, a partir de 1993 os navegadores passaram a ter interface gráfica, suportavam novos elementos, formulários e facilidade de instalação. Possuem motores de renderização modernos (transformam o conteúdo *markup* (HTML) em algo que possa ser exibido na tela) e em constante atualização, permitindo melhores experiências do usuário. Com o uso da internet via *smartphones*, surge a necessidade de navegadores para dispositivos móveis.

É possível dizer que o mercado de celulares iniciou uma nova era após o lançamento do primeiro *smartphone* da Apple, o *iPhone*, em 2007, criando tendência de formato e aplicações para esses tipos de aparelhos. Com isto, o desenvolvimento *web* mudou consideravelmente, os navegadores passaram a renderizar as páginas com maior precisão e os sites se adaptaram ao tamanho da tela dos dispositivos

(design responsivo). Porém, não para por aí e a *web* está em evolução constante, trazendo padronização e mais possibilidades para as aplicações web em dispositivos móveis, com acesso a recursos como GPS, acelerômetro e câmera, por exemplo.

Observa-se que a evolução da web e dos navegadores foi um condutor para deixar a internet próxima dos usuários. Os usuários não são mais consumidores passivos e sim espectadores participativos desse processo, que combina dispositivos modernos e a construção colaborativa da *web*. Com o objetivo de uma *web* única e que funcione em qualquer tipo de dispositivo, começa a se deslocar do universo dos computadores para se reinventar nos dispositivos móveis.

5.3 Tipos de desenvolvimento de aplicações móveis

Ao se construir uma aplicação para dispositivo móvel, a escolha do tipo de aplicativo é fundamental para a estratégia do negócio. Nesse processo de escolha, deve-se levar em conta a plataforma de desenvolvimento e os recursos necessários. Dentre os principais tipos temos o nativo, híbrido e *web apps*, que serão apresentados nesta seção.

5.3.1 Aplicações nativas

Uma aplicação nativa é codificada especialmente para uma determinada plataforma, composta de diversas tecnologias, como: sistema operacional, linguagens de programação e IDEs³ (Ambiente de Desenvolvimento Integrado). Há diversos tipos de plataformas para celulares, tais como: *Android (Google)*, *IOS (Apple)*, *Windows Phone (Microsoft)*, dentre outras, e cada uma necessita que o aplicativo nativo seja construído na linguagem de programação específica. Há APIs⁴ específicas para cada plataforma, que oferecem acesso a todo o hardware do

³ IDE - *Integrated Development Environment* (Ambiente de Desenvolvimento Integrado): conjunto de software que consolida as ferramentas básicas necessárias para escrever e testar outro software. Fonte: ISBRASIL.

⁴ *Application Programming Interface* (Interface de Programação de Aplicativos) tem a função de conectar sistemas, softwares, aplicativos e bancos de dados que usam linguagens de programação semelhantes ou diferentes e entregar uma experiência de uso mais familiar para as pessoas. Fonte: Zenvia.

dispositivo. Com essa diferença de plataformas, desenvolver aplicativos nativos para diversos tipos de sistemas operacionais exige uma equipe de trabalho com conhecimento sobre os diversos tipos de tecnologias.

As aplicações nativas são disponibilizadas nas lojas de aplicativos de acordo com a plataforma para a qual foram desenvolvidas, como *Play Store (Android)*, *Apple Store (IOS)* e *Microsoft Store (Windows Phone)*, e são instaladas pelos usuários e acessadas a partir de um ícone fixado na tela do dispositivo móvel.

Muraroli e Girotti definem aplicações nativas como:

As aplicações nativas são aquelas desenvolvidas especificamente para serem executadas no sistema operacional do dispositivo móvel e armazenadas localmente. Desta forma, são desenvolvidas nos ambientes e linguagens compatíveis com cada SO, o que torna o projeto exclusivo para cada situação, inclusive em termos da linguagem de programação utilizada. [...] Neste tipo de aplicação é possível explorar plenamente todos os recursos dos aparelhos, como GPS, câmera, acelerômetro, catálogo de contato, agenda etc. e é permitido se ter o controle completo do dispositivo. As APIs nativas são livres para acesso pelo desenvolvedor, possibilitando acessar as funcionalidades de cada SO. [...] Com as aplicações nativas pode-se trabalhar com a notificações (Push Notifications) de forma facilitada, através das quais os usuários podem ser informados em seu aparelho sobre novidades e receber mensagens de alerta. Ainda, pode-se criar o comportamento offline, armazenando os dados e o próprio programa localmente, dispensando a conexão permanente com a internet (MURAROLI, GIROTTI, 2015, p.23).

5.3.2 Aplicações híbridas

As Aplicações híbridas são construídas com tecnologias baseadas na *web* (HTML, CSS e JavaScript), mas que são executadas dentro de um ambiente de processo nativo. As APIs são responsáveis pela comunicação entre os ambientes web e nativo. Os aplicativos desenvolvidos de modo híbrido podem ser utilizados em ambientes *mobile* e *desktop*, são executados pelo controle do navegador e não possuem barra de endereços ou qualquer outro tipo de controle do navegador.

Por utilizarem as tecnologias da web, as aplicações híbridas são mais eficazes em termos de custo e têm a vantagem adicional de reutilizarem o mesmo código para diversas plataformas. Possuem manutenção mais fácil e mais acessível em termos de tempo e custo, principalmente em casos de aplicações multiplataforma. Os aplicativos são distribuídos em lojas oficiais e o processo de

atualização para as novas versões pode ser demorado, pois há a necessidade de adaptação das novas tecnologias.

Para Murarolli e Girotti, aplicações híbridas são:

As aplicações híbridas (também chamadas de mashups), como o nome sugere, são uma combinação dos dois tipos de aplicações [...]: aplicações nativas e aplicações web. A ideia é uma página web (desenvolvida em HTML5, CSS3 e JavaScript) empacotada e que pode ser distribuída via lojas de aplicativos. [...] utiliza as linguagens da web (em geral, mais fáceis que as linguagens nativas), e multiplataforma e fornece acesso aos sensores do dispositivo através das APIs nativas. Com isso, tende-se a dispende menos tempo e, consequentemente, ter menor custo. A maior parte da infraestrutura é tipo web e é necessário o uso de ferramentas (frameworks de desenvolvimento) que fazem o empacotamento e geração dos executáveis para as plataformas desejadas (MURAROLLI, GIROTTI, 2015. p.26).

5.3.3 Aplicações web apps

Aplicações *web apps* têm como objetivo desenvolver aplicações móveis que serão executadas no browser do aparelho celular; utilizam técnicas padrão da *web*, como linguagens HTML, CSS e JavaScript, para construir aplicações com a aparência de um aplicativo nativo. Pelo fato de serem *sites* interpretados pelos navegadores dos dispositivos móveis, possuem pouco acesso aos recursos de *hardware* do dispositivo.

Por ser uma aplicação web, o sistema operacional não possui controle sobre os conteúdos e a funcionalidades do *app* e, com isto, pode ser visualizado em sistemas operacionais diferentes, isto é, em qualquer plataforma. Não necessita de loja de aplicativos, já que o seu acesso é por URL⁵ e as alterações podem ser feitas a qualquer momento, com efeito imediato aos usuários. Há também a opção de instalar o app na tela inicial do aparelho, criando um atalho para acessá-lo.

As autoras Murarolli e Girotti consideram que:

As aplicações Web (ou WebApps) consistem basicamente em uma aplicação web (hospedada em um servidor, como um site) que será acessada através do navegador (browser) dos dispositivos móveis. É um software desenvolvido para ser acessível exclusivamente pela internet. Estas aplicações se popularizaram conforme a internet se tornava acessível a mais pessoas e a banda para acesso crescia. Uma característica interessante é que se pode atualizar a aplicação no servidor e todos os

⁵Uniform Resource Locator (URL) é o Localizador Padrão de Recursos, o endereço eletrônico de um site ou a página que permite a sua localização na internet. Fonte: Neipatel.

clientes passam a usufruir das novidades, sem a necessidade de o usuário realizar instalações em sua máquina (MURAROLLI, GIROTTI, 2015. p.22).

5.4 *Progressive Web Apps*

5.4.1 Definição

Com a evolução da *web* nos últimos anos, já é possível utilizar os recursos de hardware dos dispositivos móveis. As APIs da *web* estão em constante aprimoramento, permitindo desenvolver aplicações cada vez mais complexas com o uso de linguagens simples. Todo esse avanço da *web* permite levar essas funcionalidades para a plataforma *mobile* e é aí que entram os PWAs.

O termo *Progressive Web Apps*, ou simplesmente PWA, lançado em 2015 pela *Google*, pode ser considerado como um novo passo para a evolução da *web*, principalmente para o desenvolvimento *móvil*; pode-se dizer que representam uma nova filosofia para a criação de aplicativos da *web*.

O conceito *Progressive Web Apps* foi cunhado por Frances Berriman e Alex Russel, da equipe de desenvolvedores da *Google*, em 2015, com a proposta de entregar uma *web app* com a interface de um aplicativo e a praticidade de um *site*, porém, os recursos utilizados vão muito além disso. Para a *Google Developers*, PWAs são experiências de usuários que têm o alcance da *web* e são confiáveis, rápidos e engajáveis. Podem ser executados a partir da *homescreen* dos *smartphones* e *desktops*, conseguem acessar recursos do dispositivo, oferecem suporte para notificações e podem trabalhar de modo *offline*.

Segundo a *Mozilla Developer*, os PWAs não são criados com uma única tecnologia. Eles representam um novo modelo para a criação de aplicativos da *web*, envolvendo alguns padrões específicos, APIs e outros recursos, aproveitando os recursos da *web* e dos aplicativos nativos.

Progressive web apps are a new breed of web apps that combine the benefits of a native app with the low friction of the web. Progressive web apps start off as simple websites, but as the user engages with them, they progressively acquire new powers. They transform from a website into something much more like a traditional, native app (ATER, 2017, p.11).

Ater (2017) aponta que os *Progressive Web Apps* criam confiança em seus usuários e adquirem novos poderes, conforme necessário, e, com seu modelo progressivo, substituem a natureza binária instalada/não instalada de aplicativos nativos.

