



Programación multimedia y dispositivos móviles

Jacinto D. Cabrera Rodríguez



Contenidos digitales
www.sintesis.com



Programación multimedia y dispositivos móviles

Consulte nuestra página web: www.sintesis.com
En ella encontrará el catálogo completo y comentado



Queda prohibida, salvo excepción prevista en la ley, cualquier forma de reproducción, distribución, comunicación pública y transformación de esta obra sin contar con autorización de los titulares de la propiedad intelectual. La infracción de los derechos mencionados puede ser constitutiva de delito contra la propiedad intelectual (arts. 270 y sigs. Código Penal). El Centro Español de Derechos Reprográficos (www.cedro.org) vela por el respeto de los citados derechos.

Programación multimedia y dispositivos móviles

Jacinto D. Cabrera
Rodríguez



ASESOR EDITORIAL:

Juan Carlos Moreno Pérez

© Jacinto D. Cabrera Rodríguez

© EDITORIAL SÍNTESIS, S. A.
Vallehermoso, 34. 28015 Madrid
Teléfono 91 593 20 98
<http://www.sintesis.com>

ISBN: 978-84-135750-4-9

Impreso en España - Printed in Spain

Reservados todos los derechos. Está prohibido, bajo las sanciones penales y el resarcimiento civil previstos en las leyes, reproducir, registrar o transmitir esta publicación, íntegra o parcialmente, por cualquier sistema de recuperación y por cualquier medio, sea mecánico, electrónico, magnético, electroóptico, por fotocopia o por cualquier otro, sin la autorización previa por escrito de Editorial Síntesis, S. A.

índice

PRESENTACIÓN	13
1. TECNOLOGÍAS MÓVILES	15
Objetivos	15
Mapa conceptual	16
Glosario	16
1.1. Introducción	17
1.2. Tecnologías móviles: características y limitaciones	17
1.2.1. Características de los dispositivos móviles	17
1.2.2. Tipos de dispositivos móviles	18
1.2.3. Tecnología de comunicación móvil	20
1.2.4. Limitaciones de los dispositivos móviles	22
1.3. Sistemas operativos móviles	23
1.3.1. Android	24
1.3.2. iOS	25
1.3.3. Windows Phone	25
1.3.4. BlackBerry OS	26
1.3.5. Symbian	26
1.3.6. Palm OS P (WebOS)	26
1.3.7. Firefox OS	26
1.3.8. Ubuntu Touch	27
1.3.9. Harmony OS	27
1.4. Lenguajes de programación para dispositivos móviles	27
1.4.1. Desarrollo nativo	27
1.4.2. Desarrollo multiplataforma compilado a nativo	27
1.4.3. Desarrollo multiplataforma basado en HTML5	28

1.5. Entornos integrados de desarrollo de aplicaciones móviles	28
1.5.1. Symbian	29
1.5.2. BlackBerry OS	29
1.5.3. Windows Phone	29
1.5.4. iOS	29
1.5.5. Android	30
Resumen	30
Actividades de autoevaluación	31
2. INTRODUCCIÓN A ANDROID	33
Objetivos	33
Mapa conceptual	34
Glosario	34
2.1. Introducción	35
2.2. Fundamentos de una aplicación en Android	35
2.3. Componentes de una aplicación	36
2.4. Android Manifest	36
2.5. Recursos de una aplicación	38
2.5.1. Cómo acceder a los recursos	40
2.6. La clase R	41
2.7. Estructura de un proyecto	42
2.7.1. Instalación de Android Studio	42
2.7.2. Iniciando el entorno de desarrollo	44
2.7.3. Elementos de un proyecto	46
2.7.4. Configurar el dispositivo donde probar las aplicaciones	47
2.8. Los permisos en Android	48
Resumen	50
Actividades de autoevaluación	50
3. LAS ACTIVIDADES EN ANDROID	53
Objetivos	53
Mapa conceptual	54
Glosario	54
3.1. Introducción	55
3.2. Las actividades	55
3.3. Ciclo de vida de una aplicación	57
3.4. ¡Hola mundo de Android!	59
3.5. Conservar el estado de una aplicación	62
3.6. Intents y filtros	63
3.6.1. Intents explícitos	64
3.6.2. Intents implícitos	65
3.6.3. Elementos	65
3.6.4. Propagación	67
3.6.5. Filtros	69
3.6.6. PendingIntent	69
Resumen	70
Actividades de autoevaluación	71

4. INTERFACE DE USUARIO: LOS LAYOUTS	73
Objetivos	73
Mapa conceptual	74
Glosario	74
4.1. Introducción	74
4.2. Interfaces de usuario. Clases asociadas	75
4.3. Layouts	77
4.4. LinearLayout	77
4.5. FrameLayout	78
4.6. AbsoluteLayout	79
4.7. RelativeLayout	80
4.8. TableLayout	81
4.9. GridLayout	83
4.10. ConstraintLayout	84
Resumen	86
Actividades de autoevaluación	87
5. INTERFACE DE USUARIO: CONTROLES BÁSICOS	89
Objetivos	89
Mapa conceptual	90
Glosario	90
5.1. Introducción	91
5.2. TextView	91
5.2.1. Cambiar las propiedades del texto	92
5.2.2. Añadir nuevo texto	93
5.2.3. Uso de fuentes variadas	93
5.2.4. Animación del texto	94
5.3. Button	97
5.4. ToggleButton	99
5.5. ImageButton	100
5.6. EditText	100
5.7. AutoCompleteTextView	103
5.8. MultiAutoCompleteTextView	103
5.9. Spinner	104
5.10. CheckBox	105
5.11. RadioButton	106
5.12. Switch	107
5.13. SeekBar	108
5.14. RatingBar	109
5.15. ProgressBar	109
Resumen	110
Actividades de autoevaluación	111
6. LISTADOS Y MENÚS	113
Objetivos	113
Mapa conceptual	114
Glosario	114
6.1. Introducción	115

6.2. Los listados	115
6.3. ListView	115
6.4. GridView	117
6.5. Spinner	119
6.6. Trabajando con adaptadores	120
6.6.1. ArrayAdapter	121
6.6.2. BaseAdapter	124
6.7. OptionsMenu	126
6.8. Submenú	127
6.9. Menú contextual	128
6.10. Menú contextual en listas	129
Resumen	130
Actividades de autoevaluación	131
7. PREFERENCIAS, DIÁLOGOS Y NOTIFICACIONES	133
Objetivos	133
Mapa conceptual	134
Glosario	134
7.1. Introducción	135
7.2. Preferencias	135
7.2.1. Categorías	135
7.2.2. CheckBoxPreference	136
7.2.3. EditTextPreference	136
7.2.4. ListPreference	136
7.2.5. MultiSelectListPreference	137
7.2.6. Acceso a las preferencias	137
7.3. Toast	138
7.4. Diálogos	139
7.4.1. DatePickerDialog	142
7.4.2. TimePickerDialog	143
7.5. Notificaciones	143
7.5.1. Crear la notificación	144
7.5.2. Establecer la actividad de destino	145
7.5.3. Construir y lanzar la notificación	145
Resumen	147
Actividades de autoevaluación	147
8. MATERIAL DESIGN	149
Objetivos	149
Mapa conceptual	150
Glosario	150
8.1. Introducción	151
8.2. Elementos del diseño	151
8.2.1. La superficie y la luz	151
8.2.2. Elevación	151
8.2.3. Colores	152
8.2.4. Movimiento	152
8.2.5. Tipografías	153
8.2.6. Iconografía	153

