

Progressive Web Apps: Concepts and Features

João Carneiro

Assistant Professor at ISTECS - joao.carneiro@my.istec.pt

Resumo: As aplicações móveis progressivas (PWA) apresentam-se hoje como a alternativa às aplicações nativas a que nos temos habituado. O desenvolvimento multiplataforma sem prejuízo da funcionalidade, performance ou acessibilidade representam importantes vantagens. Este artigo resume as principais características e conceitos base das PWA.

Palavras-chave: *Aplicações Web Progressivas, PWA, Aplicações Móveis, Web, Multiplataforma.*

Abstract: *Progressive mobile apps (PWA) can be seen today as the alternative to the native apps we became used to. The use of cross-platform development without compromising features, performance or accessibility are an important advantage. This article presents the main characteristics and concepts that support PWA.*

Keywords: *Progressive Web App, PWA, Mobile Applications, Web, Cross-Platform.*

I. Introduction

Traditionally, creating interoperable mobile applications implies important difficulties in the development process. Publishing an application in multiple mobile platforms or operating systems, requires the development of separate independent development projects for each target system, and therefore also different development environments, tools and resources with adequate competencies to tackle each system [1][2].

For companies, this reality translates into big challenges such as the difficulty of having and maintaining specialized resources (human, physical and licenses) for each platform and the difficulty of ensuring a timely time to market in multiple systems. Due to these difficulties the

multiplatform development, where a single source code is produced, being immediately compatible with different platforms has been progressively gaining attention throughout the years [3][4].

When referring to multiplatform development, the following three methods exist:

- 1) Hybrid – the apps are developed using web technologies (such as HTML and CSS) and displayed on the device using a WebView (e.g. Ionic and PhoneGap);
- 2) Interpreted – the apps are executed over a JavaScript interpreter that resides on the device which will then generate the corresponding native components (e.g. React Native);
- 3) Multiplatform compilation – the source code is separately compiled to each of the target platforms producing the corresponding binaries (e.g. Xamarin) [6].

The web applications have been widely used as a mean to distribute multiplatform computer software across different computer architectures. However, until recently, these were limited and could not compete with native apps. The Google Web Fundamentals group has been pushing the development of a set of relevant features such as the support for offline mode, synchronization in the background or application installation into devices. The appearance of these features allowed for significant advances in web apps and allowed the appearance of what we currently define as Progressive Web Apps (PWA). The evolution of web applications to PWA has been proving the ability of web technologies in offering similar features, and therefore competing, against native apps, with the added advantage of a requiring one single (multiplatform compatible) software development source code. Table 1 presents a comparison of features offered by native apps

and each of the three multiplatform development methods that we previously described.

	Interpret.	PWA	Hybrid	Native
Installable	Y	Y	Y	Y
Offline mode	Y	Y	Y	Y
No installation	N	Y	N	N
Marketplace available	Y	Y	Y	Y
Push notifications	Y	Y	Y	Y
Multiplatform	Y	Y	Y	N
Hardware and system API	Y	Y	Y	Y
Background synchronization	Y	Y	Y	Y

Table 1: Feature availability in Interpreted, Progressive Web Apps, Hybrid and Native apps [5]

PWA provide the same level of features as Native apps as well as Interpreted and Hybrid apps, however, PWA are the only type of apps which can be used without installing. Due their capabilities, PWA are currently seen as a strong candidate for future replacement of native apps.

II. Features and Concepts

PWA use the latest advances in web technology in order to provide a similar experience to the experience native apps offer to users. PWA are web applications installed and configured into web servers, accessible through an URL and a browser and indexed by the search engines.

In order to be considered a PWA, a web application has to offer at least the following minimum feature requirements:

- a) Progressive – developed over the base principles of progressive applications, ensuring accessibility to all users, independently of the used browser;
- b) Responsive – adapted to any screen format and any platform including desktop, laptop, tablet or mobile phone;

- c) Connectivity independent – uses Service Workers to ensure a correct behavior even when the network connection is limited or does not exist [7];
- d) Native app similarity – uses the AppShell model in order to offer similar navigation and interactivity as native apps [7][8];
- e) Updated – uses Service Workers to ensure continuous content update [7][8];
- f) Safe – served over HTTPS to ensure security in all data connections;
- g) Easy discovery – identified as a PWA using a WebApp Manifest and Service Worker registration allowing indexing of search engines [7][8];
- h) Engaging – use of features such as push notifications for higher interaction and user engagement;
- i) Installable – allow that users store the applications in their desktop or devices without having to rely on app marketplaces;
- j) Easy connection – allow an easy sharing through a simple URL without the need to go through the installing process.

The following fundamental core technologies enable the development of the above mentioned PWA features: Service Workers, Application Shell, Web App Manifest and HTTPS.