O *Google* já definiu o PWA como confiável, rápido e engajável, e é fundamentado nas expectativas do usuário, suas experiências e necessidades. Por ser desenvolvido com tecnologias da *web* é hospedado em servidores remotos e exibido como aplicativo pelo navegador da *web*, utilizando recursos avançados, dependendo basicamente de HTML, CSS e *JavaScript*. Isto faz com que funcione em diversas plataformas, como Android, IOS, Windows Phone, Mac, Windows, com base em um único código. Por meio de padrões abertos, implementados diretamente no *browser*, recursos como câmeras, geolocalização, notificações *push*, utilização *offline* e realidade virtual, antes utilizados somente por aplicações nativas, já estão disponíveis.

Os *PWAs* são definidos por um conjunto de conceitos e palavras-chave, incluindo: conectividade progressiva (para qualquer usuário, independente do *browser*); responsivo (elaborado para qualquer dispositivo *desktop* ou *mobile*); *app-like* semelhante a um aplicativo – (o usuário se sente em um aplicativo nativo); atual (não há a necessidade de fazer atualizações, pois, como todos os dados estão na *web*, quando abrir o *app* a nova versão já será carregada); descobrível - *SEO Friendly*⁶ – (os mecanismos de busca conseguem encontrar o conteúdo dos aplicativos); reengajável (através de *push notifications*, o usuário pode ser constantemente engajável); instalável (podem ser adicionados à *home screen* do dispositivo móvel); e linkável (mais fácil de compartilhar conteúdo ao enviar o *link* para alguém).

Essa unificação da experiência móvel, proporcionada pelas aplicações construídas com os recursos do PWA, traz algumas vantagens para os seus usuários, como:

- Disponibilidade independente da conexão: os aplicativos da *web* progressivos podem ser carregados independentemente do estado da rede. Possuem uma camada de aplicação chamada *Service Worker* que pré-armazena os

⁶ É um *site* que respeita uma série de padrões considerados como “boas práticas” pelos buscadores e, assim, ele se torna mais “amigável” a estes mecanismos de busca. Fonte: Intermídias.

principais recursos, eliminando a dependência da rede, garantindo uma experiência instantânea e confiável aos usuários.

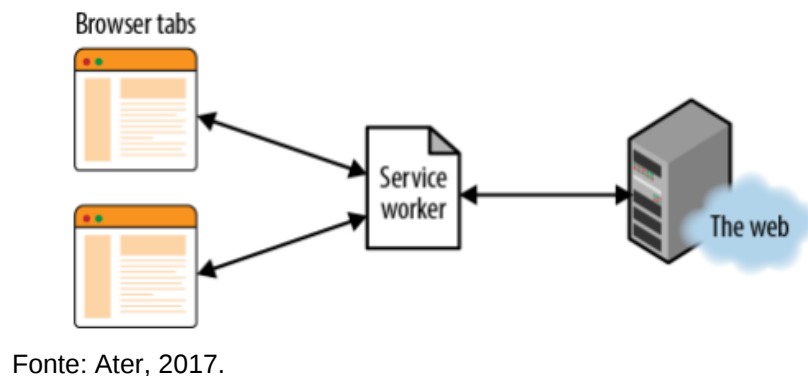
- Carregamento rápido: criando aplicações com a utilização dos *service worker* e boas técnicas de desempenho, os usuários podem carregar instantaneamente seu pwa, e ter experiências mais rápidas que aplicativos nativos.
- Engajamento: aplicativos web progressivos podem enviar notificações para os usuários. As notificações por *push* da web possibilitam envolver os usuários de forma eficaz, com conteúdo relevante e personalizado. As notificações dos *progressive web apps* despertam em seus usuários sensações completamente nativas.
- Instalável: analisando o engajamento do usuário com a aplicação, o navegador sugerirá automaticamente a possibilidade de instalação de um ícone na tela inicial do dispositivo.
- Aparência Nativa: quando instalados na tela inicial do dispositivo, podem ter uma aparência totalmente nativa e semelhante a um aplicativo. Podem ser iniciados no modo de tela cheia, sem a interface do navegador, e conseguem alterar a orientação da tela.
- Independente de Plataforma: por ser desenvolvido utilizando as tecnologias da web e interpretado pelo navegador, pode ser acessado por qualquer tipo de dispositivo, *mobile* ou *desktop*, por meio de sua URL e não precisam estar em lojas de aplicativos.

5.4.2 Service Workers

Para a *Google Developers*, o *service worker* é um *script* que o navegador executa em segundo plano, separado da página web. Inclui recursos como notificação *push* e sincronização em segundo plano. A sincronização em segundo plano é uma API da *web* que permite adiar ações até o usuário possuir conectividade estável, garantindo assim o envio de suas solicitações. Um dos principais recursos do *service worker* é a capacidade de interceptar e gerenciar solicitações de rede, o que permite experiências *offline*. Para Ater (2017), *service worker* são:

A service worker is a script that can be registered to control one or more pages of your site. Once installed, a service worker sits outside of any single browser window or tab. From this place, a service worker can listen and act on events from all pages under its control. Events such as requests for files from the web can be intercepted, modified, passed on, and returned to the page (ATER, 2017, p. 17).

Figura 2 - Funcionamento do Service Worker



Conforme observado na Figura 2, pode-se perceber que o *service worker* está no centro do *progressive web apps*, formando uma camada entre a página e a web, assegurando que qualquer ação executada pelo usuário seja entregue ao servidor. Com isto, consegue responder solicitações independentemente de conexão (*offline*, conexões lentas e perda de conectividade) e retornar o conteúdo armazenado em cache.

O Google destaca algumas características importantes de um *service worker*: é um *JavaScript Worker*⁷, portanto não consegue acessar o DOM⁸ diretamente; é um *proxy* de rede programável, o que permite controlar como as solicitações de rede da página serão gerenciadas; é encerrado quando ocioso e novamente reiniciado quando necessário; e usam muito as promessas⁹.

Um *service worker* tem um ciclo de vida totalmente separado da página da web. Para instalá-lo é necessário fazer o registro no próprio código *JavaScript* da

⁷ API para gerar scripts de segundo plano no aplicativo da web. Fonte: HTML5 Rocks.

⁸ O Modelo de Objeto de Documento - DOM - é uma interface de programação para documentos HTML, XML e SVG. Ele fornece uma representação estruturada do documento como uma árvore. Fonte: Mozilla.

⁹ É uma abstração para programação assíncrona, que recebe algum *input* (dados) e retorna uma promessa do *output* (resultado) final. Diferente de um sistema baseado em *callbacks*, que recebe um *input* e um *callback* (função), que é executado passando algum *output*. Fonte: iMasters.

página, e isso faz com que o navegador realize a instalação em segundo plano. Nesse processo, alguns recursos estáticos são armazenados em cache, indicando que o procedimento está correto. Em seguida, é realizada a etapa de ativação, o que permite o *service worker* controlar todas as páginas dentro do escopo.

O *service worker* é compatível com os navegadores *Chrome*, *Firefox* e *Opera*. Consta no *site* da *Google Developers* que o navegador *Microsoft Edge* já está implementando suporte à tecnologia, e o *Safari* possui interesse em desenvolvimento futuro¹⁰.

5.4.3 App Manifest

O *Web App Manifest* é um arquivo texto JSON simples, que fornece informações sobre o aplicativo (como nome, autor, ícone e descrição) e como ele deve se comportar quando “instalado” no dispositivo móvel. O manifesto informa detalhes para *sites* instalados na tela inicial de um dispositivo, proporcionando aos usuários um acesso mais rápido e uma experiência mais rica. Com os arquivos de manifesto, pode-se especificar se um aplicativo da *web* deve ser aberto no modo de tela cheia, possuir uma cor personalizada na barra de endereços e até definir ícones que podem ser inseridos na tela inicial dos dispositivos. O *web app manifest* faz parte de uma coleção de tecnologias da *Progressive Web Apps* que permite a sua instalação, na tela inicial de um dispositivo, sem precisar estar em uma loja de aplicativos, além de recursos como trabalhar *offline* e receber notificações *push*.

5.4.3.1 JSON

Entende-se *JSON – JavaScript Object Notation* – como um modelo para armazenamento e transmissão de informações no formato texto. Apesar de muito simples, tem sido bastante utilizado por aplicações *Web* devido à sua capacidade de estruturar informações de uma forma mais compacta do que a conseguida pelo modelo XML¹¹, tornando mais rápida a análise dessas informações. Dados representados em JSON podem ser acessados por qualquer linguagem de

¹⁰ Fonte: Google Developers.

¹¹ *Extensible Markup Language* é uma linguagem de marcação que define um conjunto de regras para codificação de documentos. Fonte: Rock Content.

programação, por meio de APIs específicas; isto explica o fato de empresas que possuem grandes volumes de dados para transmissão adotarem o JSON.

5.4.4 HTTPS

HTTPS é a sigla em inglês de *Hyper Text Transfer Protocol Secure*, que em português significa “Protocolo de transferência de hipertexto seguro”; é uma implementação do protocolo HTTP¹² sobre uma camada adicional de segurança e utiliza o protocolo *SSL/TLS*¹³.

O HTTPS é utilizado para fornecer segurança e integridade dos dados de aplicações e informações pessoais dos usuários, mesmo que essas não sejam sigilosas. Muitos recursos dos novos navegadores passaram a exigir a utilização do HTTPS, como é o caso dos *service workes*, em aplicações web progressivas.

5.4.5 HTML

O *HTML* (*HiperText Markup Language* - Linguagem de Marcação de Hipertexto), por ser uma linguagem de marcação, possibilita organizar e formatar documentos por meio da estrutura de *tags* e atributos, ou seja, marcas inseridas nas páginas que definem cabeçalhos, parágrafos, tabelas, listas e vários outros elementos. Assim como a internet e a *web* evoluíram, a linguagem HTML também passou por diversas versões, até chegar no HTML 5, que é a versão mais recente da linguagem, com novos elementos, atributos, comportamentos e tecnologias capazes de criar aplicações e *websites* diversificados e poderosos.

