
Programación de Servicios y Procesos

Publicación 1.0

Oscar Gomez

19 de September de 2014

1. Análisis de tecnologías para aplicaciones en dispositivos móviles:	3
1.1. Puntos iniciales	3
1.2. Sistemas operativos para dispositivos móviles. Características.	3
1.3. Limitaciones	4
1.4. Entornos integrados de trabajo.	5
1.5. Módulos para el desarrollo de aplicaciones móviles.	5
1.6. Emuladores.	5
1.7. Configuraciones. Tipos y características. Dispositivos soportados.	5
1.8. Perfiles. Características. Arquitectura y requerimientos. Dispositivos soportados.	5
1.9. Ciclo de vida de una aplicación	5
1.10. Modificación de aplicaciones existentes.	5
1.11. Utilización del entorno de ejecución del administrador de aplicaciones.	5
2. Programación de aplicaciones para dispositivos móviles	7
2.1. Herramientas y fases de construcción.	8
2.2. Interfaces de usuario. Clases asociadas.	8
2.3. Servicios en dispositivos móviles.	8
2.4. Proveedores de contenido.	8
2.5. Gestión de recursos y notificaciones.	8
2.6. Contexto gráfico. Imágenes.	8
2.7. Eventos del teclado.	8
2.8. Técnicas de animación y sonido.	8
2.9. Descubrimiento de servicios.	8
2.10. Bases de datos y almacenamiento.	8
2.11. Persistencia.	8
2.12. Modelo de hilos.	8
2.13. Comunicaciones: clases asociadas. Tipos de conexiones.	8
2.14. Gestión de la comunicación inalámbrica.	8
2.15. Seguridad y permisos.	8
2.16. Envío y recepción de mensajes texto.	8
2.17. Envío y recepción de mensajería multimedia. Sincronización de contenido.	8
2.18. Manejo de conexiones HTTP y HTTPS.	8
2.19. Empaquetado y despliegue de aplicaciones para dispositivos móviles.	8
2.20. Centros de distribución de aplicaciones.	8

2.21. Documentación de aplicaciones de dispositivos móviles.	8
3. Utilización de librerías multimedia integradas	9
3.1. Conceptos sobre aplicaciones multimedia.	9
3.2. Arquitectura del API utilizado.	9
3.3. Fuentes de datos multimedia. Clases.	9
3.4. Datos basados en el tiempo.	9
3.5. Procesamiento de objetos multimedia. Clases. Estados, métodos y eventos. . .	9
3.6. Reproducción de objetos multimedia. Clases. Estados, métodos y eventos. . .	9
3.7. Depuración y documentación de los programas.	9
4. Análisis de motores de juegos	11
4.1. Animación 2D y 3D.	11
4.2. Arquitectura del juego. Componentes.	11
4.3. Motores de juegos: Tipos y utilización.	11
4.4. Áreas de especialización, librerías utilizadas y lenguajes de programación . . .	11
4.5. Componentes de un motor de juegos.	11
4.6. Librerías que proporcionan las funciones básicas de un Motor 2D/3D.	11
4.7. APIs gráficos 3D.	11
4.8. Estudio de juegos existentes.	11
4.9. Aplicación de modificaciones sobre juegos existentes.	11
5. Desarrollo de juegos 2D y 3D	13
5.1. Entornos de desarrollo para juegos.	14
5.2. Integración del motor de juegos en entornos de desarrollo.	14
5.3. Conceptos avanzados de programación 3D.	14
5.4. Fases de desarrollo:	14
5.5. Propiedades de los objetos: luz, texturas, reflejos, sombras.	14
5.6. Aplicación de las funciones del motor gráfico. Renderización.	14
5.7. Aplicación de las funciones del grafo de escena.	14
5.8. Tipos de nodos y su utilización.	14
5.9. Asociación de sonidos a los eventos del juego.	14
5.10. Análisis de ejecución. Optimización del código.	14
5.11. Documentación de la fase de diseño y de desarrollo.	14
6. Sistemas basados en localización	15
6.1. Tecnologías de localización (GPS, A-GPS,...).	15
6.2. Servicios de localización, mapas y geocodificación.	15
6.3. Emuladores para simular las ubicaciones.	15
6.4. Visualización la información geolocalizada.	15