III. Performance

An essential metric for evaluating the different multiplatform app technologies is their performance. In [5], the authors compare the apps from each method in terms of publication size, startup time and rendering time. Table 2 summarizes the results.

	Hybrid	Interpreted	PWA
Size	4.53MB	16.39MB	104KB
Startup Time	860ms	246ms	230ms
Rendering Time	9242.1ms	862ms	1319ms

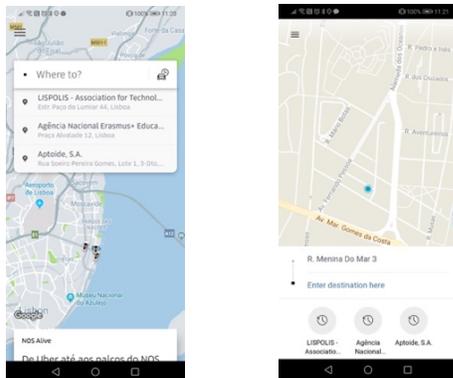
Table 2: Performance metrics (size, startup and rendering times) for each type of multiplatform application

IV. Examples

Today, multiple examples of PWA versions of well-known applications already exist, having been developed offering the benefits discussed in the previous sections. Some examples are Starbucks, Lyft, Twitter, Uber, Trivago, Instagram, Google Photos or Outlook.

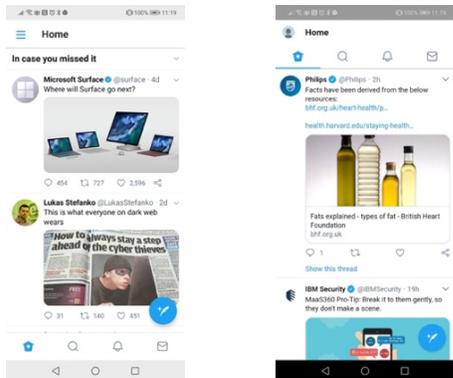
Next, we provide side-by-side screenshots showing examples of equivalent native and PWA interface and icons of two well-known apps (Uber and Twitter) installed and running in an Android phone. For the given examples, one should notice the inexistence of relevant difference of the PWA when compared to its corresponding native apps.

a) Uber



Picture 1: Screenshots of Uber Native (left) and PWA (right)

b) Twitter



Picture 2: Screenshots of Twitter Native (left) and PWA (right)

c) Icons



Picture 3: Icons for Uber and Twitter native (left) and PWA (right)

V. Conclusion

We are seeing relevant advances in the field of PWA and the mobile community has been discussing on its adoption as viable alternative to the well-established native apps. The multiplatform compatibility offers important economic advantages of developing a single source code while maintaining access to the whole mobile platform market share and even extending to other ecosystems.

The increasing use of PWA by top companies such as Google, Microsoft, Twitter or Uber show a clear trend in the adoption of the technology. However multiple challenges still need to be tackled such as the dependencies and limitations of some browsers [9], mobile data costs, methods for new app discovery, new business models and the impact on user interactions.

VI. References

- [1] Heitkötter H., Hanschke S., Majchrzak T.A. (2013) Evaluating Cross-Platform Development Approaches for Mobile Applications. In: Cordeiro J., Krempels KH. (eds) Web Information Systems and Technologies. WEBIST 2012. Lecture Notes in Business Information Processing, vol 140. Springer, Berlin, Heidelberg.
- [2] Henning Heitkötter, Tim A. Majchrzak, and Herbert Kuchen. 2013. Cross-platform model-driven development of mobile applications with md2. In Proceedings of the 28th Annual ACM Symposium on Applied Computing (SAC '13). ACM, New York, NY, USA, 526-533. DOI: <https://doi.org/10.1145/2480362.2480464>.
- [3] Raj, C. R., & Tolety, S. B. (2012, December). A study on approaches to build cross-platform mobile applications and criteria to select appropriate approach. In India Conference (INDICON), 2012 Annual IEEE (pp. 625-629). IEEE.
- [5] Biørn-Hansen, A., Majchrzak, T.A., & Grønli, T. (2017). Progressive Web Apps: The Possible Web-native Unifier for Mobile Development. WEBIST.
- [6] Latif, M., Lakhri, Y., Nfaoui, E. H., and Es-Sbai, N. (2016). Cross platform approach for mobile application development: A survey. In 2016 Int. Conf. on Information Technology for Organizations Development (IT4OD), pages 1-5. IEEE.
- [7] Gaunt, M. (2016). Service Workers: an introduction.
- [8] Osmani, A. and Gaunt, M. (2017). Instant loading web apps with an application shell architecture.
- [9] Malavolta, I. (2016). Beyond native apps: Web technologies to the rescue! (keynote). Pro. 1st Int. Workshop on Mobile Development. ACM.

