

# Progressive Web Apps: uma alternativa às Apps móveis nativas

## Progressive Web Apps: an alternative to the native mobile Apps

David Fortunato

Departamento de Engenharia Informática e Sistemas  
Instituto Superior de Engenharia de Coimbra  
Coimbra, Portugal  
a21170225@alunos.isec.pt

Jorge Bernardino

Departamento de Engenharia Informática e Sistemas  
Instituto Superior de Engenharia de Coimbra  
Coimbra, Portugal  
jorge@isec.pt

**Resumo** — As aplicações móveis, vulgarmente conhecidas por Apps, têm alcançado um enorme sucesso no mercado de dispositivos móveis. Essa oportunidade atraiu muitas empresas interessadas em ter as suas próprias aplicações móveis otimizados para todos os principais sistemas operativos móveis. Contudo, estes desenvolvimentos podem ser demasiado dispendiosos, particularmente no caso de se desenvolver nativamente para cada plataforma. Os novos desenvolvimentos Web, permitiram obter mais funcionalidades e capacidades, que anteriormente apenas eram possíveis com aplicações nativas, o que começa a possibilitar o desenvolvimento de aplicações móveis centralizado nas Web Apps, que são aplicações para executar nos *browsers*. Este artigo analisa as evoluções, capacidades e limitações existentes no desenvolvimento de um aplicativo da Web para ser executado em todos os dispositivos. Apresentamos o novo conceito de Progressive Web App, criado pelo Google, de forma a normalizar todos os desenvolvimentos da Web. Também são apresentadas as principais vantagens no desenvolvimento de aplicativos centralizados como um Progressive Web App, comparando com o desenvolvimento da mesma solução para cada plataforma móvel diferente. É também descrito o estado atual das tecnologias Web e em que cenários as Progressive Web Apps são já uma boa alternativa às aplicações móveis nativas.

**Palavras-chave**—Web Apps, Progressive Web Apps, Android, iOS, Windows, Mobile Native Apps

**Abstract** — The mobile apps have been reaching a huge success on the mobile market. This opportunity attracted a lot of interested companies to have their own optimized mobile apps for all major mobile operation systems. However, these developments are expensive when developed natively for each mobile platform. New improvements done on the web technologies, allowed more

features and capabilities than previously was only possible on apps that was developed natively. This started new possibilities on consolidate all developments only on web apps, that are apps that runs on web browsers. This paper intends to understand which evolutions, capabilities and limitations exists on developing a web app to run in all devices. We present the new concept of Progressive Web App, created by Google, in a way to normalize all web developments. It will be introduced the major advantages on developing the apps centralized as a Progressive Web App, comparing on developing the same solution for each different mobile platform. It will be also described the current state of web technologies and in which preferable scenarios the Progressive Web Apps are a strong alternative to the mobile native apps.

**Keywords**— Web Apps, Progressive Web Apps, Android, iOS, Windows, Mobile Native Apps

### I. INTRODUÇÃO

As aplicações móveis mudaram o mercado das tecnologias de informação, criando novos mercados, maior acessibilidade para qualquer utilizador, novos empregos e conteúdos sempre acessíveis em qualquer momento. Esta acessibilidade fez com que os utilizadores queiram todo o seu conteúdo digital disponível em qualquer dispositivo com acesso à Internet, seja no telemóvel, computador, tablet, ou televisão, entre outros. Este facto faz com as empresas sejam obrigadas a ter as suas aplicações disponíveis online para qualquer dispositivo independentemente do seu sistema operativo, tamanho do ecrã ou outra qualquer condição do dispositivo. Esta imposição (do mercado) naturalmente implica desenvolvimentos, muitas vezes, extremamente dispendiosos, no caso de ter de se desenvolver o produto nativamente para cada plataforma. Para isso as empresas começaram a procurar soluções mais económicas e que ofereçam igualmente boa qualidade ao produto final [1].

