

DESARROLLO DE VIDEOJUEGOS PARA DISPOSITIVOS MÓVILES

PROYECTO EDITORIAL: TECNOLOGÍAS DIGITALES



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.

DESARROLLO DE VIDEOJUEGOS PARA DISPOSITIVOS MÓVILES

Daniel Burón García
Mario Manzano Vázquez
Maximiliano Paredes Velasco



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

© Daniel Burón García
Mario Manzano Vázquez
Maximiliano Paredes Velasco

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

ISBN: 978-84-1357-058-7
Depósito Legal: M-82-2021

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 cualquier otro, sin la autorización previa por escrito de Editorial Síntesis, S. A.

Contenido

<i>Introducción</i>	11
1. Videojuegos: definición y clasificación	13
1.1. Concepto de juego	13
1.1.1. <i>La búsqueda de una definición formal</i>	14
1.1.2. <i>La búsqueda de la definición de juego en el mundo del videojuego</i>	15
1.1.3. <i>Definición de videojuego en el ámbito de este libro</i>	17
1.2. Clasificación	17
1.2.1. <i>Plataforma</i>	18
1.2.2. <i>Modelo de negocio</i>	18
1.2.3. <i>Regional y contenido</i>	19
1.2.4. <i>Género</i>	20
1.2.5. <i>Jugadores</i>	24
1.2.6. <i>Estilo visual</i>	25
1.2.7. <i>Representación</i>	26
1.2.8. <i>Realidad extendida</i>	27
1.2.9. <i>Otras clasificaciones</i>	27
1.3. Componentes y arquitectura	29
1.3.1. <i>Arquitectura genérica</i>	30
1.4. Resumen	34
2. Características específicas de los dispositivos móviles	35
2.1. Ecosistema de los dispositivos móviles	35
2.1.1. <i>Formatos de hardware</i>	36
2.1.2. <i>Sistemas operativos</i>	36

2.2.	Hardware específico	38
2.3.	Publicación y distribución	44
2.4.	Software específico: servicios de Google Play	45
2.5.	Fundamentos de desarrollo con Android Studio	47
2.5.1.	<i>Inicializando Android</i>	47
2.5.2.	<i>Creación de un proyecto</i>	48
2.5.3.	<i>Interfaz de usuario principal</i>	50
2.5.4.	<i>Depuración y pruebas</i>	51
2.5.5.	<i>Generación del fichero .apk</i>	53
2.6.	Resumen	53
3.	<i>El proceso de desarrollo</i>	55
3.1.	Metodologías de desarrollo de videojuegos	55
3.2.	Análisis	60
3.2.1.	<i>Definiendo el objetivo</i>	61
3.2.2.	<i>Análisis conceptual</i>	62
3.2.3.	<i>Análisis de usuarios</i>	63
3.2.4.	<i>Análisis de la experiencia</i>	65
3.2.5.	<i>Análisis técnico</i>	65
3.2.6.	<i>Análisis de mercado</i>	66
3.3.	Diseño	66
3.4.	Implementación	68
3.4.1.	<i>El ciclo de desarrollo</i>	69
3.4.2.	<i>Principios y patrones de diseño software</i>	71
3.5.	Testing	73
3.6.	Publicación	74
3.7.	Resumen	76
4.	<i>Motores de videojuegos</i>	77
4.1.	Funcionalidades de los motores de videojuegos	77
4.2.	Motores de videojuegos para dispositivos móviles	79
4.3.	Arquitectura y componentes en Phaser	81
4.3.1.	<i>El desarrollo web</i>	81
4.3.2.	<i>El entorno de desarrollo en Phaser</i>	84
4.3.3.	<i>Estructura</i>	86
4.3.4.	<i>Módulos (Modules)</i>	87
4.3.5.	<i>Juego (Game)</i>	87
4.3.6.	<i>Escenas (Scenes)</i>	88
4.3.7.	<i>Objetos de juego (GameObjects)</i>	88

4.4.	Arquitectura y componentes en Unity	89
4.4.1.	<i>Escenas</i>	90
4.4.2.	<i>Objetos</i>	90
4.4.3.	<i>Componentes</i>	91
4.4.4.	<i>Prefabs</i>	92
4.4.5.	<i>Materiales</i>	92
4.4.6.	<i>Materiales físicos</i>	93
4.5.	Resumen	93
5.	Marcianos: <i>desarrollo de juego con Android Studio</i>	95
5.1.	Configuración del entorno	95
5.2.	Análisis	96
5.3.	Diseño	99
5.4.	Implementación	101
5.4.1.	<i>Pantalla de inicio</i>	101
5.4.2.	<i>Objetos del juego</i>	106
5.4.3.	<i>Motor del juego</i>	112
5.4.4.	<i>Gestión del control de la nave</i>	124
5.4.5.	<i>Gestión de colisiones</i>	127
5.4.6.	<i>Integrando sonido</i>	133
5.4.7.	<i>Finalizando el juego</i>	136
5.4.8.	<i>Integrando Google Play Game Services API</i>	136
5.5.	Testeo	148
5.6.	Publicación	150
5.7.	Resumen	152
6.	Deliveracer: <i>desarrollo de juego con Phaser</i>	153
6.1.	Configuración del entorno	154
6.1.1.	<i>Editor de código</i>	155
6.1.2.	<i>Navegador web</i>	155
6.1.3.	<i>Servidor web</i>	156
6.1.4.	<i>Creación de aplicación móvil</i>	157
6.1.5.	<i>Configuración de Android Studio y de desarrollo Android</i>	158
6.1.6.	<i>Crear el proyecto</i>	159
6.2.	Análisis	161
6.2.1.	<i>Objetivo</i>	161
6.2.2.	<i>Análisis conceptual</i>	162
6.2.3.	<i>Análisis de usuarios</i>	162

