

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CÂMPUS CORNÉLIO PROCÓPIO
DIRETORIA DE GRADUAÇÃO E EDUCAÇÃO PROFISSIONAL
DEPARTAMENTO DE COMPUTAÇÃO
ENGENHARIA DE SOFTWARE

LUIZ GUILHERME DEVIDE SPIRITO

APLICAÇÃO PWA

TRABALHO DE CONCLUSÃO DE CURSO

CORNÉLIO PROCÓPIO

2018

LUIZ GUILHERME DEVIDE SPIRITO

APLICAÇÃO PWA

Trabalho de Conclusão de Curso apresentada ... da
Universidade Tecnológica Federal do Paraná como
requisito parcial para obtenção do grau de ... em

Orientador: Diogo Cezar Teixeira Batista

CORNÉLIO PROCÓPIO

2018



TERMO DE APROVAÇÃO

Aplicação PWA

por

Luiz Guilherme Devidé Spirito

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de “... em ...” e aprovado em sua forma final pelo Programa de Graduação em Engenharia de Software da Universidade Tecnológica Federal do Paraná.

Cornélio Procópio, XX/XX/XXXX

Prof. Titulação, Nome Professor Orientador
Universidade Tecnológica Federal do Paraná

Prof. Titulação, Nome professor membro da
banca
Universidade Tecnológica Federal do Paraná

Prof. Titulação, Nome professor membro da
banca
Universidade Tecnológica Federal do Paraná

RESUMO

DEVIDE SPIRITO, Luiz Guilherme. Aplicação PWA. 22 f. Trabalho de Conclusão de Curso – Engenharia de Software, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2018.

Texto do resumo (máximo de 500 palavras).

Palavras-chave: Palavra-chave 1, Palavra-chave 2, XXXXXXXX, teste de software

ABSTRACT

DEVIDE SPIRITO, Luiz Guilherme. Title in English. 22 f. Trabalho de Conclusão de Curso – Engenharia de Software, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2018.

Abstract ... (maximum of 500 words).

Keywords: Keyword 1, Keyword 2, ...

LISTA DE FIGURAS

FIGURA 8	– Exemplo de um arquivo manifest	14
----------	--	----

LISTA DE TABELAS

SUMÁRIO

1	CONTEXTO	8
1.1	DELIMITAÇÃO DO TEMA	8
1.2	PROBLEMA	10
1.3	OBJETIVOS	10
1.3.1	Objetivo Geral	10
1.3.2	Objetivos Específicos	10
1.4	ORGANIZAÇÃO DO TEXTO	11
2	FUNDAMENTAÇÃO TEÓRICA	12
2.1	PWA	12
2.2	MANIFEST	13
2.3	SERVICE WORKER	15
2.4	TRABALHOS RELACIONADOS	15
3	DESENVOLVIMENTO	16
4	RESULTADOS E DISCUSSÕES	17
5	CONCLUSÃO	18
	REFERÊNCIAS	19
	Apêndice A – NOME DO APÊNDICE A	20
	Apêndice B – NOME DO APÊNDICE B	21
	Anexo A – NOME DO ANEXO	22

1 CONTEXTO

Em 1990, Tim Berners-Lee desenvolveu o primeiro navegador web da história, o WorldWideWeb. Feito que no final desse mesmo ano se uniu a outras criações de Lee, como o servidor e as primeiras páginas web, marcando assim a história como o primeiro serviço publicado na internet.

Nos anos seguintes, com o anúncio feito pela CERN, em 1993, de que a internet seria liberada para todos, foi o estopim para que o mundo visse todo o potencial tecnológico que ali existia. Dessa maneira novas tecnologias passaram a serem criadas para facilitar todo esse processo, sendo as principais delas o CSS, em 1996, e o JavaScript em 1995.

Com a união do HTML, CSS e JavaScript temos a base dos sistemas Webs usados até hoje, porém de formas mais avançadas.

Diferente de outras linguagens, o HTML é uma linguagem de marcação. Isso quer dizer que com ela podemos marcar elementos de nosso código, para assim definir o que deve ser mostrado na página. Porém a forma que os elementos são mostrados na página é de uma maneira muito simples, então para isso temos o CSS para nos auxiliar.