HTML, ou HyperText Markup Language, é uma linguagem de marcação utilizada para criar páginas acessadas a partir de um navegador. A característica principal dessas páginas é que elas utilizam hipertexto para viabilizar a navegação. Todos os elementos que compõem uma página são posicionados por meio de comandos específicos da linguagem, denominados TAGs. Uma TAG é uma palavra específica, definida em HTML, envolta por sinais de “menor que” (<) e “maior que” (>) (MILETTO, BERTAGNOLLI, 2014, p.72).

¹² É um protocolo de camada de aplicação para transmissão de documentos hipermídia, como o HTML. Foi desenvolvido para comunicação entre navegadores web e servidores web. Fonte: Mozilla.

¹³ Protocolos criptografados que garantem segurança de comunicação ao usuário assim que ele acessa o ambiente virtual, como navegação em páginas (http), e-mail (SMTP) e demais meios para a transferência de dados. Fonte: solutiresponde.

Com a publicação do HTML 5, pela W3C¹⁴, a *web* avançou enormemente, tornou-se mais multimídia, com elementos de áudio e vídeo nativos, sem a necessidade de *plug-ins*. Elementos semânticos foram criados, proporcionando significado às áreas dos *sites*, e tornando a *web* mais inteligente. Os formulários foram modificados, possibilitando que navegadores e desenvolvedores fossem utilizados de forma independente. Vários outros elementos foram atualizados ou descontinuados, pois o HTML sofre alterações significativas desde a sua criação e continuará a crescer, paralelamente ao crescimento da *web*.

As marcações do HTML 5 aumentaram a capacidade semântica do código, o que gerou poder de representação e significado. Mecanismos de busca se aperfeiçoaram, estabelecendo relações de relevância e hierarquia no conteúdo, com elementos semânticos que carregam um nível de significado mais rico.

Elementos como *article*, *section* e *nav* (artigo, seção e navegação) fazem sentido não só por para sintaxe do código, mas também para a interpretação humana, pois possuem um significado que transcende a linguagem de máquina e estabelece uma relação direta com a nossa maneira de organizar o conteúdo para a web (FLATSCHART, 2011, n.p.).

Os avanços da linguagem HTML 5 proporcionam facilidade na manipulação e modificação de elementos, de forma não intrusiva e transparente. O HTML 5 permite que *websites* ou aplicações fiquem leves e funcionais, por meio da utilização de APIs. Sua padronização nas aplicações organiza as informações na página, tornando-a mais semântica e com menos código.

5.4.6 CSS

Cascading Style Sheets – Folha de Estilo em Cascata -, ou simplesmente CSS, é uma linguagem de estilo responsável pela formatação e apresentação do conteúdo, utilizada para estilizar os elementos escritos pela linguagem HTML. O CSS separa o conteúdo da apresentação visual do *site*. Com ele é possível alterar as propriedades da fonte, cor de texto e fundo da página, espaçamento entre letras, palavras e parágrafos, entre outros, como também é possível criar variações de *layouts*, ajustar imagens, criar tabelas, além de muitas outras funcionalidades. O uso

¹⁴ É uma organização internacional que desenvolve padrões para assegurar a normalização e a evolução da *web*.

do CSS propõe flexibilidade e modularidade ao fluxo de trabalho, por permitir que conteúdo e apresentação sejam trabalhados de forma independente.

As folhas de estilo em cascata (ou CSS – Cascading Style Sheets) mudam a forma de organização das páginas. O HTML passa a ser utilizado somente como elemento para estruturar as páginas, e o CSS é utilizado na formatação da aparência das páginas. Com o CSS, é possível definir em um único local a formatação que será utilizada por cada TAG. Com isso, apenas um arquivo é alterado, sendo que a mudança é automaticamente propagada a todas as páginas que compõem o site. As folhas de estilo possibilitam criar estilos personalizados para títulos, listas, imagens, etc., além de permitirem a definição de cores, fontes, bordas, alinhamentos, entre outras características vinculadas à aparência das páginas Web (MILETTO, BERTAGNOLLI, 2014, p.80).

5.4.7 Linguagem *Javascript*

Surgiu em 1995 e é considerada uma linguagem interpretada e executada diretamente em navegadores *web*, ou melhor, no navegador do computador do usuário e, uma vez carregada, não necessita de novas requisições ao servidor Web, por isso é um tipo de linguagem *front-end* ou *client-side*. Ela permite implementar funcionalidades mais complexas às aplicações *web*, como mostrar conteúdos atualizados em tempo real, mapas interativos, animações, etc. Sempre que as informações aparecerem de forma não estática, certamente o *JavaScript* está presente.

JavaScript é uma das linguagens mais populares da Web e se caracteriza por ter tipagem dinâmica, por ser baseada em objetos, orientada a eventos (p.ex., movimentos do mouse, pressionar botão, arrastar e soltar, etc.) e realizar avaliação em tempo de execução. JavaScript é padronizada pela ECMA International (European Computer Manufacturers Association) nas especificações ECMA-2623 e é baseada em ECMA Script (MILETTO, BERTAGNOLLI, 2014, p.106).

Para o *Mozilla Developers*, *JavaScript* é uma linguagem de programação dinâmica (operações feitas em tempo de compilação podem ser feitas em tempo de execução), com muitos recursos e que, quando aplicada a um documento HTML, fornece interatividade aos *sites*, podendo criar até jogos 2D e 3D (animados), aplicativos abrangentes com banco de dados e muito mais. *JavaScript* é uma linguagem compacta e flexível, com uma diversidade de funcionalidades que incluem:

- APIs integradas em navegadores web, fornecendo funcionalidades dinâmicas ao HTML e ao CSS, coletar e manipular vídeo da webcam, gerar gráficos 3D e amostras de áudio.
- APIs de terceiros permitem incorporar funcionalidades aos sites de outros provedores de conteúdo, como Facebook, Twitter, entre outros.
- Estruturas e bibliotecas de terceiros, que podem ser aplicadas ao HTML para criar sites e aplicativos de forma mais rápida.

Por ser uma linguagem orientada a objetos, possui tipos e operadores, objetos e métodos, porém não possui classes, o que faz com que suas funcionalidades sejam realizadas por protótipos de objetos. Sua sintaxe é herdada das linguagens Java e C, e, com isto, muitas estruturas dessas linguagens se aplicam ao *JavaScript*.

A linguagem *JavaScript* não possui conceito de entrada e de saída. Foi projetada para funcionar como linguagem de script em ambientes de terceiros (hospedeiro), sendo o mais comum o navegador, e cabe a ele fornecer mecanismos para a comunicação com outros objetos.

5.4.8 Linguagem de Programação Java

Java é uma linguagem de programação orientada a objetos, criada na década de noventa pela *Sun Microsystems* e adquirida pela *Oracle*, em 2008. É considerada como uma plataforma de desenvolvimento de *software* por possuir um conjunto de APIs que facilitam o desenvolvimento e atende a diferentes necessidades de aplicações.

A plataforma Java é a base para praticamente todos os tipos de aplicações em rede por possuir um padrão de desenvolvimento e distribuição. Por essa versatilidade é multiplataforma, atendendo, assim, as necessidades dos desenvolvedores, como: desktop (Java SE), composta por APIS e bibliotecas básicas para possibilitar o desenvolvimento de aplicativos de linhas de comando; e móvel (Java ME), plataforma voltada para dispositivos móveis e embarcados, com capacidade de processamento reduzida e corporativa (Java EE), a qual fornece

especificações e APIs para *containers*, e direcionada para o desenvolvimento de soluções *web* e corporativas.

5.4.9 Vue.js

O *Vue.js* é um *framework JavaScript* de código aberto, possui uma estrutura progressiva para o desenvolvimento de *interfaces* com os usuários e aplicativos de página única (spa). Sua principal biblioteca é voltada para a camada de visualização e possui facilidade de integração com outras bibliotecas ou projetos existentes.

A arquitetura do *Vue.js* pode ser adotada de forma incremental, orientada à renderização declarativa e na composição de componentes. Para funcionalidades mais avançadas, utilizadas em aplicações complexas, possui bibliotecas e pacotes de suporte mantidos oficialmente.

5.4.10 Quasar

O Quasar é um *framework* desenvolvido sobre a estrutura do *Vue.js*, que permite ao desenvolver criar sites e aplicativos responsivos de vários tipos: SPAs (aplicativo de página única); SSR (aplicativo renderizado ao lado do servidor) (+ aquisição opcional do cliente PWA); PWAs (aplicativos da *Web* Progressivos); Aplicativos móveis e aplicativos de desktop multiplataforma.

Com a utilização do Quasar é possível desenvolver apenas uma base de código e implantá-la, simultaneamente, em um site, um aplicativo móvel ou desktop, usando uma CLI (Interface de linha de comando) de última geração, utilizando melhores práticas e componentes da Web Quasar.

5.4.11 SQL Server

O *SQL Server* é um SGDB, isso é, um Sistema Gerenciador de Banco de Dados, desenvolvido e mantido pela *Microsoft*. É um software, que tem como função o armazenamento, a atualização e a recuperação de dados computacionais, incluindo os relacionamentos, permitindo toda esta administração de forma remota (por meio de rede/internet).

O *Microsoft SQL Server*, como o próprio nome sugere, utiliza-se do padrão de linguagem de pesquisa declarativa SQL (*Structure Query Language* – Linguagem de Consulta Estruturada), linguagem universal para consultas e operações em banco de dados.

5.4.12 *Spring Boot*

O *Spring Boot* é uma ferramenta que visa facilitar o processo de configuração e publicação de aplicações que utilizem o ecossistema *Spring*¹⁵.

O *Spring Boot* fornece a maioria dos componentes baseados nos *Spring* necessários em aplicações em geral, de maneira pré-configurada, tornando possível que uma aplicação “rode” rapidamente, com o esforço mínimo de configuração e implantação.

¹⁵ O *Spring Framework* fornece um modelo abrangente de programação e configuração para aplicativos corporativos modernos baseados em Java - em qualquer tipo de plataforma de implantação. Fonte: Spring.

