



# Enfoques de Desarrollo de Aplicaciones Móviles Multiplataforma

*Mg. Pablo Thomas*

[pthomas@lidi.info.unlp.edu.ar](mailto:pthomas@lidi.info.unlp.edu.ar)

# Cronograma

Clase	Fecha	Contenidos/Actividades
1	20-08-2025	Introducción al Desarrollo de Aplicaciones Móviles
2	27-08-2025	Aplicaciones Nativas – Android + Kotlin
3	03-09-2025	Aplicaciones Nativas – Android + Kotlin
4	10-09-2025	Aplicaciones Interpretadas – React Native
5	17-09-2025	Aplicaciones Interpretadas – React Native
6	24-09-2025	Aplicaciones Híbridas - Ionic
7	01-10-2025	Aplicaciones Híbridas – Ionic
8	08-10-2025	PWA
9	22-10-2025	Aplicaciones generadas por Compilación Cruzada – Net Maui
10	29-10-2025	Aplicaciones generadas por Compilación Cruzada - Net Maui
11	05-11-2025	BD NoSQL
12	12-11-2025	Uso de BD en Apps
		<b>Examen: 03-12-2025 - Trabajo Final Promoción a definir</b>

## Enfoques de Desarrollo de Aplicaciones Móviles Multiplataforma

Curso

Buen comienzo del curso. Esperamos sea de su agrado.

Inicio	▼
<b>Bienvenida</b>	
Itinerario	▼
Propuesta	
Materiales y Actividades	6
Comunicación	▼
Mensajería	52
Cartelera de Novedades	2
Foro de Discusion	0
Colaboración	▼
Compartir Archivos	0

# Agenda

- “Software”
- Introducción al desarrollo de Apps
- Aspectos de Ingeniería de Software
- Tipos de aplicaciones móviles
- Emuladores

# Software

- Alma y cerebro de una computadora
- El **conocimiento** capturado acerca de un área de aplicación
- Colección de los programas y datos necesarios para convertir a una computadora (de propósito general) en una máquina de propósito especial diseñada para una aplicación particular.
- Información (documentación) producida durante el desarrollo de un sistema software-intensivo.

# ¿Por qué el software es único?

- Es intangible → dificulta su control y medición
- Posee un alto contenido intelectual
- No se lo reconoce como un activo contable
- Su proceso de desarrollo es mano de obra intensivo, basado en equipos y por proyectos → multiplica el problema de la comunicación, un aspecto clave de los proyectos
- Generalmente no hay separación entre R&D y producción
- Potencialmente es modificable hasta el infinito → genera problemas a la hora de la evolución del software liberado
- No se degrada
- No hay costo de reproducción

En resumen el Software es

**CONOCIMIENTO EMPAQUETADO**

# **Es fácil construir Software ?**

La respuesta es....



# Desarrollo de Software, una actividad compleja



# Agenda

- “Software”
- **Introducción al desarrollo de Apps**
- Aspectos de Ingeniería de Software
- Tipos de aplicaciones móviles
- Emuladores

# Introducción

- Con el creciente desarrollo tecnológico cada vez más personas acceden a dispositivos móviles.
- Han evolucionado constantemente en capacidades de servicio, ofreciendo funciones de comunicación y procesamiento de datos, que van mucho más allá que las simples llamadas telefónicas.



# Introducción

Un “**dispositivo móvil**” es un término que abarca una amplia gama de aparatos electrónicos surgidos en los últimos años, que se caracterizan por:

- tamaño reducido,
- su conectividad,
- memoria limitada,
- capacidad de procesamiento y almacenamiento de datos.

Por ejemplo: teléfonos móviles más evolucionados (smartphones), smartwatch, Tablet, etc.



# Introducción



LOS DISPOSITIVOS HAN EVOLUCIONADO

# Introducción

Los dispositivos móviles forman parte de la vida cotidiana y son cada vez más sofisticados.