8.3. Trabajar con Material Design	153
8.4. Layouts y ToolBar	154
8.4.1. ToolBar	155
8.5. TabLayout	157
8.6. TextInputLayout	159
8.7. Floating Action Button (FAB)	161
8.8. Snackbar	163
8.9. CardView y RecyclerView	164
8.9.1. RecyclerView	166
8.10. Navigator Drawer	170
Resumen	172
Actividades de autoevaluación	172
9. ANDROID AVANZADO	175
Objetivos	175
Mapa conceptual	176
Glosario	176
9.1. Introducción	177
9.2. Los fragmentos	177
9.2.1. Construyendo un fragmento	178
9.2.2. Intercambio de fragmentos	180
9.2.3. Fragmentos especiales	181
9.3. Timer, hilos y procesos	184
9.3.1. Hilos	185
9.3.2. Hilos secundarios	185
9.3.3. Tareas asíncronas	190
9.3.4. Timer/TimerTask	191
9.3.5. Planificador de tareas (Scheduler)	192
9.4. Los servicios	193
9.4.1. Ciclo de vida de los servicios	193
9.4.2. Started Service	194
9.4.3. Bound Service	196
9.4.4. Intent Service	197
9.4.5. Servicios en primer plano	199
9.5. Receptores de Broadcast	199
9.5.1. Envío de Broadcast	201
Resumen	202
Actividades de autoevaluación	203
10. PERSISTENCIA	205
Objetivos	205
Mapa conceptual	206
Glosario	206
10.1. Introducción	207
10.2. Preferencias compartidas	207
10.3. Almacenamiento en la memoria interna del dispositivo	209
10.4. Lectura de un recurso de la aplicación	210
10.5. Persistencia en la tarjeta de almacenamiento externo	210

10.6. Persistencia en bases de datos (SQLite)	212
10.6.1. Insertar registros	214
10.6.2. Eliminar registros	214
10.6.3. Actualizar registros	214
10.6.4. Consultar registros	215
10.6.5. Movimientos del cursor	216
10.7. Almacenamiento en la Red	217
10.7.1. Conexión HTTP	217
10.7.2. Conexión HTTPS	220
10.7.3. Conexión por Sockets	222
10.8. Proveedores de contenidos	224
10.8.1. Crear un proveedor de datos (Content Provider)	225
10.8.2. Acceder a datos de otra aplicación (Content Resolver)	228
Resumen	232
Actividades de autoevaluación	232

11. LIBRERÍAS MULTIMEDIA EN ANDROID 235

Objetivos	235
Mapa conceptual	236
Glosario	236
11.1. Introducción	237
11.2. Las imágenes en Android	237
11.2.1. La pantalla	237
11.2.2. Las unidades de medida en Android	239
11.2.3. La clase ImageView	239
11.2.4. La clase BitMap	240
11.3. Los dibujos en Android	241
11.3.1. Vistas personalizadas	241
11.3.2. La clase Canvas y la clase Paint	242
11.3.3. La clase Path	245
11.3.4. La clase Drawable	246
11.4. Grabación/reproductor de audio	246
11.4.1. La clase SoundPool	246
11.4.2. La clase MediaPlayer	248
11.4.3. La clase MediaRecorder	250
11.5. Grabación/reproductor de vídeo	251
11.5.1. La clase VideoView	252
11.5.2. La clase MediaController	252
11.5.3. La clase MediaPlayer	253
11.5.4. La clase MediaRecorder	254
11.6. La clase MediaStore	255
11.6.1. Obtener información de MediaStore	256
11.6.2. Añadir contenidos a MediaStore	257
11.7. La pantalla táctil	257
11.7.1. Eventos de pantalla	258
11.7.2. Eventos en pantallas Multi-Touch	259
11.7.3. Las gesturas	259
11.8. Los sensores	259
11.8.1. Acelerómetro	261
11.8.2. Giroscopio	262

11.8.3. Magnetómetro	262
11.8.4. Sensor de orientación	263
11.8.5. Sensor de proximidad	263
11.8.6. Sensor de iluminación	263
11.9. La geolocalización	264
11.9.1. Obtener la ubicación del dispositivo mediante la API de Android	264
11.9.2. Google Maps	267
Resumen	270
Actividades de autoevaluación	271
 12. LOS JUEGOS EN ANDROID	275
Objetivos	275
Mapa conceptual	276
Glosario	276
12.1. Introducción	277
12.2. Animación 2D y 3D	277
12.2.1. Animación 2D	277
12.2.2. Animación 3D	278
12.3. La animación en Android	279
12.3.1. Transiciones de escenas	281
12.3.2. Animar drawables	281
12.3.3. Animación del Canvas	283
12.4. Los sprites	284
12.5. Las colisiones	285
12.6. Propiedades de los objetos: materiales, texturas, luces y sombras	286
12.6.1. Los materiales	286
12.6.2. Las texturas	286
12.6.3. Las luces	287
12.6.4. Las sombras	288
12.7. La clase SurfaceView	288
12.8. La clase GLSurfaceView	289
12.9. Arquitectura de un juego	293
12.10. Motores de juegos	296
12.10.1. Framework libGDX	296
12.10.2. Otros motores de juegos	300
12.11. Entornos de desarrollo para juegos	302
Resumen	303
Actividades de autoevaluación	304
 BIBLIOGRAFÍA	307



Este libro contiene material adicional para ampliar información y realizar las actividades propuestas

Presentación

Los avances tecnológicos y la agilización de las redes han contribuido a la aparición de potentes dispositivos móviles, con el consecuente auge de aplicaciones desarrolladas para este tipo de equipos. Esto ha favorecido que se genere un mercado increíblemente rentable, en el que las empresas potencian la labor de los desarrolladores mediante la producción de herramientas avanzadas y abiertas para todos los niveles. De hecho, en la actualidad, las compañías de videojuegos basan una gran parte de su negocio en los juegos móviles, tanto en Android como para otros sistemas operativos.

A finales del pasado año había 5.000 millones de usuarios con un smartphone y se descargaron 197.000 millones de aplicaciones, de entre los 2.800 millones existentes (en marzo de ese año) en Google Play y 2.200 millones en la App Store. Con un promedio de 22 millones de personas utilizando aplicaciones móviles todos los días, se espera que en 2020 este mercado genere alrededor de 189 mil millones de dólares de ingresos.

Además, el sector del videojuego facturó en España, el pasado año, 1.530 millones de euros (un 12,6% más que el año anterior), siendo, actualmente, en nuestro país la primera opción de ocio por encima de la industria del cine y de la música grabada.

Esta tendencia no parece que vaya a ser temporal, garantizando con ello el futuro a largo plazo de estas aplicaciones y las tecnologías de desarrollo a ellas ligadas. Por ello, es impensable formar a profesionales de la programación sin incluir en su currículu un profundo conocimiento de dichas tecnologías, especialmente las que contienen multimedia y posibilitan el desarrollo de juegos, lo cual justifica su inclusión en los planes oficiales de estudio.

Por otro lado, como en esta obra queda patente, son múltiples las posibilidades entre las que se puede optar a la hora de desarrollar una aplicación para móvil. De entre todas ellas, gracias a su popularidad, difusión, mercado y, especialmente, costo de licencias, se ha optado por utilizar Android nativo, fundamentado en Java, como lenguaje base sobre el que desarrollar el currículu regulado en la normativa vigente.

