

# PWA, Native, Cross-Platform IEEE-830.

By Guillermo Salas Diaz

10-B

Progressive web applications

Professor: Dr.Ray Brunette Parra Galaviz

1/09/2024

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Purpose . . . . .	2
1.2	Research scope . . . . .	2
1.3	Definitions, Acronyms and Abbreviations . . . . .	3
1.4	References . . . . .	3
1.5	Vision General of the Document . . . . .	3
<b>2</b>	<b>PWA</b>	<b>3</b>
2.1	Characteristics of PWA . . . . .	4
2.2	Characteristics . . . . .	4
2.3	Advantages . . . . .	5
2.4	Disadvantages . . . . .	5
<b>3</b>	<b>Cross-Platform Applications</b>	<b>5</b>
3.1	Characteristics . . . . .	5
3.2	Advantages . . . . .	6
3.3	Disadvantages . . . . .	6
<b>4</b>	<b>Native Applications</b>	<b>6</b>
4.1	Characteristics . . . . .	6
4.2	Advantages . . . . .	7
4.3	Disadvantages . . . . .	7
<b>5</b>	<b>Service-oriented architecture</b>	<b>7</b>
5.1	Characteristics . . . . .	8
5.2	Advantages . . . . .	8
5.3	Disadvantages . . . . .	8
<b>6</b>	<b>Diferrence PWA, Native an Cross-Platform</b>	<b>8</b>

# 1 Introduction

The following document present the research on PWA, Native and multi platform.

In the current panorama technological in the development application has become a important pillar to offer immersive and efficient digital experiences. In this context, three different focus emerged that have set the course for application development: Progressive Web Apps(PWA), native applications and cross-platform applications.

The web applications are computer programs allow working with technologist web services, similar to HTML, CSS, JS and PHP. This tools allow access to application through web navigator and made more easier to create and administrate content in line.

## 1.1 Purpose

In this work going to explain the context of different types applications, the difference and objectives, is important differentiate this types of applications cause' in we career we must decide to use between each one.

The document it is aimed at a school audience whit the propose to offer different perspective on the use of web applications for each type of developer who is beginning to develop systems.

## 1.2 Research scope

The functionality of research on PWA, Native and multi platform it fundamental for the professionals in develop systems of software and designers in UX/UI.

Learning about Progressive Web Apps, native apps, an cross-platform apps provides software development professionals with a solid foundation to address various challenges and opportunities in the ever-dynamic field of technology

- **Informed decision making:** By understanding the characteristics, advantages, and disadvantages of each approach, developers and decision-makers can choose the most appropriate strategy for a particular project. This knowledge involves considering factors such as the type of app, deadlines, target audience, and required functionalities.
- **Flexibility in development:** With knowledge of PWAs, native, and cross-platform applications, developers have the ability to adapt to different contexts and project requirements.
- **User experience:** Understanding the specific characteristics of each type of application helps in designing a program that assists the user in achieving more with less effort.
- **Technological evolution:** In the world of software development, technologies are constantly evolving. Being in contact with new technologies allows you to stay competent.

It is expected to learn about Progressive Web Apps, native applications and multi platform applications for the development of more specific applications that allow the user to have more comfort.

The expected goal is to learn and differentiate between each one, to apply it in our project in a way that can benefit us.

Frameworks like React help to do PWA, and others helping to user designer to do more easier applications, that tools are important to the development and the objective to learn more others types of applications it is to find more tools.

### 1.3 Definitions, Acronyms and Abbreviations

- PWA: Progressive Web Application.
- APPS: Applications.
- UE: User Experience.
- UI: User Interface.
- HTML: Hypertext Markup Language.
- JS: JavaScript.
- PHP: Hypertext Processor.
- WEB: World Wide Web.
- SOA: Service oriented-architecture.
- CSS: Cascading Style Sheets.

### 1.4 References

Applications web. (2020, 4 de noviembre). Postgradoingenieria. <https://postgradoingenieria.com/question-aplicaciones-web/>

Making a Progressive Web App | Create React App. (2021, 4 de noviembre). Create React App. <https://create-react-app.dev/docs/making-a-progressive-web-app/>

### 1.5 Vision General of the Document

This section provides a brief description of the contents. Its main objective is to provide readers with an overview that allows them to understand the structure and purpose of the document.

This document has been developed in order to comprehensively the types of development forms. The information here covers aspects of each type of application.

The organizational structure of this document follows IEEE830 documentation best practices, allowing easy navigation and cross-reference between the different sections. Readers are encouraged to follow the proposed sequence to obtain a complete understanding of the system requirements.

## 2 PWA

PWA stands for "Progressive Web App" (Progressive Web Application). A Progressive Web App is a type of web application that uses modern web technologies to offer a user experience similar to that of a native application. The main idea behind PWAs is to provide an application that is reliable, fast and attractive, regardless of the browser or device being used.