6.2.4.	<i>Análisis de la experiencia</i>	163
6.2.5.	<i>Análisis técnico</i>	163
6.2.6.	<i>Análisis de mercado</i>	164
6.3.	Diseño	165
6.3.1.	<i>Escenas e interfaz de usuario</i>	165
6.3.2.	<i>Personajes</i>	167
6.3.3.	<i>Datos de juego</i>	167
6.3.4.	<i>Niveles</i>	169
6.3.5.	<i>Aspectos visuales</i>	170
6.3.6.	<i>Aspectos auditivos</i>	171
6.4.	Implementación	171
6.4.1.	<i>El archivo config.xml y los complementos</i>	172
6.4.2.	<i>Recursos visuales y auditivos</i>	174
6.4.3.	<i>Resumen del código</i>	175
6.4.4.	<i>Index.html</i>	177
6.4.5.	<i>Utils.js</i>	179
6.4.6.	<i>index.js</i>	191
6.4.7.	<i>Boot.js</i>	192
6.4.8.	<i>Preloader.js</i>	193
6.4.9.	<i>MainMenu.js</i>	195
6.4.10.	<i>Settings.js</i>	198
6.4.11.	<i>Upgrades.js</i>	199
6.4.12.	<i>Levels.js</i>	201
6.4.13.	<i>Game.js</i>	203
6.4.14.	<i>Achievements.js</i>	230
6.5.	Testeo	231
6.5.1.	<i>Pruebas en el navegador</i>	232
6.5.2.	<i>Pruebas en dispositivos</i>	234
6.5.3.	<i>Generación de archivo de instalación</i>	234
6.6.	Publicación	235
6.7.	Resumen	237
7.	Pangkanoiders: desarrollo de juego con Unity	239
7.1.	Configuración del entorno	240
7.2.	Análisis	243
7.2.1.	<i>Configuración para dispositivos móviles</i>	243
7.2.2.	<i>Escenas</i>	243
7.2.3.	<i>Interfaz de usuario y almacenamiento de datos</i>	244
7.2.4.	<i>Área de juego</i>	244

7.2.5.	<i>Enemigos</i>	245
7.2.6.	<i>Jugador</i>	245
7.2.7.	<i>Bloques</i>	246
7.2.8.	<i>Explosiones</i>	246
7.2.9.	<i>Gestión de partida</i>	246
7.3.	Diseño	246
7.3.1.	<i>Materiales</i>	247
7.3.2.	<i>Materiales físicos</i>	248
7.3.3.	<i>Área de juego</i>	249
7.3.4.	<i>Jugador</i>	250
7.3.5.	<i>Bolas</i>	252
7.3.6.	<i>Bloques</i>	253
7.3.7.	<i>Plataformas</i>	254
7.3.8.	<i>Bombas</i>	254
7.3.9.	<i>Balas</i>	255
7.3.10.	<i>Explosiones</i>	256
7.4.	Implementación	257
7.4.1.	<i>Scripts auxiliares</i>	259
7.4.2.	<i>Escena Level</i>	264
7.4.3.	<i>Escena Main</i>	281
7.4.4.	<i>Escena Ranking</i>	283
7.5.	Testeo y publicación	285
7.5.1.	<i>Pruebas con Unity Remote</i>	285
7.5.2.	<i>Generación de archivo de instalación</i>	286
7.6.	Resumen	287
<i>Referencias</i>		289

2

Características específicas de los dispositivos móviles

Para desarrollar videojuegos dirigidos a dispositivos móviles debemos conocer sus características y peculiaridades frente a otras plataformas. Su rendimiento creciente y su gran versatilidad, gracias a su pequeño formato y sus múltiples sensores, hacen de ellos un sistema muy recomendable para jugar. Para sacar el máximo provecho de estas circunstancias, debemos conocerlas y apoyarnos en las herramientas actuales que se ofrecen a los desarrolladores, de forma que podamos crear juegos con funciones avanzadas en menos tiempo y más fácilmente.

En este capítulo vamos a tratar el entorno de los dispositivos móviles desde varios puntos de vista. Inicialmente consideraremos de forma general los diversos dispositivos existentes en el panorama actual. En el primer apartado veremos el hardware concreto que podemos encontrarnos en estos dispositivos. Posteriormente mostraremos los canales que tenemos a nuestra disposición para llevar los videojuegos a estos dispositivos. Las dos últimas secciones se centran en el caso concreto de los dispositivos Android. Por un lado, los servicios que aporta Google para ayudar al desarrollo y por otro lado los principios de éste con Android Studio.

