APLICACIONES NATIVAS/NO NATIVAS

DISEÑO DE APLICACIONES

RODRIGUEZ CACHO XIMENA C.

UNIVERSIDAD TECONOLOGICA DE TIJUANA

CONTENTS

1. Native Applications: 3
Advantages:.....3
2. Non-Native or Hybrid Applications: 3
Advantages:.....3
3. Multiplatform Applications: 3
REFERENCES.....4

There are different approaches to developing mobile apps, and they can generally be classified into three categories: native apps, non-native apps (also known as web or hybrid apps), and cross-platform apps.

1. Native Applications:

- **Definition:** Native applications are designed and developed specifically for a particular platform or operating system, such as iOS or Android.

Advantages:

- Optimal performance: They take full advantage of the specific features and functions of the platform.
- Full access to operating system APIs.
- User experience consistent with platform design guidelines.

2. Non-Native or Hybrid Applications:

- Definition: These applications are built using standard web technologies (HTML, CSS, JavaScript) and are wrapped in a native container that allows it to run as an application on a mobile device.

Advantages:

- Faster and cheaper development by sharing code between platforms.
- Easier maintenance, since changes are applied to all platforms simultaneously.
- Access to some native APIs through frameworks such as Apache Cordova or React Native.

3. Multiplatform Applications:

- **Definition**: These applications are built using a development framework that allows the development of applications that run on various platforms.

They can be native applications written in a language that is compiled for various platforms or packaged web applications.

- **Advantages:**

- More efficient development, since a large amount of code can be shared between platforms.
- Easier maintenance.
- Some cross-platform applications offer performance close to that of native applications.

Some examples of tools and frameworks for cross-platform development are:

- **React Native:** Uses JavaScript and React to build native applications.
- **Flutter:** Uses Dart and allows the creation of attractive and high-performance user interfaces.
- **Xamarin:** Uses C# and allows the development of native applications for iOS and Android.

Choosing between native, non-native, or cross-platform applications often depends on the specific requirements of the project, the preferences of the development team, and the long-term goals for the application. Each approach has its advantages and disadvantages, and the decision is made considering factors such as performance, costs, development time, and user experience.

REFERENCES

- **[Apple Inc., 2016]** Apple Inc. (2016). Jump right in. https://developer.apple.com/library/archive/referencelibrary/GettingStarted/DevelopiOSAppsSwift/

- **[Biørn-Hansen et al., 2017]** Biørn-Hansen, A., Majchrzak, T. A., y Grønli, T.-M. (2017). Progressive web apps: The possible web-native unifier for mobile development. En International Conference on Web Information Systems and Technologies, volumen 2, pp. 344–351. SCITEPRESS.

- **[Biørn-Hansen et al., 2020]** Biørn-Hansen, A., Rieger, C., Grønli, T.-M., Majchrzak, T. A., y Ghinea, G. (2020). An empirical investigation of performance overhead in cross-platform mobile development frameworks.