Com esse objetivo, e também por forma a possibilitar a unificação destes desenvolvimentos (minimizando os custos de sincronização entre equipas), a Google lançou um novo conceito chamado de Progressive Web Apps (PWA). Esta nova tecnologia permite que uma aplicação possa estar disponível em qualquer dispositivo com acesso a um *Web Browser*, sem a necessidade de desenvolver a aplicação nativamente especificamente para determinado dispositivo ou sistema operativo. Ainda assim, essa aplicação deve oferecer uma experiência de utilização semelhante a uma aplicação nativa. O objetivo é que um utilizador comum não sinta a diferença de estar a executar uma aplicação nativa ou web.

Neste artigo investigamos este tipo de soluções, estudando a sua evolução e analisando os próximos avanços desta tecnologia, tentando assim perceber se este tipo de soluções poderá mesmo terminar com os desenvolvimentos nativos. São apresentadas as vantagens e as evoluções existentes nas tecnologias web de forma a perceber qual a melhor solução para cada tipo de projeto.

O artigo está estruturado da seguinte forma. Na seção II é feita uma introdução às aplicações móveis. Na seção III são apresentadas as Progressive Web Apps, analisando alguns trabalhos realizados sobre o tema. Na seção IV são apresentados alguns trabalhos relacionados. Na seção V são mostradas as especificações técnicas das Progressive Web Apps. Na seção VI é feita uma pequena introdução às frameworks de desenvolvimento de Web Apps e na seção VII uma comparação entre essas frameworks. Por fim, na seção VIII são apresentadas as conclusões, assim como algumas sugestões para trabalho futuro.

## II. IMPORTÂNCIA DAS APPS MÓVEIS

Qualquer empresa que pretenda ver o seu produto publicado online, sabe que para que este tenha sucesso depende muito da sua acessibilidade por parte dos utilizadores e que estes possam aceder ao produto em qualquer dispositivo com ligação à Internet. Atualmente estima-se que os utilizadores de smartphones passem aproximadamente 67% do seu tempo online alocado a aplicações móveis [2].

Na Figura 1 é apresentado o crescimento do tempo gasto pelos consumidores de conteúdo digital. Como se pode constatar, tem-se verificado um grande crescimento por parte dos utilizadores na utilização de Apps móveis (+111%) e Web Apps (+62%). Outro facto interessante destes resultados é verificarmos que o consumo a partir de desktop começa a baixar, aumentando a tendência para esse consumo ser efetuado via smartphone. Daí se compreende o constante interesse de qualquer empresa querer ver os seus produtos expostos com sucesso no mercado móvel. Compreende-se também o porquê das empresas que procuram obter o maior proveito de tráfego móvel preferirem focar-se nos desenvolvimentos de aplicações nativas especificamente para cada plataforma (geralmente para Android e iOS), sem olhar para as Web Apps como uma alternativa válida. Talvez um dos principais motivos seja devido ao mercado móvel ser atualmente tão rentável que ainda compensa ter equipas especializadas em cada plataforma de forma a obter o melhor

resultado possível. Contudo, o mercado começa a ficar saturado, com muita oferta e por vezes a inovação é o fator que distingue um produto de outros.

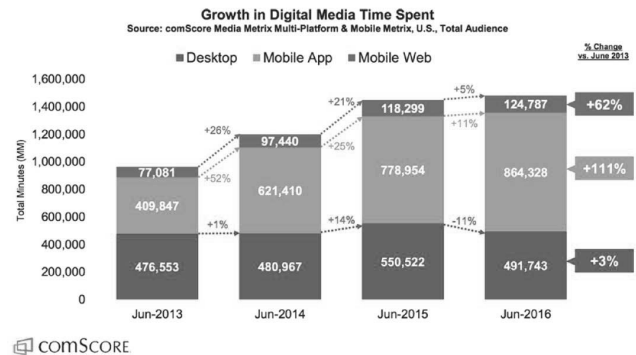


Figura 1: Tempo despendido com Apps móveis (fonte: comScore)

Na Figura 2 é apresentada a média de visitantes mensais nos maiores sites e aplicações móveis. Apesar do grande crescimento verificado no consumo nas aplicações móveis, o consumo via Web App continua em grande crescimento.

Isto indica claramente que para um qualquer serviço online ter sucesso no mercado móvel, não bastará focar-se apenas nas aplicações nativas, mas sim ter uma presença em web mobile muito forte e otimizada. Por exemplo, com base nestes dados se estivermos a produzir uma determinada Web App para mobile, teremos aproximadamente três vezes mais visitantes do que se estivéssemos a produzir uma aplicação móvel nativa. Assim, para uma qualquer empresa ter sucesso, tem que se preocupar com a presença do seu website no mercado, porque esse poderá ser muito provavelmente o maior ponto de acesso por parte dos utilizadores.