O CSS é formado por um conjunto de regras formando uma folha de estilo que irá alterar a apresentação do nosso HTML. Porém temos algumas limitações usando apenas o CSS, como a falta de interação com o usuário e incapacibilidade de se fazer alterações dinâmicas.

Como o próprio nome diz, JavaScript é uma linguagem baseada em scripts, sendo utilizada pelo lado do cliente. Isso se deve ao fato de ser executada diretamente no browser, não tendo acesso ao servidor. Com ela podemos fazer alterações dinâmicas na nossa página, validar campos entre outras funcionalidades.

1.1 DELIMITAÇÃO DO TEMA

Com a facilidade do desenvolvimento o web, começamos a ver seu uso sendo implementado em qualquer tipo de objeto utilizado, fenômeno conhecido como a internet das coisas.

Ideia que apoia o fato de ser possível acessar a internet através de qualquer dispositivo. Dessa forma o principal alvo foram os telefones celulares.

Esse foco nos aparelhos móveis acabou acarretando na criação dos smartphones, aparelhos multifuncionais. E para melhorá-los ainda mais, era necessário o desenvolvimento de novas aplicações exclusivas para os mobiles.

Atualmente existem dois sistemas operacionais mais utilizados e mobiles, o Android e o iOS, exclusivo da Apple. Nos dois é possível através de uma loja virtual, fazer os downloads dos apps mais recentes. Aplicativos esses que geralmente são criados de forma nativa, isso quer dizer que são desenvolvidos na linguagem nativa do sistema operacional.

O desenvolvimento nativo é uma técnica muito utilizada, porém com grandes desvantagens em relação, por exemplo, ao desenvolvimento híbrido. Pois no nativo será necessário uma equipe para o desenvolvimento do app para apenas um sistema operacional, tornando assim o desenvolvimento muito custoso.

Por esses motivos uma nova tecnologia vem criando muita força no mercado. Esse novo tipo de desenvolvimento web é conhecido como PWA, ou Progressive Web Apps. Onde com essa tecnologia é possível desenvolver a sua aplicação web, fazendo-a funcionar tanto como um site, e como um aplicativo mobile.

Tecnologia, essa conhecida por ter algumas características obrigatórias:

1. Progressivo: Deve funcionar em qualquer navegador utilizado pelo usuário.
2. Responsivo: Deve se adequar aos formatos da tela sempre. Mesmo quando aberto em um desktop, celular ou tablet.
3. Offline: O sistema deve estar sempre disponível, mesmo que de forma parcial.
4. Instalável: Ao abrir a página web no navegador do celular, é preciso mostrar uma mensagem de instalação do aplicativo para o usuário.

Ao se cumprir todos esses requisitos, temos uma aplicação web, que pode ser vista como um site em seu navegador, ou pode ser baixada e utilizada como um aplicativo em seu dispositivo móvel.

1.2 PROBLEMA

Em 2016 haviam aproximadamente 8.05 milhões de estudantes matriculados em cursos ofertados por 2.407 instituições de ensino superior. Número que se comparado ao ano de 2006, mostra um aumento de 62.8% da taxa de matrículas (??).

Esse grande número de universitários, em relação ao pouco número de instituições nos mostra que esses ambientes precisam ser muito grande, para comportar a alta quantidade de estudantes.

Portanto, o estudante acaba por não ter informação de todos os eventos que estão acontecendo ao seu redor, pelo simples motivo de não conhecer aquele seu ambiente completamente.

Dessa forma, a aplicação visa dar a oportunidade, para o nosso principal usuário, o estudante universitário, de saber de tudo o que está acontecendo ao seu redor de uma forma bem simples. Acessando o site, ou usando o aplicativo, ele poderá saber em tempo real se há algum evento em sua volta.

Além do fato de poder descobrir novos eventos, será possível realizar a criação deles também. Atraindo mais pessoas para uma área de seu interesse. Como por exemplo, a apresentação de um TCC, onde pessoas que se interessam por aquele assunto poderão saber que ela está acontecendo. Algo que geralmente não é muito divulgado.

1.3 OBJETIVOS

1.3.1 OBJETIVO GERAL

O principal objetivo desse trabalho será o desenvolvimento de um site, usando PWA (Progressive Web App) para isso. Esse site poderá ser usado como um aplicativo em dispositivos móveis. Sendo sua principal função, a de permitir que o usuário tenha conhecimento de eventos que estão acontecendo ao seu redor. Sua primeira versão será funcional apenas no campus de Cornélio Procópio, da Universidade Tecnológica Federal do Paraná.