Es evidente el crecimiento de la telefonía celular y otros dispositivos móviles como los GPS's, tablets, , etc





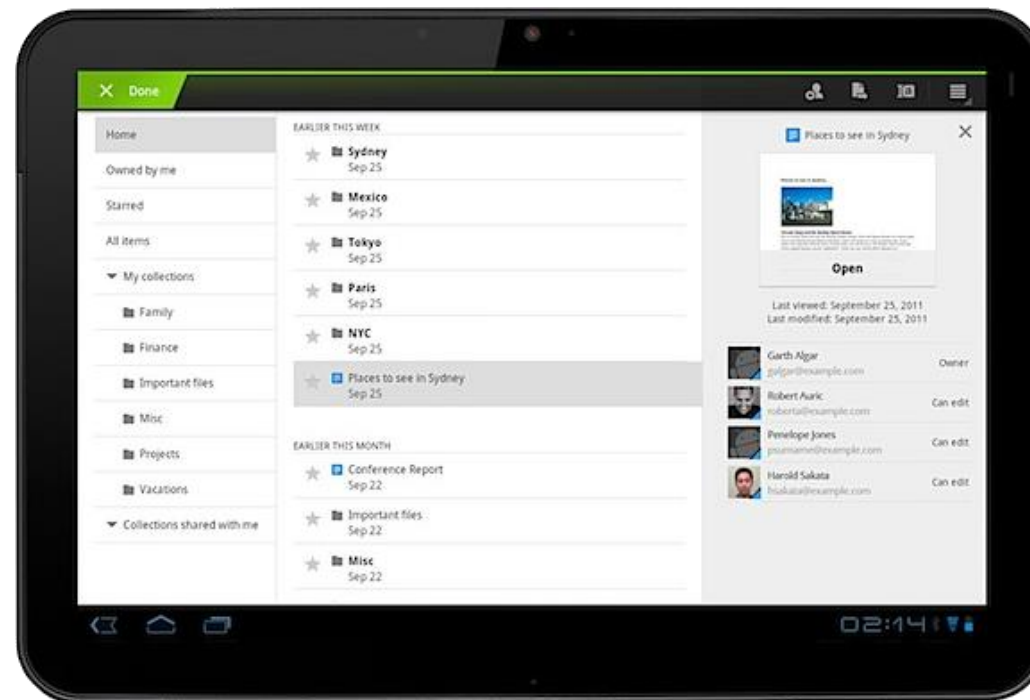
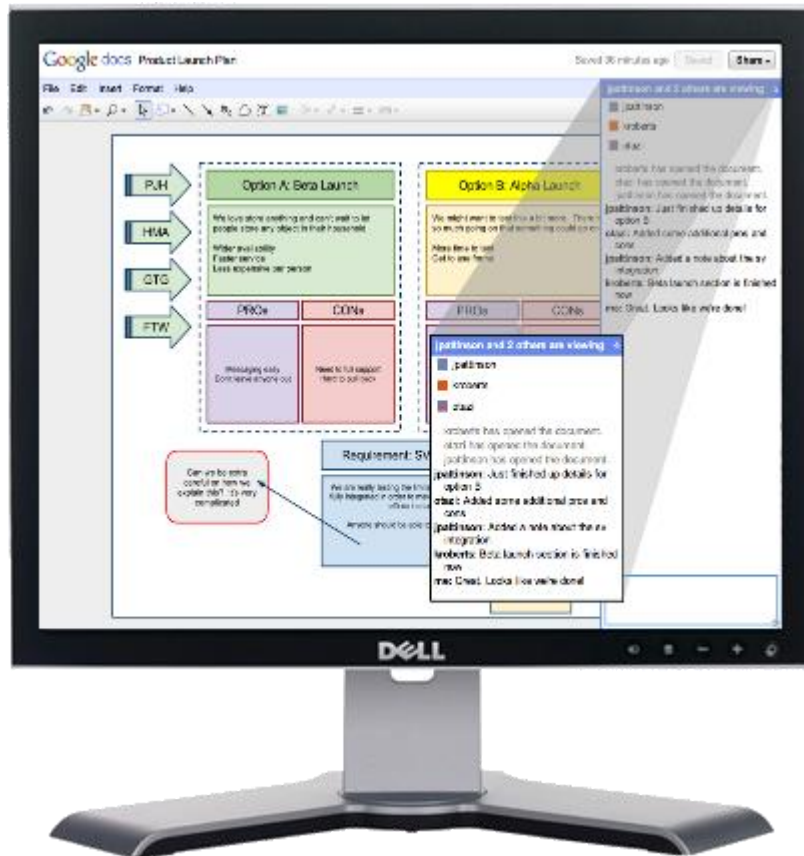
# Introducción