De ahí que este libro resulte particularmente indicado, como manual de trabajo, para el módulo Programación Multimedia y Dispositivos Móviles del ciclo formativo de grado superior

de Técnico Superior en Desarrollo de Aplicaciones Multiplataforma, cuyo currículo queda recogido en el Real Decreto 450/2010, de 16 de abril. Además, con los conocimientos aquí aportados, pueden adquirirse las cualificaciones profesionales Programación en Lenguajes Estructurados de Aplicaciones de Gestión IFC155_3 (R.D. 1087/2005, de 16 de septiembre), Programación con Lenguajes Orientados a Objetos y Bases de Datos Relacionales IFC080_3 (R.D. 295/2004, de 20 de febrero) y colaborar en la adquisición de otras unidades de competencia, como la UC0964_3: Crear elementos software para la gestión del sistema y sus recursos.

Por motivos fundamentalmente pedagógicos, se han dividido los cinco bloques temáticos que propone el Real Decreto 450/2010 en doce capítulos. En el primer capítulo se hace un recorrido por las diferentes tecnologías para móviles, tipos, limitaciones y evolución, así como por los sistemas operativos existentes en el mercado, los lenguajes para programarlos y las herramientas existentes para tal fin.

Los dos capítulos siguientes introducen al desarrollador en la programación nativa en Android. Se estudian los componentes y fundamentos de una aplicación en Android y los elementos fundamentales sobre los que se sustenta un proyecto para este sistema operativo móvil (el *AndroidManifest*, la clase R, los recursos, las actividades, los *intent*...).

En los capítulos 4, 5 y 6, con un enfoque eminentemente práctico merced a numerosas actividades e incontables ejemplos, se trabajan extensamente y en profundidad los elementos del *interface*, iniciando el recorrido por los *layouts*, controles básicos, listados y menús, estudiando, en cada caso, clases, métodos, tipos, variantes y arquitectura de construcción. El capítulo 7 está dedicado a las preferencias (categorías, acceso y construcción), *toast*, diálogos y notificaciones.

Casi de manera monográfica, el capítulo 8 se dedica a Material Design, un nuevo concepto de diseño basado en objetos materiales, presentado por Google en 2014, utilizado en Android, pero también en la web y en cualquier otra plataforma.

Fragments, *timer*, hilos, procesos, servicios y *broadcast* quedan englobados en el siguiente capítulo, un estudio de diferentes clases diseñado para su fácil comprensión, a pesar de incluirse en el apartado Android avanzado.

Bajo el título de “Persistencia” se incluyen las variadas formas que Android nos propone para almacenar y recuperar la información (memoria interna, recursos, tarjeta, bases de datos...), incluidas las que usa la red (HTTP, HTTPS, *sockets*). También se abordan en este capítulo los proveedores de contenidos como forma de distribuir y acceder a la información.

En el capítulo 11 se incluye un detallado estudio de las librerías multimedia, imágenes, dibujos, audio, video... Además, por cercanía y compatibilidad de uso, se incluyen también en este bloque, la clase MediaStore, los eventos de pantalla, los sensores y la geolocalización.

Por último, cierra el libro con el mundo de los juegos en los dispositivos móviles, haciendo un recorrido por las técnicas de animación 2D y 3D, y algunos conceptos necesarios a la hora de elaborar juegos y animaciones (materiales, texturas, luces, sombras, *sprites*, colisiones...). Se practica la animación en Android con Drawables y Canvas y se usa SurfaceView y GLSurfaceView como superficies de reproducción. Finalizan el capítulo la arquitectura de juegos, los motores de juegos (libGDX, AndEngine...) y el estudio de los entornos de desarrollo de juegos más populares del mercado.

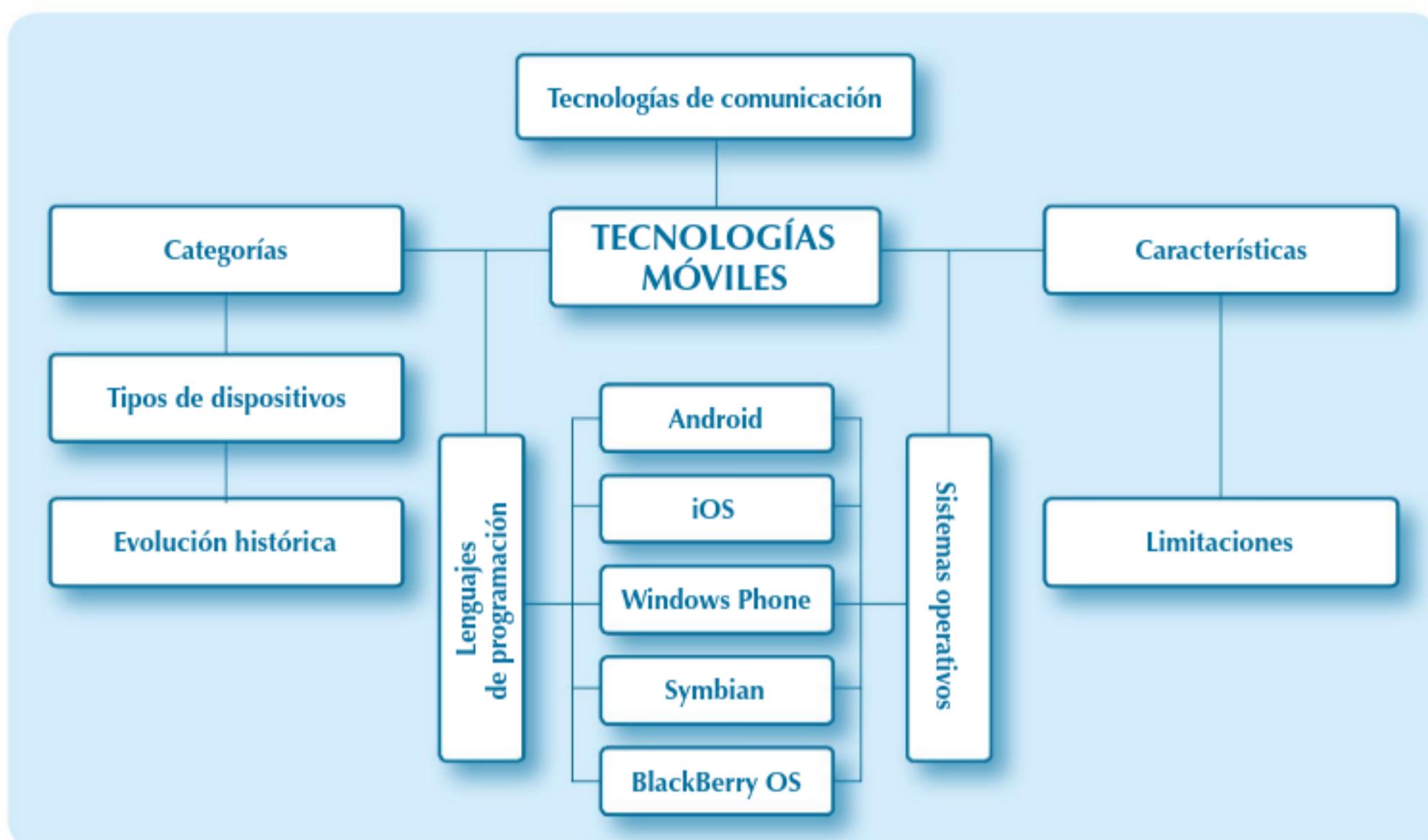
Todos los capítulos han sido elaborados con un enfoque práctico, con ejemplos, tutoriales, material didáctico y la propuesta de numerosas actividades, intentando facilitar al lector la consecución de las destrezas necesarias que el actual mercado de desarrollo de aplicaciones nativas en Android demanda. Se incluyen, asimismo, una serie de recursos digitales, disponibles en la plataforma de Editorial Síntesis (www.sintesis.com), identificados con el icono .