VI MATERIAIS E MÉTODOS

6.1 Tecnologias e Recursos utilizados

A elaboração do protótipo proposto deverá utilizar tecnologias que explorem a aplicabilidade dos conceitos fundamentais do *progressive web apps* e o desenvolvimento de aplicações móveis. Dentre elas estão: linguagens básicas de desenvolvimento web – HTML, CSS e *JavaScript*; DevTools do *Chrome* -, conjunto de ferramentas para criação e depuração de páginas *web* que permite aos desenvolvedores acesso aprofundado na parte interna do navegador e sua aplicação; a linguagem Java será utilizada no *back-end*, em conjunto com a ferramenta *Spring Boot*, responsável por facilitar o processo de configuração e de publicação das aplicações.

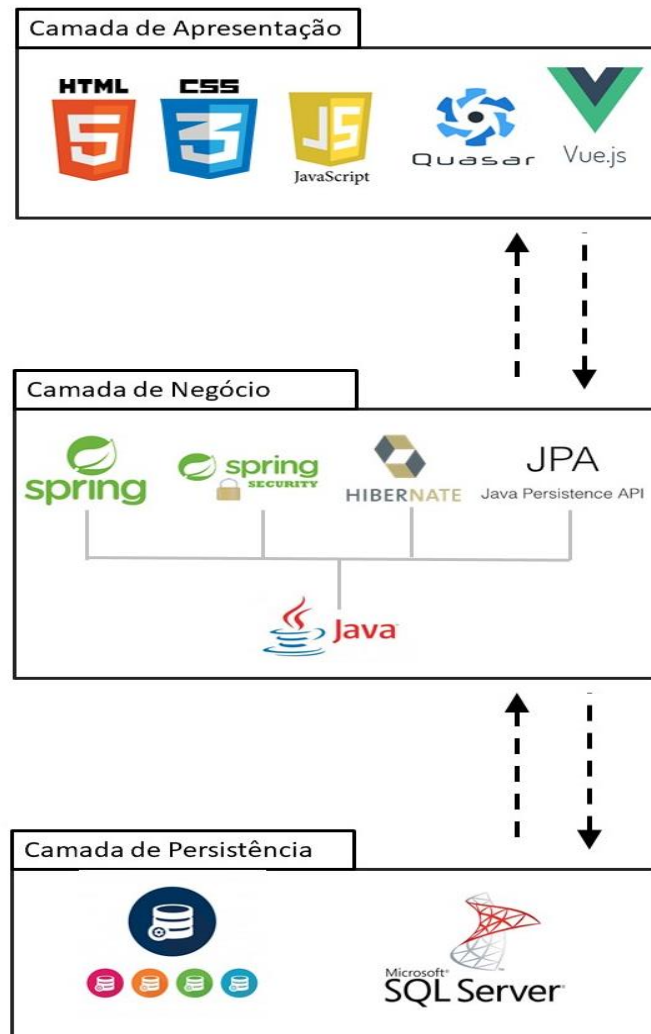
O armazenamento dos dados será via *SQL Server*, *software* de criação e gerenciamento de banco de dados. Para o *design* serão utilizados os *frameworks* *Vue.js* e *Quasar*, por possuírem uma interface de usuário de última geração para *sites* e aplicativos, além de possuir suporte para navegadores de desktop e móveis.

6.1.1 Arquitetura e Tecnologias

A arquitetura de *software* é o processo que define quais tecnologias serão empregadas e quais os conhecimentos técnicos serão necessários para o seu desenvolvimento. Nessa etapa são definidos os recursos de apoio ao desenvolvimento, como *frameworks* e IDE.

A arquitetura de *software* utilizada nesta aplicação está dividida em três camadas. Cada camada possui uma responsabilidade única no sistema, porém interligada à camada adjacente. A Figura 3 demonstra graficamente como será realizada esta interação, bem como os componentes e as tecnologias utilizadas em cada uma das camadas.

Figura 3 - Arquitetura em 3 camadas



Fonte: Elaborado pela autora.

6.1.1.1 Camada de Apresentação

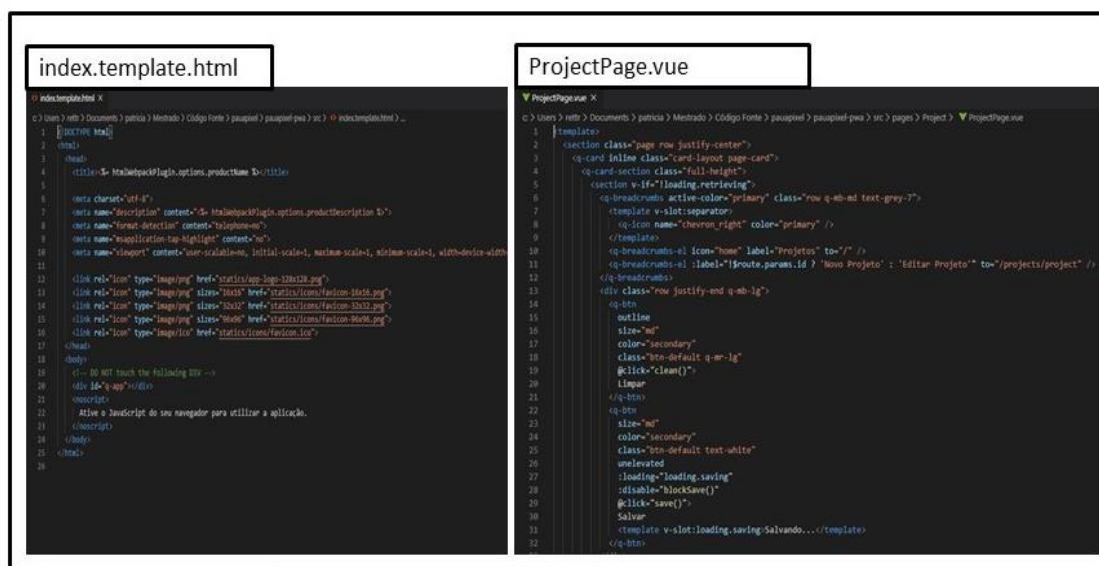
A camada de apresentação representa a Interface Gráfica do Usuário – GUI (*Graphical User Interface*). Nesta camada, o usuário interage diretamente com a interface da aplicação por meio de seu dispositivo.

Havia sido estabelecida a utilização da linguagem de *design materialize*, para auxiliar a construção desta camada. No entanto, pela necessidade de componentes mais leves, com foco em desenvolvimento em PWA, optou-se pelos *frameworks* *Vue.js* e *Quasar*, modificando, assim, a ferramenta original para desenvolvimento desta camada.

6.1.1.2 Camada de Negócio

A camada de negócio é responsável por toda a estrutura lógica da aplicação. Contém os dados e a lógica de negócio. É independente da interface com o usuário e dos dados persistentes, sendo responsável por processar as requisições vindas dos clientes, manipular os dados necessários e retorná-los. Dentro desse processo, torna a camada de apresentação mais leve, contendo apenas o código necessário para chamar as interfaces de negócio e para exibir ou gerar resultados.

Figura 4 – Trecho de código Spring Boot



Fonte:

Elaborado pela autora.

Utilizando o *framework Spring Boot*, escrito sobre a linguagem Java, foi desenvolvida uma API que possibilita o acesso da camada de apresentação por meio de requisições RESTfull, *pattern* de comunicação sob o protocolo http.

Neste processo de implementação, foram utilizadas as bibliotecas do próprio *Spring*, que possui métodos específicos para a criação de APIs, com destaque para métodos de autenticação, validação de dados e tratamento de permissões.

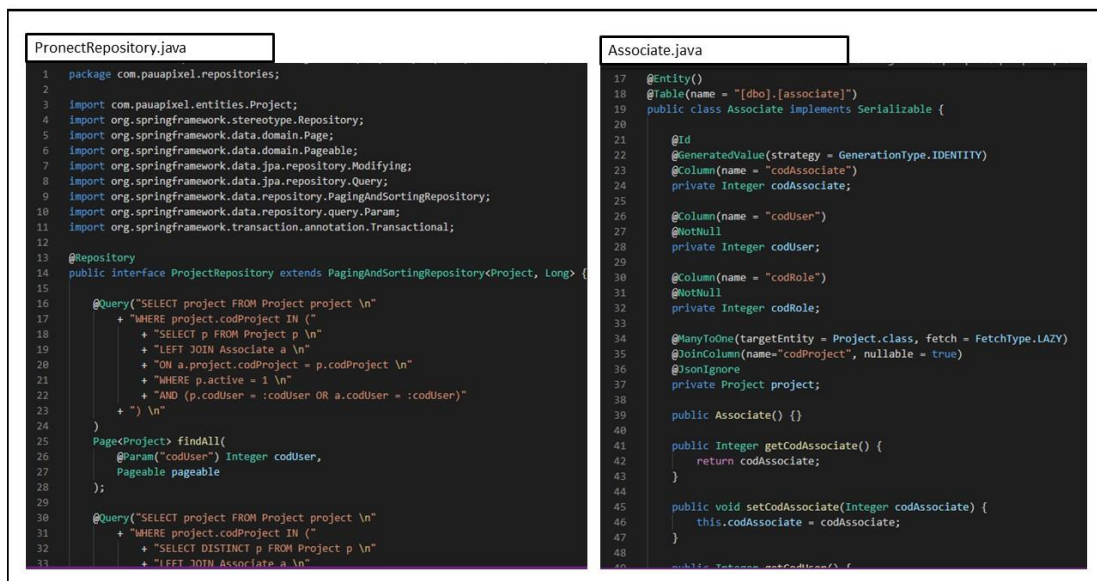
6.1.1.3 Camada de Persistência

Essa camada é responsável pela comunicação com recursos externos, como sistemas de armazenamento de dados. A camada de persistência recebe requisições da camada de negócios e seus métodos executam essas requisições

em um banco de dados. O *Spring Boot* possui o *Spring Data*, que é um modelo de programação baseado em *Spring* e consistente para acesso a dados. Esta técnica facilita o uso de tecnologias de acesso a dados, banco de dados relacionais e não relacionais.