APLICACIONES EN DISTINTOS  
DISPOSITIVOS



# Introducción

## APLICACIONES EN DISTINTOS DISPOSITIVOS



Google docs



# Introducción

En la comunicación humana, al expresarnos en una conversación normal:

- hablamos (sonido),
- escribimos (texto),
- observamos (vídeo)
- gesticulamos (animación)

Todo esto se ha trasladado al campo de los dispositivos móviles ofreciéndonos múltiples aplicaciones.

# Introducción

## Ventajas

- Movilidad
- Conectividad
- Aplicaciones
- Extensión de nuestra computadora
- GPS – Cámara – Video Cámara – Micrófono
- Reconocimiento de voz

# Introducción

## Desventajas

- Pérdida de señal
- Limitaciones de interfase → Limitaciones de usabilidad
- Las aplicaciones pueden hacer que el dispositivo funcione más lentamente para sus funciones básicas
- Limitaciones de batería
- Pérdida, rotura del dispositivo, etc.
- Industria fragmentada: variedad de estándares, protocolos y tecnologías

# COMPUTADORAS

Actividades de larga duración

Actividades de exploración

Casa / Dpto



# DISPOSITIVOS MÓVILES



Actividades de corta  
duración

Actividades con foco

Lugar en movimiento

# ASPECTOS A TENER EN CUENTA

## 1 O 2 MANOS



# ASPECTOS A TENER EN CUENTA DÓNDE SE VA A UTILIZAR





# ASPECTOS A TENER EN CUENTA

# TIEMPO DE RESPUESTA



# Agenda

- “Software”
- Introducción al desarrollo de Apps
- **Aspectos de Ingeniería de Software**
- Tipos de aplicaciones móviles
- Emuladores



# Aspectos de IS

En muchos aspectos, el desarrollo de aplicaciones móviles es similar que utilizar la Ingeniería de Software para el desarrollo de aplicaciones clásicas.

¿Qué tienen en común?

Integración con dispositivos de hardware, así como las propiedades tradicionales de seguridad, performance, confiabilidad y limitaciones de almacenamiento.

# Aspectos de IS

- Potencial interacción con otras aplicaciones
- El manejo de sensores (acelerómetro, cámara, micrófono, etc)
- Familias de plataformas de hardware y de software
- Seguridad
- Interfaces de usuario

# Aspectos de IS

Algunos autores sostienen que la movilidad representa una “crisis” total para todo lo asumido y estable en la IS.

Las limitaciones de la IS incluyen:

- la maximización y el balance entre el poder de procesamiento del dispositivo y la carga de batería,
- el tamaño del dispositivo así como el tamaño de su pantalla

# Aspectos de IS

El desarrollo de aplicaciones móviles está caracterizado por 3 factores:

- 1) La madurez de la infraestructura de redes móviles
- 2) Los avances del hardware móvil
- 3) La creciente “demanda” de aplicaciones móviles

# Requerimientos No Funcionales más relevantes

- **Performance** (uso eficiente de los recursos del dispositivo, capacidad de respuesta, escalabilidad)
- **Tamaño**
- **Consumo de batería**
- **Confiabilidad** (robustez, conectividad, estabilidad)
- **Seguridad**
- **Usabilidad**

# Requerimientos no funcionales más relevantes

## Usabilidad

- Completar una tarea en un contexto específico (eficacia)
- Completar una tarea con velocidad y precisión (eficiencia)
- Nivel de confort y agrado percibido al usar la aplicación (satisfacción)
- Facilidad de aprendizaje
- Capacidad de memorizar como usar eficientemente una aplicación
- Cantidad de procesamiento cognitivo requerido por el usuario para usar la aplicación (carga cognitiva)

