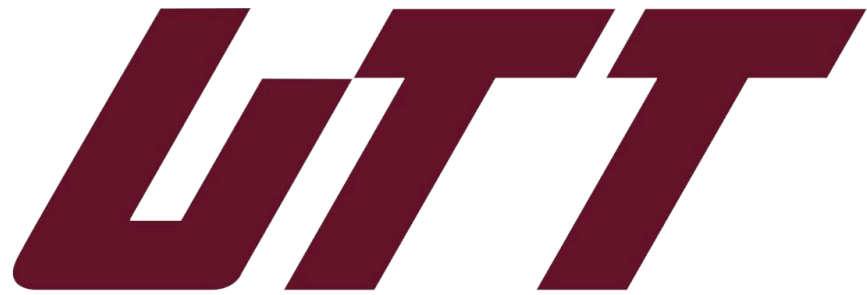


UNIVERSIDAD TECNOLÓGICA DE TIJUANA



Universidad Tecnológica de Tijuana

TSU EN TECNOLOGÍAS DE LA INFORMACIÓN

ÁREA DESARROLLO DE SOFTWARE

MULTIPLATAFORMA

MATERIA:

DISEÑO DE APPS

ESTUDIANTES:

ESTRADA ZARATE DIEGO

DOCENTE:

PARRA GALAVIZ RAY BRUNETT

GRUPO:

4-D BIS

**NATIVE, NON-NATIVE AND CROSS-PLATFORM
APPLICATIONS**

¿Qué son las Native Applications?

Son aquellas diseñadas específicamente para un sistema operativo o plataforma en particular, como **iOS**, **Android**, **Windows**, o **macOS**.

Características principales:

Lenguaje de programación específico: Usan lenguajes y herramientas compatibles con el sistema operativo (por ejemplo, Swift para iOS o Kotlin para Android).

Rendimiento optimizado: Funcionan de manera eficiente ya que están hechas para aprovechar al máximo el hardware y las capacidades del dispositivo.

Acceso completo a características del dispositivo: Como cámara, GPS, almacenamiento, notificaciones push, etc.

Por ejemplo las aplicaciones de IOS está creada por Swift y Xcode y no se puede correr en un sistema operativo android porque solo está hecha para sistema operativo IOS.

¿Qué son las non-native Applications?

Estas aplicaciones no están diseñadas para una plataforma específica. Usualmente, se ejecutan en un entorno externo al sistema operativo, como un navegador son multiplataforma ya que solo se necesita un navegador en el celular, laptop o tablet.

Características principales:

- **Basadas en web o en tecnología de terceros:** Suelen ser apps web o apps híbridas que dependen de tecnologías como HTML5, CSS, y JavaScript.
- **No están completamente optimizadas:** Pueden ser menos eficientes y tener menos acceso a las capacidades avanzadas del dispositivo.
- **Ejemplo:** Una app web progresiva (PWA) que funciona en cualquier dispositivo con navegador como ya mencione, GeoGebra es un buen ejemplo de esto ya que no ocupa instalarse (la versión de web por que existe para android también)

¿Qué son las Cross-Platform Applications ?

Son aplicaciones diseñadas para funcionar en múltiples sistemas operativos (como Android, iOS, Windows, y macOS) utilizando un solo código base. Estas aplicaciones son desarrolladas con frameworks como **Flutter**, **React Native**, o **Xamarin**, que permiten escribir el código una vez y ejecutarlo en distintas plataformas.

Ventajas:

- **Compatibilidad múltiple:** Puedes usar la misma app en diferentes dispositivos y sistemas operativos sin desarrollarla por separado para cada uno.
- **Reducción de costos y tiempo:** Al reutilizar un código base, se ahorra tiempo en desarrollo y mantenimiento.

Desventajas:

- **Rendimiento más lento:** Como el código se traduce en tiempo real para adaptarse al sistema operativo, puede ser menos eficiente que una aplicación nativa.
- **Acceso limitado a funciones avanzadas del dispositivo:** Algunas características específicas del hardware, como sensores o notificaciones avanzadas, podrían no estar completamente optimizadas.

Características principales:

- **Librerías o frameworks específicos:** Usan herramientas como **Flutter**, **React Native**, o **Xamarin**.
- **Ahorro de tiempo y recursos:** Como tienen un solo código base, los desarrolladores no necesitan escribir código separado para cada plataforma.
- **Rendimiento decente:** Aunque no siempre son tan rápidas como las nativas, logran un equilibrio entre compatibilidad y rendimiento.
- **Ejemplo:** Una app creada con Flutter que funciona tanto en Android como en iOS