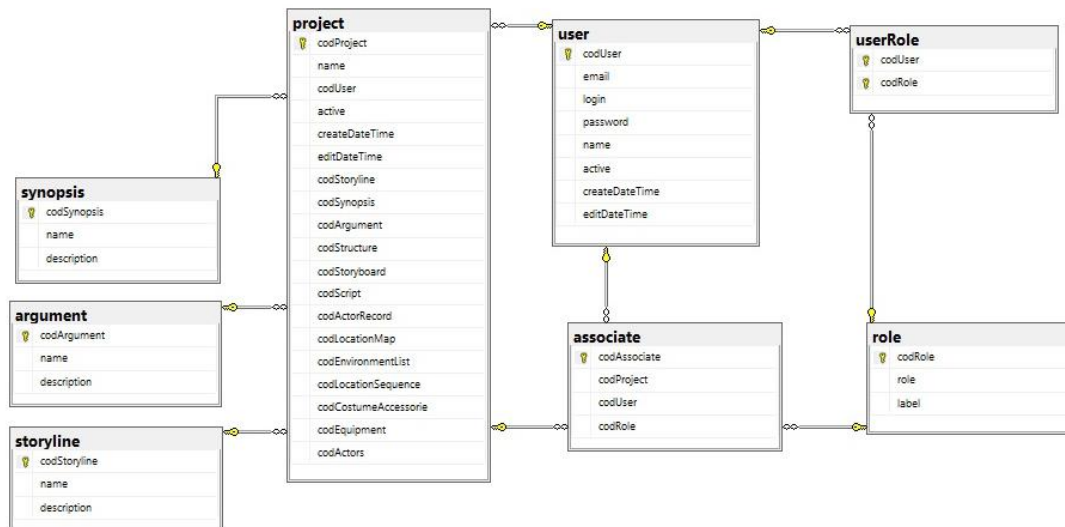
Combinando com o *Spring Data JPA*, é possível utilizar o *Spring Boot* para uma configuração rápida e fácil, disponibilizando um repositório de dados com uma API, permitindo, assim, a execução da aplicação sem a necessidade de nenhuma ferramenta externa. A Figura 5 mostra a criação de um modelo de classe e de um repositório, enquanto a Figura 6 mostra o Diagrama de Entidade e Relacionamento, utilizado para representar a estrutura dos dados.

Figura 5 - Código Repositório de Classe



Fonte: Elaborado pela autora.

Figura 6 - Diagrama de Entidade e Relacionamento



Fonte: Elaborado pela autora.

6.2 Descrição da Prova de Conceito (PoC)

O projeto apresentará, como produto final, um protótipo de *software* caracterizado como Prova de Conceito ou PoC. Uma PoC é a evidência documentada de que um software pode ser bem-sucedido, e com ele é possível identificar erros técnicos que possam interferir no funcionamento e nos resultados esperados. Uma Prova de Conceitos, realizada no desenvolvimento de um *software*, visa determinar se o software pode ser mesmo construído de fato (viabilidade técnica).

6.3 Aplicação de Software

O desenvolvimento da aplicação tem como base os fundamentos do *Progressive Web Apps*. Os conceitos da tecnologia *Progressive Web Apps* têm sido fortemente utilizados em aplicações de *e-commerce* e redes sociais, porém, com o aprimoramento constante dessa tecnologia, optou-se por vislumbrar novas possibilidades de utilização. Para isto, o desenvolvimento deste trabalho está alicerçado na elaboração de uma PoC, com o desenvolvimento de um protótipo que explore os recursos da tecnologia *progressive web apps* e abarque o processo de produção audiovisual.

Segundo Marques, a realização de um filme obedece a três fases principais, que se sucedem:

- A pré-produção ou preparação: nesta fase são realizados: a contratação dos técnicos, dos atores, dos cenários e figurinos; o roteiro, orçamento, plano de produção, escolha de locações e de estúdio, negociação com laboratório de imagem e com os fornecedores, também acontecem nesta fase;
- A produção: nesta fase é realizada a gravação propriamente dita;
- A pós-produção e/ou finalização: começa após o término da gravação, com a realização da edição.

Para que a aplicação atenda às necessidades da produção audiovisual, serão seguidas as instruções de Aída Marques:

Para se chegar a um texto tão completo, que sirva a todos as pessoas envolvidas no filme e de onde cada um possa tirar as informações que lhe serão relevantes, em geral é necessário passar por várias fases desde aquela ideia inicial até este último documento: o roteiro. Normalmente, sua elaboração passa pelas seguintes etapas: Storyline, Sinopse, Argumento, Escaleta, Roteiro, Lista de Diálogos, Decupagem técnica (MARQUES, p.23).

Considerando estes aspectos, determinou-se que os itens da produção audiovisual a serem inseridos no protótipo serão: roteiro, *storyboard*, *animatic*, roteiro decupado ou técnico, continuidade, ficha do ator, mapa de locação e de sets, lista de ambientes, locação por sequência, boletim de câmera, objetos de cena e *props*, figurino e acessórios, plano de gravação ou ordem do dia, equipamentos, lista de equipe, atores, *chat*.

Para a criação do roteiro será utilizado o padrão *Master Scenes*. Esse formato apresenta as informações de maneira rápida e de fácil interpretação, permite ainda que seja controlado todo o tempo de duração de uma história narrada. O padrão de roteiro *Master Scenes* é dividido em quatro partes: Cabeçalho de cena, Ação, Diálogos e Transição, cada qual tendo uma característica gráfica específica.

A proposta da aplicação tem por objetivo a elaboração de um protótipo que atenda às necessidades acadêmicas para a elaboração da produção audiovisual, incluindo a pré-produção e a produção.

6.3.1 Proposta de implementação do PWA *Pau a Pixel Design Movies*

Este trabalho visa o desenvolvimento de um protótipo de aplicação *web* progressiva, denominada *Pau a Pixel Design Movies*, voltado para a área de produção audiovisual, com o objetivo de disponibilizar aos usuários a criação de roteiros de filmes com base nos elementos já definidos, como *storyline*, sinopse, argumento e escaleta, a composição dos cenários, figurinos e acessórios e o compartilhamento de informações entre os envolvidos no projeto, por meio de um *chat*.

Por meio de uma pesquisa em aplicativos semelhantes e o conhecimento profissional da prof^a. Dr^a. Letícia Passos Affini, foram levantadas algumas funcionalidades e requisitos essenciais para o desenvolvimento da aplicação.

6.3.1.1 Sistemas Semelhantes

Para auxiliar no levantamento dos requisitos do sistema, fez-se necessário pesquisar e analisar sistemas que têm como foco a produção audiovisual e afins.

Final Draft: aplicativo de roteiro compatível com iPad, iPod touch e iPhone e requer iOS 13.0 ou superior. Permite escrever novos *scripts* ou editar os existentes, alternar para diferentes cenas de um *script*. Possui recursos para lembrar os nomes dos personagens e títulos de cenas. Inclui recursos de produção como numeração de cenas, omissão de cenas, páginas coloridas e bloqueadas. Este aplicativo requer pagamento para a licença de uso.

Artemis Pro: aplicativo de gravação de vídeo compatível com iPad, iPod touch e iPhone e requer iOS 11.0 ou superior. Os vídeos podem ser editados, adicionando cartões de títulos personalizados, bem como molduras para qualquer tamanho, forma ou proporção que o usuário desejar. Possui uma série interna de “visuais”, que fornecem uma maneira de pré-visualizar as ideias de classificação de

cores, possui integração com outros aplicativos do fabricante e com o Photoshop. Requer pagamento para a licença de uso.

MovieSlate 8: aplicativo de produção audiovisual compatível com iPad, iPod touch e iPhone e requer iOS 11.0 ou superior. Considerado “all-in-on digital slate”, por agregar itens como claquete, registro de gravações e bloco de notas para filmes, TV, documentários, videoclipes e entrevista. Captura/edita imagens, permitindo fazer anotações ao mesmo tempo, para posterior relatório. Aplicativo com licença de uso e sistema de versões, necessita de pagamento para maiores funcionalidades.

Clapperboard Pro – Premium Slate: aplicativo de vídeo e foto compatível com sistema Android 4.4 ou superior. Possui recursos de salvar e restaurar projetos diferentes; além disso, possui registro de captura, com exportação no modo CS para Excel ou LibreOffice Calc, tanto no modo retrato quanto paisagem. Funcionalidades como cena, foto, tomada, título e etiquetas personalizadas também estão presentes. Requer pagamento para a licença de uso.

6.3.1.2 Modelagem

Modelagem de *software* é a atividade de construir modelos que expliquem as características ou o comportamento de um software ou de um sistema de software. Na construção do *software*, os modelos podem ser usados na identificação das características e funcionalidades que o software deverá prover (análise de requisitos), e no planejamento de sua construção (WIKIPEDIA).

6.3.1.2.1 Estratégia

O objetivo da aplicação é disponibilizar aos usuários ferramentas que facilitem a criação de roteiros e todo o processo de produção audiovisual, em qualquer tipo de plataforma de *software* e tipo de dispositivo, móvel ou desktop.

O público-alvo deste produto são estudantes da área de Cinema, Rádio e TV interessados em realizar projetos práticos na área de produção audiovisual, que incluam a pré-produção e produção.

Pelo fato de a implementação ser feita com a utilização de ferramentas e softwares sem licença de uso, a aplicação será disponibilizada como freeware, isto é, sua utilização não implica pagamento de licenças de uso ou *royalties*.

6.3.1.2.2 Escopo

A partir da análise dos sistemas semelhantes e dos conhecimentos práticos da prof^a. Dr^a. Letícia Passos Affini, foram levantados os requisitos funcionais, apresentados no Quadro 1.

