January 10, 2024

Nombre: Toledo Perez Cristian Alejandro
Grupo: 10B
Materia: Aplicaciones Web Progresivas
Actividad: Tarea 1: What is pwa?
Docente: Ray Brunet Parra Galaviz
Fecha: 10 de enero de 2024

Contents

1 Introduction

In this document, we aim to explore the landscape of web-based applications, service-oriented applications, native applications, and delve deeper into the distinctive characteristics of Progressive Web Apps (PWAs). We will discuss their advantages, disadvantages, and the underlying technologies that power each type of application.

1.1 Purpose

The purpose of this document is to provide an in-depth analysis and comparison between web applications, service-oriented applications, native applications, and Progressive Web Apps. This analysis will shed light on their functionalities, target platforms, and the user experiences they offer.

1.2 System Scope

The scope of this document encompasses a comprehensive examination of web-based applications, service-oriented architecture, native applications, and a detailed exploration of the key features that define Progressive Web Apps.

1.3 Acronyms

- HTML: HyperText Markup Language
- CSS: Cascading Style Sheets
- JS: JavaScript
- PWA: Progressive Web App
- SOA: Service-Oriented Architecture
- UX: User Experience
- API: Application Programming Interface
- GPS: Global Positioning System
- JSON: JavaScript Object Notation
- HTTPS: Hypertext Transfer Protocol Secure
- CPU: Central Processing Unit
- RAM: Random Access Memory
- UI: User Interface
- CLI: Command Line Interface
- SDK: Software Development Kit
- IDE: Integrated Development Environment

1.4 References

References

[1] Gillis, A. S. (2022, December 8) native app. Software Quality.January 9, 2024, from `https://www.techtarget.com/searchsoftwarequality/definition/native-application-native-app`

[2] Fernández, C. (2022, March 18). Cross-platform app development. Characteristics. ABAMobile. January 9, 2024, from `https://abamobile.com/web/cross-platform-app-development-characteristics/`

[3] aws (n.d.). what is SOA? - SOA architecture explained. Amazon Web Services, Inc. January 9, 2024, from `https://aws.amazon.com/what-is/service-oriented-architecture/`

[4] vuestorefront (n.d.). what is PWA? Progressive Web Apps Explained | Vue Storefront. Vue Storefront. January 9, 2024, from `https://vuestorefront.io/blog/pwa`

1.5 Overview of the Document

The document is structured to provide insights into different types of applications prevalent in modern software development. We will delve into the specifics of each category, their strengths, weaknesses, and their implications in today's digital landscape.

2 What is PWA?

A Progressive Web App (PWA) is an application developed using web technologies such as HTML, CSS, and JavaScript, aiming to offer a seamless and engaging User Experience (UX) similar to that of a native application. PWAs leverage modern web capabilities to function across various platforms and devices, employing a single codebase.

2.1 Concepts from pwa

- Service Workers: Service workers represent a fundamental element within PWAs. These are JavaScript-based scripts operating discreetly in the background, decoupled from the web page. Their primary role encompasses managing tasks like caching, push notifications, and offline functionality. Their implementation ensures the PWA's robustness, enabling reliable operation, even in scenarios with limited or no network connectivity.

- Web App Manifest: The web app manifest stands as a JSON file providing crucial metadata about the PWA, comprising details such as its name, description, icons, and designated start URL. This file serves to instruct the browser on how to treat the web app, allowing it to be perceived as an installable application and supporting features such as home screen addition.

- Responsive Design: Responsive design constitutes an integral aspect ensuring the adaptability and aesthetic appeal of PWAs across a spectrum of devices and screen sizes. Commonly utilizing CSS media queries and flexible layouts, this feature guarantees an optimal visual experience.

- HTTPS: The utilization of a secure HTTPS connection is a mandatory requirement for PWAs. It serves as a protective measure ensuring the confidentiality and integrity of data exchanged between users and servers. HTTPS is pivotal for enabling critical PWA functionalities like service workers and push notifications.

- App Shell Architecture: The app shell denotes the minimal HTML, CSS, and JavaScript essential for rendering the foundational user interface of the PWA. Its prompt loading facilitates a swift and dependable initial user experience, with subsequent content loaded dynamically as required.

- Caching Strategies: The implementation of caching mechanisms is a cornerstone of PWAs, facilitating offline functionality and expedited loading times. Developers employ caching strategies through service workers to store and retrieve assets like images, stylesheets, and scripts.

- Offline Capabilities: Designed to ensure uninterrupted user experiences, PWAs adeptly operate even in offline or suboptimal network conditions. This resilience is achieved through the strategic utilization of service workers and caching strategies, enabling seamless functionality sans an active internet connection.