PWA stands for "Progressive Web App" (Progressive Web Application). A Progressive Web App is a type of web application that uses modern web technologies to offer a user experience similar to that of a native application with some differences. The main idea behind PWAs is to provide an application that is reliable, fast and attractive, regardless of the browser or device being used with the aim of covering a wider range of users.

## 2.1 Characteristics of PWA

1. Responsive: PWAs adapt to different devices, crucial with the predominant role of smartphones. They automatically adjust to any format, browser, or device, considering variations in measurements and resolution, especially given their mobile nature.
2. Updated: PWAs always display their latest version to the user through automatic updates, ensuring constant and instant updates without the need for manual downloads.
3. Safe: PWAs use the secure HTTPS protocol, employing technologies such as TLS for web encryption.
4. Quick: PWAs are generally optimized for speed, both in terms of loading and browsing.
5. Offline: PWAs must allow access, either partially or completely, even in the absence of an internet connection or under low-quality conditions. Service workers and caching of essential information enable the app to serve content to users who are offline, starting from the first time it is opened.
6. Multi-platform: In their development, the technology used allows execution on various devices, operating systems, and browsers.
7. Native appearance: The user interface and overall appearance of a PWA closely resemble that of native apps, both in aesthetics and in the way users interact and navigate through it.

Features of a native App With the evolution of PWAs, they have been acquiring options that were previously reserved only for native Apps, such as access to different functions of the device. Progressive Web Apps can, for example, access the device's geolocation, Bluetooth, sync in the background or send push notifications (even when the PWA is not open). These notifications are a powerful communication tool that informs the user and invites them to access, increasing visits and, consequently, conversions. It should be considered that these possibilities are not available for all browsers.

## 2.2 Characteristics

Some of the features progressive web applications are as follows:

1. All PWAs are designed to function in any browser that adheres to appropriate web standards.
2. Facilitates developers in creating cross-platform applications more efficiently compared to native applications.
3. Employs a progressive enhancement web development strategy.
4. Some PWAs adopt a design architectural style known as the App Shell Model.

## 2.3 Advantages

In the evolution of web development, Progressive Web Apps (PWAs) take center stage, promising rapid performance and an uninterrupted user experience.

1. **Faster performance:** PWAs are designed to load faster and provide the best user experience.
2. **Security:** Through the use of HTTPS, PWAs offer a higher level of security.
3. **Less storage requirements:** PWAs require less storage space for data.
4. **Offline operation:** PWAs can use cached content and allow interaction even when offline.
5. **No installation required:** They can be accessed directly from the internet without the need for installation.

## 2.4 Disadvantages

Some disadvantages of using PWA in application development could be the following, it does not mean that they cannot be covered with their advantages, but it was the most relevant.

1. **Browser Dependency:** Some websites may function differently across browsers, necessitating additional adjustments during development.
2. **Reduced Visibility in App Stores:** Despite being indexable by search engines, PWAs may have lower visibility compared to apps listed on popular app stores.
3. **Monetization Complexity:** Monetizing PWAs can pose challenges, often being more intricate than native apps.

# 3 Cross-Platform Applications

Development cross-platform is the practice to do application so than can be distributed on more than platform, like mobile, tablet or computer all the same time.

This type of development focuses on focusing the accessibility of the application taking resolutions, devices and requirement of each devices to access them.

Traditionally, application development was done custom for each platform. This mean that to launch an IOS app, it was necessary write the app code in native IOS language. Actually you can write the same code for devices to access regardless of operating system.

## 3.1 Characteristics

**Technological flexibility:** In some cases, developers are allowed to use common technologies such as HTML, CSS and JavaScript, which facilitates the creation of applications more quickly.

**Speed when developing applications:** Once the base components of an application have been created correctly, we could use them in other applications without having to recreate or create for a specific operating system.

**Efficient Maintenance:** Considering that we have a single code base, we could say that maintenance would be very fast, because it would be updated on all platforms.

Support: We can create applications for different platforms and the most interesting thing is, some of them would be Android, Windows, macOS and more, using a coherent approach.

### 3.2 Advantages

1. **Compatibility:** Cross-platform applications are crafted for efficient use on various devices, regardless of the operating system.
2. **Code Reuse:** A crucial characteristic of Cross-Platform development is code reuse, enabling programmers to save time by utilizing a single codebase.
3. **Cost Effectiveness:** Developing a unified codebase for multiple platforms can be more cost-effective than creating separate applications for each platform.
4. **Uniform User Experience:** Multiplatform applications aspire to deliver users a consistent and enhanced access experience across all their devices, ensuring efficiency with the same application at any given time.

### 3.3 Disadvantages

1. **In some cases, higher consumption:** The additional layer of abstraction could negatively affect performance on older devices or devices with limited hardware. This would be a long-term limitation when wanting to develop for devices that have a lower reputation.
2. **Optimization problems for specific devices:** It is true that specific optimization for a device or platform can be more challenging compared to native development. This could affect the user experience when wanting to develop for a specific device; we found that limit.
3. **Personalization:** By wanting to cover so many devices, we will encounter the limitation of customization. In some devices, the functionality will be seen at its maximum performance capacity, focusing on a more standardized system.

