

Módulo 8

Código: 489

Programación multimedia y dispositivos móviles

Técnico Superior en Desarrollo de
Aplicaciones Multiplataforma



Análisis de tecnologías para aplicaciones en dispositivos móviles

Contenidos teóricos



1. Objetivos generales de la unidad
2. Competencias y contribución en la unidad
3. Contenidos conceptuales y procedimentales
4. Evaluación
5. Distribución de los contenidos
6. Sesiones

1. Objetivos generales de la unidad

Objetivos curriculares

1. Describir la evolución, características y limitaciones de los dispositivos para el desarrollo de aplicaciones móviles.

2. Dar una visión global de la Computación Ubicua.

3. Analizar las tecnologías disponibles, los lenguajes y los entornos integrados de trabajo y compilación.

4. Proporcionar los conceptos básicos y las herramientas que permitan al alumno adquirir las competencias necesarias para trabajar con los distintos entornos de desarrollo de aplicaciones para dispositivos móviles.

2. Competencias y contribución en la unidad



Analizando las tecnologías para crear aplicaciones en dispositivos móviles, seleccionando la más adecuada para nuestro cometido, instalando el IDE correspondiente y configurando el emulador y el entorno de una manera correcta.

3. Contenidos

| CONTENIDOS CONCEPTUALES | CONTENIDOS PROCEDIMENTALES |
|---|---|
| <ul style="list-style-type: none">• WAP• UMTS• IDE• Emulador• SDK• NFC | <ul style="list-style-type: none">• Limitaciones que plantea la ejecución de aplicaciones en dispositivos móviles• Sistemas operativos para dispositivos móviles. Características• Entornos integrados de trabajo• Módulos para el desarrollo de aplicaciones móviles.• Emuladores. Configuraciones. Tipos y características• Ciclo de vida de una aplicación• Modificación de aplicaciones existentes• Utilización del entorno de ejecución del administrador de aplicaciones |

4. Evaluación

a) Se han analizado las limitaciones que plantea la ejecución de aplicaciones en los dispositivos móviles.

b) Se han identificado las tecnologías de desarrollo de aplicaciones para dispositivos móviles.

c) Se han instalado, configurado y utilizado entornos de trabajo para el desarrollo de aplicaciones para dispositivos móviles

d) Se han identificado configuraciones que clasifican los dispositivos móviles en base a sus características.

4. Evaluación

e) Se han descrito perfiles que establecen la relación entre el dispositivo y la aplicación.

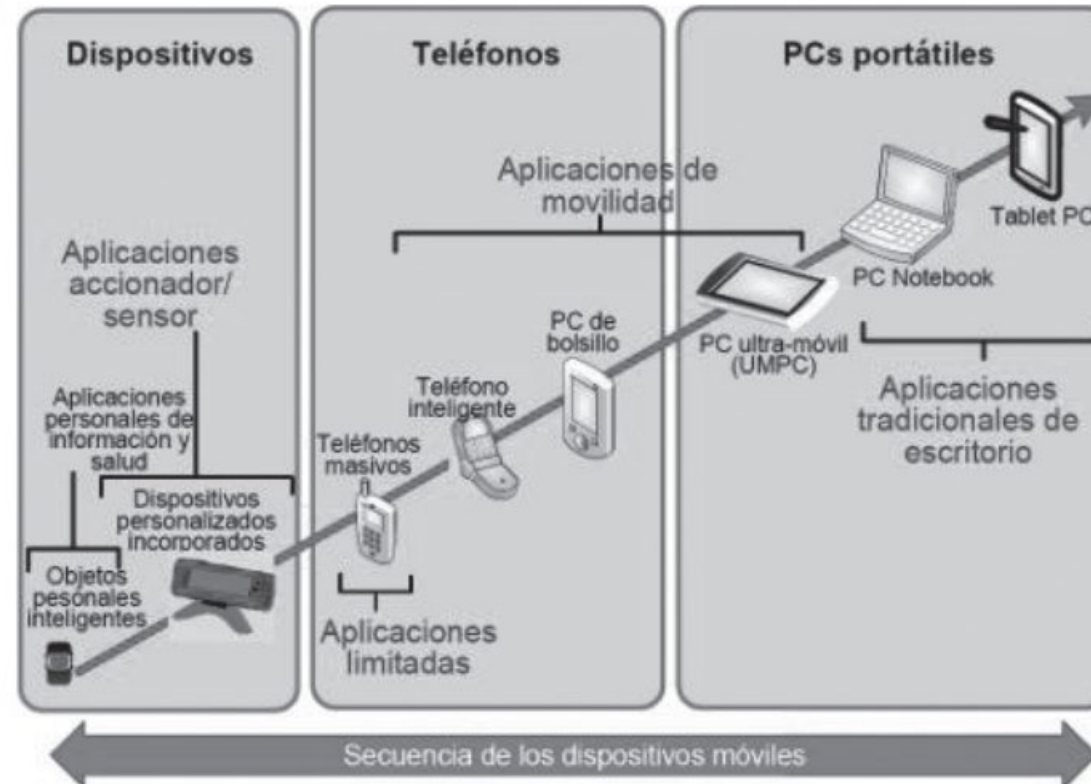
f) Se ha analizado la estructura de aplicaciones existentes para dispositivos móviles identificando las clases utilizadas

