

## PROGRAMACIÓN PARA DISPOSITIVOS MÓVILES

del 04 al 22 de julio de 2022

PROGRAMA OPERATIVO FSE CASTILLA Y LEÓN 2014-2020.  
ACTUACIÓN COFINANCIADA POR EL FONDO SOCIAL EUROPEO.

Formador: Israel Parra Conejero



# INTRODUCCIÓN A LOS DISPOSITIVOS MÓVILES

## TABLA DE CONTENIDO

### Contenido

<b>INTRODUCCIÓN A LOS DISPOSITIVOS MÓVILES .....</b>	<b>3</b>
INTRODUCCIÓN E HISTORIA.....	3
TIPOS DE APLICACIONES MÓVILES .....	3
COMPONENTES DE UN DISPOSITIVO MÓVIL.....	4
FAMILIAS DE DISPOSITIVOS.....	5
<b>INTRODUCCIÓN A ANDROID .....</b>	<b>5</b>
INTRODUCCIÓN A ANDROID .....	5
ENTORNO DE DESARROLLO.....	5
FUNDAMENTOS DE LAS APLICACIONES ANDROID.....	6
INTERFACES DE USUARIO EN ANDROID .....	6
RECURSOS DE APLICACIONES EN ANDROID .....	6
<b>INTRODUCCIÓN A IPHONE .....</b>	<b>7</b>
INTRODUCCIÓN A LA FAMILIA DE IOS .....	7
ENTORNO DE TRABAJO.....	7
HERRAMIENTAS BÁSICAS.....	8
APLICACIONES BÁSICAS .....	8
<b>PUBLICACIÓN Y LOS MARKETS .....</b>	<b>9</b>
ESTUDIO INICIAL DEL MERCADO.....	9
APLICACIONES SIMILARES Y DIFERENCIACIÓN DE NUESTRO PRODUCTO .....	9
DESPLIEGUE, VERSIONES Y ACTUALIZACIONES.....	9
MODELOS DE NEGOCIO .....	10
<b>GESTIÓN DE PROYECTOS MÓVILES .....</b>	<b>10</b>
METODOLOGÍAS TRADICIONALES .....	10
METODOLOGÍAS ÁGILES.....	11
SELECCIÓN DE PLATAFORMA Y PÚBLICO.....	11
SEGUIMIENTO Y MANTENIMIENTO .....	11

# INTRODUCCIÓN A LOS DISPOSITIVOS MÓVILES

## INTRODUCCIÓN E HISTORIA

Las aplicaciones han recorrido un largo camino desde que vivían en lo que se considera el primer asistente digital personal (PDA).

En la década de 1980, Psion desarrolló el modelo Psion Organizer I, que se calificó como la "primera computadora de bolsillo práctica del mundo" y se completó con una calculadora, un reloj y otras aplicaciones conocidas.

Alrededor de una década más tarde, llegó el primer teléfono inteligente del mundo de IBM en 1993, que estaba equipado con muchas de las mismas aplicaciones utilitarias como Psion.

El siguiente salto en la tecnología de aplicaciones se produjo en otra década cuando se lanzó el teléfono inteligente BlackBerry en 2002 cuando la empresa introdujo capacidades de correo electrónico en su teléfono.

Entonces, en 2010, las aplicaciones se pusieron a la vanguardia cuando Apple anunció su infame tienda de aplicaciones que se lanzó un año después.

En su primera década, App Store ha superado todas nuestras expectativas más salvajes, desde las aplicaciones innovadoras que los desarrolladores han soñado hasta la forma en que los clientes han hecho que las aplicaciones formen parte de su vida diaria, y esto es solo el comienzo", dijo Phil. Schiller, vicepresidente sénior de marketing mundial de Apple en 2018. "No podríamos estar más orgullosos de lo que han creado los desarrolladores y de lo que nos deparan los próximos 10 años.

Solo en el tercer trimestre de 2021, hubo casi 34 000 millones de descargas de aplicaciones en la tienda de aplicaciones.