Tecnologías móviles

Objetivos

- ✓ Reconocer los tipos de dispositivos móviles, sus características y limitaciones.
- ✓ Identificar la evolución de la tecnología de comunicación móvil.
- ✓ Tener una visión histórica de la evolución de los dispositivos móviles.
- ✓ Conocer los sistemas operativos existentes para móviles.
- ✓ Familiarizarse con los tipos y variedades de lenguajes de programación.

Mapa conceptual



Glosario

API (Application Programming Interface). La interfaz de programación de aplicaciones es un conjunto de funciones y procedimientos cuyo uso permite a los desarrolladores interactuar con determinados elementos del sistema operativo o de otros programas.

APP. Acrónimo de aplicación informática.

Framework. Entorno o marco de trabajo formado por un conjunto de elementos o módulos de software que sirven como base para la organización y desarrollo de software.

Gadget. Dispositivo, generalmente tecnológico y novedoso, de pequeñas proporciones que tiene función específica y práctica.

Kernel. Núcleo, parte central y fundamental del sistema operativo. Suele ser responsable de la comunicación entre el software y el hardware del sistema informático.

Runtime. Intervalo de tiempo en el que un programa de computadora se ejecuta en un sistema operativo.

SMS (Short Message Service). Servicio de mensajes, disponible en los teléfonos móviles, que permite el envío de mensajes cortos entre dispositivos móviles.

UI (User Interface). Interfaz del usuario. Es la vista que permite al usuario interactuar de manera efectiva y cómoda con un sistema.

WAP (Wireless Application Protocol). Protocolo de aplicaciones inalámbricas. Es un estándar abierto internacional para aplicaciones y dispositivos que utilizan comunicaciones inalámbricas.

1.1. Introducción

Aunque los rápidos cambios a los que nos tienen acostumbrados las tecnologías, especialmente las de la información, han hecho que sepamos adaptar, a vertiginoso ritmo, nuestros hábitos y rutinas de vida, quizás ninguno de ellos ha producido una variación tan radical en nuestra mentalidad y conducta como la consecuente a la gran revolución que han supuesto las redes móviles y, especialmente, la conjugación con ellas de teléfonos inteligentes, capaces de simplificar en un solo dispositivo multitud de tareas para las que hasta ahora necesitábamos variados y complejos aparatos.

1.2. Tecnologías móviles: características y limitaciones

Antes de adentrarnos en el desarrollo de aplicaciones móviles, es conveniente conocer las tecnologías móviles que pueden encontrarse en el mercado, dispositivos que las soportan y aquellos otros aspectos que pueden ser relevantes a la hora de programar una aplicación móvil.

1.2.1. Características de los dispositivos móviles

Una primera aproximación al concepto de *dispositivo móvil* nos podría hacer definirlo como aquél aparato con capacidad de procesado y almacenamiento, de pequeño tamaño, portable, autonomía de funcionamiento en cuanto a batería, conexión permanente o semipermanente a una red de comunicaciones y con variadas funciones, si bien una de ellas es para la cual fue fundamentalmente diseñado. Esta definición nos muestra ya alguna de sus características generales, que podríamos agrupar en cuatro grandes bloques.

1. *Capacidad de procesado*: puesto que, en la actualidad, relacionamos el concepto de dispositivo móvil, en cualquiera de sus variantes, con el de microordenador, con cierta especificidad de funciones. Esto lo asocia a capacidad de cálculo (en forma de velocidad de procesado) y almacenamiento, que incluiría el necesario para el procesado y la persistencia de la información en él depositada (no menos importante).
2. *Tamaño*: el concepto de portabilidad lo llevamos a sus máximos extremos, alejándonos de los ordenadores portátiles, en cuanto les exigimos que estos puedan compartir con el usuario su vida diaria, independientemente de la actividad que se esté realizando, aunque esta sea deportiva.
3. *Movilidad*: asociada a la anterior característica, pero también a la no dependencia de cableado para su alimentación o comunicación. Es decir, pequeño tamaño, batería duradera y comunicación inalámbrica son los tres pilares sobre los que se asienta esta propiedad.
4. *Conectividad*: si bien no es novedosa la comunicación inalámbrica entre teléfonos, utilizando variadas tecnologías (muchas de ellas ya obsoletas), han sido los avances en las redes móviles y la masiva incorporación del uso de redes de datos lo que ha creado un nuevo paradigma a la hora de definir estos dispositivos.

Además de estas propiedades, podríamos incluir muchas otras, como sistema operativo, diseño, ergonomía, tamaño de pantalla, gadgets incorporados..., todos ellos importantes y posiblemente, en algunos casos, decisivos a la hora de adquirir uno de estos aparatos, pero consideradas como propiedades secundarias a la hora de su clasificación.

1.2.2. Tipos de dispositivos móviles

Basándonos en las anteriores características, podemos agruparlos en tres categorías, según los estándares que en 2005 propusieron T38 y DuPont Global Mobility Innovation Team:

- *Limited Data Mobile Device (Dispositivo Móvil de Datos Limitados)*: dispositivos de pantalla pequeña, generalmente tipo texto, y servicios de datos limitados a SMS y acceso WAP.
- *Basic Data Mobile Device (Dispositivo Móvil de Datos Básicos)*: pantalla de mediano tamaño (entre 120×120 y 240×240 píxeles), con menú de navegación basado en interface gráfica, posibilidad de correo electrónico y navegación web.
- *Enhanced Data Mobile Device (Dispositivo Móvil de Datos Mejorados)*: pantallas de medianas a grandes, con las mismas características que los anteriores, a las que se añaden variadas aplicaciones nativas, corporativas y multitud de gadgets, como cámara, GPS, sensores.

En cuanto a los tipos, cuando utilizamos el término *dispositivo móvil* nos referimos a grupo amplio de aparatos electrónicos. Teniendo en cuenta lo anteriormente visto y desde un punto de vista de clasificación, podemos agrupar los dispositivos móviles en los siguientes tipos:

- a) *Teléfonos móviles*: incluimos en este apartado aquellos teléfonos cuya funcionalidad se limita básicamente a la comunicación por voz, si bien en algunos casos pueden incluir operatividad SMS, cámara fotográfica, agenda.
- b) *Handhelds*: con variado diseño, inicialmente con pantalla plegable sobre el teclado, fueron utilizados como agenda/organizador con posibilidad de ejecutar algunas aplicaciones (científicas o financieras). Lanzados por Psion, Casio, tenían sus propios sistemas operativos de baja compatibilidad, hasta que Microsoft lanzó Windows CE, que fue adoptado por la mayoría de las compañías. Durante un tiempo, fueron utilizados como complemento de los ordenadores portátiles, pero a pesar de la evolución de los sistemas operativos (Windows Mobile, CE 202), la mayoría de las compañías fueron abandonando este formato de dispositivos.
- c) *Netbooks*: lanzados al mercado en 1999 por Psion y con buena acogida hasta 2007, se diseñaron originariamente como ordenadores compactos de pequeño tamaño y bajo precio con suficiente poder de cálculo como para darle uso ofimático y tener acceso a Internet.



PARA SABER MÁS

El término *nettop* se aplica a aquellos equipos de características similares en cuanto al tamaño y las prestaciones a los netbooks.

No son portátiles, sino de escritorio.