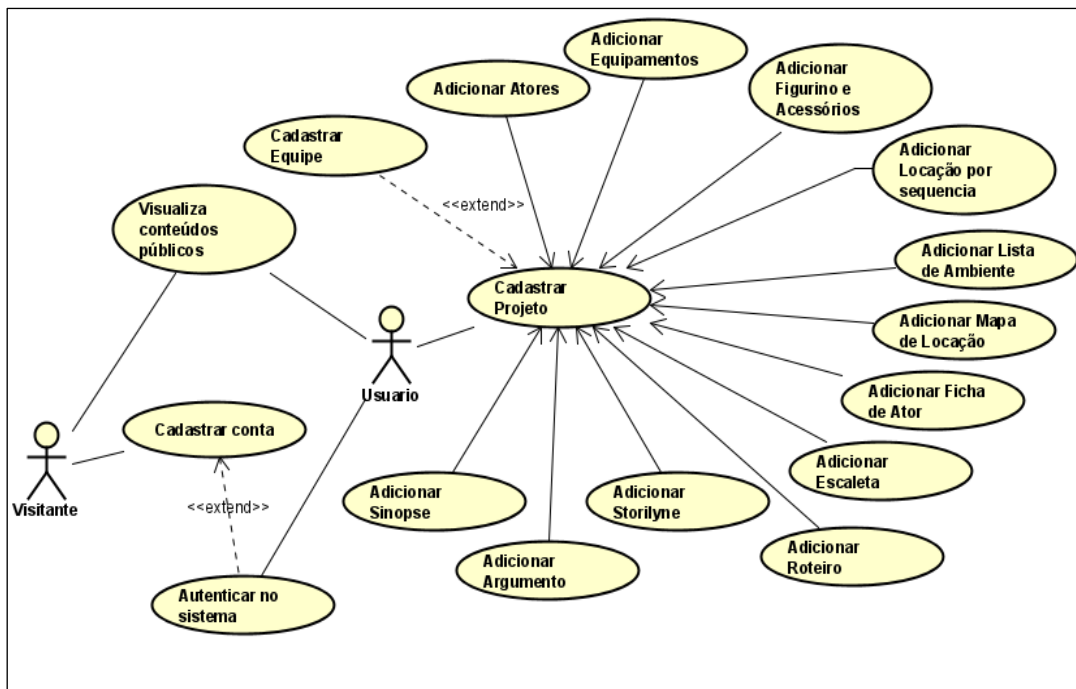
Quadro 1 – Requisitos funcionais do projeto

R1	Cadastro dos usuários para acessar o sistema.
R2	Cadastro dos projetos de produção audiovisual.
R3	Cadastrar equipes participantes dos Projetos (lista de equipe), bem como a função que cada um exercerá.
R4	Cadastro do Roteiro – incluindo <i>storyline</i> , sinopse, argumento, escaleta, roteiro decupado ou técnico, <i>animatic</i> .
R5	Permitir criação de ficha de ator, mapa de locação e de sets, lista de ambientes, objetos de cena e <i>props</i> , figurino, acessórios, plano de gravação ou ordem do dia, lista de equipamentos dentro do Diário de Projetos.
R6	Permitir acesso à câmera para gravação de vídeo e foto
R7	Criação de <i>chat</i> para lista de equipe participante de um Diário de Projeto.
R8	Permitir acesso <i>offline</i> a algumas funcionalidades da aplicação

Fonte: Elaborado pela autora.

A figura 7 demonstra o Diagrama de Caso de uso do sistema, referenciado nos requisitos funcionais elencados.

Figura 7 - Diagrama de Caso de Uso do projeto



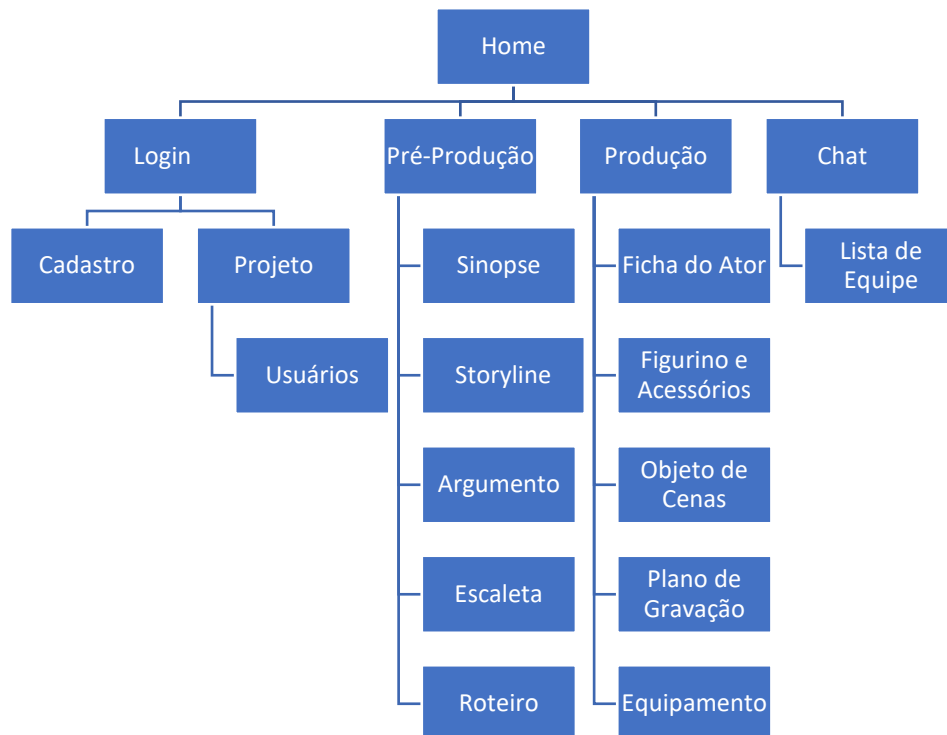
Fonte: Elaborado pela autora.

6.3.1.2.2 Arquitetura da Informação

A Arquitetura da Informação permite organizar as informações disponibilizadas pelos sites e aplicativos por meio da estruturação adequada de navegação e busca de conteúdos, que permitirão a visualização e as ações para acesso às informações, funcionalidades e ajuda aos usuários.

Pretende-se, com a Arquitetura da Informação, definir a estrutura do sistema e a forma como as informações devem ser acessadas, apresentadas e interligadas, com o objetivo de facilitar o acesso do usuário ao conteúdo, em um contexto de uso.

A partir das análises e das necessidades levantadas em um sistema de produção audiovisual, a estrutura da aplicação será apresentada por meio de um fluxograma, conforme demonstrado pela figura 8, facilitando, assim, a compreensão para as seções da aplicação/site.

Figura 8 - Fluxograma da Aplicação

Fonte: Elaborado pela autora.

6.4 Metodologia utilizada

Para que o trabalho de pesquisa consiga atingir os objetivos propostos, algumas ações devem ser planejadas para que o problema detectado encontre as suas respostas e resultados. Desse modo, o projeto de pesquisa conta essencialmente com a pesquisa exploratória, realizada por levantamento bibliográfico, aprimorando, assim, o conhecimento sobre o tema estudado e culminando com a pesquisa aplicada, ao elaborar o protótipo de uma aplicação pwa destinada à produção audiovisual. Por meio da elaboração de uma aplicação *mobile*, caracteriza-se como uma PoC o desenvolvimento de um aplicativo multiplataforma com conceitos *mobile first*, visando explorar a tecnologia *progressive web apps*.

As etapas necessárias para o desenvolvimento da aplicação foram constituídas nos seguintes passos:

1. Pesquisa exploratória e bibliográfica sobre o tema da pesquisa;
2. Construção do referencial teórico e exploração das aplicações similares já desenvolvidas;
3. Definição do objeto e problema de pesquisa;
4. Levantamento de requisitos a fim de determinar as funcionalidades inerentes à aplicação de *software*;
5. Elaboração do projeto de *software*;
6. Implementação da aplicação de *software*.

6.4.1 Etapas do processo de pesquisa e desenvolvimento

1- Pesquisa exploratória e bibliográfica sobre o tema da pesquisa:

Nesta etapa, buscou-se a exploração sobre o tema geral da pesquisa: *Progressive Web Apps*. Por meio de leitura de livros, revistas, artigos, trabalhos publicados em eventos, *sites* de conteúdo e outros, foi realizada uma exploração sobre o conteúdo relacionado ao tema. Com isso, pode-se criar um melhor embasamento teórico e auxiliar na escolha dos elementos a serem pesquisados.

2- Construção do referencial teórico e exploração sobre aplicações similares já desenvolvidas:

Para a pesquisa bibliográfica, serão utilizados os conceitos de Tal Alter (2017), Sérgio Lopes (2016), Reinaldo Ferraz (2018), Murarolli e Girotti (2015), Miletto e Bertagnolli (2014), Kurose e Ross (2013), entre outros. A inter-relação dos conceitos engloba a evolução da *web*, redes e dispositivos móveis, linguagem de programação, HTML, CSS, JavaScript, aplicativos, inovação e tecnologia, todos fundamentais para potencializar o desenvolvimento prático proposto pelo presente trabalho. Dessa maneira, os apontamentos realizados pelos autores aqui referenciados contribuem para a concepção desta pesquisa, que busca explorar os recursos de uma tecnologia inovadora e emergente, por meio do desenvolvimento de um protótipo que evidencie suas funcionalidades. Decidiu-se também pela inclusão de fontes não tradicionais de pesquisas, provenientes de *sites*, comunidades de desenvolvedores,

blogs, e as relacionadas exclusivamente com o objeto da pesquisa, o *progressive web apps*.

- 3- **Definição do objeto e problema de pesquisa:** Após o levantamento bibliográfico e a escolha dos autores, foi possível, dentro do tema geral da pesquisa, a delimitação do objeto a ser pesquisado e, conseqüentemente, quais os problemas que a pesquisa busca abordar.
- 4- **Levantamento de requisitos a fim de determinar as funcionalidades inerentes à aplicação de software:** Definidos o objeto e o problema de pesquisa, foi possível a realização do processo de levantamento de requisitos para determinar quais as funcionalidades serão executadas pela aplicação de software. Esta é a primeira etapa do ciclo de vida de desenvolvimento de software, onde são definidos as funcionalidades e o escopo do projeto.
- 5- **Elaboração do projeto de software:** A elaboração do projeto de software é uma etapa essencial para o correto desenvolvimento do produto final do *software*. Nele, pode ser definido o domínio da aplicação, a estrutura dos dados, bem como as interações necessárias para o fluxo de funcionamento. Existem inúmeros padrões e metodologias. Neste projeto, foi abordado o modelo de programação orientada a objetos e, com isso, foi utilizado o diagrama de Caso de Uso, conforme apresentado na Figura 7, para uma melhor visualização e compreensão do projeto de software.
- 6- **Implementação da aplicação do software:** Uma das fases mais críticas do processo de criação do *software* é a sua codificação. Após escolhida a tecnologia a ser utilizada, é necessário um estudo sobre as melhores abordagens e técnicas. É um processo longo, que envolve o uso de uma IDE e de recursos auxiliares para o desenvolvimento. Tais recursos podem variar de uma simples técnica de otimização de códigos até *frameworks* com bibliotecas avançadas, que agilizam todo o processo de codificação.