# Alternativas de desarrollo

- Desarrollar solamente para una plataforma
- Desarrollar aplicaciones nativas para cada plataforma
  - Android
  - iOS
- Desarrollar aplicaciones web para móviles
- Utilizar una o más capas de abstracción que puedan mapear “lo programado una sola vez” en programas nativos que se ejecuten sobre múltiples plataformas

# Agenda

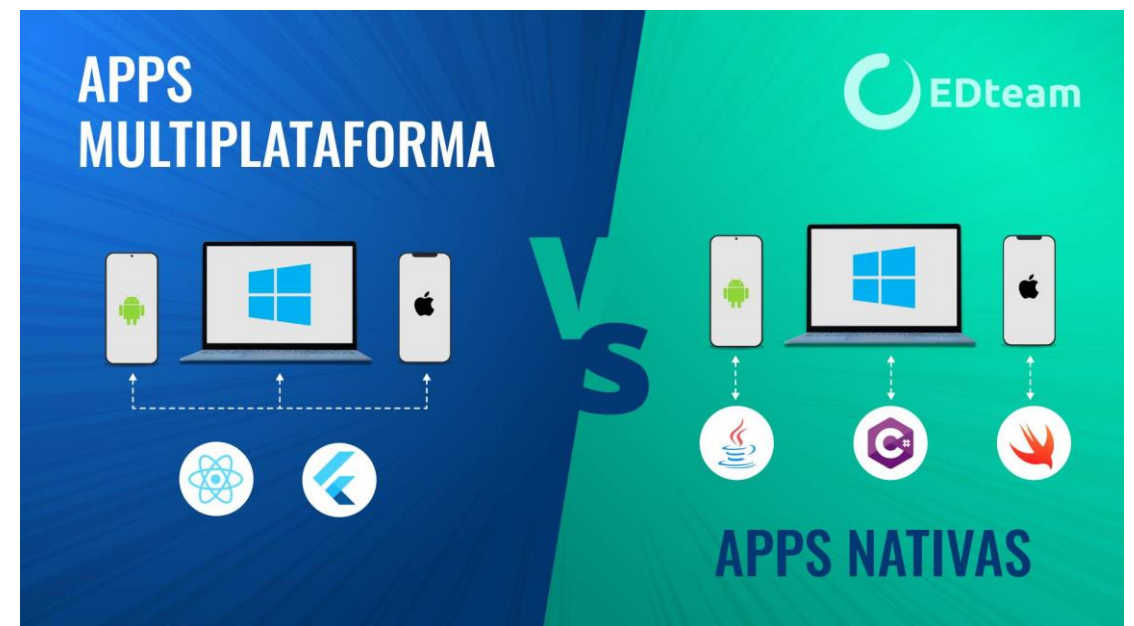
- “Software”
- Introducción al desarrollo de Apps
- Aspectos de Ingeniería de Software
- **Tipos de aplicaciones móviles**
- Emuladores



# Tipos de Aplicaciones Móviles

## Enfoques de desarrollo de aplicaciones móviles

- Nativo
- Multiplataforma
  - Sitios web móvil - PWA
  - Aplicaciones híbridas
  - Aplicaciones interpretadas
  - Aplicaciones generadas por compilación cruzada





# Aplicaciones Nativas



- Ejecutadas en un dispositivo y sistema operativo específico.
- El código fuente de estas aplicaciones se escribe en función del dispositivo.
- El código fuente se compila a un ejecutable. Es un proceso similar al de las tradicionales aplicaciones de escritorio.
- Cuando la aplicación está lista para ser distribuida, es subida a las App stores (tiendas de aplicaciones) específicas de cada dispositivo.
- Una vez subida la aplicación, las App stores tiene un proceso de auditoría para evaluar si se adecúa a los requerimientos del sistema.