Índice:

Análisis de tecnologías para aplicaciones en dispositivos móviles:

1.1 Puntos iniciales

- En primer lugar, en <http://10.8.0.253> se puede encontrar un servidor.
- Los apuntes también se podrán encontrar a diario en <http://oscarmaestre.github.io>
- Existen fotocopias con la programación, criterios, etc... en la mesa del profesor. En cualquier caso, están colgadas en la página del centro <http://www.iesmaestredecalatrava.es> (buscar el apartado “Presentaciones”)
- Si se desea acceder a algún fichero individual de los apuntes puede hacerse en las página siguiente
 - <https://github.com/OscarMaestre/Moviles>
 - <https://github.com/OscarMaestre/ServiciosYProcesos>

1.2 Sistemas operativos para dispositivos móviles. Características.

El desarrollo para la telefonía móvil es un campo que se encuentra en plena expansión. El número de teléfonos no deja de crecer y las necesidades de programación de los mismos tampoco. En ese sentido existen diversas plataformas de desarrollo a tener en cuenta al empezar a programar.

- Android: es el más numeroso de lejos. La mayor parte del mercado usa esta plataforma. El hecho de que Google ofrezca *completamente gratis* el sistema operativo para los fabricantes y el entorno para los programadores lo ha hecho crecer hasta desbancar a su competidor. Google solo se ocupa de la venta de apps y ese es su nicho de beneficios.
- iOS: Utiliza una filosofía completamente distinta que es controlar todo el proceso desde el desarrollo hasta la distribución de aplicaciones. Ese control se hace por medio del pago de licencias y de la unicidad de la distribución.

- Windows Phone: es el sistema de Microsoft que ha lanzado más o menos recientemente y que aún está por ver si consigue una cuota de mercado significativa o no.
- FirefoxOS: se centra principalmente en mercados emergentes (con el objetivo a largo plazo de ganar cuota de mercado) su filosofía es la misma del software libre.
- Bada: Es la plataforma de Samsung creada exclusivamente para sus teléfonos. Es muy poco usada fuera de Corea del Sur.
- Symbian: surge de un antiguo sistema creado para las PDA, está prácticamente descatalogado en el desarrollo para telefonía móvil.
- Tizen: una plataforma relativamente nueva. Al igual que Android es gratis y además tiene el respaldo de Intel.
- Jolla: surge en los países nórdicos con una filosofía similar al software libre pero con la salvedad de que, de momento, solo se ejecuta en sus teléfonos (que son de gama alta)

En cuanto a la tecnología hay diferencias sustanciales entre ellas:

- Android: pensado principalmente para ser programado en Java (aunque se puede llegar a usar C++ con un kit aparte).
- iOS: usa Objective-C que lo separa mucho del resto de plataformas. El entorno exige el pago de una licencia, el SO exige una licencia y el poner aplicaciones a la venta exige otra.
- Windows Phone: usa Visual Studio que es una herramienta muy potente y usa la plataforma .NET.
- FirefoxOS: usa HTML y Javascript.
- Symbian y otros también permiten el uso de HTML y JS.
- Tizen permite dos opciones: C++ o HTML/JS
- Jolla/Sailfish: usa C++.
- Bada usa C++.

En este curso se usará Android con Java como lenguaje de desarrollo.

1.3 Limitaciones

Programar un teléfono móvil implica preparar nuestro programa para situaciones que no son de importancia en los ordenadores de escritorio o que incluso no existen.

