

Universidad Nacional de La Matanza



Docentes

- Bayarri Brian
- Hoz Aylene

Dinámica de cursada

- Clases presenciales y virtuales - Sábados a las 14 hs.
- Asistencia obligatoria.
- 1 parcial
- 1 recuperatorio
- 1 Trabajo Práctico Integrador

Introducción

- Enfoques posibles
 - Desarrollo “nativo”
 - Desarrollo Híbrido
 - Desarrollo Cross-Platform
 - Desarrollo Multiplataforma

Desarrollo "nativo"

Se programa de acuerdo a los lenguajes, herramientas y APIs propuestos por Google (Android) y Apple (iOS)



Desarrollo "nativo"

- Método más popular de desarrollo.
- Las apps tienen la mejor calidad posible, siempre y cuando se utilicen los SDKs adecuadamente.
- Siempre vamos a contar con actualizaciones, documentación y soporte.
- Soporte *instantáneo* para utilizar nuevos features de los sistemas operativos.
- Desventaja: Hay que mantener 2 código fuente distintos, ya que no se comparte nada de código entre plataformas.

Desarrollo Híbrido

- Ionic
- Cordova

Desarrollo Híbrido

- Combina técnicas de desarrollo nativo con programación web.
- Las aplicaciones son en realidad una **WebView** de la plataforma que carga contenido Web.
- Se desarrollan utilizando Javascript, HTML, CSS.
- Se puede acceder al hardware del dispositivo (cámara, sensores, etc)
- La desventaja más importante de este enfoque es el **rendimiento**, que depende de la implementación del **WebView**.

Desarrollo Híbrido

- Si tu producto ya tiene una versión web...
- Ventaja: una web puede actualizarse sin lanzar una nueva versión de las app a las tiendas.
- Ejemplos: General Electric, Electronic Arts...

Desarrollo Cross-Platform

- React-Native
- Xamarin
- Flutter
- Unity3D

Desarrollo Cross-Platform

- El enfoque es “programar una vez, ejecutar en todos lados”.
- Estas técnicas son la tendencia creciente en el mercado.
- Es posible abrir el proyecto en forma **nativa** e implementar algunas partes específicamente con los SDK oficiales.
- No es necesario tener “un equipo de desarrollo por plataforma”.
- Desventaja: El soporte para nuevos features de los sistemas operativos es más lento.
- Ejemplos: Facebook, Google, Tesla, Pinterest, Skype, eBay, Disney...

Desarrollo Multiplataforma

- Kotlin Multiplatform (KMP) y Kotlin Multiplatform Mobile (KMM).

Desarrollo Multiplataforma

- El enfoque es compartir la lógica de negocio y datos.
- Las interfaces gráficas se desarrollan con alguno de los otros enfoques (preferentemente nativo).
- Ventaja: Lo mejor de los dos mundos. Poca repetición de código, mejor rendimiento en la UI.
- Otros equipos podrían participar del desarrollo móvil. Ejemplo: desarrollando la lógica de los requests/responses.
- Desventaja: Es la menos madura de las técnicas de desarrollo.
- Ejemplos: Phillips, Netflix, VMWare, CashApp

Cuestiones generales

¿Cuál elegir?

Influyen muchos factores

- Necesidad de negocio.
- Mercado al que se apunta.
- Conocimiento previo del equipo.
- Escasez de recursos.
- Necesidad técnica real.
- Y más...

Experiencia previa

- Caso AirBnb (Nativo -> React Native -> Nativo)
- Caso LinkedIn (Híbrido -> Nativo)
- Caso Facebook (Híbrido -> Nativo -> React Native)
- Caso The Podcast App (Nativo -> Nativo)
- Caso NuBank (Nativo -> Cross-Platform)

¿Qué vamos a aprender este
cuatrimestre?

Kotlin Multiplatform Mobile

- ... y nociones de desarrollo iOS.
- Dificultad para enseñar iOS.
- Ventaja de los que ya manejan Kotlin.
- Ventaja competitiva: saber desarrollo nativo **y** algún otro enfoque.

¿Preguntas?

¡Muchas gracias!