# Aplicaciones Nativas

## Ventajas

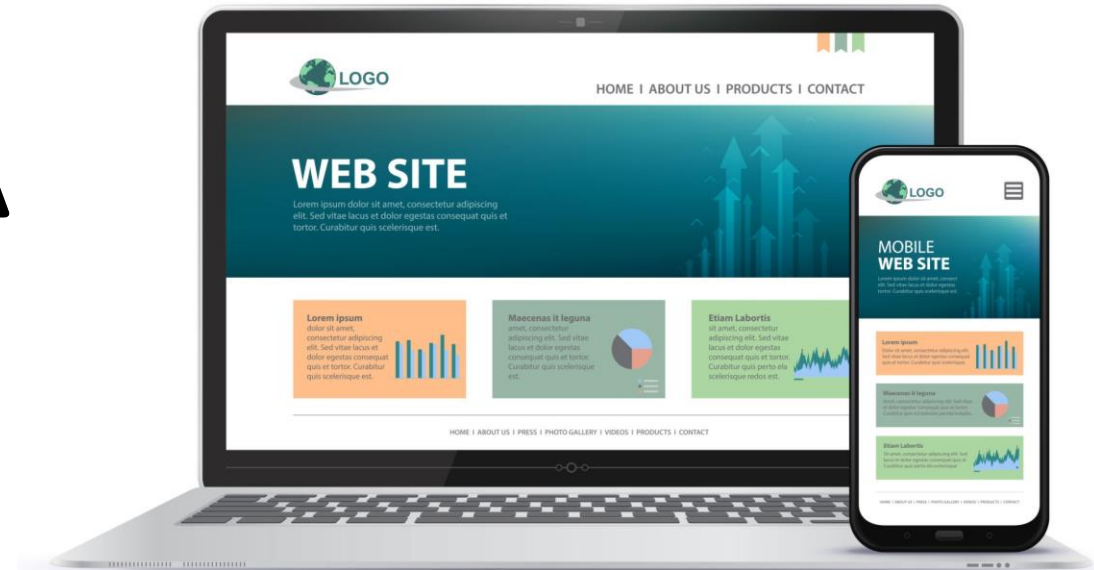
- Aplicación integrada al dispositivo, que puede interactuar con otras aplicaciones del dispositivo.
- Acceso total a todas las utilidades del dispositivo: almacenamiento, cámara, GPS, acelerómetro, agenda, etc.
- Puede ejecutarse off line.
- Una aplicación puede utilizar toda la pantalla del dispositivo y proporcionar una mejor interfaz al usuario (consistente con el resto de las app del dispositivo)
- La aplicación puede correr en background. Recuperar información de un servidor en background y notificar al usuario cuando la información esté disponible

# Aplicaciones Nativas

## Desventajas

- Cada plataforma tiene su propio lenguaje de programación
- El desarrollo es más costoso que el desarrollo de un sitio web móvil
- Las aplicaciones no están disponibles para ser descargadas en todos los dispositivos móviles
- Es más compleja la actualización de la aplicación
- Fragmentación del mercado → desarrollar para una plataforma puede significar quedar afuera de un alto % del mercado
- Fragmentación interna → Por ej: Android tiene un gran % del mercado pero actualmente existe varias versiones en uso.
- Para que un usuario pueda utilizar una aplicación primero debe descargársela, lo que requiere un mayor conocimiento y lealtad a la marca

# Sitios Web Móvil / PWA



## Ventajas

- No es necesaria una instalación (Excepto PWA). Se utiliza el navegador del dispositivo.
- Es accesible por todos los dispositivos con acceso a Internet.
- Las actualizaciones pueden realizarse en cualquier momento y estar disponibles de inmediato.
- Requiere menos recursos que una aplicación no web.
- Es fácil y rápido poner en marcha
- El proceso de desarrollo es más sencillo → emplean tecnologías ya conocidas como HTML , Javascript y CSS
- No necesitan de la aprobación de ningún fabricante para ser publicados

# Sitios Web Móvil

## Desventajas

- Algunos de los navegadores móviles son extremadamente simples
- Nunca será tan complejo y dinámico como una aplicación nativa
- Un sitio web móvil será menos atractivo
- Necesita conexión a internet para funcionar
- Los tiempos de respuesta decaen por la interacción cliente-servidor

