

## PWA: Tecnologia da informação a serviço de pequenas empresas

Camila Brandina Crispin de Souza<sup>1</sup>

Fausto Gonçalves Cintra<sup>2</sup>

### Resumo

O presente trabalho tem por objetivo contextualizar de forma exemplificativa sobre o desenvolvimento de um *software* compatível com PWA (*Progressive Web Application*). As ferramentas utilizadas para o desenvolvimento deste tipo de sistema não diferem muito de um sistema *web*; no entanto, suas funcionalidades trazem uma gama de benefícios ao usuário final e ao desenvolvedor. Não se pretende disponibilizar um passo a passo para a criação de um PWA, mas oferecer uma base de conhecimentos para que um desenvolvedor iniciante tenha as primeiras noções do que esperar e do que procurar para construir sua aplicação progressiva, desde a ideia até a escolha da estratégia. Para tanto, é utilizado como exemplo um sistema de vendas consignadas.

**Palavras-chave:** *Progressive Web Application*. PWA. Controle de vendas consignadas. Análise e desenvolvimento de sistemas.

### Abstract

*The paper aims to contextualize in an explanatory way the development of a software compatible with PWA. The tools used for the development of this type of system do not differ much from a web system; nevertheless, its features bring a range of benefits to the final user and to the developer. We do not intend to provide a step-by-step approach to creating a PWA, but to offer a knowledge basis for a beginner developer to have the basic notions of what to expect and what to look for to build his/her progressive application, from the project initial idea to the strategy choice. Thus, an assigned sales system is an example.*

**Keywords:** *Progressive Web Application*. PWA. Consignment sales control. Systems analysis and development.

## 1 Introdução

O PWA (*Progressive Web Application*) é um modo de desenvolvimento que une as benesses de uma aplicação *mobile* com um sistema *web*. Por ser um modo

---

<sup>1</sup> Graduanda em Análise e Desenvolvimento de Sistemas pela Fatec Dr Thomaz Novelino - Franca/SP. Bacharela em Direito pela Universidade Federal de Goiás-UFG. Endereço eletrônico: kmila.rpsp@gmail.com.

<sup>2</sup> Docente da Fatec Franca, Mestre em Desenvolvimento Regional pelo Centro Universitário Municipal de Franca, Bacharel em Ciência da Computação com ênfase em Análise de Sistemas pela Universidade de Franca. Endereço eletrônico: fausto.cintra@fatec.sp.gov.br.

de desenvolvimento relativamente novo, o surgimento de algumas dúvidas quanto à maneira de colocá-lo em prática pode ser comum.

O presente trabalho possui uma característica exemplificativa, com o objetivo de tornar o primeiro contato com o PWA mais tangível para o desenvolvedor iniciante. O uso de um exemplo, neste caso, uma aplicação para vendas consignadas, busca deixar mais claro o método, a escolha da estratégia e perpassar por algumas dificuldades neste tipo de desenvolvimento, além de deixar claro como este tipo de aplicação pode ser benéfica também para pequenas/micro empresas que geralmente não utilizam *softwares* de usos específicos em suas rotinas diárias.

### 1.1 Histórico

Em plena Quarta Revolução Industrial, em que a combinação entre máquinas e sistemas digitais rouba a cena e domina o cenário econômico mundial, muitas inovações surgem a todo momento com o objetivo de facilitar a vida das pessoas e tornar o acesso a dados, agora essenciais, mais rápidos e seguros.

Não raro é possível encontrar pessoas em diversos ramos de indústria ou comércio que utilizam um sistema para uma finalidade específica, seja ele *desktop*, *web* ou uma aplicação *mobile*. Fato é que a evolução dos *hardwares* seguida pela evolução dos *softwares* e, principalmente, a evolução da internet, gerou uma mudança drástica no ramo da tecnologia nas últimas décadas.

Se entre os primeiros computadores criados, o Mark I em 1944 e o ENIAC, lançado em 1946, e a popularização dos computadores domésticos, na metade da década de 1990, se passaram quase cinquenta anos, a evolução da internet foi muito mais rápida. Desde a criação da ARPANET, criada para uso político e militar no final da década de 1960, até a popularização da internet no meio da década de 1990 nota-se um intervalo de tempo bem menor. (Kleina, 2011, *online*)

A evolução da internet representou um marco para o desenvolvimento tecnológico no mundo. As páginas estáticas, que usavam protocolos como o antigo HTML (*Hyper Text Markup Language*), adotaram um visual mais interativo e dinâmico através de tecnologias como o Ajax (*Asynchronous Javascript and XML*) e páginas responsivas.

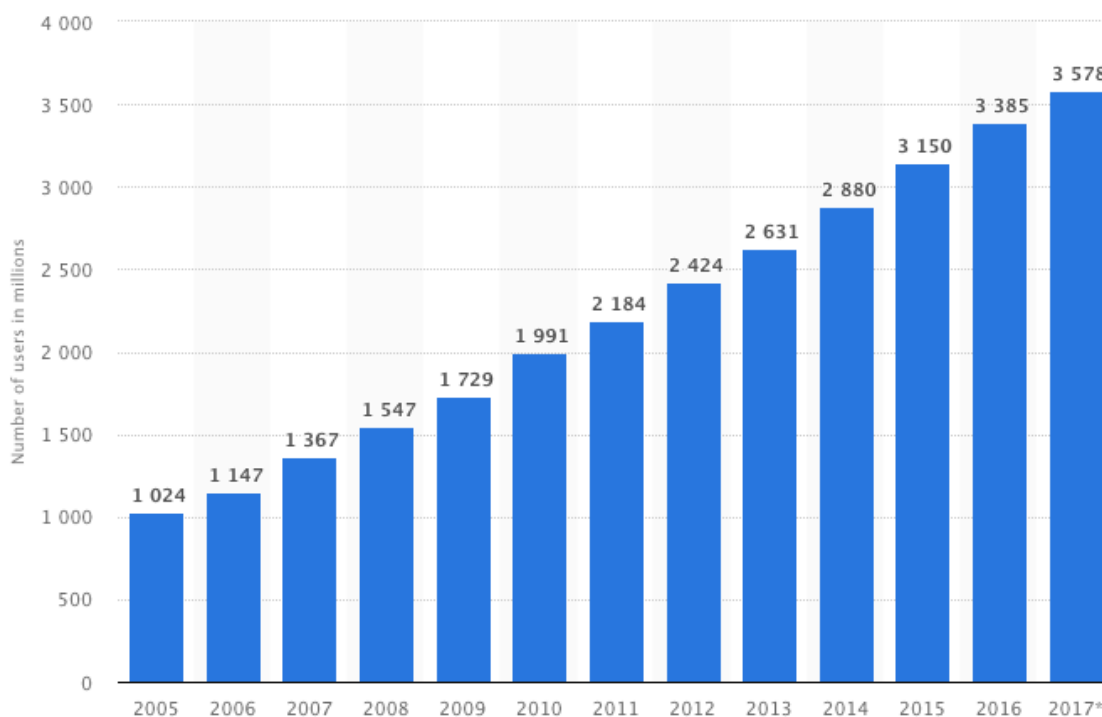
Some-se a esse cenário o lançamento do primeiro *smartphone* pela *Apple* em 2007, tecnologia que se alastrou para outras empresas do ramo e conquistou o