1.3.2 OBJETIVOS ESPECÍFICOS

1. Disponibilizar o mapa, através do Google Maps, com a localização atual do usuário, mostrando assim os eventos ao seu redor. Esse mapa, assim como todo sistema, estará em apenas uma página.
2. Ter o seu desenvolvimento em PWA irá garantir uma maior flexibilidade para o usuário.

Tendo em vista que o foco do projeto é ajudar os estudantes a terem informações de todos os eventos acontecendo no interior do campus. Portanto o acesso poderá ser feito através de um notebook, ser baixado como um aplicativo, e até ter elementos carregados de forma offline.

3. O layout da aplicação deverá ser idêntico em todas as plataformas utilizadas, se comportando sempre da mesma forma.

1.4 ORGANIZAÇÃO DO TEXTO

2 FUNDAMENTAÇÃO TEÓRICA

2.1 PWA

Dados comparados entre os anos de 2015 e 2016, mostram que o número de downloads de aplicativos móveis diminuiu em 20%. Esse fato, somado ao dado de que o usuário gasta 80% do seu tempo em apenas 5 aplicativos(Tabeless), mostra que esse é um mercado muito concorrido, e já dominado por grandes empresas.

Devido a esses números algumas empresas se sentem receosas ao desenvolver uma aplicação. Principalmente se ela for feita de forma nativa, onde será necessário o uso de pelo menos duas equipes de desenvolvimento, uma para sistemas Android, e outra para sistemas iOS.

Uma aplicação nativa traz algumas desvantagens, como um custo de desenvolvimento maior, e até mesmo uma incerteza sobre a aprovação da mesma em ser publicadas nas principais lojas de aplicativos(UseMobile).

É nesse momento que surge a opção do PWA, um modelo de desenvolvimento que combina o melhor do web, com o melhor dos aplicativos. Bastando assim apenas o desenvolvimento de tipo web, para se ter um aplicativo funcional, assim como um site.

Mas para ser considerado um PWA, este aplicativo precisa obedecer algumas regras, que são(WillianJusten):

1. Progressivo: Deve funcionar para todo e qualquer tipo de usuário, independente do navegador utilizado. Um exemplo disso é o site do NY Times, que é um PWA, e funciona em qualquer navegador.
2. Responsivo: O layout de sua página deve se comportar de forma funcional, independente do dispositivo usado. Como um celular, tablet, notebook.
3. Independente de conectividade: Uma aplicação PWA deve ter pelo menos uma funcionalidade disponível de forma offline, e aprimorada para funcionar em redes de baixa

qualidade. Um exemplo para isso é um site de notícias, onde mesmo estando offline, o usuário poderia ter acesso as notícias previamente carregas em seu último acesso.

4. Semelhante a aplicativos: Necessário que se pareça com aplicativos nativos, com interações e até emissões de notificações. Isso só será possível pelo fato de ser compilado no modelo shell de aplicativo.
5. Atual: Deve-se manter a aplicação sempre atualizada.
6. Seguro: A sua página deve ser no formato HTTPS, mantendo assim a sua segurança.
7. Descobrível: Em seus manifestos e service worker, deverá ser definido que ele pode ser identificável como um aplicativo, permitindo assim a sua instalação.
8. Reenvolvente: Deve permitir o uso de notificações push.
9. Instalável: Permite que o usuário instale a aplicação, porém sem ser preciso acessar alguma loja de aplicativos, como a Play Store ou a App Store.
10. Linkável: Pode ser acessado por uma URL, podendo assim ser acessado via navegador, e aplicação. Quando solicitada, a instalação deve ser executada de forma simples.

Apesar de todas essas regras serem necessárias, algumas delas serão essenciais no desenvolvimento da aplicação. Começando pela responsividade, pois em uma aplicação SPA(Single Page Application), em que o principal foco é um mapa, o modo em que a página é mostrada em diferentes tipos de tela faz uma grande diferença.

Outro grande foco será a parte de instalação, onde o maior o objetivo é que a aplicação depois de instalada seja extremamente parecida com uma aplicação nativa.