- d) *E-Book Readers*: con variadas denominaciones (libro electrónico, eBook, libro digital), son dispositivos usados para leer ficheros de documentos (generalmente libros) electrónicos. Suelen ser ligeros, con alta capacidad de almacenaje, bajo consumo y normalmente utilizan una tecnología de pantalla denominada tinta electrónica.

- e) *Tablets*: con tamaño intermedio entre el ordenador y el móvil, tienen como principales características la ligereza, autonomía y fácil manejo (pantalla táctil). Aunque cuentan con algún antecedente histórico, se considera que su irrupción en el mercado data de principios de este siglo, con los lanzamientos de Microsoft Tablet PC, la Nokia 510 webtablet o el iPad de Apple.

Variantes de estas son las minitablets, que tienen un tamaño de 7 u 8 pulgadas, los “tabléfonos”, popularmente denominados a los smartphones de gran tamaño. El portátil convertible cuyo teclado se desliza (en algunos incluso puede separarse) debajo de la pantalla, que suele ser táctil, pudiéndose utilizar como una tableta. Los booklets, dispositivos con dos pantallas, una de las cuales es táctil, que se comporta como teclado virtual.



Figura 1.1
Tipos de dispositivos móviles.

- f) *Personal digital assistant*: más conocidos como PDA, son dispositivos que intentan combinar variadas prestaciones con la pretensión de funcionar como un organizador digital. Inicialmente manejados con un lápiz (stylus), fueron evolucionando en la calidad y resolución de pantalla. Los más conocidos fueron los Palm y el Pocket PC, sin embargo, tanto uno como otro decayeron con la eclosión en el mercado de los teléfonos móviles inteligentes.

- g) *Smartphones*: dispositivos cuya principal funcionalidad es la de teléfono móvil, pero que por sus características, muy similares a las de un ordenador personal, y la variedad de recursos electrónicos que progresivamente se le han ido añadiendo, han hecho que la mayoría de los usuarios recurran a ellos para obtener prestaciones, en muchos casos distintas a las de comunicación telefónica. Suelen ser características comunes a todos ellos las pantallas de alta calidad, la múltiple conectividad, la gran capacidad de procesamiento y almacenaje, y la riqueza en dispositivos incorporados (cámara, sensores, GPS).

Pueden encontrarse en múltiples sistemas operativos y versiones de los mismos, pero en todos los casos tienen la posibilidad de incorporar nuevas aplicaciones que generalmente se suelen descargar de repositorios públicos.

- h) *Gadgets*: dispositivos electrónicos con una función específica, generalmente pequeños, prácticos y novedosos. Tradicionalmente se han englobado en ellos algunos de los ya vistos, como PDA, móviles, smartphones, reproductores mp3, pero cada vez es más frecuente el utilizar este término para dispositivos que aplican las últimas tecnologías a la electrónica de consumo.

Actividad propuesta 1.1



Localiza a través de Internet cinco E-Books y realiza una tabla donde se muestren las siguientes características: fabricante, tipo de pantalla, tamaño de la pantalla, conectividad, memoria del almacenamiento, autonomía.

1.2.3. Tecnología de comunicación móvil

La red de comunicación móvil ha ido evolucionando con el paso del tiempo. Cada generación aporta cambios a la anterior:

1. *Generación 0.* Aunque existe cierta discrepancia sobre su consideración como dispositivos móviles, se incluyen aquí aquellos dispositivos que utilizaban ondas de radio para comunicarse, como los Walkie Talkie. Los estándares de comunicación en esta generación son:
 - PTT: siglas de Push to Talk, *pulsar para hablar*.
 - IMTS: Improved Mobile Telephone System.
2. *Primera generación (tecnología 1G).* Surge a partir de 1977. Se trata de equipos analógicos de gran tamaño y peso, mecanismo de transmisión y recepción con ondas de radio, que solo podían ser utilizados para la transmisión de voz y tenían baja seguridad. La primera red celular se implantó en el año 1977 en Chicago, siendo Japón el primer país con red 1G nacional. Fue la primera generación considerada realmente como de teléfonos móviles, que contaba con los siguientes estándares:
 - AMPS (American Advanced Mobile Phone Service): en América del Norte. Introducido en 1987.
 - NMT (Nordic Mobile Telephone): para los países nórdicos. Introducido en 1979.
 - TACS (British Total Access Communication Systems): en Inglaterra. Introducido en 1983.
 - NAMTS (Nippon Advanced Mobile Telephone Systems): en Japón. Introducido en el 1981.
3. *Segunda generación (tecnología 2G).* Tuvo su inicio sobre el año 1990, pasando así de la tecnología analógica a la digital. Surge ante la necesidad de transmitir datos y voz. Se mejora el proceso de llamadas y se integran otros servicios adicionales a la voz, de entre los que destaca el Servicio de Mensajes Cortos (Short Message Service). Una de las características principales del estándar GSM es el Módulo de Identidad del Suscriptor, conocido comúnmente como tarjeta SIM, tarjeta que contiene la información del usuario, datos de red y directorio telefónico. Las tecnologías predominantes en esta generación son:
 - GSM (Global System for Mobile Communications): utilizado en Europa.
 - IS-136 (conocido también como TIA/EIA136 o ANSI136): utilizado en EE.UU.

- CDMA (Code Division Multiple Access).
 - PDC (Personal Digital Communications): utilizado en Japón.
4. *Segunda generación y media (tecnología 2.5G-2.75G)*. Se crea ante la necesidad de incrementar la velocidad del tráfico de datos, facilitando la navegación por Internet. Pueden incluirse aquí algunos teléfonos móviles que incorporan algunas de las mejoras y tecnologías del estándar 3G. Los estándares más utilizados son:
- GPRS (General Packet Radio Service): mejora la velocidad de transmisión de datos.
 - EDGE (Enhanced Data for Global Evolution): también conocida como EGPRS.
5. *Tercera generación (tecnología 3G)*. Tiene sus orígenes en octubre del 2001 en Japón. Supone un salto cualitativo en el mundo de las telecomunicaciones, al poder, a través de un dispositivo móvil, navegar con comodidad por Internet. Esta generación está basado en los UTMS (Universal Mobile Telecommunications System). Caracterizada por la convergencia de la voz y alta transmisión de datos con acceso. Este sistema alcanzaba velocidad de transmisión de hasta 2Mbps, por lo que se usó el esquema de Acceso Múltiple por División en Códigos (CDMA).
6. *Tercera generación y media (tecnología 3.5G-3.75 G)*. Una modificación de la 3G, cambiando la tecnología UMTS a la llamada HDSPA (High-Speed Downlink Packet Access), que permite que su velocidad de transmisión llegue a los 14Mbps. Esta generación también se conoce como 3G+.

Figura 1.2
Comparación entre las diferentes generaciones de redes móviles.

1G 1980 Solo voz Analógico	2G 1990 Voz y SMS Digital	3G 2001 Voz y datos Multimedia	4G 2010 Datos Protocolo IP	5G 2020 Datos Banda ancha	6G ? Datos Satelital
2,4 Kb/s	64 Kb/s	2 Mb/s	100 Mbps	1 Gbps	?