mundo em poucos anos. Os telefones passaram a assumir funções muito além das simples ligações e serviços de mensagens de texto (SMS), e o acesso à internet por meio dos telefones celulares, que antes era uma tarefa sofrível, se tornou mais rápido, fácil e prazeroso.

Numa análise atual, *a priori*, ambos os períodos evolutivos dos *hardwares* e *softwares* são demasiados grandes, se observados pela ótica atual, em que as informações trafegam cada vez mais rápido, e os dados são cada vez mais complexos. Não se pode negar que a sociedade tem acompanhado tais evoluções, assim como as tem incorporado no dia a dia. Os dados corroboram tal afirmação.

O Gráfico 1 ilustra o crescimento do número de pessoas que utilizaram a internet entre os anos de 2005 e 2017.

**Gráfico 1:** Número de usuários de internet.



Fonte: TRACTO (2017, *online*).

Note-se que em dez anos, de 2007 a 2017, foi registrado um aumento de mais de 260% no acesso à internet no mundo. Em estudo realizado em 2016 pelas empresas Cisco Systems e Sandvine (Micali, 2016, *online*) as companhias previram um aumento ainda mais significativo até o ano de 2020.

Segundo relatório das Nações Unidas, UNCTAD (ONU, 2017, *online*) a penetração da internet no mundo corresponde a 81% em países desenvolvidos, 40% em países em desenvolvimento e 15% em países subdesenvolvidos/pobres.

As Américas contam com 65% da população utilizando a internet. Já no Brasil, 59% da população tinha tal acesso, seja por meio de computadores, *tablets*, *notebooks*, *smartphones* entre outros.

Em pesquisa de nível nacional, o IBGE (Instituto Brasileiro de Geografia e Estatística) publicou em 2016 uma pesquisa realizada pela Pesquisa Nacional por Amostra de Domicílios Contínua (PNAD C)<sup>3</sup>, em que constatou que cerca 116 milhões de brasileiros, ou seja, em torno de 64,7% da população com idade acima de 10 anos, acessaram a internet no ano de 2016. O aparelho mais utilizado para tal acesso foi o *smartphone*, que representava cerca de 94,6% da fatia de internautas.

Levando em consideração que boa parte da faixa etária do público em questão é apta ao trabalho, não é exagero entender que o *smartphone* pessoal também tem sua utilidade no âmbito profissional. Este é um *link* importante para compreender um dos intermináveis ramos onde um sistema PWA pode ser aplicado.

O relatório da Cisco Systems (Belluno, *sd, online*) aponta tendências voltadas para o ramo da tecnologia *mobile*. Segundo a empresa, os *smartphones*, que em 2016 representavam 8% das conexões à internet, representarão cerca de 30% delas em 2020.

A evolução da internet somada às mudanças significativas nos aparelhos smartphones tem gerado uma demanda de *softwares* cada vez mais leves e rápidos, compatíveis com os gostos e usos de clientes cada vez mais exigentes.

Nesse cenário, não é de se admirar que grandes empresas tendam a pensar em soluções que reúnam as facilidades de sistemas web com aplicações *mobile*, alcançando tanto os computadores quanto os smartphones. Assim, o desenvolvimento de aplicações híbridas, eficiente em ambos os dispositivos, ganha forma.

---

<sup>3</sup> PNAD Contínua TIC 2016, ou, Pesquisa Nacional por Amostra de Domicílio Contínua (PNAD C) Pesquisa Nacional por Amostra de Domicílio Contínua (PNAD C) Tecnologias da Informação e Comunicação. (IBGE, 2016, *online*)

## 2 Referencial teórico e trabalho correlatos

Como os dados já demonstraram, atualmente é difícil encontrar pessoas que não utilizem um *smartphone* e suas aplicações, que sem dúvida, facilitam na organização do dia a dia com despertadores, calendários, agendas, aplicativos bancários, entre outros, além de proporcionar maior interação entre pessoas, próximas ou distantes, seja por uma simples ligação ou por aplicativos de redes sociais e mensagens. Atualmente, é possível fazer praticamente tudo com um *smartphone*.

No âmbito empresarial, as facilidades de um *smartphone* nem sempre alcançam os níveis gerenciais. Essa fatia pertence aos sistemas *desktop* e *web*. Ou seja, se os *smartphones* apresentam grande avanço em termos de aplicações móveis de uso majoritariamente pessoal, os sistemas *web* englobam uma boa fatia de mercado profissional.

Em um mundo ideal, a junção de ambos seria bastante benéfica, principalmente se o usuário precisasse ter controle de determinado sistema, mas não dispusesse de um computador ou um dispositivo com hardware de alto nível ou internet de boa qualidade no momento.

A proposta apresentada pelas aplicações PWA vai ao encontro com tal idealização. O motivo é simples: o PWA é uma combinação de uma aplicação *web* com uma aplicação *mobile*, utilizando o melhor de ambas.

Criado pela Google e o W3C (*World Wide Web Consortium*), assimilado pela Microsoft e, ultimamente, pela Apple, tal metodologia tem tudo para se tornar tendência nos próximos anos, e é fácil compreender o porquê.

A Figura 1 traz as dez características principais presentes em um PWA.

Figura 1: Características PWA



Fonte: GRONER (2017, *online*).