- Push Notifications: Enabling real-time updates and user engagement, push notifications are a pivotal feature of PWAs. Service workers play a key role in managing and displaying these notifications, ensuring user interaction even when the app is not actively accessed.

- IndexedDB and Local Storage: Leveraging technologies such as IndexedDB and local storage, PWAs efficiently store data on the client side. This capability enables data persistence and empowers offline functionality.
- Cross-Browser Compatibility: PWAs are meticulously crafted for compatibility across diverse browsers. Developers adhere to standardized web APIs and features universally supported by modern browsers, ensuring seamless performance irrespective of the browser being utilized.
- Automatic Updates: PWAs exhibit the capability to autonomously update, ensuring users consistently access the latest version without necessitating manual intervention. This self-updating functionality is executed through service workers and a sophisticated cache update strategy.

2.2 What's the difference between Progressive Web Apps and Native Mobile Apps?

Native apps have dominated for a decade, but there is now a shift towards a unified experience across all platforms and devices with Progressive Web Apps (PWAs). PWAs are mobile-first oriented and significantly lighter compared to native apps, while still delivering a high level of interactivity. Although major players like Alibaba, Pinterest, and Twitter have already embraced and implemented PWAs on their sites, the term "Progressive Web App" remained somewhat vague for smaller entities for years. Many of them, recognizing the importance of a mobile-first approach, have been convinced of the necessity of Responsive Web Design or native apps However, awareness of the advantages of PWAs is steadily increasing, as evidenced by Google's statistics and the growing number of live projects using Vue Storefront. This suggests a shifting perception towards these progressive applications.

3 SOA (Service-Oriented Architecture)

Service-Oriented Architecture (SOA) stands as a strategic methodology in software development, employing program elements termed "services" to craft robust business applications. SOA's primary intent lies in the reusability of services across diverse systems or their aggregation to execute intricate tasks. Operating as an architectural paradigm, SOA enables the construction of applications through a consortium of services fostering seamless communication. This interaction could encompass simple data transmission or orchestrate collaborative activities among two or more services. Establishing interconnections among these services remains imperative for SOA's functionality.

3.1 advantages

SOA has the following advantages:

- Service Reusability: Services can be used across various applications, reducing redundancy and improving development efficiency.
- Flexibility: Easily adapts to changes in business requirements by modifying or adding services.
- Interoperability: Facilitates communication between heterogeneous systems, enabling the integration of diverse technologies.
- Scalability: Can be easily scaled by adding new services or replicating existing services as needed.
- Simplified Maintenance: Modifying one service does not impact other parts of the system, facilitating maintenance.

3.2 disadvantages

SOA has the following disadvantages:

- Initial Complexity: Implementing SOA can be complex and require careful planning, potentially increasing initial costs.
- Version Management: Managing service versions can become complicated, especially in environments with multiple applications and services.
- Performance: Communication between services may introduce some latency, affecting performance compared to monolithic architectures.
- Security: Managing security can be challenging, as service communication often occurs over networks.

3.3 Characteristics of SOA

- Reusability: Services can be reused in different applications and contexts, facilitating development efficiency.
- Interoperability: Enables effective communication between different services and applications, even if developed on different platforms.

- Service Discovery: Services can be easily discovered and accessed, facilitating their integration into new applications.
- Abstraction: Services are designed to be independent of underlying implementations, allowing changes in implementation without affecting the service interface.
- Composition: Allows the combination of services to create more complex and functional applications.

4 Native app and Cross-platform

4.1 Native app

A native application refers to a tailored software application meticulously crafted by developers for a distinct platform. Tailored specifically for a particular device and its operating system, native apps harness the distinct hardware and software attributes of the device. These apps offer optimized performance, leveraging the device's unique capabilities, such as GPS integration, in contrast to generic web or mobile cloud applications designed to cater to multiple systems.

4.2 Native app characteristics

- Tailored for a specific platform like Android, iOS, Windows, or Blackberry.
- Developed using a programming language specific to the targeted platform.
- Utilizes the latest mobile device technology, including GPS and camera features.
- Available for download and installation from designated app stores like the Apple App Store or Google Play Store.
- Capable of functioning offline without requiring an internet connection.

4.3 Cross-platform

Cross-platform software stands as a versatile application compatible with multiple operating systems. It isn't bound to a singular platform or device, running seamlessly across Windows, Mac OS X, and Linux environments. Often known as multi-platform software or platform-independent software, cross-platform applications exhibit the following characteristics:

5 Appendices