Types of applications

Higuera Sanchez Dulce Mariela

February 3, 2024

The present document describes the types of applications that exist, mentioning the advantages and disadvantages of each, as well as the operating systems in which they are commonly implemented.

1 Introduction

An application is defined as a computer program designed to perform a specific function or series of functions for the user. These functions can vary widely, from performing simple mathematical calculations to managing complex tasks such as email, photo editing, or internet browsing. Applications can run on a variety of devices, such as computers, smartphones, tablets, smartwatches, and other electronic devices, and can be developed for different operating systems such as iOS, Android, Windows, among others.

2 Native apps

Native applications are designed specifically for a particular platform, such as iOS (for Apple devices) or Android (for Android devices). These applications are developed using the programming languages and development tools recommended by the platform manufacturer.

2.1 Advantages

- Optimized Performance: They are specifically designed for a platform, allowing for fast and efficient performance.
- Full Access to Device Features: They can leverage all hardware and operating system capabilities.
- Superior User Experience: Optimized for a specific platform, they offer a more intuitive and seamless experience.
- Better Ecosystem Integration: They can seamlessly integrate with other system apps and services.

2.2 Disadvantages

- Development Cost and Time: Developing native apps for different platforms can be expensive and time-consuming, requiring separate versions for each operating system.
- Limited Reach: Designed for a specific platform, native apps

may have limited reach and can only be used by users with compatible devices.

- Complicated Maintenance: Maintaining multiple versions of a native app can be complicated, as any updates or bug fixes must be separately implemented on each platform, increasing the workload for developers.

- Learning Curve: Developers need specific knowledge of each platform to create native apps, which may require a steep learning curve and make managing multidisciplinary teams challenging.

3 Non-Native Apps

Also known as hybrid applications, they are software programs developed using standard web technologies such as HTML, CSS, and JavaScript. Unlike native applications, which are designed specifically for a particular platform (such as iOS or Android), non-native applications are designed to run on multiple platforms. These applications are often developed using frameworks such as Apache Cordova, Ionic, or React Native, which allow writing the code once and then packaging it as a native application for each specific platform.

3.1 Advantages

- Faster Development: They can be developed more quickly by sharing much of the base code across platforms.

- Lower Cost: Developing and maintaining a single app for multiple platforms can be more economical than developing separate native apps.

- Flexibility: They can run on multiple platforms, reaching a wider audience.

- Easy Updates: Changes and updates to the base code are automatically reflected across all platforms.

3.2 Disadvantages

- Inferior Performance: Non-native apps may experience inferior performance compared to native apps due to reliance on standard web technologies and the need to run a web container for operation.

- Limited Device Feature Access: Non-native apps may struggle to access certain device features, such as sensors, cameras, or specific hardware, limiting their functionality compared to native apps.

- Suboptimal User Experience: Due to their hybrid nature, non-native apps may offer a less smooth and consistent user experience compared to native apps, especially in terms of performance and responsiveness.

- Dependency on Third-Party Updates: Non-native apps often depend on third-party frameworks and libraries for their operation, which can lead to compatibility and security issues if these tools are not regularly updated or are abandoned by their developers.

4 Cross-Platform Apps

Multiplatform applications are software programs designed to run on multiple platforms, such as iOS, Android, Windows, among others. These applications are developed using a single set of technologies and tools, allowing the same codebase to be used across different operating systems. To develop multiplatform applications, frameworks or development platforms are used to facilitate writing the code once and executing it on different operating systems.

4.1 Advantages

- Shared Codebase: Allows writing and maintaining a single set of code for multiple platforms, reducing development time and costs.
- Wide Reach: Can reach a broader audience by running on multiple operating systems.
- Available Tools and Frameworks: There are numerous frameworks and tools available for cross-platform development, facilitating the development process.
- Simplified Maintenance: Updates and bug fixes can be implemented more efficiently across all platforms.

4.2 Disadvantages

- Performance and Optimization: Due to the need to adapt to different operating systems, cross-platform apps may experience inferior performance and reduced optimization compared to native apps, affecting the user experience.
- Limited Device Feature Access: Often, cross-platform apps may struggle to access certain device features or fully leverage specific hardware, limiting their functionality compared to native apps.
- Design and User Experience Compromises: The need to maintain visual and functional consistency across different platforms may lead to compromises in design and user experience, resulting in a less intuitive and appealing app.
- Development and Maintenance Complexity: While cross-platform development may reduce development time and costs by sharing a codebase, it can also introduce greater complexity in the development and maintenance process, especially as the app grows in size and complexity.

5 Conclusion

Native apps may require more development time and cost by needing separate versions for each platform, non-native apps may experience inferior performance and limitations in accessing certain device features, while cross-platform apps may compromise performance and user experience as they need to adapt to different operating systems, which could result in lesser optimization and functionality.

6 References

Gunka, S. (2016, september). "Apps nativas: ¿Qué son y qué ventajas ofrecen? Studio G. https://gunkastudios.com/apps-nativas-que-son-y-que-ventajas-ofrecen/

Gillis, A. (2023, October 18th). "Definition native app ". TechTarget https://www.techtarget.com/searchsoftwarequality/definition/native-application-native-app

Kotlin, E. (2024, January 22th). "What is cross-platf orm mobile development?". https://kotlinlang.org/docs/cross-platform-mobile-development.html/