As informações foram elencadas pela Google e explanada pelo desenvolvedor Pete LePage (2017, *online*). De maneira mais clara, significam:

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

**Descobrível** - Pode ser identificado como "aplicativo" graças aos manifestos W3C e ao escopo de registro do *service worker*, que permitem que os mecanismos de pesquisa os encontrem.

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

**Responsivo** - Se adequa a qualquer formato: *desktop*, celular, *tablet* ou o que for inventado a seguir.

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

**Sempre atualizado** - Sempre atualizado graças ao processo de atualização do *service worker*.

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

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

**Independente de conexão** - Aprimorado com *service workers* para trabalhar *offline* ou em redes de baixa qualidade.

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

Do ponto de vista do desenvolvedor, o PWA possui maior abrangência pois, teoricamente, não obriga a criação de uma aplicação nativa para cada plataforma já

que as aplicações PWA deverão ser compatíveis com os principais navegadores do mercado (Google Chrome, Microsoft Edge e Apple Safari).

De maneira geral, o Quadro 1 é bastante explicativo no que tange às diferenças entre tais tipos de aplicação.

**Quadro 1:** Recursos Sistemas.

Recurso	Website	PWA	Mobile App
Usar quando estiver online	Não	Sim	Sim
Precisa baixar em apps store	Não	Não	Sim
Implementa Design Responsivo	Sim	Sim	Sim
Indexável por mecanismos de busca	Sim	Sim	Não
Notificações locais	Não	Sim	Sim
Notificações Push	Não	Sim	Sim
Precisa baixar para instalar	Não	Não	Sim
Atualizações rápidas e on-demand	Sim	Sim	Não

**Fonte:** adaptado de VIEIRA (2017, *online*).

Observa-se no Quadro 1 que o PWA, quando comparado ao sistema *web* ou aplicação *mobile* tem mais benefícios no quesito recursos. Se um sistema *web* se mantém atualizado por funcionar totalmente online, peca no quesito de ser inutilizável quando offline. Já a aplicação *mobile* carece de *download* prévio utilizando parte da memória do dispositivo, enquanto o sistema *web* utiliza de memória *cash*. Ambos os sistemas demonstram falhas. Assim, o PWA se mostra mais vantajoso porque unindo as características dos demais, apara as lacunas deixadas pelos demais *softwares* e gera maior benefício ao usuário final.

Alguns sistemas *web* oferecem uma versão *mobile* para seus clientes, mas não há justificativa para usar duas aplicações diferentes se o usuário pode utilizar a mesma aplicação tanto no *desktop/web* quanto no seu *smartphone* utilizando menos memória interna e podendo acessar sua aplicação mesmo estando *offline*.

Trazendo a teoria para a aplicação de que trata o presente trabalho, é fácil compreender os benefícios. O controle de venda consignado poderia ser realizado facilmente por um sistema *web*, *desktop* ou *mobile*, porém, uma aplicação nativa ou *desktop* carece de maior espaço para respectivo armazenamento no aparelho ou computador, enquanto um sistema *web* necessita de constante acesso à internet.

Vale salientar também que os sistemas *mobiles* e *Web* não possuem abertura para acessar recursos do *smartphone*, como, por exemplo, a câmera para registro

de foto do vendedor no momento do cadastro (*online* ou *offline*). Imagine que o controle consignado, apesar de simples, deve ser preciso, assim como qualquer sistema de controle em geral. Some-se a isso, o fato de que estar refém da internet poderia prejudicar o relacionamento da empresa com seu vendedor (picolezeiro) autônomo.

A empresa de picolés em que se foca o presente sistema conta com três espaços físicos diferentes. Uma fábrica, um ponto de distribuição e uma loja, que também distribui picolés aos vendedores ambulantes (picolezeiros).

A rotina de venda da empresa é bastante simples. Os atores principais deste processo são basicamente dois, o funcionário responsável pela entrega e recebimento do acerto junto aos vendedores autônomos, e o administrador que terá acesso ao CRUD (Acrônimo de *Create*, *Read*, *Update* e *Delete* ou Inserir, Ler, Alterar e Deletar), que define as quatro operações básicas de um banco de dados relacional, relatórios e demais mudanças que afetarão os demais usuários.

O atual passo a passo do processo de venda se inicia quando o vendedor (picolezeiro) solicita, pessoalmente, o carrinho para vendas. O atendente recebe os pedidos e adiciona ao carrinho as quantidades e tipos de produtos solicitadas. Os dados são adicionados em uma lista detalhada contendo a data da retirada, o nome do vendedor, a descrição e quantidades respectivas de cada tipo de produto. Vale destacar que cada ponto de distribuição possui anotações de vendas própria.

As vendas e seus respectivos acertos ocorrem, geralmente, no mesmo dia. Assim, no final da tarde, quando da devolução dos produtos não vendidos, há uma comparação entre as quantidades retiradas e as quantidades devolvidas, resultando no cálculo final.

O controle de contas realizado via anotação manual em caderno de vendas segue o modelo tal qual a Tabela 1:



**Tabela 1:** Controle de venda manual.

<b>Data</b>	15/02/2018				
<b>Vendedor</b>	José Marcio				
<b>Produto</b>	<b>Qtd Retirada</b>	<b>Qtd Devolvida</b>	<b>Qtd Vendida</b>	<b>Valor Un.</b>	<b>Valor Total</b>
Tradicionalis	150	45	105	R\$ 0,80	R\$ 84,00
Skimo	9	2	7	R\$ 1,85	R\$ 12,95
Brigadeiro	3	1	2	R\$ 2,50	R\$ 5,00
Fruta	2	0	2	R\$ 2,40	R\$ 4,80
Açaí	2	1	1	R\$ 2,40	R\$ 2,40
<b>Total Geral</b>					<b>R\$ 109,15</b>

Fonte: Elaborado pelos autores.

Respeitadas as devidas proporções, erros básicos no dia a dia, podem gerar grandes problemas ao final de um mês ou um ano. A utilização de um sistema se torna mais que uma opção, passa a ser uma necessidade prioritária.

O objetivo do sistema PWA é facilitar o controle das vendas consignadas por meio de *smartphone*, usando o próprio aparelho como instrumento de validação das vendas, assim como para manter base de dados cadastrais de vendedores atualizado, além de gerar relatórios para futuras análises de venda. Tais recursos são essências para este tipo de controle de vendas e dados cadastrais.