2.1. Ecosistema de los dispositivos móviles

Gracias a la evolución tecnológica, contamos cada vez con aparatos electrónicos más potentes y compactos. Dentro de los dispositivos móviles tenemos una enorme variedad y hablar de todos ellos sería prácticamente imposible, incluso si nos limitáramos a los disponibles en el mercado en una gama concreta. Para dar una idea general, vamos a clasificarlos en grupos, basados en su formato y su sistema operativo.

2.1.1. Formatos de hardware

Aunque hoy en día podemos encontrar multitud de dispositivos móviles de diversos tamaños, potencias y casos de uso, en el sector específico del videojuego para móviles tenemos estos grandes grupos:

- *Smartphones*: los principales dispositivos móviles por su importancia en el mercado y enorme popularidad. Aúnan en el mismo dispositivo no solo un sistema de comunicación por voz, sino también prácticamente un ordenador completo con el que jugar, navegar, chatear, ver películas, escuchar música, etc. Todo en un tamaño compacto que podemos llevar cómodamente el bolsillo. En este caso, los cinco primeros fabricantes, que controlan aproximadamente un 75 % del mercado son: Samsung (32 %), Apple (23 %), Huawei (10 %), Xiaomi (8 %) y Oppo (4,5 %).
- *Tablets*: similares a los anteriores en características técnicas. Al utilizar componentes parecidos, la gran diferencia con los smartphones se encuentra en la pantalla, donde su tamaño y resolución suelen ser la prioridad (normalmente entre 7 y 12", y con resoluciones por encima de Full HD). En este caso, los 5 principales fabricantes son: Apple (73 %), Samsung (15 %), Amazon (3 %), Asus (1 %) y Huawei (1 %).
- *Phablets*: mención especial tienen las phablets, cuyo nombre viene de combinar *phone* y *tablet*, y que normalmente se definen como smartphones de gran pantalla en los que predomina su capacidad multimedia frente a su portabilidad. Actualmente se han integrado prácticamente dentro del formato smartphone, con tamaños cada vez mayores.
- *Consolas*: diseñadas específicamente para jugar, suelen contar con una mayor capacidad gráfica y mandos de control físicos, mientras que sus características en cámaras, sensores y comunicación suelen ser más limitadas, ya que no son su prioridad. Los principales fabricantes eran Nintendo y Sony, y ya han decidido discontinuar sus últimas consolas portátiles. En la actualidad, queda en el mercado únicamente la consola Nintendo Switch, pensada tanto para sobremesa como portátil.

2.1.2. Sistemas operativos

Al margen de su formato, hay que tener en cuenta el software que los gestiona. Ya comentamos los sistemas operativos móviles brevemente en el capítulo 1, pero vamos a refrescar ahora con algo más de detalle los grandes grupos existentes en la actualidad. Hay que tener en cuenta, además, que en este capítulo y en el resto del libro nos cen-

tramos sobre todo en el desarrollo de juegos para Android, debido a los motivos que veremos a continuación. A pesar de ello, en los capítulos 6 y 7, comprobaremos que a través del desarrollo web e híbrido es fácil portar los juegos a otras plataformas. Los principales sistemas operativos son los siguientes:

- Android*: desarrollado por Google y basado en el kernel de Linux. Tenemos dos vertientes principales, por un lado, la de núcleo abierto Android Open Source Project (AOSP), que permite a cualquier fabricante crear su propia variante, y por otro lado la de software privativo con Google Mobile Services (GMS). El lenguaje oficial de desarrollo de Android es Java, aunque también puede desarrollarse en Kotlin, C y C++, además de contar con otras opciones como ya veremos en el capítulo 4, 6 y 7. La última versión disponible en su momento era Android 10 (octubre de 2020) y cuenta con más del 75 % de la cuota del mercado y con más de 2 000 millones de usuarios activos mensuales.

Salvo Apple y Nintendo, prácticamente el resto de fabricantes usa Android, por lo que tenemos una enorme variedad de dispositivos de diversas gamas, precios y características. Debido a la enorme variedad del hardware, pueden existir además algunas incompatibilidades en algunas aplicaciones y juegos, aunque se compensa con mucha más libertad y variedad para el usuario. Android cuenta con casi 3 millones de aplicaciones en Google Play, su tienda de aplicaciones. Gracias a la apertura de Android, existen otras tiendas de aplicaciones, destacando la de Amazon, y es más fácil para los desarrolladores publicar sus creaciones, pero por el contrario la fragmentación y libertad supone más competencia y dificultad para sacar un mayor beneficio económico.
- iOS*: creado por Apple para sus smartphones iPhone y tablets iPad, utiliza el kernel XNU y es el gran rival de Android. Aunque solo cuenta con menos del 23 % de la cuota de mercado de smartphones, controla el 73 % de la cuota de tablets. En su caso, el lenguaje oficial de desarrollo es Objective-C, y aunque hay otras alternativas como en Android, no podremos desarrollar en Windows o Linux, y estaremos limitados a un ordenador macOS o un servicio externo. Su última versión disponible en su momento era iOS 13 y solo se utiliza en los dispositivos propios de Apple, que cuentan con hardware específico y controlado. Contar con solo unos pocos dispositivos y que la misma compañía controle el hardware y software supone menos libertad para los usuarios y desarrolladores, pero a cambio reduce al mínimo las incompatibilidades y simplifica la experiencia. Esto se extiende a su vez a su tienda de aplicaciones Apple Store, la única oficial y legal disponible para iOS, con un férreo control por parte de Apple y que dispone de casi 2 millones de aplicaciones. La App Store genera muchos más ingresos que Google Play, a pesar de tener menos dispositivos, y

las aplicaciones suelen ser más rentables para los desarrolladores que Android, pero por el contrario es mucho más difícil publicar y actualizar.

