APP DESIGN

TSU EN TECNOLOGÍAS DE LA INFORMACIÓN ÁREA DESARROLLO DE SOFTWARE MULTIPLATAFORMA

Activity:
Native, Non-native, and Cross-platform Applications

Teacher:
Dr. Ray Brunett Parra

Student:
Garcia Yañez Yesenia Gpe

4B

1. Native Applications:
   - Definition: Native apps are developed for a specific platform (iOS, Android, etc.) using platform-specific programming languages and tools, such as Swift for iOS or Kotlin/Java for Android.
   - Advantages:
     - Performance: Since they are built specifically for the platform, they offer the best performance and speed.
     - User Experience (UX): Native apps provide a more fluid, intuitive, and responsive user experience, aligning perfectly with platform-specific design guidelines.
     - Access to Device Features: Native apps can fully access device hardware (e.g., camera, GPS, sensors) and platform-specific APIs.
   - Disadvantages:
     - Cost and Time: Developing separate apps for each platform can be expensive and time-consuming.
     - Maintenance: Updates and bug fixes need to be done separately for each platform.
2. Non-native Applications (Web Apps):
   - Definition: Non-native apps are web-based applications that run within a browser or a webview, rather than being installed on the device. They are typically developed using web technologies like HTML, CSS, and JavaScript.
   - Advantages:
     - Platform Independence: Web apps can run on any device with a browser, regardless of the operating system.
     - Easy Updates: Updates and maintenance are handled on the server-side, so users always access the latest version.
     - Cost-Effective: There's no need to develop separate versions for different platforms.
   - Disadvantages:
     - Performance: Web apps tend to be slower and less responsive than native apps, as they rely on the internet connection and browser capabilities.
     - Limited Device Access: Web apps have limited access to device hardware and features, which can limit their functionality.
     - User Experience: The user experience might not feel as integrated or polished as that of a native app.
3. Cross-platform Applications:
   - Definition: Cross-platform apps are developed using frameworks and tools (e.g., React Native, Flutter, Xamarin) that allow a single codebase to be used across multiple platforms, such as iOS and Android.
   - Advantages:
     - Code Reusability: Developers can write the app once and deploy it on both iOS and Android, reducing development time and cost.
     - Faster Development: Cross-platform tools often come with pre-built components, speeding up development.
     - Wider Reach: A cross-platform app can reach a larger audience across both major mobile platforms.
   - Disadvantages:
     - Performance: Cross-platform apps may not perform as well as native apps, especially for complex or resource-intensive tasks.
     - Limited Native Feel: While cross-platform frameworks aim to replicate native experiences, they may fall short in providing the exact user experience of a native app.
     - Platform-Specific Tweaks: Some platform-specific features might require additional code or customization.