O cadastro dos vendedores, realizado de forma manual, e armazenado em ficha cadastral deixa a desejar por ser descentralizado, gerando perda de tempo na busca por dados, inseguro, pois, quando se trata de armazenamento de dados físicos, o risco de perda ou extravio de informações é muito maior, pessoas são falhas por natureza, e documentos e anotações se perdem com frequência, principalmente quando não são tratados como prioridade. Se um funcionário é realocado para outro ponto de distribuição e não tem conhecimento de todos os vendedores desse novo local, antes de distribuir os produtos, terá que entrar em contato com o administrador ou o funcionário que, normalmente, atende no local, para confirmar a identidade do vendedor, outro quesito facilmente resolvido com uma aplicação PWA, afinal mesmo sem o acesso momentâneo à internet, pode, o funcionário novato/remanejado, acessar os dados e fotos do vendedor em questão e atestar a autenticidade de sua identidade e liberá-lo para a venda, sem precisar envolver mais pessoas no processo de autenticação.

Ao cadastrar um novo produto, ou alterar o valor de venda ou comissão do vendedor, o administrador tem por obrigação repassar as novas informações para cada um dos pontos de distribuição e seus respectivos funcionários. No caso da aplicação, tal medida se tornaria desnecessária, já que uma simples alteração na aplicação PWA deixaria todos os usuários a par das alterações.

Os dois pontos de distribuição não possuem um local de centralização de dados que reúna informações de vendas diárias. O controle manual gera certo atraso no acesso às informações. Se o administrador precisa saber qual vendedor saiu em determinado dia, terá que entrar em contato via telefone ou mensagem, onde o funcionário local terá que localizar a informação no caderno de anotações e, só então, repassar a informação almejada. Note-se que, na existência de uma aplicação PWA, a informação buscada seria alcançada muito mais rapidamente, envolvendo menos pessoas e tempo, gerando economia para empresa, seja no quesito pecuniário (pelas ligações ou mensagens), seja pela otimização do tempo de ambas as pessoas envolvidas no processo de busca manual, que poderia utilizá-lo para o desenvolvimento de outras tarefas prioritárias.

Partindo da análise, mesmo que simplista, não resta dúvida de que este tipo de desenvolvimento é mais vantajoso às partes envolvidas, seja do ponto de vista de acesso à aplicação ou à sua manipulação. O PWA é mais benéfico para os usuários da empresa pois lhes assegura maior portabilidade e funcionalidades, tal qual uma aplicação *mobile*, e, mais favorável ao administrador, que terá uma economia de tempo e recursos de telefonia e internet.

### 3 Materiais e métodos ou desenvolvimento

O ramo da tecnologia, principalmente do desenvolvimento de *softwares* não conta com grande volume de publicações impressas, boa parte dos conteúdos voltados para o desenvolvimento podem ser encontrados em *sites* com tutoriais e fóruns para tirar dúvidas junto a outros desenvolvedores como o *Devmedia*, *Imasters* ou *Tabeless*, *blogs* de desenvolvedores como o *RB Tech*, sites com cursos de desenvolvimento como a *Udemy* ou *Casa do Código*, aulas práticas em canais do YouTube como o *Alura* ou *BrazilJS*, além de repositórios de desenvolvedores disponíveis no *GitHub* ou o *BitBucket*. Estas são apenas algumas fontes que serviram de base para o presente trabalho, porém, a internet é quase infinita em

possibilidades, e as informações necessárias podem estar em muitas outras páginas disponíveis virtualmente.

O desenvolvimento PWA não difere muito de um sistema *web* ou uma aplicação nativa com *Ionic*, *framework* para desenvolvimento de aplicações *mobile* híbridas, por exemplo. As ferramentas para desenvolver o sistema de controle de vendas consignadas para distribuidora de picolés são, de modo simplista, uma ferramenta de codificação, como o *Visual Studio Code*, uma ou mais linguagem de programação, de preferência, orientada a objetos, como o *JavaScript*, um navegador de internet para visualizar e testar a aparência do seu projeto, como o *Chrome*, que já possui algumas extensões bem úteis para o desenvolvimento PWA, um ou mais frameworks para auxiliar no desenvolvimento e aparência da aplicação, como o *Angular*. Também será necessário o uso de Banco de Dados para disponibilizar e registrar os arquivos gerados pela aplicação. Dependendo do objetivo do projeto, pode ser necessário o uso de *hosting* para hospedar a aplicação, armazenamento para hospedar arquivos e autenticação para autorizar o acesso ao aplicativo. Neste caso, o *Firebase*, da Google, é uma boa opção para quem está iniciando. Outra ferramenta da Google bastante útil para um iniciante é o *Web Server for Chrome*, que é um *Service Worker*, Script executado em segundo plano pelo navegador que possibilita a manipulação de recursos da aplicação mesmo offline, responsável pelas interações *offline*.