- *Windows*: si bien Microsoft dejó de desarrollar Windows 10 Mobile en 2017 y a finales de 2019 finalizó su soporte, en el caso de las tabletas es muy habitual encontrarlas con Windows 10. Se trata de dispositivos más cercanos a ordenadores y por tanto no nos centraremos en ellos al quedar fuera del ámbito de este libro.
- *Otros*: aunque los anteriores son los principales sistemas y controlan prácticamente el 99 % del mercado, existen también otros sistemas operativos menos populares, tanto de código abierto como cerrado. En algunos casos son de uso generalista, mientras que en otros están diseñados especialmente para jugar a videojuegos o para máquinas concretas, como en el caso de las consolas portátiles.

2.2. Hardware específico

Los dispositivos móviles han conseguido llevar la informática prácticamente a todo el mundo, no solo por permitir que podamos llevar siempre con nosotros prácticamente un ordenador completo con multitud de elementos y posibilidades, sino también por simplificar su uso y hacerla más asequible. Mejoras en la eficiencia de sus componentes hardware y las baterías permiten una renovación continua de las capacidades y formatos de los dispositivos móviles, de manera que sus especificaciones mejoran cada año, fomentadas además por la gran competencia entre fabricantes ante unas expectativas crecientes por parte de los usuarios.

Esta sección hace un breve resumen de los principales elementos de hardware de un dispositivo móvil actual, pero no tiene en cuenta las gamas y opciones concretas existentes debido a la enorme variedad existente. Dependiendo del juego que desarrollemos y de sus necesidades, deberemos tener en cuenta el rango de gamas que podrán ejecutarlo, con las posibles optimizaciones y ajustes para adaptarlo a una mayor audiencia potencial:

- *System on a Chip (SoC)*: el cerebro del dispositivo, integra varios módulos en un solo chip, lo que simplifica su fabricación y diseño. Su enorme variabilidad dificulta conocer su potencia real solo con sus datos técnicos, por lo que es recomendable apoyarse en pruebas de rendimiento comparativas, como Geekbench, GFXBench, 3DMark o AnTuTu. Destacan los Snapdragon de Qualcomm, Exynos de Samsung, Kirin de HiSilicon y los Apple Ax. Sus elementos principales son:
 - *Central Processing Unit (CPU)*: ejecuta el software del dispositivo. Se define por su modelo, número de núcleos, frecuencia de reloj y proceso

de manufactura, con los modelos basados en la arquitectura ARM como los más habituales. Podemos encontrarnos CPU de 2 a 8 núcleos, con frecuencias de 1 a 3 gigahercios (GHz) y procesadores de 22 a 7 nanómetros (nm). Hay que destacar que más núcleos y una mayor frecuencia no significan siempre un mejor rendimiento, ya que el modelo concreto de procesador es muy importante, por lo que suelen usarse pruebas de rendimiento para compararlos.

- *Graphics Processing Unit* (GPU): optimizada para operaciones específicas relacionadas con los gráficos. No son tan versátiles como las CPU, pero a cambio procesan rápidamente grandes cantidades de datos en paralelo. Aunque se pueden definir por sus características como las CPU, sus detalles técnicos quedan fuera del alcance de este libro.
 - *Otros*: junto con los anteriores, es muy habitual que un SoC integre también otros componentes, como el módem, GPS, procesador de imagen para las cámaras y otros.
- *Memoria*: junto al SoC, la memoria es imprescindible para el funcionamiento del dispositivo. Distinguimos entre dos tipos:
 - *Almacenamiento*: sirve para almacenar los datos, ya sean los propios de su sistema y aplicaciones, como imágenes o vídeos que captamos con sus cámaras y documentos de todo tipo. Utiliza módulos de memoria flash de estado sólido. Se mide en gigabytes (GB), y su capacidad varía normalmente entre 8 y 512 GB, con 32 GB como más habitual y la posibilidad de ampliarla usando tarjetas microSD.
 - *Memoria de acceso aleatorio* (RAM): carga el kernel del sistema operativo y las instrucciones que ejecuta el SoC. Es una memoria temporal, sus datos se pierden al apagar el dispositivo, pero también muy rápida, algo esencial para el rendimiento. También se mide en GB, con valores actuales entre 1 y 12 GB, siendo 4 GB lo más habitual.
 - *Pantalla*: componente básico de cualquier dispositivo móvil para mostrar la imagen y ofrecer interacción a través de un panel táctil, normalmente de 10 puntos y con tecnología capacitiva. Sus principales características son:
 - *Tamaño*: se define a partir de las pulgadas que mide su diagonal y en la actualidad puede ir de 4 a 7 pulgadas para smartphones, con 5,5 pulgadas como más habitual, y de 7 a 12 pulgadas para tablets, predominando las de 10 pulgadas.

