

Nombre: Toledo Perez Cristian Alejandro

Grupo: 10B

Materia: Aplicaciones Web Progresivas

Actividad: Tarea 2: Development and Execution of PWAs

Docente: Ray Brunet Parra Galaviz

Fecha: 10 de enero de 2024

Contents

1	Introduction	3
1.1	Purpose	3
1.2	System Scope	3
1.3	Definitions, Acronyms, and Abbreviations	3
1.4	References	4
1.5	Overview of the Document	4
2	General Description of the Document	5
2.1	Product Perspective	5
2.2	Product Functions	5
2.3	User Characteristics	5
2.4	Constraints	5
2.5	Assumptions and Dependencies	5
2.6	Future Requirements	6
3	Specific Requirements	7
3.1	External Interfaces	7
3.2	Functions	7
3.3	Performance Requirements	7
3.4	Design Constraints	8
3.5	System Attributes	8
3.6	Other Requirements	8
4	Appendices	9

1 Introduction

Progressive Web Applications (PWAs) revolutionize user experiences by operating seamlessly across devices. These applications hinge on fundamental technical elements like web manifest files, HTTPS security, and service worker integration. While PWAs install like native apps, their functionality may vary across devices and browsers. This document navigates through PWA essentials: technical requisites, installation processes, compatibility considerations, and crucial development tools.

1.1 Purpose

This document aims to provide a comprehensive understanding of Progressive Web Applications (PWAs), focusing on their technical requirements, installation procedures, compatibility nuances, and the essential tools required for their successful development. By exploring these facets, the aim is to equip developers and stakeholders with the knowledge necessary to leverage the potential of PWAs in delivering seamless and engaging user experiences across diverse platforms and devices.

1.2 System Scope

This document encompasses a detailed exploration of Progressive Web Applications (PWAs), delving into their technical architecture, installation mechanisms, compatibility considerations, and a curated selection of pivotal tools and technologies. The scope includes understanding the foundational aspects of PWAs and their practical implementation across various platforms and devices, aiming to assist developers and stakeholders in harnessing the potential of these applications for seamless and versatile user experiences.

1.3 Definitions, Acronyms, and Abbreviations

In this document, the following terms, acronyms, and abbreviations are utilized:

- **PWA:** Progressive Web Application. Web applications employing modern web capabilities to offer a native-like user experience.
- **Web Manifest:** A JSON file providing metadata to the browser about the application, enabling features like 'Add to Homescreen.'
- **HTTPS:** HyperText Transfer Protocol Secure. A protocol ensuring secure communication between the user's browser and the web server.
- **Service Worker:** A script running in the background, facilitating event-driven functionalities and offline capabilities for PWAs.
- **Application Shell:** A foundational structure of a PWA, providing essential UI elements using HTML and CSS, enhancing user experience.
- **Compatibility:** The ability of a PWA to function optimally across various browsers and operating systems.

Polymer, Webpack Module Bundler, Magento PWA Studio, ScandiPWA, PWA Builder, Ionic, Angular, React PWA Library, VueJS, Lighthouse: Various tools and technologies leveraged in the development and optimization of PWAs.

These definitions, acronyms, and abbreviations are crucial for understanding and discussing the components and requirements integral to Progressive Web Applications throughout this document.

1.4 References

References

- [1] Love, C. (2003, February 7). Progressive web application development by example. O'Reilly Online Learning. Retrieved January 9, 2024, from <https://www.oreilly.com/library/view/progressive-web-application/9781787125421/e89ac7bb-8251-41ed-ad3f-6ed793a73b9d.xhtml>
- [2] MDN Web Docs. (2023, November 17). How to make PWAs installable - Progressive Web Apps. Retrieved January 9, 2024, from https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps/Tutorials/js13kGames/Installable_PWAs
- [3] Herrera, I., Herrera, I. (2023, July 27). Compatibilidades de las PWA según el navegador y el sistema operativo. ttandem.com. Retrieved January 9, 2024, from <https://www.ttandem.com/blog/desarrollo-que-son-las-pwa-o-progressive-web-applications/compatibilidades-de-las-pwa-segun-el-navegador-y-el-sistema-operativo/>
- [4] Raghavan, R. (2022, February 18). Best 10 Tools to leverage for Progressive Web App Development. Codemotion Magazine. Retrieved January 9, 2024, from <https://www.codemotion.com/magazine/frontend/web-developer/best-10-tools-to-leverage-for-progressive-web-app-development/#:~:text=React%20PWA%20Library,-The%20ReactJS%20framework&text=ReactJS%2C%20as%20one%20of%20the,ideal%20solution%20for%20building%20PWAs>

1.5 Overview of the Document

This document serves as a comprehensive guide to Progressive Web Applications (PWAs), outlining their essential technical requirements, installation procedures, compatibility considerations, and the key tools and technologies pivotal for successful development. It begins by elucidating the fundamental requisites for PWAs, emphasizing the significance of web manifest files, HTTPS security, and service worker integration. The document then delves into the installation process of PWAs, elucidating how supporting browsers prompt users for installation. Furthermore, it discusses the adaptability of PWAs across various browsers and operating systems, emphasizing their evolving compatibility landscape. Additionally, the document offers insight into ten influential tools and technologies deemed imperative for leveraging PWA development effectively, encompassing frameworks like Polymer, Magento PWA Studio, and Ionic, among others. Overall, this document aims to equip developers and stakeholders with a comprehensive understanding of PWAs, facilitating their seamless integration and harnessing their potential for delivering superior user experiences.