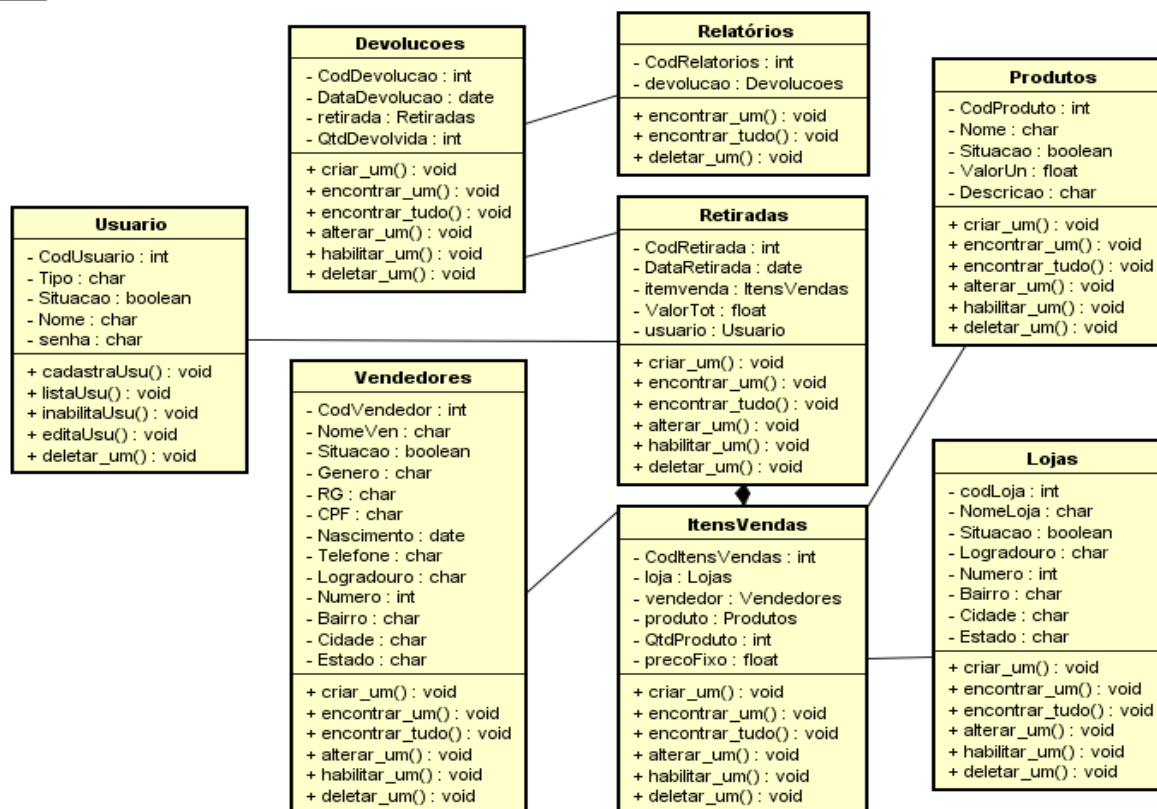
Apesar de não ter como objetivo a demonstração de um passo a passo para a construção de aplicação PWA, é importante compreender o exemplo utilizado para demonstrar as primeiras impressões de um projeto. No caso em tela, as funções da aplicação são basicamente um CRUD, onde dados dos vendedores, das lojas distribuidoras, dos produtos e dos usuários serão manipulados.

A Figura 2 (Diagrama de Classes<sup>4</sup>) traz as tarefas a serem desempenhadas dentro de cada classe do sistema.

---

<sup>4</sup> “[...]é uma representação da estrutura e relações das classes que servem de modelo para objetos.[...]”. (TYBEL, 2016, *online*).

Figura 2: Diagrama de Classe

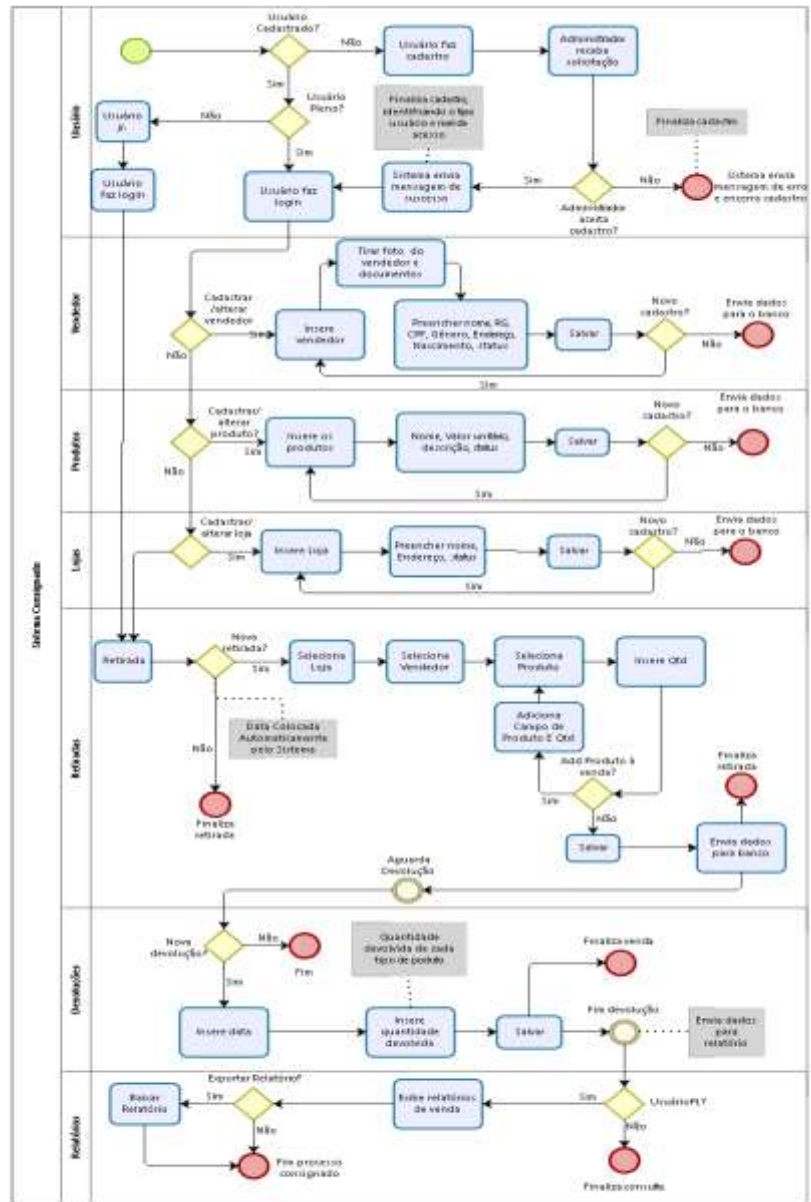


Fonte: Elaborado pelos autores.

Além do CRUD, a função “relatórios” também deverá estar disponível para o usuário, assim como o armazenamento de imagens a partir da câmera do dispositivo.

Outra ferramenta que também auxilia na ilustração do desenvolvimento do sistema é o BPMN (*Business Process Model and Notation*), uma ferramenta de metodologia de gerenciamento de processos de negócio que facilita o entendimento do usuário. A Figura 3 traz o fluxo de processos a serem realizados na manipulação do sistema.

Figura 3: BPMN Sistema Consignado



Fonte: Elaborado pelos autores.

A Figura 3 traz o passo a passo do sistema, nela é possível compreender quem são os atores do sistema, no caso, o usuário primário ou administrador, que tem a tarefa de realizar o CRUD de vendedores, lojas, produtos e relatórios, além de controlar quem serão os usuários secundários, já aos usuários secundários cabe o controle sobre as retiradas e devoluções realizadas diariamente junto aos vendedores/picolezeiros ambulantes.