7. *Cuarta generación (tecnología 4G)*. Empezó a desplegarse a partir del año 2010. Ofrece un mayor ancho de banda que permite la recepción de televisión en alta definición. Long Term Evolution (LTE) es reconocida como 4G y mejora los sistemas basados en UTMS (Universal Mobile Telecommunications System), además, utiliza Wimax (Worldwide Interoperability for Microwave Access), que permite velocidades de descarga de hasta 60 Mbps y envíos de paquetes de información de hasta 40 Mbps.
8. *Cuarta generación y media (tecnología 4G+)*. Aparecida el 2016. Supone un avance en la velocidad de transmisión denominada 4G+ que puede superar los 400 Mbps.
9. *Quinta generación (tecnología 5G)*. Presentada durante el Mobile World Congress celebrado en Barcelona en febrero del 2017, inicia su lanzamiento comercial en 2019, fijando para 2020 su despliegue generalizado. Esta es una tecnología que multiplica por diez la velocidad del 4G (llegando a 1 Gbps de velocidad de transmisión). La quinta generación, conocida también como Real Wireless World System, integra diferentes sistemas

inalámbricos, como Wi-Fi, redes celulares, sistemas de corto alcance con redes de sensores inalámbricos (WSN), comunicaciones de máquina a máquina (M2M), y genera el llamado *Internet de las Cosas* (IoT).

10. *Sexta generación (tecnología 6G)*. La sexta generación implementará los sistemas satelitales con la quinta generación, con lo cual se puede conseguir una cobertura global. Un sistema basado en redes satelitales de navegación de posicionamiento global, utilizadas en la telecomunicación para la telefonía.

Actividades propuestas



- 1.2. Entra en la siguiente dirección web y localiza la cobertura 5G en tu región y quiénes son los proveedores que la dan: <https://www.speedtest.net/ookla-5g-map>
- 1.3. Realiza una búsqueda a través de Internet de la evolución de las ventas de teléfonos móviles de Apple, Samsung y Huawei durante los tres últimos años.



RECURSO DIGITAL 1.1

Para saber más, véase el recurso digital *Evolución de los dispositivos móviles*.

1.2.4. Limitaciones de los dispositivos móviles

La programación de aplicaciones para móviles tiene muchos aspectos similares al desarrollo de aplicaciones para escritorio, pero es necesario entender que para conseguir que los programas funcionen correctamente en un amplio y variado número de dispositivos se debe conocer cuáles son las limitaciones o restricciones a las que estamos sujetos.

La mayoría de las **limitaciones** estarán relacionadas con las **características hardware**. Aunque el mercado ha evolucionado de manera importante, se han de tener en cuenta las variaciones que se pueden encontrar en la capacidad de procesado y almacenamiento, **evitando**, en la medida que sea posible, **la excesiva carga en elementos multimedia**.

Debemos apartarnos de la tendencia a programar para dispositivos de alta gama, ya que si se logra que una aplicación funcione bien en teléfonos con pocos recursos, seguramente lo hará aún mejor en otros de gama superior. Siendo aconsejable, en cualquier caso, optimizar el uso del procesador para una rápida ejecución de nuestro código y respuesta de la aplicación.

La diversidad de tamaño de pantalla también puede hacernos modificar más de una vez una aplicación, ya que diferentes resoluciones harán obtener resultados muy lejanos a los deseados por el programador. Además, no todos los equipos están dotados de sensores y las tecnologías que son consideradas básicas, lo que puede provocar que ni siquiera admitan la instalación de la aplicación. Es importante conocer las diferencias entre plataformas operativas y las versiones que existan de ellas, ya que nunca se tendrá garantizada la retrocompatibilidad u operatividad futura del software desarrollado. También hay que tener cuidado al incorporar herramientas y librerías de terceras partes, puesto que no son admitidas por todos los equipos.

TOMA NOTA



Nunca un procesador más potente o una mayor existencia de memoria será la solución a una aplicación mal diseñada.

Por último, a pesar de la generalización de las redes móviles y con ella la conectividad de casi la totalidad de los dispositivos, las posibles limitaciones del ancho de banda y las desconexiones temporales de la red por falta de cobertura pueden llevar al fracaso de una aplicación si su lógica demanda una conexión constante a la red.

Por todo lo anterior, es recomendable que antes de llevar una aplicación a la fase de explotación, se pruebe y verifique en el más variado número de dispositivos posibles, utilizando, en caso de no poder hacerlo con equipos reales, los emuladores que suelen suministrarte los entornos de desarrollo.



Actividad propuesta 1.4

BQ Aquaris, Energy Phone, Wolder WIAM son algunos de los productos nacionales en el mundo de los smartphone. Localiza qué empresa nacional sigue comercializando dispositivos de marca propia, qué modelos ofertan y cuáles son sus características.

1.3. Sistemas operativos móviles

Al igual que en los ordenadores personales, el sistema operativo móvil es el conjunto de programas de bajo nivel que permiten la abstracción de las características propias del hardware y provee servicios a las aplicaciones que se ejecutan sobre él. La mayoría de los sistemas operativos móviles están basados en el modelo de capas con variada complejidad y multiplicidad a medida que se alejan de los elementos hardware. Simplificando cada una de ellas, podemos encontrar las siguientes:

- *Núcleo o kernel.* Proporciona el acceso a los elementos del hardware, ofreciendo servicios a las capas superiores a través de los controladores o drivers, la gestión de procesos, el sistema de archivos y la gestión de la memoria.
- *Middleware.* Conjunto de módulos que posibilitan la existencia de aplicaciones. Es transparente para el usuario y ofrece servicios como el motor de mensajería y comunicaciones, códec multimedia, intérpretes web, gestión del dispositivo y seguridad.
- *Entorno de ejecución de aplicaciones.* Gestor de aplicaciones e interfaces abierto que permite la programación por parte de los desarrolladores para la creación de software.
- *Interfaces de usuario.* Facilitan la relación con el usuario y se encargan de la presentación visual de la aplicación. Incluyen los componentes gráficos (botones, pantallas, listas, etc.) y el marco de interacción.
- *Aplicaciones nativas.* Propias de cada uno de los modelos y fabricantes.

1.3.1. Android

Lanzado en 2007, es el sistema operativo líder en el mercado móvil. Fue desarrollado por la Open Handset Alliance (OHA), una agrupación de 78 compañías lideradas por Google. Su principal cualidad es su carácter abierto, ya que se distribuye bajo dos tipos de licencias, una que abarca el código del kernel, la GNU GPLv2; y otra licencia para el resto de componentes del sistema, que se licencia bajo APACHE v2. La arquitectura de Android tiene las siguientes capas:

- a) *Kernel de Linux.* El tiempo de ejecución de Android (ART) se basa en el kernel de Linux, con funcionalidades como la generación de subprocessos y la administración de memoria de bajo nivel.



Investiga

Consulta las diferencias que existen entre ART (Android Runtime) y Dalvik, que fue la máquina virtual utilizada originalmente por Android.

- b) *Capa de abstracción de hardware (HAL).* Proporciona interfaces que enlazan las capacidades de hardware del dispositivo al framework de nivel más alto. Posee varias bibliotecas, con una interfaz para cada tipo de componente (cámara, bluetooth...).
- c) *Runtime.* Cada APP ejecuta sus propios procesos con sus propias instancias del tiempo de ejecución de ART, ya que está diseñado para ejecutar varias máquinas virtuales en dispositivos de memoria baja ejecutando archivos DEX, un formato de código de bytes específico de Android, y optimizado para ocupar un espacio de memoria mínimo.
- d) *Bibliotecas C/C++ nativas.* Muchos componentes y servicios se basan en código nativo, que requiere bibliotecas nativas escritas en C y C++. La plataforma Android proporciona la API para posibilitar la funcionalidad de estas bibliotecas.
- e) *Framework de la Java API.* Todo el conjunto de funciones del SO Android está disponible mediante API escritas en el lenguaje Java.
- f) *Apps del sistema.* Android incluye un conjunto de apps nativas para correo electrónico, mensajería SMS, calendarios, navegación en Internet.