Para facilitar que esses objetivos sejam alcançados, existirá um arquivo chamado manifest no projeto.

2.2 MANIFEST

Um dos principais arquivos existentes em uma aplicação PWA é o manifest.json, pois através dele o seu browser irá reconhecer o site como um PWA e dando assim a opção de instalação para o usuário(Willian Justen).

O seu formato de arquivo é um JSON (JavaScript Object Notation), tipo muito usado para a transmissão de texto devido a sua simplicidade e portabilidade em relação ao JavaScript(D. Crockford).

Abaixo podemos ver um exemplo de um arquivo manifest.json, e quais suas principais regras:

Figura 8: Exemplo de um arquivo manifest

```
{
  "name": "PWA Aplicação",
  "short_name": "A-PWA",
  "theme_color": "#09ffa",
  "background_color": "#000000",
  "display": "fullscreen",
  "scope": "/",
  "start_url": "/pwa-aplicação",
  "lang": "pt-BR",
  "orientation": "any",
  "icons": [
    {
      "src": "/assets/img/icons/icone.png",
      "sizes": "512x512",
      "type": "image/png"
    }
  ]
}
```

Fonte: Autoria Propria

1. name: Esse será o nome da aplicação.
2. short-name: O nome que irá aparecer no ícone do aplicativo.
3. theme-color: Define a cor tema da aplicação, como por exemplo a cor da barra de ferramentas.
4. background-color: Cor de fundo da aplicação, item obrigatório.
5. display: Define de que modo a aplicação apresentada na tela.
6. starturl: Com ela é possível saber se a página foi aberta via app.
7. lang: Define em que língua o aplicativo será utilizada.

8. orientation: Define se a aplicação será utilizada em telas na vertical ou na horizontal.
9. icons: Define as dimensões da imagem a ser usada com ícone. Essa imagem deve ser do tipo PNG.(Google Dev.)

Após a criação do manifest, é necessário o adicionar em seu projeto. Para isso é necessário fazer a sua chamado no head do projeto, utilizando a tag link.

2.3 SERVICE WORKER

2.4 TRABALHOS RELACIONADOS

3 DESENVOLVIMENTO

4 RESULTADOS E DISCUSSÕES

5 CONCLUSÃO

Espera-se que o uso do estilo de formatação \LaTeX adequado às Normas para Elaboração de Trabalhos Acadêmicos da UTFPR (`normas-utf-tex.cls`) facilite a escrita de documentos no âmbito desta instituição e aumente a produtividade de seus autores. Para usuários iniciantes em \LaTeX , além da bibliografia especializada já citada, existe ainda uma série de recursos (??) e fontes de informação (????) disponíveis na Internet.

Recomenda-se o editor de textos Kile como ferramenta de composição de documentos em \LaTeX para usuários Linux. Para usuários Windows recomenda-se o editor \TeX nicCenter (??). O \LaTeX normalmente já faz parte da maioria das distribuições Linux, mas no sistema operacional Windows é necessário instalar o software \MiKTeX (??).

Além disso, recomenda-se o uso de um gerenciador de referências como o JabRef (??) ou Mendeley (??) para a catalogação bibliográfica em um arquivo \BIBTeX , de forma a facilitar citações através do comando `\cite{}` e outros comandos correlatos do pacote \ABNTTeX . A lista de referências deste documento foi gerada automaticamente pelo software \LaTeX + \BIBTeX a partir do arquivo `reflatex.bib`, que por sua vez foi composto com o gerenciador de referências JabRef.

O estilo de formatação \LaTeX da UTFPR e este exemplo de utilização foram elaborados por Diogo Rosa Kuiaski (diogo.kuiaski@gmail.com) e Hugo Vieira Neto (hvieir@utfpr.edu.br), com contribuições de César Vargas Benitez. Sugestões de melhorias são bem-vindas.

REFERÊNCIAS

APÊNDICE A – NOME DO APÊNDICE A

Use o comando `\appendice` e depois comandos `\chapter{}` para gerar títulos de apêndices.

APÊNDICE B – NOME DO APÊNDICE B

Use o comando `\appendice` e depois comandos `\chapter{}` para gerar títulos de apêndices.

ANEXO A – NOME DO ANEXO

Use o comando `\anexo` e depois comandos `\chapter{}` para gerar títulos de anexos.