Para além das ferramentas utilizadas numa aplicação *web* tal qual *JavaScript*, *HTML5*, *CSS* entre outras, a aplicação *PWA* conta ainda com (GRONER, 2017):

- O *Manifest*: Tem a função de controlar dados de um código de acordo com o dispositivo que acessa a aplicação. Tem base personalizável e deve ser importado no arquivo *HTML* para que esteja à disposição do usuário. Também permite que a aplicação seja instalável de acordo com a vontade do usuário. Ou seja, permite dar a uma aplicação *web* a cara de um *app* para ser usado em um *smartphone*. Já existem serviços de geração de *manifest* na *web* como o *App Manifest Generator* do *Firebase*.
- O *App Shell*: Base da aplicação. Quando a aplicação é acessada *offline*, o desenvolvedor pode oferecer ao usuário pelo menos o básico (estático) para que este interaja com a aplicação. Esta função tem a aparência de um aplicativo para *smartphone*.
- *Service Worker/Cache*: Trata-se de um versionador de cache. Responsável por sua capacidade de funcionar *offline*.

Considerando a importância do *Service Worker* em uma aplicação *PWA*, é importante compreender seu funcionamento. O *Service Worker* trabalha como um *proxy*, onde toda requisição é interceptada e cabe ao desenvolvedor elaborar o que acontecerá a partir de cada uma delas, sendo este o passo mais trabalhoso aos olhos de quem desenvolve. Precisa ser instalado na aplicação, e depois da instalação, ativado, além de ser importado no arquivo *HTML*.

Groner (2017) identifica ainda os seguintes passos para colocar um *Service Worker* em funcionamento:

- Registrar: é basicamente a verificação do navegador para saber se ele está habilitado a receber uma aplicação *PWA*. Atualmente, o *Chrome* da *Google* e o *Edge* da *Microsoft* já possuem tal suporte, e o *Safari* da *Apple* está caminhando no mesmo sentido.
- Ciclo de vida: Instalar, Ativar, Buscar, Verificar no *Browser*, *Cache App Shell*. Dar uma aparência inicial para a aplicação quando não estiver *online*. Neste passo, será apresentada a base e implementada a estratégia básica da aplicação.
- Estratégia: como tratar as requisições *online* ou *offline* e o que fazer em cada uma delas. A sincronização dependerá de qual estratégia foi

utilizada pelo desenvolvedor, como por exemplo, qual a requisição será carregada primeiro.

Mesmo se tratando de tema recente, muitos sistemas conhecidos já foram convertidos em PWA, e apesar de parecer bastante complexo, os passos podem ser gerados quase que completamente por ferramentas web como *Workbox* e *PWABuilder*, o que só é indicado para profissionais que já compreendem o fluxo de desenvolvimento da aplicação, ou seja, as ferramentas devem ser utilizadas para tornar mais hábil o tempo de elaboração do sistema. Tentar usar ferramentas para construir um PWA completo pode ser um erro, já que qualquer sistema que se preze deve ter uma assinatura, deve ser elaborada de acordo com as especificidades exigidas por cada cliente/usuário e com a estratégia escolhida.

Retomando o exemplo do sistema consignado, imagine-se que o acesso à câmera é essencial para que o cadastro de determinado vendedor seja realizado, esta é uma parte a ser considerada na estratégia desta aplicação. Outro ponto importante é a anotação de saída de venda e seu respectivo retorno. Portanto, este também é um ponto crucial que deve, necessariamente, estar disponível *offline* para o usuário. Já informações como os relatórios de vendas podem ser solicitados *online first* e, caso a aplicação esteja *offline*, usar *cache* para acessar os últimos dados guardados. A ideia de manter as duas opções e não apenas a versão *online* com consulta inicial se deve ao fato de que algumas informações guardadas no *cache* da aplicação devem subir antes que uma nova consulta seja realizada, sendo assim, toda vez que a aplicação transitar de *offline* para *online*, é uma boa prática dar prioridade para o *upload* dos dados em *cache*, sob risco de perder informações importantes.

Ao final do desenvolvimento, aos olhos de quem acessa sua versão via computador ou *notebook*, pouca diferença seria notada. Porém, espera-se que, quando acessado por um *smartphone* ou até mesmo um *desktop* por meio de *link*/atalho, a agilidade na velocidade das respostas, além do funcionamento *offline*, com certeza, demonstrem a eficácia do método. Isso porque as informações prioritárias estarão carregadas e disponíveis, independente de conexão, e atualizadas sempre que o acesso à internet esteja presente.

Para além da estratégia e codificação, caso exista interesse em publicar uma aplicação PWA em alguma loja oficial, deve-se considerar, antes, o teste da aplicação. Para isso, existem ferramentas interessantes. Nesse ponto, vale destacar,

que, como a Google tem se empenhado em expandir o conceito de PWA, já era de se esperar que ferramentas para sua implementação e testes estivessem inclusas no rol de extensões de seu navegador, é o caso do *Lighthouse*, “(...) uma ferramenta automatizada de código aberto que aprimora a qualidade de apps da Web.” (GOOGLE DEVELOPERS, 2017, *online*).

Esta ferramenta faz uma Verificação *Baseline Progressive Web App*, que é basicamente a certificação de que todos os requisitos exigidos, para declarar que a aplicação está dentro dos padrões PWA, estão sendo seguidos. São eles:

**Quadro 2:** Lista de Verificação *Lighthouse*

1	Site é veiculado por HTTPS
2	As páginas são responsivas em tablets e dispositivos móveis
3	Todos os prompts de instalação de aplicativos não são usados excessivamente
4	Metadados fornecidos para Adicionar à tela inicial
5	Primeira carga rápida mesmo em 3G
6	O site funciona em vários navegadores
7	As transições de página não parecem ser bloqueadas na rede
8	Cada página tem um URL
9	O conteúdo do site é indexado pelo Google
10	Os metadados do Schema.org são fornecidos quando apropriado
11	Metadados sociais são fornecidos quando apropriado
12	URLs canônicos são fornecidos quando necessário
13	Páginas usam a API de histórico
14	O conteúdo não pula quando a página é carregada
15	Retornar de uma página de detalhes mantém a posição de rolagem na página da lista anterior
16	Quando tocadas, as entradas não são obscurecidas pelo teclado na tela
17	O conteúdo é facilmente compartilhável no modo autônomo ou de tela inteira
18	O site é responsivo em todos os tamanhos de tela de telefone, tablet e desktop
19	Todos os prompts de instalação de aplicativos não são usados excessivamente
20	O prompt Adicionar à tela inicial é interceptado
21	Primeiro carregue muito rápido mesmo em 3G
22	Site usa rede de cache primeiro
23	O site informa adequadamente o usuário quando ele está off-line
24	Fornecer contexto ao usuário sobre como as notificações serão usadas
25	A interface do usuário que incentiva os usuários a ativar as notificações por push não deve ser excessivamente agressiva.
26	O site escurece a tela quando a solicitação de permissão está sendo exibida
27	As notificações por push devem ser oportunas, precisas e relevantes
28	Fornecer controles para ativar e desativar notificações
29	O usuário fez login em vários dispositivos por meio da API de gerenciamento de credenciais
30	O usuário pode pagar facilmente por meio da interface do usuário nativa da Payment Request