## 4 Native Applications

The term native app refers to an app that you can download and install on a device. A native mobile application is developed specifically for a mobile device. The terms native app, native mobile app, and mobile app are often used interchangeably to refer to the same type of software.

### 4.1 Characteristics

Service-oriented architecture provides a modular and reusable approach to software development, fostering flexibility and efficiency. Native mobile applications allow users to interact with the operating systems and internal hardware of devices. You can grant users access to native features, such as:

1. Device location tracking
2. Device cameras and microphone

3. User contact lists
4. Touch gestures, device tilt, and other user interactions
5. Device security features, such as fingerprint scanner or facial recognition

## 4.2 Advantages

1. Native applications are developed focusing solely on a single device, they offer several advantages compared to the competition.
2. Optimized performance: They are developed to provide the best optimization according to the device being used, measures are implemented to use all the capabilities that the device has.
3. Full access to the device on which it is developed: The execution of these applications normally asks for access to some features to provide the best experience, compared to some PWAs.
4. Complementation with the operating system: They integrate very well with the operating system that is being used in the specific development of the device.
5. UI/UX: By working with a dedicated device, it allows us to guarantee a better experience by giving you access from your mobile or computer, in a more specific way that will not be affected in its performance.

## 4.3 Disadvantages

1. In some cases, higher consumption: The additional layer of abstraction could negatively affect performance on older devices or devices with limited hardware. This would be a long-term limitation when wanting to develop for devices that have a lower reputation.
2. Optimization problems for specific devices: It is true that specific optimization for a device or platform can be more challenging compared to native development. This could affect the user experience when wanting to develop for a specific device; we found that limit.
3. Personalization: By wanting to cover so many devices, we will encounter the limitation of customization. In some devices, the functionality will be seen at its maximum performance capacity, focusing on a more standardized system.

# 5 Service-oriented architecture

Service-oriented architecture (SOA) is a software development methodology that leverages modular components called services to construct business applications. These services can communicate across diverse platforms and languages, fostering re usability and flexibility in application development. This approach aims to streamline processes by breaking down complex tasks into manageable, independent services.

SOA involves creating services, each catering to a specific business capability, which can be utilized across different systems. It promotes the reuse of services, reducing development time and costs. An example includes consolidating user authentication functionality into a single service that can be shared among various business processes.

## 5.1 Characteristics

1. **Reusable Services:** SOA encourages the creation of modular and reusable services.
2. **Cross-Platform Communication:** Services can communicate seamlessly across different platforms and programming languages.
3. **Flexibility:** Developers can combine independent services to perform intricate tasks.
4. **Business Capability Focus:** Each service corresponds to a specific business capability.

## 5.2 Advantages

1. **Faster Time to Market:** Reusing services across various business processes accelerates application development.
2. **Efficient Maintenance:** Modifying or updating individual services is simpler compared to monolithic applications, minimizing the impact on overall functionality.
3. **Greater Adaptability:** SOA supports the integration of older systems into newer applications, making it adaptable to technological advancements.

## 5.3 Disadvantages

1. **Limited Scalability:** The scalability of the system may be constrained when services share resources and coordination is necessary.
2. **Increasing Interdependencies:** As SOA systems grow, interdependencies between services can complicate modifications and debugging.
3. **Single Point of Failure:** Implementations with an Enterprise Service Bus (ESB) may introduce a centralized point of failure, disrupting communication if the ESB goes down.

# 6 Diferrence PWA, Native an Cross-Platform

An application is software that allows you to exchange information with customers and help them complete specific tasks. The different types of applications, or apps, are based on the development method and internal functionality. Web applications are offered in an internet browser. Users do not have to install them on their devices. For their part, native applications are designed for a specific platform or type of device. The user must install the appropriate software version on the device of their choice. Hybrid apps are native apps with an embedded web browser.

Progressive Web Apps (PWA), native apps, and cross-platform apps are distinct approaches to app development:



Table 1: Summary of Differences: Web Apps vs. Hybrid Apps vs. Native Apps

<b>Features</b>	<b>Web Application</b>	<b>Hybrid Application</b>	<b>Native Application</b>
Use	Users can access directly from a browser	Users need to install the application on their chosen device	Users need to install the application on their chosen device
Internal Operation	Client code in the browser communicates with databases and remote server code	Client and browser code are included in a native container or interpreter	Client code is written in technology and language specific to the device or platform it will run on
Native Device Features	Not accessible	Accessible	Accessible
User Experience	Inconsistent and dependent on the browser used	Consistent and interactive	Consistent and interactive
Access	Limited by browser and network connectivity	One-step access with offline features	One-step access with offline features
Performance	Slow and less responsive	Faster, but may consume more battery	Performance can be optimized according to the device
Development	Cost-effective with a faster time to market	Cost-effective with a faster time to market	Expensive with a slower time to market