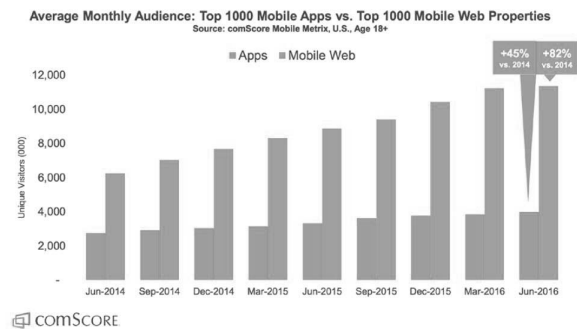


Figura 2: Média de vistas nas 1000 maiores aplicações móveis e web

## III. O QUE SÃO AS PROGRESSIVE WEB APPS?

Uma Progressive Web App (PWA) tem como foco o desenvolvimento progressivo e independente da plataforma que irá executar a aplicação. Este foi talvez o primeiro grande passo, para uniformizar os desenvolvimentos de aplicações para que estas estejam disponíveis em qualquer dispositivo e plataforma com um *Web Browser*. Este novo conceito deu-nos a possibilidade de chegar a um ponto de sermos capazes de fornecer funcionalidades semelhantes e uma interface de utilizador muito aproximada aquela encontrada nas melhores

aplicações nativas de Android ou iOS. Uma interface que anteriormente era difícil de fornecer nas tecnologias Web [2].

A vantagem mais importante desta abordagem é também a mais aparente, é que os programadores não precisam de desenvolver várias aplicações específicas para diferentes plataformas móveis. Mesmo para aqueles que desenvolvem apenas para iOS e Android, isso significa grandes economias em tempo e esforço no desenvolvimento de aplicações. As PWA oferecem a capacidade de desenvolver uma única aplicação e que funciona perfeitamente em todas as plataformas e dispositivos.

Ao contrário de uma aplicação Web convencional, as PWAs podem ser executadas em modo offline. Para isso basta adicionar a aplicação ao ecrã inicial do dispositivo (desktop ou smartphone) e que este tenha ligação à Internet de forma a instalar os dados necessários localmente no dispositivo. O modo offline é útil no caso de jogos ou conteúdo que se pretende consultar mais tarde. Quando esta aplicação é adicionada ao ecrã inicial deixa de ter o aspeto de um site, passando a dar a sensação de ser uma aplicação nativa. Este tipo de desenvolvimento permite também que a aplicação seja responsiva em diferentes tamanhos de ecrã, ou seja, o design adapta-se consoante o tamanho e orientação de ecrã.

Na Figura 3 temos o exemplo da aplicação *Trivago.pt*, em que a Trivago foi um dos primeiros parceiros da Google a terem uma aplicação PWA pronta a mostrar ao público. Na figura, as duas primeiras imagens representam a Web App da Trivago. A primeira aberta diretamente no *browser* e a segunda foi adicionada ao ambiente de trabalho do dispositivo e posteriormente aberta a partir do ambiente de trabalho, em que podemos ver que com isso a barra superior para inserir o endereço URL desapareceu, dando mais a ideia de esta ser uma aplicação nativa. A terceira imagem diz respeito à aplicação nativa. Ao nível de *user interface*, é muito difícil para um utilizador perceber que uma PWA adicionada ao dispositivo é na verdade uma Web App, porque esta sendo progressiva, adapta-se ao sistema em que foi instalada.