- *Resolución*: indica la cantidad de píxeles que forman su imagen dando su dimensión vertical y horizontal, lo que define su relación de aspecto. Al igual que el tamaño, es muy variable y actualmente las pantallas suelen ser entre 1 280x720 píxeles (HD) y 3 840x2 160 píxeles (4K), predominando la relación de aspecto 16:9 o similares.
 - *Densidad*: marcada por la relación entre el tamaño y resolución, se suele definir en píxeles por pulgada (ppi) y varía de 200 a 500 ppi.
 - *Tecnología*: tenemos dos opciones principales en la actualidad. Por un lado, Liquid Crystal Display (LCD), más barata y que requiere de una fuente de luz, con la variante IPS-LCD de mayor calidad y gran popularidad. Por otro lado, Organic Light-Emitting Diode (OLED), más cara y en la que cada punto emite luz, por lo que no requiere una fuente extra, con sus variantes P-OLED y AMOLED, de mayor calidad y más costosa.
- *Batería*: esencial para usar un dispositivo sin cables que lo conecten a la red eléctrica. Las más utilizadas actualmente son de iones de litio (Li-Ion), más baratas, seguidas de las de polímeros de litio (LiPo), más caras y avanzadas. Con un voltaje de 3,7 voltios (V) por celda, su capacidad se mide en miliamperios hora (mAh), y va de los 2 000 mAh a 5 000 mAh en smartphones, y de los 4 000 mAh a 12 000 mAh en tablets. Aunque en algunos dispositivos son removibles, normalmente las baterías están integradas. Para cargarlas se usa un conector por cable, aunque también pueden cargarse inalámbricamente, generalmente con el estándar Qi.
 - *Cámaras*: cada vez más importantes en los dispositivos móviles. Permiten tomar fotos y vídeos, o incluso la captura de datos biométricos y del entorno. Es muy habitual que cuenten con varias, al menos una en el frente con la pantalla y otra en la parte trasera, pero hoy en día hay dispositivos con cinco o más. Sus características vienen definidas por:
 - *Lente*: se encarga de trasladar la imagen del mundo real al sensor, y es la parte más relevante para la calidad de la misma. Normalmente tiene varios componentes ópticos y marca por un lado la apertura, que define la cantidad de luz que deja pasar y su profundidad de campo. La apertura se expresa a través del número f , que relaciona la distancia focal y apertura para indicar la luminosidad del objetivo, siendo $f/2,0$ más luminoso y con mayor apertura que $f/2,8$. La profundidad de campo indica el área de espacio enfocada en la imagen, y a mayor apertura una profundidad de campo menor. Por otro lado, la lente define el ángulo de visión y distancia focal, contando con lentes que van del gran angular, con una visión más

- extensa del entorno, a teleobjetivos, con una visión más estrecha y ampliación de la imagen.
- *Sensor*: convierte en señales eléctricas la imagen mostrada por la lente. Viene definido inicialmente por su tecnología, con el dispositivo de carga acoplada (CCD) por un lado, más caro, de mayor consumo y lectura de la matriz simultánea. Por otro lado, el semiconductor complementario de óxido metálico (CMOS), más barato y eficiente, pero de lectura secuencial, y con la variante de iluminación posterior (BSI), que mejora su luminosidad y rendimiento con poca luz. Otra característica importante es su resolución, expresada en megapíxeles, es decir, su número total de píxeles, con valores entre 3 y 40 megapíxeles (MP). Por último, su tamaño es importante, ya que un sensor mayor es capaz de captar más luz y reduce el ruido. Suele expresarse relativo a su resolución a través del tamaño de píxel, que normalmente va de 1 a 2 micras.
 - *Procesador de señal de imagen (ISP)*: se encarga de procesar las señales eléctricas del sensor a su formato digital final. El ISP analiza, modifica y optimiza la señal usando complejos algoritmos y procesamiento en paralelo. Su rendimiento y velocidad es crucial para la calidad de las imágenes y el vídeo, ya que a pesar de que el sensor sea de muchos megapíxeles, es el ISP el que define la resolución y tasa máxima posible de fotogramas por segundo, yendo de formatos 640x480 (VGA) a 4K, con velocidades de 24 a 60 fotogramas por segundo (fps) para vídeos normales, y de 120 a 960 fps para cámara lenta.
 - *Extras*: cuentan con características como flash LED, autoenfoco por detección de contraste (CDAF), detección de fase (PDAF) o láser, así como estabilización de imagen óptica o electrónica.
 - *Uso*: dependiendo del tipo de cámara y la función a la que está destinada, puede ver el espectro visible o infrarrojo, captar la profundidad o el calor, centrarse en huellas dactilares, la captura del iris, o las facciones de la cara, entre otros.
- *Audio*: para la comunicación por voz y disfrutar de contenidos multimedia es esencial contar con sistemas de audio.
 - *Altavoces y auriculares*: es muy común que existan altavoces integrados, o al menos un puerto por cable o inalámbrico, generalmente Bluetooth, para conectar auriculares.
 - *Micrófonos*: según la importancia de las comunicaciones en el dispositivo, podemos contar con tan solo un micrófono o multitud de ellos, dispuestos para mejorar la calidad del sonido captado y la reducción del ruido.