Actividad propuesta 1.5



Localiza en qué versión de Android se dio soporte nativo a los sensores y en cuáles otras se incorporan Material Design, Digital Wellbeing o el Pixel Themes.

1.3.2. iOS

Desarrollado por Apple y originariamente denominado iPhone OS, es un derivado de Mac OS X. Su principal fuerte es la combinación, casi perfecta, entre hardware y software, así como el manejo de la pantalla multitáctil que lo convierte en un sistema operativo basado en la manipulación directa. Su arquitectura está basada en capas: las capas más altas contienen los servicios y tecnologías más importantes para el desarrollo de aplicaciones y las capas más bajas controlan los servicios básicos.

- *Cocoa Touch*: capa superior y más importante para el desarrollo de aplicaciones iOS. Es la que los usuarios utilizan para interactuar con las aplicaciones, es decir, la capa visible. Esta capa está formada fundamentalmente por dos Frameworks: UIKit (clases para el desarrollo de una interfaz de usuario) y Foundation Framework (acceso y manejo de objetos, servicios del SO).
- *Media Services*: provee los servicios de audio, gráficos y multimedia a la capa superior.
- *Core Services*: proporciona los servicios imprescindibles del sistema para poder ser utilizados por todas las aplicaciones (base de datos, acceso a la red).
- *Core OS*: núcleo del sistema con las características de bajo nivel (manejo de memoria, seguridad, drivers del dispositivo).



Figura 1.3
Arquitectura iOS.

Recurso web

www

En este código QR encontrarás un vídeo de Xataka TV con información sobre Fuchsia OS, un sistema operativo alternativo a Android en el que Google lleva trabajando desde 2016, basado en el kernel Zircon (anteriormente, Magenta):

1.3.3. Windows Phone

Desarrollado por Microsoft (antes llamado Windows Mobile), fue lanzado en el año 2010. Es un SO móvil compacto en el núcleo del sistema operativo Windows CE. Con un diseño similar a las versiones de escritorio de Windows, tiene la ventaja de la gran oferta de software de terceros. Entre sus novedades, destaca la denominada interfaz de usuario *Metro* basada en

el uso de mosaicos dinámicos con información útil para el usuario. Su arquitectura se basa en tres modelos:

- *Modelo de aplicación.* Las aplicaciones se despliegan en forma de paquete XAP, un archivo donde se encuentran los ensamblados y recursos originales de la aplicación.
- *Modelo de UI.* Modelo de interface que engloba el conjunto de interacciones que realiza un usuario sobre la aplicación.
- *Integración con la nube.* Integración con servicios como Exchange, Google Mail, Hotmail, Xbox Live, Skydrive, Facebook o Bing.

1.3.4. BlackBerry OS

Desarrollado por la RIM (*Research in Motion*), destaca por su capacidad de enviar y recibir correo electrónico a través de los operadores que ofrecen este servicio. Al igual que en Symbian, se pueden crear programas para este sistema operativo, pero es necesario tener cuenta de desarrollador de RIM.

Su arquitectura está basada en microkernel, que implementa una cantidad mínima de software en el núcleo, ejecutando otros procesos en el espacio de usuario que está fuera del kernel. Esto hace que sea menos vulnerable y más fácil de verificar. Otros elementos de su arquitectura son el gestor de arranque de la CPU, los controladores y servicios para soportar los subsistemas voz, datos, los servicios de plataforma y aplicaciones.

1.3.5. Symbian

Creado por la alianza de varias empresas de telefonía móvil (Nokia, Sony Ericsson, Samsung, Siemens), fue una evolución del sistema operativo Epoc, desarrollado por Psion en sus agendas electrónicas durante los 80. Entre sus características, podemos encontrar un eficiente uso de todos los recursos de la máquina (especialmente de la batería, la memoria RAM y la ROM).

Su arquitectura está basada en un microkernel: una mínima porción del sistema tiene privilegios de kernel, el resto se ejecuta con privilegios de usuario. Cada aplicación corre en sus propios procesos y tiene acceso solo a su propio espacio de memoria. Este sistema operativo perdió protagonismo con la llegada de iOS y Android, por lo que en la actualidad ha quedado relegado a un sector minoritario.

1.3.6. Palm OS P (WebOS)

Desarrollado originalmente en 1996 por la empresa Palm Inc para PDA's y conocido también como LG webOS, open webOS y HP webOS, es un SO multitarea basado en Linux, con una interfaz gráfica apoyada en tecnologías web (HTML5, JavaScript y CSS). Las aplicaciones se basan en el marco de una interfaz de usuario y de funciones establecidas para los servicios de acceso.

1.3.7. Firefox OS

Desarrollado por Mozilla Corporation, tiene núcleo Linux y está basado en HTML5. Está pensado para dispositivos móviles de gama baja y permite que las aplicaciones HTML5 puedan comunicarse directamente con el hardware usando JavaScript y Open Web APIs.

1.3.8. Ubuntu Touch

Desarrollado por Canonical Ltd. Está basado en Linux, aunque posee una interfaz que puede ser utilizada en ordenadores de sobremesa, portátiles, netbooks, tablets.

1.3.9. Harmony OS

Novedoso sistema operativo de Huawei Technologies Co., Ltd. Distribuido y de alto rendimiento, diseñado para el interconexión en Internet de las Cosas, al ser compatible con múltiples dispositivos. Es un sistema operativo de código abierto con arquitectura basada en microkernel, centrado en servicios básicos (solo gestiona las instrucciones del chip, la comunicación entre procesos y la información de seguridad). Un motor de compilación, ARK, establece las prioridades de ejecución de tareas, anticipándose en la reserva de los límites de tiempo. El resto de los procesos se ubican fuera del núcleo, no interfiriendo entre sí cuando están en ejecución. Pueden, además, cada uno de ellos, depurarse, desarrollarse y actualizarse por separado.



Actividad propuesta 1.6

Realiza un estudio de los sistemas operativos de móviles que más frecuentemente se venden en España, en la actualidad, y compáralo con otros países, como Japón y EE.UU.

1.4. Lenguajes de programación para dispositivos móviles

El mundo de desarrollo de aplicaciones para móviles dispone de infinidad de herramientas, lenguajes y entornos para elegir antes de sentarte a codificar. La primera opción que se ha de tomar es el tipo de lenguaje de programación por el que se va a optar.

1.4.1. Desarrollo nativo

Aunque existen defensores de cada opción, el desarrollo nativo suele ser la mejor decisión, a pesar de que cada plataforma (iOS, Android...) utiliza lenguajes y paradigmas de programación diferentes y herramientas propias, pero a cambio se obtiene gran flexibilidad, adaptación total al entorno y un máximo rendimiento.

La principal desventaja de hacer que una aplicación sea compatible con más de un entorno es que se tendrá que dominar más de un lenguaje y distintas herramientas, lo que aumentará el tiempo de desarrollo. Por ello, generalmente, los programadores se suelen especializar en una plataforma, sacándole el máximo rendimiento a la programación en la misma.

1.4.2. Desarrollo multiplataforma compilado a nativo

Una opción intermedia es utilizar plataformas mixtas que permitan independizar el desarrollo del lenguaje nativo de cada plataforma móvil. Con un único lenguaje, se podrán crear

aplicaciones para todos los sistemas operativos, aunque siempre habrá que realizar pequeñas adaptaciones.