VII RESULTADOS OBTIDOS

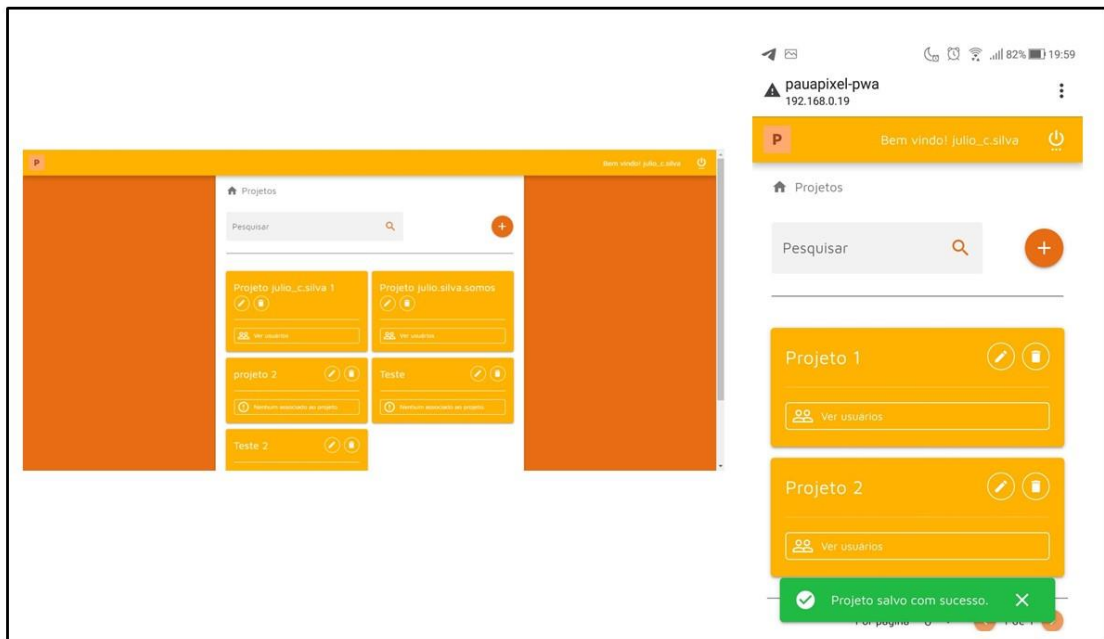
Por meio do presente estudo foi desenvolvida uma aplicação de *software* em forma de protótipo. Com isso, foi possível verificar as tecnologias existentes para o desenvolvimento de aplicações *web* progressivas.

Por ser caracterizado como protótipo, alguns requisitos do projeto ainda estão sendo finalizados. Dentro deste aspecto, as características da aplicação serão validadas com os itens finalizados, como ícone de instalação, tela *splashscreen*, modos *online* e *offline* e telas de cadastro.

Os *frameworks* *Vue.js* e *Quasar* utilizam de métodos e estilos de CSS modernos e possuem compatibilidade com a maioria dos navegadores. Apesar disso, algumas tecnologias dependem da utilização dos *Service Workers*, o que não acontece com versões antigas dos *browsers* e do navegador Safari, podendo algumas funcionalidades não ficarem disponíveis, como funcionamento *offline* e notificações. É importante ressaltar que as funcionalidades da aplicação, como cadastros e visualizações, podem ser executadas independentemente das tecnologias disponíveis no navegador. O não funcionamento de certas tecnologias apenas impacta na facilidade de realizar certas ações, e não na perda de funcionalidades.

Para o desenvolvimento das interfaces responsivas, utilizou-se das bibliotecas do *Quasar*, que já estão embutidas nas tecnologias HTML, CSS e *JavaScript*. A aplicação foi projetada utilizando a metodologia *mobile first*, iniciando-se primeiramente pelos dispositivos menores e, a partir disso, adequar a interface da tela para tamanhos maiores. A Figura 9 demonstra o comportamento responsivo da aplicação.

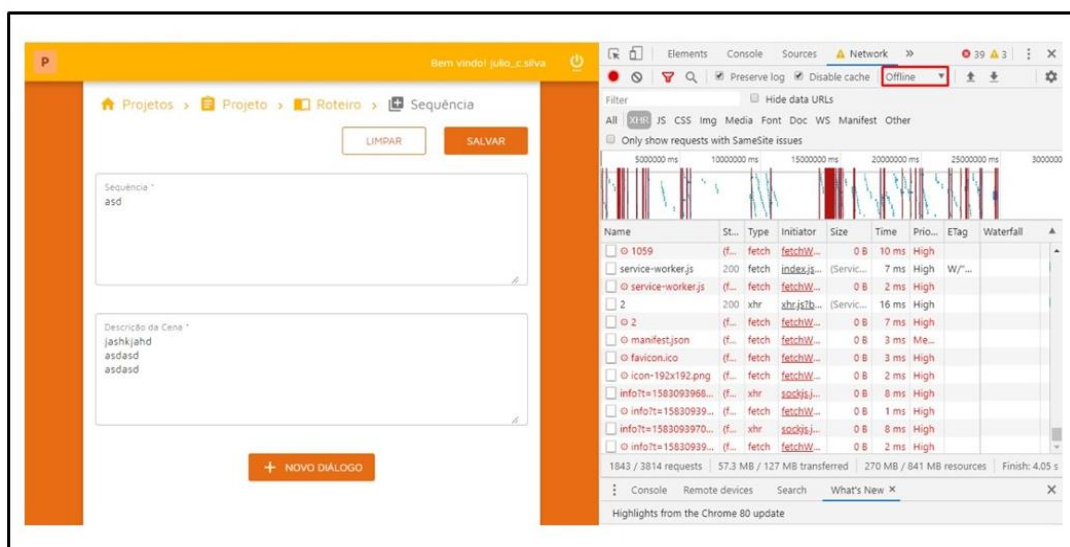
Figura 9 - Responsividade da Aplicação



Fonte: Elaborado pela autora.

A consulta às telas que já foram acessadas anteriormente fica disponível mesmo estando em modo *offline*. Funcionalidades que necessitam ser persistidas através da API ficam indisponíveis enquanto o dispositivo estiver *offline*, sendo exibida uma mensagem alertando o usuário sobre a limitação, conforme demonstrado na Figura 10.

Figura 10 - Interface no modo offline



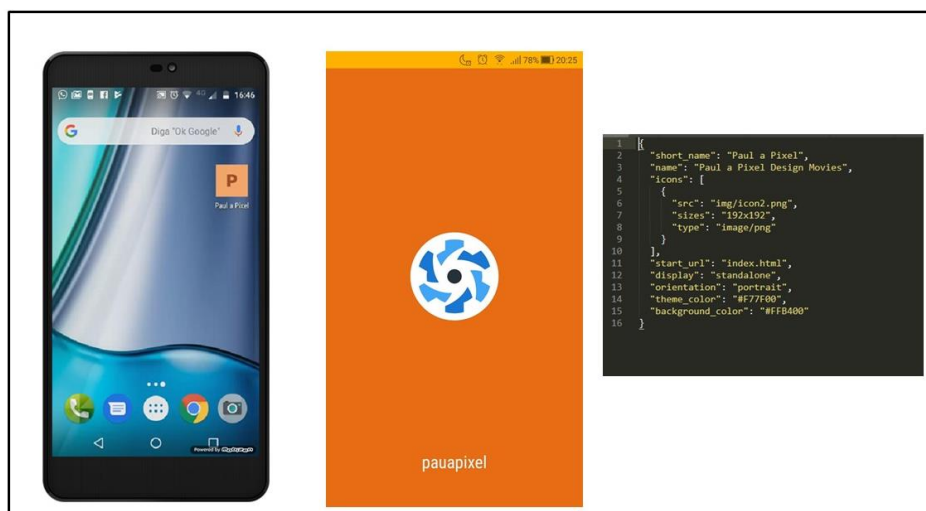
Fonte: Elaborado pela autora.

Como a aplicação está sendo desenvolvida utilizando *app shell*, a renderização dos elementos iniciais torna a aparência do aplicativo muito semelhante a uma aplicação nativa. Os efeitos de transição de tela e de interação inseridos também se assemelham aos aplicativos nativos.

Pelo fato de a aplicação ser um *website*, o *script* do *Service Worker* foi implementado para receber novas versões da mesma e replicá-las aos usuários, garantindo que, a cada nova atualização, os usuários possuam a versão atual da aplicação.

Para que o aplicativo seja instalado no dispositivo e tenha uma tela *splashscreen*, foi adicionado o arquivo *manifest.json*, conforme apresentado na Figura 11 abaixo.

Figura 11 - Ícone instalado, *splashscreen*, arquivo *manifest.json*



Fonte: Elaborado pela autora.

VIII PARCERIA INSTITUCIONAL

A pesquisadora desenvolveu a maioria das funcionalidades na UNESP, campus de Bauru - SP, o que foi de suma importância para a realização da pesquisa.

IX IMPACTOS E BENEFÍCIOS

Os impactos obtidos por este projeto baseiam-se na discussão e promoção do conhecimento sobre o tema da tecnologia *Progressive Web Apps*. O conteúdo deste trabalho buscou contribuir, de forma científica, trazendo como base uma pesquisa exploratória sobre o conceito das tecnologias progressivas e quais ferramentas podem ser encontradas atualmente no mercado. Por ser um conceito relativamente novo, não foi possível explorar todas as funcionalidades existentes, até mesmo pela bibliografia escassa; com isso, as contribuições aqui relacionadas servem como base para o conhecimento de pesquisas correlatas.