**Fonte:** adaptado de GOOGLE DEVELOPERS (2018, *online*).

Verifica-se no Quadro 2 que a gama de requisitos para que uma aplicação PWA esteja dentro dos padrões são bastante extensas, porém, apesar de parecer



algo extraordinário, boa parte dos sistemas *web* já seguem muitos dos requisitos solicitados, como o protocolo HTTPS, páginas responsivas, um url para cada página, uso de login por API (*Application Programming Interface* ou Interface de Programação de Aplicativos) de gerenciamento de credenciais, entre outros.

#### 4 Resultados e discussão

O PWA pode ser considerado novidade, e é sempre interessante que se elenque as falhas do método para que seja este melhorado e continue com seu processo evolutivo.

A adaptação dos navegadores mais utilizados para receber este tipo de tecnologia tem sido gradual. Dentre eles, o Chrome tem larga vantagem sobre os demais, e em sua loja de aplicações (*PlayStore*) é possível encontrar aplicações PWA com a simples pesquisa do termo. Infelizmente o mesmo não ocorre com os demais.

Destaque-se a falta de um padrão para o link PWA. Afinal, as aplicações devem possuir algum diferencial em sua identificação como forma de tornar a busca do usuário mais simples. Já que o objetivo é compartilhar e expandir o acesso a essas aplicações, nada mais justo que tornar sua descoberta mais fácil. A Microsoft já se atentou para o fato de que a busca por PWA's precisa ser simplificada, e tem investido na ideia por meio de sua ferramenta de pesquisa, o "Bing". O fato é que tal funcionalidade ainda não está em pleno vigor em sua loja de aplicações. Já a *Apple*, conhecida por ser mais segura no quesito aprovação de aplicações deixou claro no início de 2018 que tem intenção de participar de tal evolução, porém, não deixando de lado princípios basilares como segurança e a primordial qualidade das aplicações. Aqui vale ressaltar que as aplicações PWA não deixarão de rodar num *Apple*, apenas terão suas funcionalidades reduzidas caso não sigam o crivo da empresa para aprovação de aplicações para a *Apple Store*.

O ideal seria que as aplicações não precisassem de diferenciações nos códigos para acessar diferentes plataformas e navegadores. No entanto, enquanto o ideal não chega, o papel dos desenvolvedores é fazer com que as aplicações funcionem da melhor maneira possível dentro do universo PWA, independentemente do navegador.

A título de exemplificação, vale compartilhar o trabalho desenvolvido por Himango (2018, *online*), autor do aplicativo *Chavah Messianic Radio*. O desenvolvedor externou sua saga para publicar seu aplicativo, já consolidado, nas lojas da Google, Microsoft e Apple, onde a requisição basilar de seu *manifest* seria o controle sobre o player de música em segundo plano. Himango constatou, à época de sua publicação, que o *Chrome*, da Google, possuía uma abertura maior para controle de aplicações não nativas; o Edge da Microsoft, apesar de exigir uma API proprietária que, teoricamente, travaria parte das funcionalidades, deixou a comunicação com o PWA acontecer por meio de código *Javascript*, já o Safari da Apple foi o grande vilão no quesito compatibilidade. A não ser que o desenvolvedor burlasse o Safari com *plugins* o Apache Cordova, um *framework* para desenvolvimento de aplicações multiplataforma, a PWA não teria as permissões de uso no *smartphone*. Vale lembrar que se tratava de um produto pronto, já consolidado fora das *Stores*, e o intuito do desenvolvedor foi deixá-lo mais visível aos usuários, por isso a necessidade de passar pela análise da aplicação para disponibilizá-las nas lojas oficiais.

Todos os benefícios podem ser cercados de algumas dificuldades se o objetivo é disponibilizar nas lojas oficiais. Todavia, para quem tem como objetivo iniciar um contato com este tipo de desenvolvimento, as portas da rede estão abertas para novos desenvolvedores, e mudanças sempre estão presentes no ramo da tecnologia. O *call-back* de experiências como esta devem ser incentivadas para que o método evolua.

### Considerações finais

Vale salientar novamente que os casos citados têm a intenção de exemplificar algumas das complicações que podem surgir ao utilizar uma forma de desenvolvimento em gestação e implantação. Note-se que o projeto apresentado, o sistema para controle consignado, é exemplificativo e ainda se encontra em desenvolvimento, e esse é um ponto importante. Afinal, ao se deparar com essa nova forma de desenvolver, não há por que não expandir as primeiras impressões sobre as ferramentas e métodos para outros interessados na área.

Para o desenvolvedor iniciante, é importante ter uma base para iniciar os estudos de viabilidade de um sistema ou aplicação, neste sentido o presente

trabalho deve ser um início, não um fim em si mesmo. Afinal, depois de ter noção sobre o que é e ter um exemplo de como pode ser aplicada, é necessário o aprofundamento das ferramentas e nas estratégias a serem seguidas para finalizar uma aplicação viável.

O presente trabalho apresentou como exemplo o caso de uma empresa real e suas dificuldades habituais, elencando problemas e postulando possíveis soluções. Apesar de não apresentar um produto final, é possível compreender como pequenas empresas podem se cercar de tecnologia para dirimir dificuldades comuns, utilizando ferramentas/*softwares* que grandes empresas já têm feito uso, mas que não estão, necessariamente, fora do alcance de empresas menores. Esta é uma das vantagens das evoluções tecnológicas, a acessibilidade.