En todo 2020, se descargaron 86 700 millones de aplicaciones: "Google Play fue responsable de 108 500 millones de descargas en 2020, iOS tuvo 34 400 millones de descargas".

Ahora que cualquiera puede crear una aplicación, ya sea por su cuenta o contratando a un tercero para que la desarrolle, las aplicaciones abundan.

En los primeros tres meses de 2021, Google Play Store ofreció 3,48 millones de aplicaciones y la App Store de Apple albergaba 2,22 millones de aplicaciones disponibles para iOS.

Cuando la facilidad de creación se combina con el posible retorno de la inversión, ya sea en utilidad o ingresos, las aplicaciones son tan útiles para los creadores como para los usuarios.

## TIPOS DE APLICACIONES MÓVILES

Hay tres tipos básicos de aplicaciones móviles si las categorizamos por la tecnología utilizada para codificarlas:

- **Las aplicaciones nativas** se crean para una plataforma o sistema operativo específico, Android e iOS en este caso.
- **Las aplicaciones web** son versiones responsive de sitios web que pueden funcionar en cualquier dispositivo móvil o sistema operativo porque se entregan mediante un navegador móvil.

- **Las aplicaciones híbridas** son combinaciones de aplicaciones web y nativas, pero envueltas dentro de una aplicación nativa, lo que le da la capacidad de tener su propio ícono o descargarse de una tienda de aplicaciones.

La meta de este curso es el desarrollo de **aplicaciones híbridas**, las cuales tienen grandes ventajas si las comparamos con las nativas:

1. Se pueden escribir únicamente con JavaScript y HTML sin necesidad de conocer los lenguajes de programación nativos.
2. Se pueden publicar en las Stores sin complejas configuraciones.
3. Se pueden programar para ambas plataformas (Android e iOS) desde cualquier equipo, Windows o Mac, no sucede así en las aplicaciones nativas, ya que para iOS debemos programar desde un MAC.
4. Se pueden crear actualizaciones “en caliente” en sus respectivas Stores sin la necesidad de compilar y publicar nuevas versiones.

## COMPONENTES DE UN DISPOSITIVO MÓVIL

Las características de los teléfonos móviles son el conjunto de capacidades, servicios y aplicaciones que ofrecen a sus usuarios. Los teléfonos móviles a menudo se denominan teléfonos con funciones y ofrecen telefonía básica. [Aclaración necesaria] Los teléfonos con capacidad informática más avanzada mediante el uso de código nativo intentan diferenciar sus propios productos mediante la implementación de funciones adicionales para hacerlos más atractivos para los consumidores. Esto ha llevado a una gran innovación en el desarrollo de teléfonos móviles en los últimos 20 años.

Los componentes comunes que se encuentran en todos los teléfonos son:

- Varios semiconductores.
- Una batería (normalmente una batería de iones de litio), que proporciona la fuente de alimentación para las funciones del teléfono.
- Un mecanismo de entrada para permitir que el usuario interactúe con el teléfono. Las pantallas táctiles son las más comunes hoy en día.
- Servicios básicos de telefonía móvil para permitir a los usuarios realizar llamadas y enviar mensajes de texto.
- Tarjetas SIM para permitir que una cuenta se intercambie entre dispositivos. Algunos dispositivos CDMA también tienen una tarjeta similar llamada R-UIM.
- Todos los teléfonos móviles están diseñados para funcionar en redes celulares y contienen un conjunto estándar de servicios que permiten que teléfonos de diferentes tipos y en diferentes países se comuniquen entre sí.
- Aplicaciones como reloj, alarma, calendario, contactos, calculadora y algunos juegos.
- Enviar y recibir imágenes y videos (sin Internet) a través de MMS y para distancias cortas, p. ej. Bluetooth.
- Bluetooth es una característica común.
- Receptores de GPS integrados o conectados (es decir, usando Bluetooth) a teléfonos celulares.