g) Se han realizado modificaciones sobre aplicaciones existentes.

h) Se han utilizado emuladores para comprobar el funcionamiento de las aplicaciones.

5. Contenidos teóricos: Sesión 1

1. INTRODUCCIÓN



5. Contenidos teóricos: Sesión 1

2. COMPUTACIÓN UBICUA

Mark Weiser en 1991.

Cuatro paradigmas fundamentales:

- **Descentralización:** paso de potencia centralizada a potencia cliente-servidor.
- **Diversificación:** mismo servicio en distintos dispositivos.
- **Conectividad:** nuevos estándares: WAP, UMTS, Bluetooth...
- **Simplicidad:** interfaces de usuario intuitivas.

Además:

- Conectividad 24/7.
- Buena infraestructura.
- Acceso a la nube.
- Desarrollo de aplicaciones móviles.

5. Contenidos teóricos: Sesión 1

3. LIMITACIONES EN LA EJECUCIÓN DE APLICACIONES PARA DISPOSITIVOS MÓVILES

Una aplicación móvil no es adaptación de aplicación de escritorio a una pantalla pequeña.

Limitaciones existentes:

- **Capacidad de procesamiento y memoria:** evitar sobrecarga de elementos multimedia.
- **Tamaño de pantallas e iluminación:** los datos se insertan con una mano, en movimiento... Hay que facilitar la entrada de datos, navegación...
- **Interfaces:** han de ser intuitivas y naturales. Agrupar información por jerarquía, no mostrar información innecesaria...

5. Contenidos teóricos: Sesión 1

3. LIMITACIONES EN LA EJECUCIÓN DE APLICACIONES PARA DISPOSITIVOS MÓVILES

Limitaciones existentes:

- Batería limitada.
- Tarifa de datos: optimizar transacciones de datos.
- Otras funciones del teléfono: llamadas, alarmas... más prioridad. Mantener el estado del programa.
- Movilidad geográfica: distintas ubicaciones, mala cobertura...
- Emuladores vs equipos reales

5. Contenidos teóricos: Sesión 1

4. TECNOLOGÍAS DISPONIBLES

iOS:

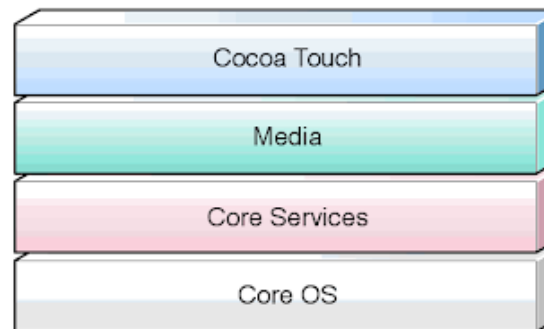
Nacimiento con el iPhone en 2007 (iPhone OS). Más adelante, iPod, iPad...

En 2008, lanzamiento del SDK y Apple Store.

Lenguaje de programación: Objective-C o Swift.

Arquitectura:

- **Cocoa Touch**: frameworks fundamentales de desarrollo.
- **Media**: servicios gráficos y multimedia
- **Core Services**: servicios fundamentales que usan todas las aplicaciones
- **Core OS**: características de bajo nivel: ficheros del sistema, manejo de memoria, seguridad, drivers del dispositivo...