- *Sensores de posición:* parte de la versatilidad de los dispositivos viene de su capacidad para conocer su posición y orientación mediante estos sensores:
 - *Inerciales:* por un lado, los acelerómetros miden la aceleración, por otro lado, los giroscopios miden la velocidad angular. Ambos miden en un eje y se usan tres de cada tipo combinados para obtener seis grados de libertad y dar la orientación y movimientos relativos.
 - *Magnetómetros:* miden la fuerza de un campo magnético y se usan para calibrar la orientación absoluta usando el campo terrestre. Se combinan con los inerciales para tener nueve grados de libertad, o con pequeños imanes para detectar el cierre de fundas o como botones.
 - *Geolocalización:* mide la posición en exteriores a través de señales de satélite y con una precisión variable que va de centímetros a metros. Aunque el sistema GPS, controlado por EE.UU., es el más habitual, existen otras alternativas como Galileo (UE), GLONASS (Rusia), BDS (China) y QZSS (Japón).
- *Sensores de entorno:* con ellos detecta el estado del ambiente que le rodea y entre los más importantes y usados nos encontramos con sensores de:
 - *Proximidad:* normalmente un sensor óptico para detectar si hay algo que bloquea o está cerca de la pantalla, o para reconocer gestos.
 - *Luz:* mide la luz ambiental y permite ajustar el brillo de la pantalla para optimizar el consumo y la comodidad.
 - *Otros:* algunos dispositivos disponen además de barómetro, para medir la presión del aire, termómetro, para la temperatura ambiental, e higrómetro para la humedad relativa.
- *Comunicaciones:* gracias a estos componentes, el dispositivo puede comunicarse con otros de forma inalámbrica, algo esencial para su portabilidad. Los principales tipos de comunicación son:
 - *Módem:* la base de la transmisión de voz y datos en móviles se lleva a cabo a través de su módem y señales de radiofrecuencia. La constante evolución de esta tecnología y la enorme variedad de estándares y frecuencias quedan lejos del alcance de este libro, pero haremos un breve resumen. Se utiliza una tarjeta SIM para identificarse en la red y las bandas más importantes son 800, 900, 1 800, 2 100, 2 600 y 3 500 megahercios (MHz) dependiendo del país y la red. La red se clasifica por su generación (2G a 5G) o el nombre concreto de su estándar (GSM, GPRS, EDGE, HSPA, LTE...). La velocidad depende de la señal y la red, que sería aproximadamente para 2G hasta 220

kpbs, 3G hasta 21 megabit por segundo (Mbps), 4G hasta 100 Mbps y 5G hasta 10 Gbps.

- *WiFi*: básica para la comunicación inalámbrica en informática. Se lleva a cabo a través de señales de radiofrecuencia en la banda de 2,4 gigahercios (GHz) y 5 GHz, basada en los estándares IEEE 802.11 a/b/g/n/ac y ax. Permite la conexión directa entre dispositivos o a través de un punto de acceso, velocidades de transferencia muy altas, normalmente hasta los 600 Mbps en 2,4 GHz y 1300 Mbps en 5 GHz, con un alcance que puede alcanzar los 100 m desde un punto de acceso.
 - *Bluetooth*: muy importante para la comunicación inalámbrica en informática. Utiliza señales de radiofrecuencia, pero en este caso solo en 2,4 GHz, con el estándar IEEE 802.15.1. Su velocidad está más limitada, máximo de 2 Mbps, y se define por: su clase, que marca su alcance entre 0,5 y 100 m, los perfiles de transmisión que permite y su variante, *Classic* y *Low Energy* o *Smart*, además de su versión específica, siendo las más habituales 2.1, 4.0 y 5.0.
 - *Near-field Communication* (NFC): permite establecer la comunicación en cortas distancias, unos pocos centímetros, en la frecuencia de 13,56 MHz y en la que uno de los dispositivos puede ser un elemento pasivo, sin necesidad de fuente de energía, muy útil para incluirlo en cualquier objeto, como etiquetas, tarjetas de crédito, etc.
 - *Infrarrojos (IR)*: tecnología de comunicación óptica basada en el espectro de infrarrojo cercano, normalmente 850 nm y 950 nm. Se utiliza sobre todo en mandos a distancia de juguetes o electrodomésticos.
 - *Otras*: existen muchas alternativas de comunicación inalámbrica, destacan ANT+, ZigBee, Lora y radio FM.
- *Puertos de conexión*: permiten conectar dispositivos externos directamente mediante un cable.
 - *Datos y carga*: para subir o descargar datos del dispositivo. Los más habituales son el estándar microUSB y el USB-C, mientras que los dispositivos de Apple usan su puerto propio Lightning. En algunos dispositivos se suele separar el puerto de datos y el de carga, pero en smartphones es habitual que sea compartido.
 - *Imagen*: permiten conectar una pantalla externa, pero no es habitual en dispositivos de pequeño tamaño. Suele usarse el formato microHDMI, MHL o USB-C.
 - *Audio*: para conectar auriculares o altavoces. Se usa generalmente el puerto minijack de 3,5 milímetros (mm) aunque algunos dispositivos lo han eliminado y usan el mismo puerto que el de datos.