## FAMILIAS DE DISPOSITIVOS

A continuación, citamos algunos de los dispositivos móviles que existen en la actualidad:

- Teléfonos inteligentes.
- Tabletas.
- Relojes inteligentes.
- Agendas digitales.
- Calculadoras.
- Videoconsolas portátiles.
- Reproductores digitales.
- Cámaras fotográficas digitales.
- Cámaras de video digitales.
- RobotsTarjetas inteligentes.

## INTRODUCCIÓN A ANDROID

### INTRODUCCIÓN A ANDROID

Android es un sistema operativo móvil basado en Linux que fue iniciado originalmente por Android Inc. con el mismo nombre. En 2005, Google adquirió Android y se hizo cargo de su trabajo de desarrollo para introducirse en el espacio móvil.

Android se hizo de código abierto y gratuito; por lo tanto, la mayor parte del código fuente de Android se publica bajo la licencia Apache de código abierto. Eso permite que cualquier persona descargue el código fuente y lo cambie según sus requisitos, por lo tanto, pueden tener su propio sabor del sistema operativo Android.

Google lanzó la primera versión beta del kit de desarrollo de software (SDK) de Android en 2007, mientras que la primera versión comercial, Android 1.0, se lanzó en septiembre de 2008.

El código fuente de Android está disponible bajo licencias de software gratuitas y de código abierto. Google publica la mayor parte del código bajo la Licencia Apache versión 2.0 y el resto, los cambios del Kernel de Linux, bajo la Licencia Pública General GNU versión 2.

### ENTORNO DE DESARROLLO

El desarrollo de aplicaciones para Android se hace habitualmente con el lenguaje de programación Java y el conjunto de herramientas de desarrollo SDK, pero hay otras opciones disponibles. En julio de 2013 existían más de 1.000.000 de aplicaciones contabilizadas para Android, con aproximadamente 25 mil millones de descargas.

La plataforma Android ha crecido hasta ser una de las preferidas por los desarrolladores para plataformas móviles.

En nuestro caso, el desarrollo se va a llevar a cabo con JavaScript y HTML, tanto para Android como para iOS. Esta es una de las grandes ventajas de trabajar con una herramienta con React Native.

Para desarrollar sólo deberemos tener instalado en nuestro equipo Node.js y Expo CLI, con eso tendremos todo lo necesario para que nuestras aplicaciones se pongan en marcha.

## FUNDAMENTOS DE LAS APLICACIONES ANDROID

Las aplicaciones Android normalmente están escritas en el lenguaje de programación Java.

El SDK de Android tiene una serie de herramientas que permitirán compilar el código, incluyendo los datos y los recursos, y lo integrará todo en un fichero APK, o también conocido como paquete Android. Este fichero será nuestro instalador.

Una vez instalada una aplicación, cada una de ellas tiene su propio sistema de seguridad, de tal modo que:

- Cada aplicación será un usuario diferente dentro de Android como Sistema Operativo basado en un sistema Linux multiusuario. Este usuario será un ID de usuario Linux único.
- Android dará permisos para todos los ficheros de una aplicación únicamente para el usuario que identifica dicha app.
- Cada proceso tiene su propia máquina virtual, por lo que la ejecución de aplicaciones es totalmente independiente.

Por defecto, cada aplicación corre en su propio proceso Linux, el cual se gestiona a nivel de Sistema Operativo.

## INTERFACES DE USUARIO EN ANDROID

La interfaz de usuario de una aplicación Android es todo aquello que el usuario puede ver y con lo que puede interactuar.

Android ofrece una variedad de componentes de IU previamente compilados, como objetos de diseño estructurados y controles de la IU que te permiten compilar la interfaz gráfica de usuario para tu app. Además, también brinda otros módulos de IU para interfaces especiales, como diálogos, notificaciones y menús.

## RECURSOS DE APLICACIONES EN ANDROID

Los recursos son los archivos adicionales y el contenido estático que usa tu código, como mapas de bits, definiciones de diseño, strings de interfaz de usuario, instrucciones de animación, etc.

Siempre debes externalizar los recursos para aplicaciones, como imágenes y strings de tu código, para que puedas mantenerlos de forma independiente.