2 General Description of the Document

The document will cover essential aspects such as the technical foundations of Progressive Web Applications (PWAs), their functionalities, user adaptability, constraints posed by varying platforms, assumptions, dependencies, and future improvements. By dissecting these elements, it aims to offer a holistic view of PWAs and their requisites for a comprehensive understanding.

2.1 Product Perspective

A Progressive Web Application (PWA) requires essential technical components for its functionality:

- **Web Manifest File:** Provides metadata to the browser, enabling features like 'Add to Homescreen.'
- **HTTPS Security:** Ensures secure communication between the user's browser and the web server.
- **Service Worker Integration:** Facilitates event-driven functionalities and offline capabilities for PWAs.
- **Application Shell or Common HTML/CSS:** Enhances user experience by providing essential UI elements.

2.2 Product Functions

The functions and capabilities of a PWA include:

- **Installation:** Supporting browsers can prompt users to install the PWA, functioning similarly to native apps.
- **Compatibility:** PWAs adapt to various browsers and operating systems, offering optimal user experiences.
- **Device-Specific Functionality:** The functionalities of a PWA may vary depending on device and browser compatibility.

2.3 User Characteristics

Users can install and use PWAs across different devices and browsers. Initially available on Android, PWAs have expanded to iOS and Windows with certain limitations.

2.4 Constraints

Not all browsers and operating systems provide the same level of compatibility with PWAs. The functionalities of PWAs might be limited based on the device and browser being used.

2.5 Assumptions and Dependencies

Changes in the organizational structure or technical details (like the system's operating system) might necessitate a review and modification of the PWA's requirements.

2.6 Future Requirements

- Enhancements in cross-platform compatibility to broaden accessibility to additional operating systems.
- Integration of emerging technologies to enhance the functionality and performance of the PWA.
- Implementation of new features and characteristics to improve user experience and overall usability

3 Specific Requirements

This section outlines the technical and functional requirements necessary for the development and operation of the Progressive Web App (PWA). The requirements focus on aspects such as external interfaces, key system functions, expected performance, design constraints, system attributes, and other relevant factors essential to ensure the effectiveness, security, and usability of the PWA. The following details these requirements in depth:

3.1 External Interfaces

This section will describe requirements affecting user interfaces, interactions with other systems (hardware/software), and communication interfaces.

- **Web Manifest File:** The system shall provide a web manifest file to enable features like 'Add to Homescreen'.
- **HTTPS Security:** All communications between the user's browser and the web server shall be served securely using HTTPS.
- **Service Worker Integration:** The system must register a service worker with a fetch event handler for event-driven functionalities and offline capabilities.

3.2 Functions

This subsection will specify all system actions or functions.

- **Installation:** Supporting browsers should prompt users to install the PWA, functioning similarly to native apps.
- **Compatibility:** The system should adapt to various browsers and operating systems, offering optimal user experiences.
- **Device-Specific Functionality:** Functionality may vary based on device and browser compatibility.

3.3 Performance Requirements

This section will detail the expected system load and data requirements.

- **User Load Expectations:** The system should handle a simultaneous user load of at least [X number] users without significant performance degradation, ensuring a smooth user experience during peak hours.
- **Response Time:** The PWA must respond to user interactions within [Y seconds] for critical functionalities (like navigation, form submissions) to maintain user engagement and satisfaction.
- **Data Throughput:** The system should support a data throughput of [Z units] per second to manage incoming and outgoing data, ensuring seamless communication between the client and the server.
- **Cache Utilization:** The PWA should effectively utilize caching mechanisms to optimize load times for subsequent visits and ensure efficient offline access.

These requirements outline performance benchmarks essential for the system to deliver a responsive, reliable, and efficient user experience. Adjust the values ([X], [Y], [Z]) based on your project's specific needs and performance goals.

3.4 Design Constraints

Hardware Limitations: The system design should accommodate low-end devices commonly used by entry-level users, ensuring the PWA's functionality and performance on devices with limited processing power and memory. **Standards Compliance:** The system design must adhere to web standards, ensuring compatibility with various browsers and ensuring compliance with industry-standard security protocols such as TLS 1.2/1.3 for HTTPS implementation.

3.5 System Attributes

- **Security:** The system must employ robust security measures, including user authentication through secure login credentials, and data encryption for sensitive user information.
- **Reliability:** The system should have a minimum uptime of 99

3.6 Other Requirements

- **Accessibility:** The system should comply with WCAG 2.1 guidelines to ensure accessibility for users with disabilities, incorporating features like screen reader compatibility and keyboard navigation.
- **Legal Compliance:** The PWA must comply with data protection laws such as GDPR or CCPA, ensuring user data privacy, and provide mechanisms for users to manage their data and consent preferences.

4 Appendices

Your PWA is progressive. It will always deliver the best possible experience given the capabilities of the browser it's running on. The more modern the browser is, the better the experience.



















	MOBILE		DESKTOP		
	 Android	 iOS	 Windows	 macOS	 Linux
 Progressive Web Apps	✓	✓	✓	✓	✓
 Service Worker	✓	✗	✓	✓	✓
 Web push notifications	✓	✗	✓	✓	✓
 Offline browsing	✓	✗	✓	✓	✓
 Background synchronization	✓	✗	✓	✓	✓
 Add to home screen	✓	✗	Soon**	Soon**	--
 App launching screen	✓	✗	--	--	--
 Bluetooth	✓	✗	✗	✓	✗
 Beacons*	✓	✗	✗	✓	✗
 Geolocation	✓	✓	✓	✓	✓
 Geofencing	✗	✗	✗	✗	✗
 Image capture	✓	✓	✓	✓	✓
 Video capture	✓	✓	✓	✓	✓

Figure 1: Browser Support & Compatibility