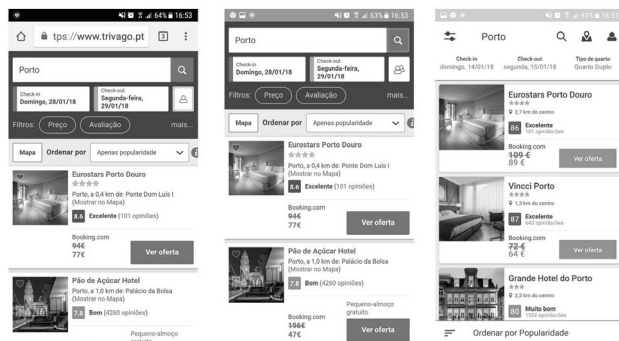


Figura 3: PWA vs App nativa (obtido através de screenshots em Android)

A Figura 4 apresenta outra novidade das PWA, em que estas ficam instaladas no sistema operativo como se de uma aplicação nativa se tratasse. Isto é, a aplicação é apresentada na lista de todas as aplicações disponíveis (não é apenas um atalho para um website) e inclusive aparece na lista de gestão de

aplicações para limpar a cache, espaço ocupado, forçar paragem ou desinstalar. Este pormenor, cria o efeito de que a aplicação está instalada no sistema operativo, contudo esta instalação não necessitou de uma aplicação nativa ou de uma *app store* para instalar essa aplicação.

Este conjunto de funcionalidades, de aproximação a uma aplicação nativa (em ambiente móvel), mas ao mesmo tempo apresentar-se como um website em browsers, é o que mais diferencia uma Web App.



Figura 4: Instalação de uma PWA no sistema operativo (obtido através de screenshots em Android)

No entanto, a possibilidade de poder ser executada em qualquer dispositivo, ser responsiva em diferentes tamanhos de ecrãs e orientação de ecrã, já existia antes do surgimento das PWAs. A diferença está na capacidade de se comportar como uma aplicação nativa quando esta é adicionada ao sistema operativo.

O conceito de progressivo vem precisamente da aplicação se adaptar e permitir diferentes funcionalidades dependendo do tipo de plataforma em que esta é executada. Dando o exemplo de uma PWA que pretenda efetuar o *upload* de uma foto. No caso de ser executada num dispositivo Android, esta aplicação terá acesso à câmara e se o utilizador pretenda, pode tirar a foto e enviar diretamente essa foto. Mas se esta aplicação seja executada no browser de um desktop, que não tenha câmara, apenas será apresentada a possibilidade do envio de ficheiro. Isto é o conceito de progressivo, em que progressivamente a aplicação se adapta ao dispositivo em que esta está a correr, não se limitando apenas à adaptação do tamanho de ecrã como se verificava nas *Web Apps*.

#### IV. TRABALHOS RELACIONADOS

Os avanços recentes da tecnologia web, permitiram a existência de novas funcionalidades que anteriormente apenas se encontravam nas aplicações nativas. As PWA procuram unificar os desenvolvimentos *web* e móvel. Com base nisso, Biørn-Hansen em [3] analisou a performance de duas aplicações *cross-platform* e uma *Progressive Web App*, que foram desenvolvidas apenas para fins comparativos. Após essa análise foi concluído que um dos grandes problemas das Progressive Web Apps está na falta de unificação dos browsers, em que cada um tem as suas funcionalidades, o que complica os desenvolvimentos de qualquer Progressive Web App [3]. No entanto, começa-se a notar uma maior unificação dos browsers nos últimos tempos.

As Mobile Web Apps representam um elevado número da utilização na Internet. Contudo, estas continuam mais lentas que as aplicações nativas ao nível de *user experience*. As Progressive Web Apps (PWA) foram introduzidas pela Google e procuram colmatar essa falha com a introdução de novas APIs conhecidas com *service workers*. Foi feita uma análise sobre o impacto que estes *service workers* têm na eficiência energética das PWAs, quando estas operam em diferentes condições de conexão e em dois telemóveis de diferentes gerações. No trabalho realizado por Malavolta et al. em [4], os resultados demonstram que os *service workers* não apresentam um impacto significativo no consumo de energia independentemente da qualidade de conexão. Os autores concluem que as PWA e os *service workers* são tecnologias promotoras em termos de eficiência energética.