Dentro do cenário tecnológico proposto pela tecnologia *Progressive Web Apps*, este projeto apresentou mais uma opção de desenvolvimento, utilizando seus conceitos, já que atualmente tem sido explorado somente em redes sociais e *e-commerce*.

Outro fator de destaque durante a implementação do projeto é que todo o processo de desenvolvimento da aplicação explorou o uso de tecnologias sem nenhum tipo de licenciamento ou compra de credenciais para o desenvolvimento ou publicação do produto.

X DIFICULDADES

A presente pesquisa aborda um objeto em pleno desenvolvimento e em constante aprimoramento e, ao elaborar a pesquisa bibliográfica sobre o tema *Progressive Web Apps*, observou-se a escassez de referencial teórico, entre eles, livros, artigos e periódicos. Com isso, foi necessário concentrar a pesquisa em *sites*, comunidades de desenvolvedores e *blogs*.

O pouco material bibliográfico encontrado, até mesmo na internet, é em língua inglesa, o que dificulta a pesquisa, pela linguagem técnica utilizada, e a tradução, em alguns momentos, fica inconsistente.

A escolha da tecnologia a ser utilizada para o protótipo foi outro fator de dificuldade, devido à diversidade de opções. Para o desenvolvimento do projeto, além das linguagens HTML, CSS e *JavaScript*, há a necessidade de incorporar banco de dados, servidor e *framework*. Dessa forma, optou-se por plataformas que não tivessem custos, como Banco de Dados *SQL Server*, linguagem *Java*, *Vue.js* e *Quasar*, e baixa complexidade no desenvolvimento.

Contudo, a tecnologia empregada está se aperfeiçoando, novas APIs e atualização de material teórico, por parte da *Google Developers*, está acontecendo. Dessa forma, busca-se, com esta pesquisa, apresentar um protótipo que contribua para a disseminação da tecnologia estudada.

XI COMENTÁRIOS GERAIS E PERSPECTIVAS

O presente trabalho tem como objetivo observar os contributos da tecnologia *Progressive Web Apps* no processo de produção audiovisual, elencando quais elementos e técnicas seriam mais relevantes para serem implementadas, independente do dispositivo que o usuário estivesse utilizando.

Com o levantamento do referencial teórico feito sobre o *Progressive Web Apps*, pode-se sugerir o desenvolvimento de uma aplicação *mobile*, que destaque as funcionalidades dessa tecnologia, como acesso *offline* e semelhança com aplicativo.

Durante o período de desenvolvimento, observou-se a necessidade de criação de dois projetos: uma interface cliente e uma API, denominadas como *front-end* e *back-end*, respectivamente. Para o *front-end* são apresentados elementos visuais, necessários para a interação do usuário com o sistema; no *back-end*, são recebidas e processadas as requisições feitas por meio dessas interações.

No processo de desenvolvimento do *front-end* da aplicação, verificou-se a necessidade de alteração de algumas tecnologias utilizadas; optou-se por utilizar os *framework* *Vue.js* e *Quasar*, por facilitarem o processo de implementação, por oferecerem uma grande quantidade de componentes de interface reutilizáveis, pelo engajamento da comunidade de desenvolvedores e pela facilidade de acesso à documentação.

Para o *back-end* foi utilizada a linguagem Java, em conjunto com o *framework* *Spring Boot*, onde foi desenvolvida a API de acesso ao *front-end*. Tal ferramenta se mostrou adequada às necessidades do projeto pela facilidade de desenvolvimento, rapidez no acesso para a busca de dados, documentação robusta e comunidade de desenvolvedores ativa.

Apesar da aplicação ainda estar em processo final de desenvolvimento – protótipo -, pôde-se observar que os requisitos funcionais elencados foram atendidos. É possível fazer inserções de conteúdo no aplicativo com certa facilidade, tanto na versão *mobile* quanto *desktop*. Os padrões utilizados na tela facilitam o entendimento das ações que devem ser executadas dentro do processo de produção audiovisual. Pode-se destacar que a aplicação é multiplataforma e sem

cobrança de licença de uso, atendendo ao público-alvo a que se destina, bem como o fato de as informações poderem ser acessadas a qualquer momento.

Pelo fato de ser uma aplicação web progressiva, algumas funcionalidades com acesso a hardware ainda não foram implementadas, porém, após análise funcional e pertinente sobre sua utilização, serão adicionadas à aplicação.

REFERÊNCIAS

ATER, T. **Building Progressive Web Apps**. Sebastopol: O'Reilly Media Inc, 2017.

CASTELLS, M. **A Galáxia da Internet**: reflexões sobre a internet, os negócios e a sociedade; tradução Maria Luiza X. de A. Borges; revisão Paulo Vaz. Rio de Janeiro: Editora Jorge Zahar, 2003.

DATAREPORTAL. Disponível em: <https://datareportal.com/reports/digital-2019-brazil>. Acesso em: 21 de outubro de 2019.

FERRAZ, R. **Tendências da Web**. Editora Senac, São Paulo: 2018.

FLATSCHART, F. **HTML 5**: Embarque Imediato. Brasport Livros e Multimídias Ltda., São Paulo: 2011.

GERHARDT, T.; SILVEIRA, D. T. **Métodos de Pesquisa**. Porto Alegre: Editora da UFRGS, 2009.

GIL, A. C. **Como elaborar Projetos de Pesquisa**. 4. ed. São Paulo: Atlas, 2002.

GOOGLE DEVELOPERS. 2018. **Progressive Web App**. (2018). Disponível em: <<https://developers.google.com/web/progressive-web-apps/>>. Acesso em: 22 março de 2018.

GOOGLE DEVELOPERS. Service Workers: an Introduction. Disponível em: <<https://developers.google.com/web/fundamentals/primers/service-workers/>>. Acesso em: 08 de junho de 2019.

HTML5 ROCKS. **The Problem**: JavaScript Concurrency. Disponível em: <https://www.html5rocks.com/en/tutorials/workers/basics/>. Acesso em: 10 de julho de 2019.

IMASTERS. O que são promessas em JavaScript? Disponível em: <https://imasters.com.br/desenvolvimento/o-que-sao-promessas-em-javascript>. Acesso em: 10 de julho de 2019.

INTERMÍDIAS. **SEO Friendly**: Importância de um site amigável aos mecanismos de buscas. Disponível em: <http://www.intermidias.com.br/seo-otimizacao-de-sites/seo-friendly/>. Acesso em: 06 de outubro de 2018.

ISBRASIL. **Dicas para escolher o melhor IDE de programação**. Disponível em: <https://www.isbrasil.info/blog/dicas-para-escolher-o-melhor-ide-de-programacao.html>. Acesso em: 01 de julho de 2019.

KUROSE, J. F.; ROSS, K. W. **Redes de Computadores e a internet**: uma abordagem top-down. Tradução de Daniel Vieira. Revisão técnica Wagner Luiz Zucchi. 6. Ed. São Paulo: Pearson Education do Brasil, 2013.

LOPES, S. **A Web Mobile** – Programe para um mundo de muitos dispositivos. Casa do Código, 2013.

MARQUES, AÍDA. **Ideias em movimento: produzindo e realizando filmes no Brasil**. Rio de Janeiro: Rocco, 2007.

MARCONI, M.A.; LAKATOS, E. M. **Fundamentos de Metodologia Científica**. ed. – 5. São Paulo: Atlas, 2003.

MATERIALIZECSS. **About**. Disponível em: <<https://materializecss.com/>>. Acesso em: 26 de junho de 2019

MDN web docs. 2019. **Aplicativos da web progressivos. (2019)**. Disponível em: <https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps>. Acesso em: 04 de março de 2019.

MILANI, A. **MySQL**: Guia do Programador. Ed. Novatec, São Paulo: 2006.

MILETTO, E. M. BERTAGNOLLI, S.C. **Desenvolvimento de Software II**: introdução ao desenvolvimento web com HTML, CSS, JavaScript e PHP. Porto Alegre: Bookman, 2014.

MOZILA. HTTP. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/HTTP>. Acesso em: 01 de julho de 2019.

MOZILA. Modelo de Objeto de Documento (DOM). Disponível em: https://developer.mozilla.org/pt-BR/docs/DOM/Referencia_do_DOM. Acesso em: 10 de julho de 2019.

MURAROLLI, P. L. GIROTTI, M.T. **Inovações Tecnológicas nas Perspectivas Computacionais**. 1. Ed. São Paulo: Biblioteca24horas, 2015.

NEIPATEL. **URL**: O Que É e Qual A Importância Para O Seu Negócio. Disponível em: <https://neilpatel.com/br/blog/url-o-que-e/>. Acesso em: 01 de julho de 2019.

ROCK CONTENT. **Entenda o que é XML e quais as utilidades dessa linguagem de marcação**. Disponível em: <https://rockcontent.com/blog/o-que-e-xml/>, Acesso em: 05 de março de 2019.

SOLUTIRESPONDE. **SSL ou TLS**: entenda a diferença entre os protocolos. Disponível em: <https://solutiresponde.com.br/ssl-ou-tls-entenda-a-diferenca-entre-os-protocolos/>. Acesso em: 01 de julho de 2019.

SPRING. **Spring Framework**. Disponível em: <https://spring.io/projects/spring-framework>. Acesso em: 15 de dezembro de 2019.

STEINBERG, GUSTAVO. **Fim da Linha/Gustavo Steinberg e Guilher Werneck; storyboard de Fabio Moon e Gabriel Bá**. São Paulo: Imprensa Oficial do Estado de São Paulo, 2008.

TAVARES, H. L. **Introdução a Desenvolvimento de Aplicações Híbridas**. Disponível em <http://www.fatecgarca.edu.br/revista/Volume6/artigos_v6/artigo17.pdf>. Acesso em: 29 de março de 2018.

WIKIPEDIA. **Shareware**. Disponível em: <https://pt.wikipedia.org/wiki/Shareware>. Acesso em: 21 de outubro de 2019.

ZENVIA. **Saiba como APIs podem ajudar na integração e desenvolvimento da sua empresa. Entenda o que é API e veja exemplos de como usar**¹. Disponível em: <https://www.zenvia.com/blog/apis-entenda-o-que-sao-e-como-funcionam>. Acesso em: 01 de julho de 2019.