- *Extras de juego*: aunque no son componentes esenciales para un dispositivo móvil, sí tienen importancia para los videojuegos y suelen ofrecerse como periféricos externos:
 - *Mandos*: son habituales en las consolas portátiles, se trata de botones o palancas, de señal discreta o continua, o sistemas inerciales extra. Existen también mandos pasivos mecánicos que se colocan sobre la pantalla y aprovechan su capacidad táctil.
 - *Visores*: imprescindibles para juegos de realidad virtual, disponen de un sistema para sujetar el móvil frente a los ojos y unas lentes especiales que forman una imagen estereoscópica para el usuario. Hay también visores de realidad aumentada, que se colocan también frente a los ojos o en soportes especiales para mandos de juego externos, como pistolas.

2.3. Publicación y distribución

El proceso de publicación y distribución tiene como objetivo lanzar el videojuego al mercado una vez que se tenga la seguridad de que el juego está validado y listo para utilizarse. Disponemos de varias tiendas para hacer esta distribución, y es necesario asumir un coste por la publicación, para lo cual se crea una cuenta de publicación y se realiza el pago de los mismos. Cuando se publica la aplicación, no solo se debe subir la aplicación a la tienda, sino que también se deben aportar varios elementos que acompañarán a la promoción del juego como son capturas de pantalla, textos descriptivos, iconos de lanzamiento, etc. Nótese que estos elementos son muy importantes ya que constituyen un medio de comunicación y promoción determinante para la descarga de la aplicación por parte del usuario.

Una vez que el juego está en el repositorio de la tienda, se pasa por un proceso de aprobación que depende de la tienda en particular y puede llegar a ser más o menos restrictivo. En caso de que la aplicación incumpla alguna política de la tienda, esta puede retirar la publicación de la misma, incluso puede llegar a suspender la cuenta de desarrollador.

Hay dos grandes tiendas para la publicación de aplicaciones para móviles: Google Play y App Store. La publicación con Google Play es sencilla. Una vez que se ha creado la aplicación en la consola de desarrollador y consignado su título y descripción, se debe cargar en el repositorio de la tienda unas capturas de pantalla, el icono de la aplicación e imágenes promocionales de la misma. Además, Google Play permite subir un vídeo de YouTube como elemento adicional promocional. Una vez subida a la plataforma de la tienda estos contenidos es necesario rellenar unos formularios describiendo los detalles de contacto y la política de privacidad, así como los idiomas de la

aplicación. A continuación, se debe subir el fichero APK de la aplicación y rellenar el formulario de calificación de contenido e indicar el precio y los países donde se quiere publicar la aplicación.

La publicación con la tienda App Store es similar a Google Play. Es necesario subir el icono de la aplicación y capturas de pantalla promocionales. App Store ofrece la utilización de la plantilla App Icon Template para ayudar a construir un icono para el juego. La plataforma permite subir hasta 5 capturas de pantallas para la promoción y, aunque no es obligatorio, sí es recomendable subir un vídeo. En tal caso el vídeo debe tener un tamaño máximo de 500 MB y no puede durar más de 30 segundos ni menos de 15. Además de este material gráfico se debe subir información adicional como el nombre, la descripción del juego, palabras clave, dirección web de soporte que integre un formulario de contacto e información de política de privacidad.

Otra opción de publicación y distribución es que el propio desarrollador o creador del juego haga él mismo el proceso de publicación y distribución. En tal caso es necesario mantener un sitio Web para vender el producto. La principal ventaja de mantener este sitio Web es que se pueden mostrar vídeos, imágenes y explicaciones para la promoción sin restricciones de extensión, como ocurre con las tiendas de Android o iOS comentadas anteriormente. En este caso es conveniente ofrecer al cliente la opción de inscribirse en una lista de correo para que éste pueda recibir fácilmente noticias del producto, actualizaciones, extras gratuitos, promociones, etc., elementos que ayudan a que el potencial cliente acabe tomando la decisión de comprar el videojuego. También conviene combinarlo con presencia en redes sociales como en Twitter o Facebook anunciando el juego. Por último, hay que tener en cuenta que uno de los aspectos claves en el desarrollo de juegos para móviles es escoger el modelo de negocio apropiado. Uno de los modelos más extendido es el modelo “de gratuito a pago” (*free-to-play*, F2P), el cual conviene establecerlo en las etapas iniciales de creación del juego. Este modelo da acceso al jugador a una parte del contenido del juego sin tener que pagar, de tal forma que a medida que el usuario se siente cómodo pueda pasar a comprarlo.