W. Jobe em [5] procurou comprovar a viabilidade das Web Apps em substituição das Apps nativas. Foram desenvolvidas duas aplicações Web, a primeira para rastreamento de corridas e a segunda para reserva de corridas. Depois de um teste de 6 meses, o autor concluiu que as aplicações Web não têm a mesma qualidade de performance em comparação com as aplicações nativas derivado da fraca qualidade do sinal de GPS, contudo, para a reserva de corridas, ambas as aplicações cumpriram com o objetivo sem grandes diferenças entre elas. No artigo concluiu-se que, sempre que é necessária uma interação com o hardware (GPS ou câmara), a tecnologia Web ainda não é viável. Contudo uma aplicação móvel que apenas necessite de apresentar uma interface nativa e apresentação de conteúdo, as soluções Web são soluções viáveis.

As PWAs são uma boa solução para o desenvolvimento de aplicações para a área da saúde. Em [6] foi desenvolvida uma aplicação PWA de autodiagnóstico de doentes. Os resultados mostraram boa satisfação por parte dos utilizadores, indicando que a Web App era muito rápida e bem organizada. Este trabalho mostra que as PWAs podem ser vistas cada vez mais como uma solução válida em qualquer área de atividade. Contudo é crucial que neste tipo de aplicações sejam seguidas as melhores práticas de design, de forma a facilitar a adaptação de qualquer utilizador e minimizar o erro humano.

Todos os artigos referidos nesta seção ajudam a perceber o potencial e a evolução que as tecnologias Web têm desenvolvido de forma a atingirem uma maior qualidade e adaptabilidade ao maior número de sistemas possíveis. No entanto falta explorar o meio envolvente das tecnologias Web, as suas ferramentas de desenvolvimento e *debug* e qual a aceitação que estas têm por parte dos programadores.

## V. ESPECIFICAÇÃO TÉCNICA DAS PWA

Tecnicamente uma *Progressive Web App* não utiliza especificamente uma determinada tecnologia. Não é uma nova *framework* de desenvolvimento, assim como também não é uma nova linguagem de programação. Na verdade, as PWA são um conjunto de estratégias, técnicas e APIs que disponibilizam ao utilizador uma experiência de utilização muito semelhante com a experiência numa aplicação nativa.

As PWA são caracterizadas como:

- **Rápidas**, geralmente renderizando o conteúdo no dispositivo do utilizador em apenas alguns segundos;
- **Estável**, mesmo em conexões de Internet de fraca qualidade ou em dispositivos mais fracos;
- **Envolvente**, ao ativar notificações, mesmo na Web App, os utilizadores podem receber notificações para serem alertados sobre alguma informação pertinente na aplicação, mesmo que o *browser* não esteja aberto.

Para desenvolver uma aplicação PWA, e apesar de toda a sua vasta compatibilidade, não necessita de ser desenvolvida em várias linguagens de programação como Objective-C, Java, C++, como as aplicações nativas. Para o desenvolvimento de aplicações PWA, basta aprender JavaScript, CSS e HTML.

Ao longo do tempo, os browsers evoluem as suas funcionalidades, no que se costuma designar por *progressive enhancement* (aprimoramento progressivo). É aqui que surge o verdadeiro potencial das PWA, os utilizadores obtêm uma melhor experiência de utilização da aplicação consoante mais atualizado estiver o seu *browser* em que esteja a executar a aplicação.

Uma das funcionalidades mais relevantes das PWAs resulta da capacidade destas conseguirem ser executadas em modo offline. Esta capacidade é obtida através dos *service workers*.

## VI. FRAMEWORKS DE DESENVOLVIMENTO

Como referido anteriormente, as Progressive Web Apps não são na verdade uma nova tecnologia ou uma linguagem de programação, nem mesmo uma framework. São na verdade um conceito e determinados procedimentos que se devem seguir no desenvolvimento de uma Web App, para que esta seja considerada uma Progressive Web App. Nesta seção serão descritas brevemente o React, Angular, Vue.js e Ionic, que são consideradas as principais frameworks utilizadas pelos programadores Web [7].

### A. React (<https://reactjs.org/>)

Esta é uma das frameworks mais conhecidas para o desenvolvimento de Web Apps. Tem tido um grande crescimento no número de programadores a escolherem esta framework no desenvolvimento das suas Web Apps. Foi criada e mantida pela equipa do Facebook [8]. A linguagem de programação utilizada é o JavaScript.