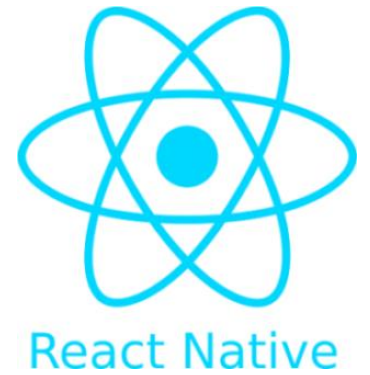


# Aplicaciones Híbridas

- Utilizan tecnologías web HTML5, Javascript y CCS como los sitios web móvil, pero con la posibilidad de acceder a las capacidades del dispositivo.
- Son ejecutadas sobre un web container del dispositivo , el cual tiene acceso a las capacidades específicas de éste, a través de una API.
- Pueden ser distribuidas a través de las tiendas de aplicaciones y ser instaladas.
- La experiencia de usuario se ve perjudicada al no utilizar componentes nativos en la interfaz.
- Frameworks populares: Phonegap, Ionic, etc.



# Aplicaciones Interpretadas



- El código de las aplicaciones se traduce en su mayor parte a código nativo mientras el resto es interpretado en ejecución.
- Se implementan de forma independiente de las plataformas utilizando diversas tecnologías y lenguajes, tales como Java, Ruby y XML, entre otros.



- Se obtienen interfaces totalmente nativas
- Appcelerator Titanium, Native Script, React son entornos de desarrollo populares para este enfoque de desarrollo





# Aplicaciones Generadas Por Compilación Cruzada



- Los proyectos multiplataformas se compilan de manera nativa creando una versión específica para cada plataforma destino.
- Según el entorno de desarrollo con el que se trabaje se pueden encontrar distintos niveles de reutilización de código entre las plataformas.
- Se pueden utilizar controles nativos sobre cada plataforma, posibilitando crear apps que son similares que las aplicaciones nativas.
- Ejemplos de entornos son Applause, Embarcadero Delphi XE6, Xamarin y .net MAUI

# Agenda

- “Software”
- Introducción al desarrollo de Apps
- Aspectos de Ingeniería de Software
- Tipos de aplicaciones móviles
- **Emuladores**

# Emuladores



Software que permite ejecutar programas en una plataforma (sea una arquitectura de hardware o un sistema operativo) diferente de aquella para la cual fueron escritos originalmente.

A diferencia de un simulador, que sólo trata de reproducir el comportamiento del programa, un emulador trata de modelar de forma precisa el dispositivo de manera que este funcione como si estuviese siendo usado en el aparato original.

# Emuladores

## Ventajas

- Rápido diseño, prueba, depuración.
- Fácil de cambiar algo y probar
- Diversos emuladores en una sola PC, sin necesidad de poseer los dispositivos móviles
- Se pueden resetear facilmente

# Emuladores

## Desventajas

- Usabilidad → El mouse y el teclado no ayudan. Escribir o apuntar con el mouse no es lo mismo que con los teclados de los teléfonos.
- El tamaño de la mano no se puede representar en una computadora.
- Es necesario el emulador específico del dispositivo a utilizar

# Bibliografía

- Anthony Wasserman , Carnegie Mellon Silicon Valley, **Software Engineering Issues for Mobile Application Development**, 2º Annual Workshop on Software Engineering for Mobile Application Development, MobiCASE ' 11, Santa Monica, California, USA, October 2011.
- An Nosseir, Derek Floord, Rachel Harrison y Osman Ibrahim, **Mobile Development Process Spiral**, 2012 7th International Conference on Computer Engineering & Systems (ICEES 2012), El Cairo, Egipto, Noviembre 2012
- Lisandro Delía, N.Galdamez, Pablo Thomas, Leonardo Corbalan, P. Pesado, **Multi-platform mobile application development analysis**, 2015 IEEE 9th International Conference on Research Challenges in Information Science (RCIS), Atenas, Grecia
- Lisandro Delía, N.Galdamez, Pablo Thomas, Leonardo Corbalan, P. Pesado, **Approaches to Mobile Application Development:Comparative Performance Analysis**, Computing Conference 2017, 18-20 July 2017, London UK.