5. Contenidos teóricos: Sesión 1

4. TECNOLOGÍAS DISPONIBLES

Android:

Desarrollado por Open Handset Alliance (OHA)

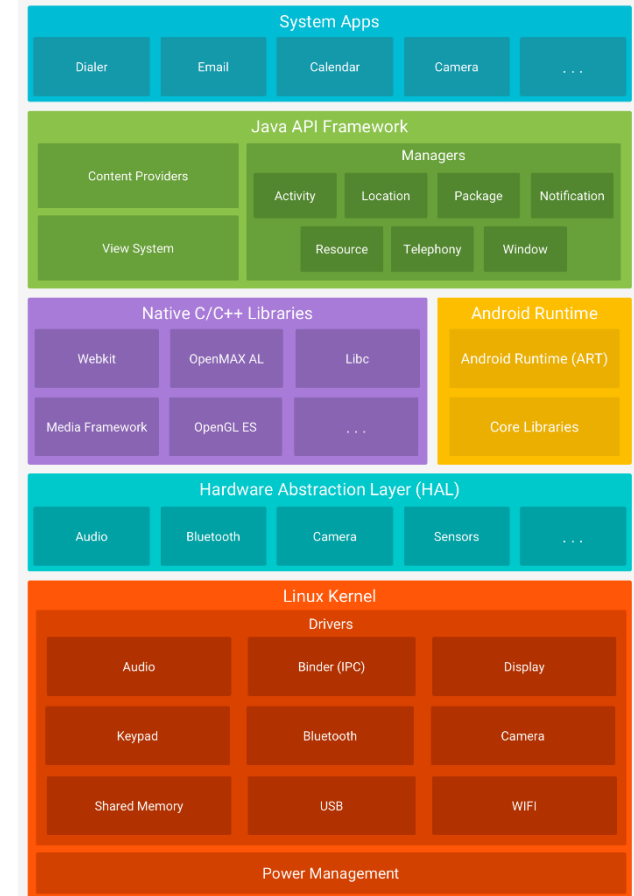
78 compañías, lideradas por Google.

Lanzado en 2007.

Lenguaje de programación: Java, C++, Kotlin

Consiste en:

- Sistema Operativo (basado en Linux).
- Entorno de ejecución de aplicaciones basado en Java.
- Conjunto de librerías de bajo y medio nivel (accesibles por las aplicaciones).
- Conjunto inicial de aplicaciones para el usuario final.
- Arquitectura basada en capas



5. Contenidos teóricos: Sesión 1

4. TECNOLOGÍAS DISPONIBLES

Otros:

Ubuntu for Android

- Variante de Ubuntu diseñado para funcionar en dispositivos móviles.
- Actualmente sin desarrollo.



Firefox OS

- Sistema operativo basado en HTML5 con núcleo de Linux.
- Permitía a aplicaciones HTML5 comunicarse directamente con el hardware del dispositivos.
- Descontinuado.

5. Contenidos teóricos: Sesión 1

4. TECNOLOGÍAS DISPONIBLES

Otros:

Tizen

- Creado por Samsung, basado en Linux.
- <https://www.tizen.org/>

Windows 10 Phone

- Sucesor de Windows 8.1 Phone
- Descontinuado en 2017



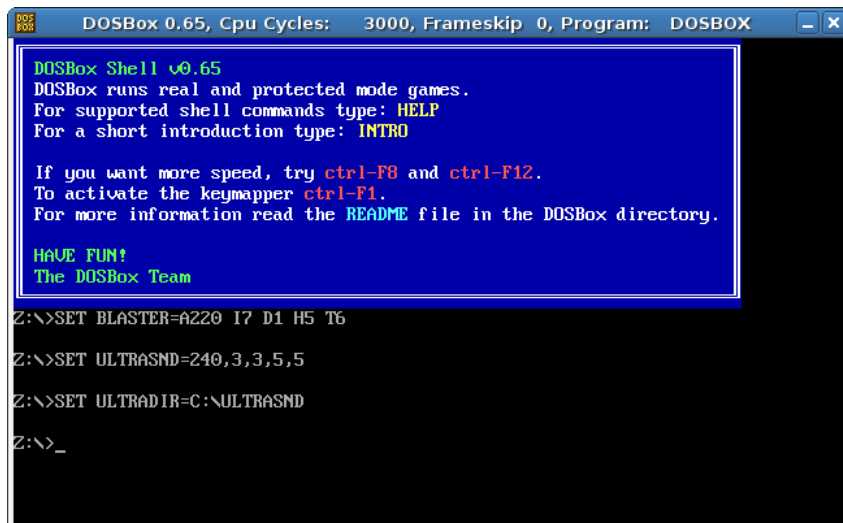
5. Contenidos teóricos: Sesión 2-4

5. EMULADORES

Software que permite ejecutar programas de una plataforma específica en otra para la que no fueron escritos originalmente.