Com base no React e tendo em conta o seu sucesso, foi disponibilizada uma nova versão do React mas com desenvolvimento de aplicações híbridas. Essa versão chama-se React Native [9], que procura obter o melhor de dois mundos, o de desenvolver uma vez e conseguir ser executada em qualquer dispositivo, mas também otimizar a usabilidade do utilizador dependendo do seu dispositivo. Isto é, no caso da aplicação correr em Android ou iOS, utilizar componentes nativos de forma a permitir a melhor usabilidade. A grande vantagem é mesmo de conseguir diminuir os custos de desenvolvimento, mas ao mesmo tempo potenciar a aplicação ao máximo em cada plataforma, por exemplo, permitir a apresentação de *Widgets* no ecrã inicial do Android (que com uma Web App não permite essa funcionalidade, por esta ser específica da própria plataforma).

### B. Angular (<https://angular.io/>)

Mais uma vez a Google procurou estar envolvida no desenvolvimento das Web Apps, e para isso uniu-se ao projeto Angular. Este projeto, inicialmente conhecido por AngularJS, foi criado para facilitar os desenvolvimentos de Web Apps e garantir que estas sejam responsivas em qualquer tamanho do dispositivo. Inicialmente a linguagem de programação desta framework era JavaScript, mas desde a versão Angular 2, que a linguagem de programação foi alterada para TypeScript. Esta é uma linguagem de script criada e desenvolvida pela Microsoft. A sintaxe é um superconjunto de JavaScript, que adiciona tipagem estática [10] opcional, isto porque todo o código JavaScript também é TypeScript válido. Esta nova linguagem mantém os atuais programadores de JavaScript e atrai programadores com maior experiência de linguagens com tipagem como o caso de Java ou Objective-C.

### C. Vuejs (<https://vuejs.org/>)

Vue.js é uma framework de desenvolvimento de aplicações Web de *front-end* em JavaScript. Foi criada para organizar e simplificar o desenvolvimento Web [11].

A *framework* centra-se na renderização declarativa e na composição dos componentes visuais, e pode ser incorporada nas páginas já existentes. Os recursos avançados necessários para aplicativos complexos, como localização, gestão de estados e ferramentas de desenvolvimento são oferecidos através de bibliotecas mantidas oficialmente pela equipa Vuejs.

Esta é muito semelhante ao React e ao Angular, mas como é mais recente, foi criada com base nas virtudes de ambas as *frameworks* e minimizando os problemas já identificados.

### D. Ionic (<https://ionicframework.com/>)

A framework Ionic, assemelha-se ao React Native, ou seja, é uma framework de desenvolvimento de aplicações híbridas. O seu objetivo é o de centralizar os desenvolvimentos de uma aplicação, poder correr em qualquer plataforma e também adicionar componentes e funcionalidades específicas em cada plataforma que esteja a correr a aplicação. As maiores diferenças é de que o desenvolvimento de aplicações em Ionic, ao contrário de todas as outras frameworks, não é gratuito, necessitando de um pagamento mensal de forma a poder ser desenvolvida uma aplicação com esta tecnologia.

Esta framework é desenvolvida em Angular, o que permite a que um programador de Angular possa facilmente começar a desenvolver aplicações em Ionic.

## VII. COMPARAÇÃO DAS FRAMEWORKS

Na Figura 5 são apresentados os dados relativos aos downloads efetuados nos últimos meses a cada uma das frameworks. Em primeiro lugar (a cor-de-laranja) encontra-se o React, com grande crescimento em comparação com os restantes. Contudo, foi necessário separar os downloads relativos à primeira versão do Angular (a vermelho) das restantes versões (a azul), porque estas foram publicadas separadamente, o que tem que se juntar ambos os resultados para se obter um valor mais aproximado à realidade. Com isto percebe-se que todas elas estão em grande crescimento, com

exceção da Ionic (a amarelo), que parece não estar a ganhar novos programadores, parecendo mesmo estar a perder quota de mercado em relação aos seus concorrentes.

Atualmente estão a aparecer cada vez mais frameworks para facilitar os desenvolvimentos de Web Apps, o que parece ser uma tendência para os próximos anos, tendo em consideração a sua evolução.