- Desconexión: un teléfono puede perder el fluido eléctrico sin aviso o perder la conexión de red de forma repentina.
- Seguridad: un teléfono puede ser accesible desde cualquier punto del planeta lo que puede poner en grave riesgo la privacidad del usuario.
- Memoria: la cantidad de memoria de estos dispositivos es mucho más reducida que los equipos de escritorio.

- Consumo batería: la cantidad de código que se ejecuta implica disminuir la cantidad de batería del usuario.
- Almacenamiento: la cantidad de espacio para almacenar ficheros en estos dispositivos es muy variable y a veces prácticamente inexistente.

1.4 Entornos integrados de trabajo.

1.5 Módulos para el desarrollo de aplicaciones móviles.

1.6 Emuladores.

1.7 Configuraciones. Tipos y características. Dispositivos soportados.

1.8 Perfiles. Características. Arquitectura y requerimientos. Dispositivos soportados.

1.9 Ciclo de vida de una aplicación

1.9.1 Descubrimiento

1.9.2 Instalación

1.9.3 Ejecución

1.9.4 Actualización

1.9.5 Borrado.

1.10 Modificación de aplicaciones existentes.

1.11 Utilización del entorno de ejecución del administrador de aplicaciones.

Programación de aplicaciones para dispositivos móviles

2.1 Herramientas y fases de construcción.

2.2 Interfaces de usuario. Clases asociadas.

2.3 Servicios en dispositivos móviles.

2.4 Proveedores de contenido.

2.5 Gestión de recursos y notificaciones.

2.6 Contexto gráfico. Imágenes.

2.7 Eventos del teclado.

2.8 Técnicas de animación y sonido.

2.9 Descubrimiento de servicios.

2.10 Bases de datos y almacenamiento.

2.11 Persistencia.

2.12 Modelo de hilos.

2.13 Comunicaciones: clases asociadas. Tipos de conexiones.

Utilización de librerías multimedia integradas

3.1 Conceptos sobre aplicaciones multimedia.

3.2 Arquitectura del API utilizado.

3.3 Fuentes de datos multimedia. Clases.

3.4 Datos basados en el tiempo.

3.5 Procesamiento de objetos multimedia. Clases. Estados, métodos y eventos.

3.6 Reproducción de objetos multimedia. Clases. Estados, métodos y eventos.

3.7 Depuración y documentación de los programas.

Análisis de motores de juegos

4.1 Animación 2D y 3D.

4.2 Arquitectura del juego. Componentes.

4.3 Motores de juegos: Tipos y utilización.

4.4 Áreas de especialización, librerías utilizadas y lenguajes de programación

4.5 Componentes de un motor de juegos.

4.6 Librerías que proporcionan las funciones básicas de un Motor 2D/3D.

4.7 APIs gráficos 3D.

4.8 Estudio de juegos existentes.

4.9 Aplicación de modificaciones sobre juegos existentes.

Desarrollo de juegos 2D y 3D

- 5.1 Entornos de desarrollo para juegos.**
- 5.2 Integración del motor de juegos en entornos de desarrollo.**
- 5.3 Conceptos avanzados de programación 3D.**
- 5.4 Fases de desarrollo:**
- 5.5 Propiedades de los objetos: luz, texturas, reflejos, sombras.**
- 5.6 Aplicación de las funciones del motor gráfico. Renderización.**
- 5.7 Aplicación de las funciones del grafo de escena.**
- 5.8 Tipos de nodos y su utilización.**
- 5.9 Asociación de sonidos a los eventos del juego.**
- 5.10 Análisis de ejecución. Optimización del código.**
- 5.11 Documentación de la fase de diseño y de desarrollo.**

Sistemas basados en localización

6.1 Tecnologías de localización (GPS, A-GPS,...).

6.2 Servicios de localización, mapas y geocodificación.

6.3 Emuladores para simular las ubicaciones.

6.4 Visualización la información geolocalizada.