La más conocida es Xamarin, basada en el lenguaje C# de Microsoft y en la plataforma .NET, que gracias a sus herramientas permite crear aplicaciones multiplataforma, reutilizando gran parte del código (a excepción de la interfaz).

Las aplicaciones escritas con Xamarin se compilan a código nativo en cada plataforma, lo que le da el mismo rendimiento que el de una aplicación nativa. También tiene acceso directo nativo a todas las API de cada plataforma, así como a los controles de interfaz de usuario nativos.

1.4.3. Desarrollo multiplataforma basado en HTML5

Últimamente es una opción muy popular, especialmente si se conoce el mundo web, el utilizar alguna herramienta basada en HTML que genere aplicaciones para todas las plataformas.

Aunque existen muchas, quizás la más conocida es PhoneGap/Apache Cordova. Además, en las apps escritas en HTML5 y compiladas con estas herramientas podemos utilizar un *skin* concreto para que el aspecto sea lo más similar posible al de las aplicaciones nativas. Se puede obtener gran parte de la funcionalidad nativa del dispositivo móvil a través de librerías JavaScript. Evidentemente, estas facilidades tienen un precio, y es que las aplicaciones desarrolladas por este sistema no tienen el mismo rendimiento que una APP nativa; tampoco se tendrá acceso a todas las API nativas de cada plataforma, aunque sí a las más importantes.



Investiga

Últimamente se habla de reemplazar las apps nativas por las *Progressive Web Apps*, una generación de aplicaciones que bordean la frontera entre aplicaciones web y apps móviles tradicionales, combinando las ventajas de ambas, pero ¿sabes en qué consisten?

Como conclusión de este apartado, y para facilitar la elección, es recomendable que si se va a programar tan solo en una plataforma y no se conoce C# ni HTML, se aprendan las herramientas y el lenguaje nativo: Objective-C o Swift y Cocoa Touch para iOS, Java y el SDK de Android para Android, y C#/XAML para Windows Phone.

En caso de querer realizar aplicaciones para todas las plataformas, según los conocimientos de C# o HTML5, se puede optar por una solución con Xamarin o PhoneGap. Se puede concretar la cuestión con los conocimientos de los que se parte y las plataformas en las que se quiere programar.

1.5. Entornos integrados de desarrollo de aplicaciones móviles

Como es fácil imaginar si se opta por la programación nativa, es aconsejable familiarizarse con las herramientas de desarrollo propias de esa plataforma. En este apartado se hará un breve recorrido por los lenguajes e IDE que es aconsejable conocer en cada caso.

 PARA SABER MÁS

AWS Amplify es una biblioteca JavaScript que consta de un marco de desarrollo y servicios, de código abierto, que ofrece una forma rápida y sencilla de compilar aplicaciones móviles (iOS, Android) y web. Las herramientas para desarrolladores incluyen la consola de AWS Amplify, que permite compilar, implementar y alojar aplicaciones web, y probar aplicaciones móviles en dispositivos reales de iOS y Android.

1.5.1. Symbian

Para programar aplicaciones en Symbian no es necesario ningún conocimiento específico de un código único para este sistema, ya que es posible programar en él a partir de lenguajes como Java, C++ Visual Basic, Python, Perl, Flash Lite.

Origo IDE es un entorno de desarrollo apropiado para la programación de móviles Symbian de última generación, que incluye un editor de código, compilador y emulador, basándose en un sencillo lenguaje de script que proporciona resultados similares al uso del C++ nativo de Symbian. Tiene el inconveniente de que es una herramienta propietaria del alto costo, aunque tiene algunas versiones demo.

1.5.2. BlackBerry OS

BlackBerry Java Development Environment es un entorno completamente integrado de desarrollo y simulación para crear BlackBerry® Java Application para dispositivos BlackBerry. Gracias a BlackBerry JDE, puedes crear aplicaciones con el lenguaje de programación Java® ME y las API extendidas de Java para BlackBerry. Este IDE incluye herramientas de edición y depuración optimizadas para el desarrollo de BlackBerry Java Application.

1.5.3. Windows Phone

Esta plataforma soporta los lenguajes C# y Visual Basic.NET. Para el diseño se utiliza el lenguaje Silverlight (también conocido como XAML), y para videojuegos se puede utilizar XNA, que genera gráficos 2D/3D.

Visual Studio 2010 es una herramienta adecuada para programar aplicaciones para esta plataforma. Se necesita tener instalado Windows Phone SDK 7.1 que suministra API y otras herramientas.

1.5.4. iOS

Swift es un innovador lenguaje de programación para Cocoa y Cocoa Touch, con mejoras sobre Objective-C, que es mucho menos transigente con los errores y una sintaxis más complicada. La combinación de Swift con Xcode facilita y agiliza el desarrollo para esta plataforma, además de poseer un intuitivo interfaz (Interface Builder, una herencia de NeXT) y buenas herramientas de depuración.

1.5.5. Android

Java es el lenguaje nativo que usa Android. Las aplicaciones que hacen uso del hardware y se comuniquen con el sistema operativo, usarán este código. Es un lenguaje multiplataforma útil para el desarrollo de apps móviles y Desktop; es robusto, orientado a objetos, de arquitectura neutral y con la gran ventaja de las numerosas librerías que puedes encontrar en los repositorios.

Paralelo a Java se ha de conocer XML, un lenguaje de marcas similar a HTML. Es una especificación de recomendación W3C como lenguaje de marcado de propósito general. Será muy útil su conocimiento para crear las vistas de las aplicaciones y gestionar los recursos.

Como herramienta de desarrollo, tiene variadas opciones, pero quizás la más adecuada sea Android Studio, un IDE basado en el software IntelliJ IDEA de JetBrains y lanzado para reemplazar a Eclipse como el IDE oficial en el desarrollo de aplicaciones para Android. Es una herramienta completa para el desarrollo y la depuración de aplicaciones Android, con un sistema flexible de compilación y despliegue, además de muchas otras utilidades que facilitan la prueba de apps en variados dispositivos mediante su herramienta AVD (Android Virtual Devices).



PARA SABER MÁS

Google está realizando una fuerte apuesta por Kotlin un lenguaje de programación desarrollado por JetBrains cuya sintaxis es más limpia que Java, con menos necesidad de código y que permite la coexistencia simultánea con Java.

Figura 1.4
Kotlin.



Resumen

- Las características fundamentales de los dispositivos móviles van a depender de la capacidad de procesamiento, tamaño, portabilidad y conectividad.
- Las categorías en las que podemos encuadrar los dispositivos móviles según los estándares de la T38 y DuPont Global Mobility Innovation Team son Limited Data Mobile Device (Dispositivo Móvil de Datos Limitados), Basic Data Mobile Device (Dispositivo Móvil de Datos Básicos) y Enhanced Data Mobile Device (Dispositivo Móvil de Datos Mejorados).
- Los tipos de dispositivos móviles que puedes encontrar en el mercado son: teléfonos móviles, handhelds, netbooks, E-Book Readers, tablets, PDAs, smartphones, gadgets.
- Las comunicaciones móviles se encuadran en generaciones, iniciadas por la 0 (ondas de radio), hasta la actual 5G, que puede llegar a 1 Gbps de velocidad de transmisión.
- Los terminales móviles han ido evolucionando con el tiempo, reduciendo el tamaño, aumentando la capacidad de cálculo, prestaciones, conectividad... En la actualidad, se ha llegado a un mercado en el que, además de los grandes fabricantes tradicionales (Apple y Samsung), se están incorporando nuevas empresas altamente competitivas en el sector.