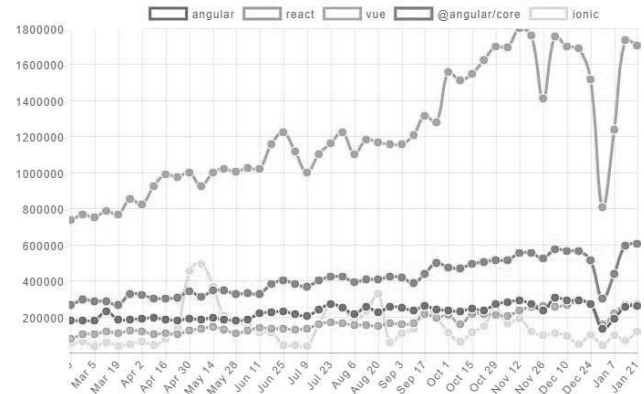


Figura 5: Instalações de cada framework nos últimos meses (fonte: [npm.trends.com](https://npm.trends.com))

Uma das grandes vantagens das PWA para o utilizador é de que este tem a garantia de estar a utilizar sempre a última versão disponível (no caso de aceder online), sem a necessidade de atualizar a aplicação a partir da *app store*. Ao contrário das aplicações nativas, em que estas necessitam de uma instalação para que o utilizador aceda à última versão da aplicação. Outra vantagem de uma PWA é o tempo de desenvolvimento envolvido na aplicação, em que uma PWA requer muito menor esforço. Isto é devido a que esta é tanto uma aplicação móvel como um website, podendo ser executada em qualquer tipo de sistema operativo.

Quando uma qualquer empresa decide desenvolver uma aplicação para o seu produto, no caso de escolher o desenvolvimento PWA, esta apenas terá de gerir uma equipa e um projeto. Quando pretende desenvolver uma aplicação nativamente para cada plataforma, os custos aumentam com o número de plataformas que se pretende suportar, assim como os custos indiretos pela manutenção de cada plataforma.

No caso de se desenvolver nativamente para cada plataforma, vai requerer no mínimo um Website responsivo (muitas vezes este website já é quase uma Progressive Web App), uma App nativa para Android e para iOS [12]. Estas geralmente oferecem as mesmas funcionalidades por uma questão de coerência, portanto no caso de gestão e sincronização de todas as plataformas para a mesma funcionalidade aumenta consideravelmente em relação ao desenvolvimento unificado de uma PWA. No caso de os desenvolvimentos serem unificados em PWA, consegue-se garantir as mesmas funcionalidades para qualquer plataforma do utilizador e ao mesmo tempo evitar elevados riscos de desenvolvimentos sem retorno.

Com base nesta análise podemos concluir que as principais vantagens no desenvolvimento de uma PWA em relação às aplicações nativas são as seguintes:

- Desenvolvimento mais barato e mais rápido;
- Gestão de equipas de desenvolvimento mais fácil;
- Implementação centralizada sempre atualizada;
- Tecnologia com grande evolução e *debug* mais fácil;
- Pode ser executada em qualquer dispositivo.

Para o desenvolvimento nativo podemos enunciar as seguintes vantagens:

- Consegue-se uma performance superior;
- Pode utilizar todas as funcionalidades disponíveis;
- Tecnologia mais estável e com maior segurança;
- Possibilidade de desenvolvimento de mais baixo nível;
- Desenvolvimentos focados para a plataforma específica;
- Instalável a partir da *app store*.

## VIII. CONCLUSÕES E TRABALHO FUTURO

Uma das principais conclusões deste estudo, foi constatar que é possível disponibilizar uma aplicação com muito boa qualidade e otimizada para qualquer dispositivo desenvolvendo apenas uma Progressive Web App (PWA). Além disso, com a possibilidade de poder chegar a todo o público online, seja por via desktop, mobile ou tablet. Pelo contrário, desenvolvendo especificamente para iOS ou Android, fica sempre a faltar a capacidade Web. Assim, uma aplicação que tenha sido apenas desenvolvida em PWA pode atingir todo o público, mas uma aplicação nativa nunca o consegue atingir. No caso de uma empresa, com orçamento limitado e que pretenda provar um determinado conceito, deverá procurar o desenvolvimento da sua solução em PWA de forma a alcançar todo o público online. Posteriormente, caso tenha obtido o sucesso esperado e sinta essa necessidade, então desenvolver essa aplicação nativamente para cada plataforma. Isto porque as funcionalidades que não existem numa PWA, são geralmente vistas como funcionalidades extras e não fundamentais para o sucesso de uma aplicação.