Usos:

- **Recreativo:** emulación de videoconsolas, ordenadores antiguos...
- **Desarrollo:** emulación de los programas escritos para otra plataforma.



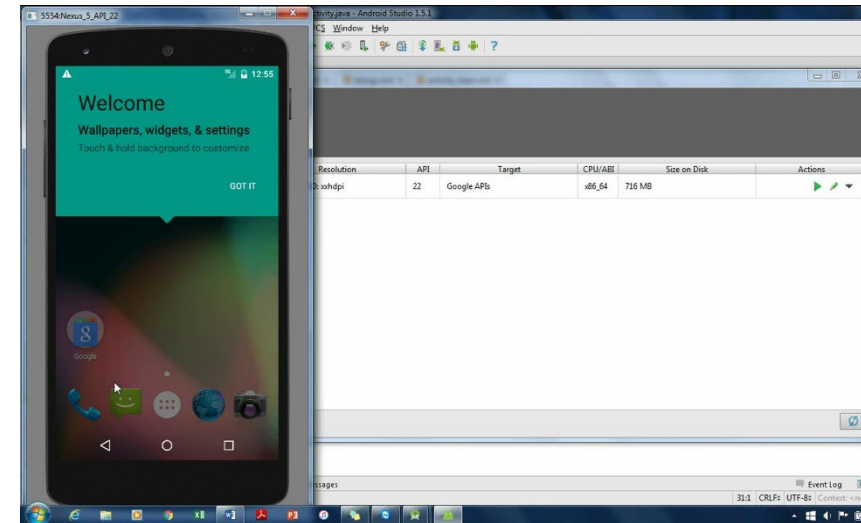
```

DOSBox 0.65, Cpu Cycles: 3000, Frameskip 0, Program: DOSBOX
DOSBox Shell v0.65
DOSBox runs real and protected mode games.
For supported shell commands type: HELP
For a short introduction type: INTRO

If you want more speed, try ctrl-F8 and ctrl-F12.
To activate the keymapper ctrl-F1.
For more information read the README file in the DOSBox directory.

HAVE FUN!
The DOSBox Team

Z:\>SET BLASTER=A220 I7 D1 H5 T6
Z:\>SET ULTRASND=240,3,3,5,5
Z:\>SET ULTRADIR=C:\ULTRASND
Z:\>_
  
```



5. Contenidos teóricos: Sesión 2-4

5. EMULADORES

En nuestro caso, el emulador que utilizaremos es el incluido en Android Studio para probar y depurar nuestras aplicaciones Android.

No obstante, pese a que el emulador es compatible con la mayoría de funciones del dispositivo, no incluye hardware virtual para lo siguiente:

- Wi-Fi
- Bluetooth
- NFC
- Inserción / expulsión de tarjetas SD
- Auriculares conectados a dispositivos
- USB

Si queremos probar funcionalidades que utilicen este hardware, tendremos que utilizar un dispositivo físico para ello.

5. Contenidos teóricos: Sesión 2-4

6. ENTORNOS DE DESARROLLO INTEGRADO

Un entorno de desarrollo integrado (IDE) es una aplicación informática que proporciona servicios integrales para facilitarle al desarrollador o programador el desarrollo del software.

En el caso de las aplicaciones móviles, tenemos:

Xcode (iOS): <https://developer.apple.com/xcode/>

Características:

- Disponible para macOS.
- Editor de código
- Editor visual
- Simulador



5. Contenidos teóricos: Sesión 2-4

6. ENTORNOS DE DESARROLLO INTEGRADO

Android Studio (Android): <https://developer.android.com/studio/>

Características:

- Multiplataforma
- Editor visual
- Emuladores
- Editor de código
- Compilación