2.4. Software específico: servicios de Google Play

Google Play Services es un conjunto de funcionalidades y utilidades para aplicaciones distribuidas mediante APKs a través del almacén de Google Play Store, tales como Google Maps, Google+ o Google Play Game Services. Aunque en el capítulo 5 se verán con más detalle los servicios de Google Play, vamos a describirlos brevemente en esta sección. Los servicios de Google Play funcionan gracias a dos componentes:

- *Google Play Services client library*: es una librería que incluye las interfaces de cada servicio Google que la aplicación va a usar. Esta librería está incluida

en el APK de la aplicación y permitirá a la aplicación acceder a los servicios usando las credenciales del usuario del dispositivo móvil donde se está ejecutando. La librería es actualizada por Google en todo momento.

- *Google Play Services APK*: este componente arranca un servicio en *background* en el sistema operativo Android del dispositivo y usando la librería cliente descrita anteriormente proporciona las funcionalidades del servicio en tiempo de ejecución. Estos APKs pueden ser descargados desde el repositorio almacén Google Play.

Como ya se ha indicado más arriba, uno de los servicios que ofrece Google Play es el servicio *Google Play Game Services* (GPGS), que proporciona funcionalidades de videojuegos como son gestión de logros, marcadores, misiones, premios, multi-jugadores, etc. GPGS puede ser usado para varias plataformas como Android, iOS y web, aunque tiene mucha mayor presencia en dispositivos Android.

A la hora de desarrollar una aplicación que utilice GPGS, debemos tener en cuenta dos aspectos: la consola de desarrollador y el código a desarrollar. La consola de desarrollador nos proporciona el poder configurar los servicios que queremos usar en el juego. En la sección de *Game services* de la consola se puede crear y configurar un nuevo juego. Esta sección lista el histórico de los diferentes juegos que el desarrollador ha activado y en qué estado se encuentra cada uno. Si se selecciona la acción de Añadir nuevo juego, aparece un cuadro de diálogo que permite configurar los servicios de juego a los que posteriormente accederá la aplicación. En este punto de registro del juego se indica el nombre del juego y el tipo que es (acción, casual, etc.). Posteriormente, el juego se debe enlazar con una aplicación. Para ello se debe indicar el nombre del paquete y el nombre de la aplicación a enlazar con el proyecto del juego que se ha creado. Esto generará una clave OAuth y un identificador unívoco del juego, el cual utilizará todas las aplicaciones que se enlacen al juego.

Los servicios de GPGS solo funcionan si el juego está publicado en Google Play. Sin embargo, se puede probar el juego sin necesidad de llegar a publicarse ya que la consola de desarrollador permite crear cuentas de usuario de pruebas y asociarlas al juego.

Con estas acciones anteriores ya se tiene preparado el juego para poder usar las funciones que proporcionan los servicios. Para ello desde la consola de desarrollador se deben dar de alta los servicios que se deseen incluir en el juego, como por ejemplo crear una tabla de posiciones o crear un logro. Una vez finalizado esta parte de configuración de la consola se debe trabajar en la configuración del código del proyecto, que es el segundo aspecto que se ha indicado más arriba y que se describe a continuación.

Lo primero que se debe hacer es añadir la librería de Servicios de Google Play al proyecto que se está desarrollando en el EDI que se esté utilizando, como por ejemplo

Android Studio. Normalmente, el EDI utilizado suele facilitar un asistente que hace esta configuración, pero es recomendable que el desarrollador revise todo el proceso de inclusión de la librería. En esta configuración se añade la dependencia de la librería en el *build.gradle* del módulo de la aplicación. También se tendrá que añadir el identificador del juego (generado en la consola del desarrollador) en el fichero *manifest* del proyecto. Finalmente, se debe realizar el registro del proyecto y alguna configuración más, terminando con programar la *Activity* de la aplicación para conectarse al juego y poder usar los servicios que el juego tiene asociado.

2.5. Fundamentos de desarrollo con Android Studio

Android Studio es el EDI oficial de Google para el desarrollo de aplicaciones Android y está basado en IntelliJ IDEA (<https://www.jetbrains.com/idea/>) que constituye un entorno de desarrollo creado por JetBrains. De este último entorno, IntelliJ IDEA, Android Studio utiliza el editor de código fuente y algunas herramientas de desarrollo genéricas, extendiendo herramientas y funciones específicas para el desarrollo Android.

2.5.1. Inicializando Android

Para que Android Studio pueda funcionar, se debe tener instalado el paquete de desarrollo JDK (Java Development Kit) de Java en la máquina de desarrollo y tener configurada la variable de entorno `JAVA_HOME` correctamente (normalmente cuando se instala el JDK el asistente de instalación crea esta variable y la configura). Posteriormente se debe descargar el paquete instalador de Android Studio conectándose al sitio desarrollador de Android (<https://developer.android.com>) y proceder a su instalación siguiendo las instrucciones del asistente, lo cual puede llevar varios minutos. Cuando se inicia Android Studio aparece una pantalla de bienvenida donde se listan proyectos recientes y se muestra la sección de *Quick Start*, la cual permite las siguientes acciones (véase figura 2.1):

- Crear un nuevo proyecto.
- Abrir un proyecto ya existente.
- Importar ejemplos, lo cual permite importar proyectos de ejemplos de código Google desde GitHub.
- Depurar apk o crear perfiles de depuración.
- Crear un nuevo proyecto importando código existente desde un sistema de control de versiones.