Para além dos custos envolvidos ao ser desenvolvida separadamente para cada plataforma, acresce os tempos de gestão e manutenção das equipas para cada plataforma. Isto porque, geralmente as equipas que desenvolvem para Web, Android e iOS são distintas, o que gera maiores tempos de gestão e manutenção, ao nível de sincronismo das equipas de desenvolvimento. Ao se projetar uma nova funcionalidade, todas as equipas têm de estar sincronizadas, de forma a evitar que determinada plataforma apresente comportamentos diferentes em comparação com outra plataforma para uma mesma funcionalidade. Isto é, ao nível de usabilidade do utilizador e algoritmos os desenvolvimentos para cada plataforma devem ser semelhantes. A isto acresce as alterações, em que sempre que haja a necessidade de alterar um comportamento da aplicação em determinada funcionalidade, esta tem de ser replicada para cada plataforma. Por isso, geralmente as empresas procuram desenvolver primeiro para uma determinada plataforma, e só depois desta estar pública e obter o *feedback* dos utilizadores, se começa a desenvolver para as restantes plataformas, de forma a minimizar o risco das

alterações. Tudo isto leva tempo, atrasando o seu desenvolvimento, que muitas vezes pode ser precioso para não se perder as janelas de oportunidade no mercado. Com os desenvolvimentos centralizados em PWA, as equipas estão juntas e trabalham por funcionalidade, o que facilita e agiliza novos desenvolvimentos do produto.

Atualmente existem enormes esforços por parte de várias empresas ligadas às tecnologias em fazer evoluir as Web Apps. O caso mais notório é a Google, que está envolvida em quase todos os grandes projetos em que se procure evoluir os desenvolvimentos Web, desde o conceito das Progressive Web Apps, a ferramenta Lighthouse, a framework Angular ou o AMP. Apesar da Google ser a detentora do Android, esta está a procurar potenciar as Web Apps, o que dá a ideia de que esta procura mesmo centralizar ao máximo os desenvolvimentos.

Como trabalho futuro pretendemos também vir a comparar os níveis de segurança, desenvolvendo uma aplicação web e outra nativamente.

## REFERÊNCIAS BIBLIOGRÁFICAS

- [1] C. Rickard, "Hybrid mobile apps: Reach more people, faster and cheaper," 2016. [Online]. Available: <http://inoutput.io/articles/development/hybrid-mobile-apps-reach-more-people-faster-and-cheaper>.
- [2] D. Sheppard, *Beginning Progressive Web App Development Beginning Progressive Creating a Native App Experience*. 2017.
- [3] A. Bjørn-Hansen, T. A. Majchrzak, and T. Grønli, "Progressive Web Apps : The Possible Web-native Unifier for Mobile Development," no. Webist, pp. 344–351, 2017.
- [4] I. Malavolta, G. Procaccianti, P. Noorland, and P. Vukmirovi, "Assessing the Impact of Service Workers on the Energy Efficiency of Progressive Web Apps," 2017.
- [5] W. Jobe, "Native Apps vs . Mobile Web Apps," vol. 7, no. 4, pp. 27–32, 2013.
- [6] M. Wahlstr, "Exploring progressive web applications for health care Developing a PWA to gather patients ' self assessments," 2017.
- [7] M. Raible, "No Title," 2017. [Online]. Available: <https://scotch.io/tutorials/the-ultimate-guide-to-progressive-web-applications>.
- [8] Facebook, "React." [Online]. Available: <https://reactjs.org/>.
- [9] Facebook, "React Native." [Online]. Available: <https://facebook.github.io/react-native/>.
- [10] "TypeScript." [Online]. Available: <https://www.typescriptlang.org/>.
- [11] "Vuejs." [Online]. Available: <https://vuejs.org/>.
- [12] Stanislava Nedyalkova and Jorge Bernardino. "Open source capture and replay tools comparison". In *Proceedings of the International C\* Conference on Computer Science and Software Engineering (C3S2E '13)*, ACM, New York, NY, USA, 117-119, 2013.