**Technological University of Tijuana**
Software Development and Management

# **Title:** Concepts and Characteristics of Progressive Web Applications (PWAs)

**Author:** Jose Eduardo Ruiz Alvarez
**Group:** 10B
**Subject:** Progressive Web Applications

**Date:** February 2, 2024

# Contents

# 1 Introduction

## 1.1 Purpose

The purpose of this document is to define the concepts and characteristics of web and service-oriented applications, native and cross-platform applications, and Progressive Web Apps (PWAs).

## 1.2 Scope

This document covers the investigation of the concepts related to web and service-oriented applications, native and cross-platform applications, and PWAs.

## 1.3 Definitions, Acronyms, and Abbreviations

- **Web Application (Web App):** A software application that runs on a web server and is accessed through a web browser.
- **Service-Oriented Application:** An application designed and structured to provide services to other applications through a network.
- **Native Application:** An application developed for a specific platform or device using the native programming language.
- **Cross-Platform Application:** An application that can run on multiple platforms with a single codebase.
- **Progressive Web App (PWA):** A type of application software delivered through the web, built using common web technologies including HTML, CSS, and JavaScript, intended to work on any platform that uses a standards-compliant browser.

## 1.4 References

1 Mozilla Developer Network (MDN). (2023, December 15). *Aplicaciones Web Progresivas | MDN.* Retrieved from `https://developer.mozilla.org/es/docs/Web/Progressive_web_apps`

2 IBM. (2022, February 10). *IBM Documentation: Service-Oriented Architecture (SOA).* Retrieved from `https://www.ibm.com/docs/en/rbd/9.5.1?topic=overview-service-oriented-`

3 Amazon Web Services, Inc. (2023). *Web Apps vs. Native Apps vs. Hybrid Apps - Difference Between Types of Web and Mobile Applications - AWS.* Retrieved from `https://aws.amazon.com/compare/the-difference-between-web-apps-native-apps-and-hybrid-ap`

## 1.5 Overview of the Document

This document will explore and define the characteristics of web and service-oriented applications, native and cross-platform applications, and PWAs.

# 2 Characteristics of Web and Service-Oriented Applications

## 2.1 Introduction

Web applications are designed for user interaction through web browsers, featuring a user-friendly interface accessible over the internet. They operate in a stateless environment, with each client request being independent. Typically using client-side scripting languages like JavaScript, web apps enhance user interactivity without complete page reloads. Communication between clients and servers primarily occurs through HTTP requests and responses.

Service-oriented applications, on the other hand, focus on breaking down functionalities into modular services that communicate with each other. They leverage services that can be independently developed, deployed, and scaled, promoting a more flexible and scalable architecture. Both types of applications aim to provide efficient and reliable solutions, but their architectures and focuses differ based on their intended use cases.

### 2.1.1 Web Application Characteristics

1. **Accessibility:** Web applications should be accessible from various devices and browsers.
2. **User Interface (UI):** Intuitive and responsive UI for seamless user interaction.
3. **Connectivity:** Reliance on internet connectivity for operation.
4. **Security:** Implementation of security measures to protect user data and transactions.

### 2.1.2 Service-Oriented Application Characteristics

1. **Modularity:** Services are modular, providing specific functionalities.
2. **Interoperability:** Services can interact and work with other services.
3. **Loose Coupling:** Minimal dependencies between services.
4. **Scalability:** Easily scalable by adding or removing services.

# 3 Characteristics of Native and Cross-Platform Applications

## 3.1 Introduction

Native applications are specifically designed for a particular platform, utilizing platform-specific languages and taking advantage of device-specific features. They offer optimal performance and seamlessly integrate with the operating system, providing a native and consistent user experience.

On the other hand, cross-platform applications are developed with a single codebase, making them more cost-effective and efficient in terms of development time. While they aim for a unified user interface, achieving a perfect native look may require additional customization. Cross-platform frameworks provide a streamlined development process, allowing updates and bug fixes to be implemented across multiple platforms simultaneously. Choosing between native and cross-platform development depends on factors such as performance requirements, budget constraints, and the desired level of platform integration.

### 3.1.1 Native Application Characteristics

1. **Platform Optimization:** Developed specifically for a single platform.
2. **Performance:** Leveraging native capabilities for optimal performance.
3. **User Experience:** Tailored user experience according to platform guidelines.
4. **Access to Device Features:** Direct access to device-specific features.

### 3.1.2 Cross-Platform Application Characteristics

1. **Code Reusability:** Codebase can be reused across multiple platforms.
2. **Cost-Efficiency:** Development and maintenance costs are generally lower.
3. **Uniformity:** Consistent user experience across different platforms.
4. **Development Time:** Faster development time due to shared codebase.

# 4 Characteristics of Progressive Web Apps (PWAs)

## 4.1 Introduction

Progressive Web Apps (PWAs) embody a modern approach to web development, providing users with a native app-like experience while maintaining the flexibility and accessibility of the web. They are designed to be cross-platform, ensuring a consistent and responsive user interface across various devices.

A key feature of PWAs is their ability to function offline or in low-network conditions, facilitated by service workers that enable caching of essential assets. PWAs prioritize fast loading times, responsive design, and secure connections through HTTPS.

They deliver app-like interactions, can be added to the home screen, and support push notifications, enhancing user engagement. With automatic updates and linkability, PWAs offer a dynamic and discoverable web experience, bridging the gap between traditional websites and native applications.

1. **Offline Functionality:** Ability to work offline or with a poor internet connection.
2. **Responsive Design:** Adaptation to various screen sizes and devices.
3. **App-Like Experience:** Providing a native app-like experience within a web browser.
4. **Installation Independent:** Can be used without traditional installation but can be installed for convenience.
5. **Push Notifications:** Capability to send push notifications to users.

# 5 Conclusion

In conclusion, Progressive Web Apps (PWAs) stand at the forefront of modern web development, seamlessly blending the advantages of native applications with the versatility of web technologies. Engineered for cross-platform compatibility, PWAs ensure a consistent and responsive user experience across diverse devices.

Noteworthy features, such as offline functionality facilitated by service workers, fast loading times, and secure connections through HTTPS, contribute to their appeal. With app-like interactions, push notification support, and the ability to be added to the home screen, PWAs elevate user engagement to a level comparable to native apps. The automatic update mechanism ensures users have access to the latest version effortlessly.

Moreover, PWAs are discoverable through search engines and shareable via URLs, embodying a linkable and dynamic web experience. As the digital landscape evolves, PWAs emerge as a powerful solution, bridging the gap between traditional websites and native applications, and offering a compelling blend of accessibility, responsiveness, and user engagement.