Por fim, é válido concluir que a evolução acontece, lenta ou rapidamente, e caso a tendência se torne realidade, é possível que uma aplicação PWA se torne tão comum quanto a internet, e se assim o for, é provável que empresas como a *Apple*, por exemplo, talvez reveja suas políticas visando a manutenibilidade de seus fiéis clientes sob pena de perde-los.

Quanto às inovações, esta é uma constante na história da humanidade, mesmo os períodos de estagnação serviram para que alguma evolução acontecesse, a terceira e quarta revolução industrial somente diminuíram o intervalo dessas inovações. O futuro próximo, apesar de uma incógnita, deve apresentar ainda mais desenvolvimento tecnológico. Onde tal progresso resultará é impossível saber, porém, em um mundo onde a Inteligência Artificial e a Internet das Coisas são aceitas como parte do futuro próximo da tecnologia, um sistema ou *site* sem características minimamente parecidas com um PWA, ou que não atendam aos mais variados calibres de usuários, muito provavelmente, se tornarão obsoletos em breve. O futuro bate à porta, e não espera que a abram para entrar.

### **Agradecimentos**

Camila: À minha filha e marido pela paciência e apoio no decorrer dos últimos três anos. Aos familiares e amigos, sou grata pelo folego nesta etapa vencida. Aos professores, meu muito obrigado pelo conhecimento repassado. À FATEC pela oportunidade apresentada.

## Referências

- ALURA. YouTube, sd. Disponível em <https://www.youtube.com/user/aluracursosonline>. Acesso em: 19 nov. 2018.
- BELLUNO. **Tendências que dominarão o mercado de Internet até 2020**. sd. Disponível em <https://www.bellunotec.com.br/conteudos/8-tendencias-que-dominarao-o-mercado-de-internet-ate-2020/>. Acesso em 02 jul. 2018.
- BITBUCKET. Sd. Disponível em <https://bitbucket.org/>. Acesso em: 19 nov. 2018.
- BRAZILJS. YouTube, sd. Disponível em <https://www.youtube.com/user/BrazilJS>. Acesso em: 19 nov. 2018.
- CASA DO CÓDIGO. Sd. Disponível em <https://www.casadocodigo.com.br/>. Acesso em: 19 nov. 2018.
- DEVMEDIA. Sd. Disponível em <https://www.devmedia.com.br/>. Acesso em: 19 nov. 2018.
- GITHUB. Sd. Disponível em <https://github.com/>. Acesso em: 19 nov. 2018.
- GOOGLE DEVELOPERS. **Progressive Web App Checklist**. sd. Disponível em <https://developers.google.com/web/progressive-web-apps/checklist>. Acesso em 15 out. 2018.
- GRONER, Loiane. **FrontInPOA: Progressive Web Apps e a evolução da Web**. YouTube, 2017. Disponível em <https://www.youtube.com/watch?v=AW-QR8F4RXs>. Acesso em 02 jul. 2018.
- HIMANGO, Judah Gabriel. **Publicar um PWA nas 3 lojas? Não é algo tão simples**. Windows Team, 2018. Disponível em <https://www.windowsteam.com.br/publicar-um-pwa-nas-3-lojas-nao-e-algo-tao-simples/>. Acesso em 09 set. 2018.
- IBGE (INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA). **Downloads**, 2015. Disponível em <https://www.ibge.gov.br/estatisticas-novoportal/downloads-estatisticas.html>. Acesso em 10 ago. 2018.
- IMASTERS. Sd. Disponível em <https://imasters.com.br/>. Acesso em: 19 nov. 2018.
- KLEINA, Nilton. **A história da internet: pré-decada de 60 até anos 80**. Tecmundo, 2011. Disponível em <https://www.tecmundo.com.br/infografico/9847-a-historia-da-internet-pre-decada-de-60-ate-anos-80-infografico-.htm>. Acesso em: 19 out. 2018.
- LEPAGE, Pete. **Seu primeiro Progressive Web App**. Web Fundamentals Google Developers, 2017. Disponível em <https://developers.google.com/web/fundamentals/codelabs/your-first-pwapp/?hl=pt-br>. Acesso em 02 jul. 2018.

MICALI, Bruno. **Segundo a Cisco, pessoas terão mais celulares do que eletricidade até 2020.** Tecmundo, 2016. Disponível em <<https://www.tecmundo.com.br/celular/95001-segundo-cisco-pessoas-terao-celulares-eletricidade-2020.htm>>. Acesso em: 19 out. 2018.

ONU (ORGANIZAÇÃO DAS NAÇÕES UNIDAS). **Brasil é o quarto país com mais usuário de Internet no mundo, diz relatório da ONU.** Nações Unidas no Brasil, 2017. Disponível em <<https://nacoesunidas.org/brasil-e-o-quarto-pais-com-mais-usuarios-de-internet-do-mundo-diz-relatorio-da-onu/>>. Acesso em 11 set. 2018.

\_\_\_\_\_. **Information Economy Report 2017: Digitalization, Trade and Development.** UNCTAD, 2017. Disponível em <[https://unctad.org/en/PublicationsLibrary/ier2017\\_en.pdf](https://unctad.org/en/PublicationsLibrary/ier2017_en.pdf)>. Acesso em 11 set. 2018.

RB TECH. Sd. Disponível em <<http://rbtech.info/>>. Acesso em: 19 nov. 2018.

TABELESS. Sd. Disponível em <<https://tableless.com.br/>>. Acesso em: 19 nov. 2018.

TRACTO. **Quantas pessoas têm acesso à internet no mundo?** 2017. Disponível em <<https://www.tracto.com.br/quantas-pessoas-tem-acesso-a-internet-no-mundo/>>. Acesso em 10 jul. 2018.

TYBEL, Douglas. **Orientações básicas na elaboração de um diagrama de classes.** Devmedia, 2016. Disponível em: <<https://www.devmedia.com.br/orientacoes-basicas-na-elaboracao-de-um-diagrama-de-classes/37224>>. Acesso em: 11 set. 2018.

UDEMY. Sd. Disponível em <<https://www.udemy.com/>>. Acesso em: 19 nov. 2018

VIEIRA, Leandro. **Web App Manifest para personalizar sua Progressive Web App.** Apiki Blog. 2017. Disponível em <<https://blog.apiki.com/2017/10/31/web-app-manifest/>>. Acesso em 14 ago. 2018.