También debes proporcionar recursos alternativos para configuraciones de dispositivos específicos, agrupándolos en directorios de recursos con un nombre especial. En tiempo de ejecución, Android utiliza el recurso adecuado según la configuración actual. Por ejemplo, puedes proporcionar un diseño de interfaz de usuario (IU) diferente según el tamaño de la pantalla o strings diferentes según la configuración de idioma. Una vez que externalizas los recursos para tu aplicación, puedes acceder a ellos mediante los ID de recursos que se generan en la clase R de tu proyecto. En este documento, se muestra cómo puedes agrupar los recursos en tu proyecto de Android, proporcionar recursos alternativos para configuraciones de dispositivos específicos y acceder a ellos desde el código de tu aplicación u otros archivos XML posteriormente.

# INTRODUCCIÓN A IPHONE

## INTRODUCCIÓN A LA FAMILIA DE IOS

iOS es un sistema operativo móvil de la multinacional Apple Inc. Originalmente desarrollado para el iPhone (iPhone OS), después se ha usado en dispositivos como el iPod touch y el iPad. Apple no permite la instalación de iOS en hardware de terceros.

La última versión del sistema operativo es iOS 15 (lanzada en septiembre de 2021)<sup>1</sup> que sustituye a iOS 14 con el objetivo principal de mejorar la experiencia del usuario, integrando cambios relevantes, entre ellos el nuevo Modo Oscuro o la posibilidad de utilizar memorias externas con la app Archivos del sistema. También con este lanzamiento existe una disputa entre Apple y Facebook por el permiso añadido para que los usuarios decidan si quieren dejar que recopilen los datos las App o no.

Los elementos de control son deslizadores, interruptores y botones. La respuesta a las órdenes del usuario es inmediata y provee una interfaz fluida. La interacción con el sistema operativo incluye gestos como deslices, toques, pellizcos, los cuales tienen definiciones diferentes dependiendo del contexto de la interfaz. Se utilizan acelerómetros internos para hacer que algunas aplicaciones respondan a sacudir el dispositivo (por ejemplo, para el comando deshacer) o rotarlo en tres dimensiones (un resultado común es cambiar de modo vertical al apaisado u horizontal).

En el marco de las filtraciones acerca de los programas de vigilancia mundial de 2013-2014 de Edward Snowden, Der Spiegel publicó que la NSA estadounidense tiene grupos de trabajo dedicados a descifrar los sistemas de seguridad de iOS; además tiene pequeños programas conocidos como scripts que permiten a la agencia vigilar a los usuarios de las distintas versiones del sistema iOS su geolocalización, notas de voz, fotos y otras aplicaciones como Google Earth, Facebook.<sup>3</sup>

iOS se deriva de macOS, que a su vez está basado en Darwin BSD, y por lo tanto es un sistema operativo Tipo Unix.

iOS cuenta con cuatro capas de abstracción: la capa del núcleo del sistema operativo, la capa de "Servicios Principales", la capa de "Medios" y la capa de "Cocoa Touch".

<https://es.wikipedia.org/wiki/IOS>

## ENTORNO DE TRABAJO

Normalmente cuando trabajamos en el desarrollo de iOS necesitamos equipos Mac ya que en Windows no es posible compilar ni desarrollar este tipo de aplicaciones debido a su arquitectura.

La herramienta utilizada para el desarrollo de aplicaciones iOS es Xcode y el lenguaje de programación es Swift, seguramente uno de los lenguajes más complejos de asimilar cuando estamos empezando debido a su sintaxis.

Por suerte para nosotros, no debemos utilizar un Mac para programar aplicaciones para iOS ni tampoco debemos escribir código Swift. El motivo es que nosotros hacemos uso de React Native, el cual nos permite desarrollar aplicaciones nativas con JavaScript y HTML, lo que denominamos JSX.

## HERRAMIENTAS BÁSICAS

Las herramientas que necesitamos para el desarrollo de aplicaciones iOS son las mismas que para Android. La única diferencia que podemos encontrar es que si deseamos probar nuestras aplicaciones en emuladores o dispositivos reales entonces sí deberemos hacerlo con un equipo Mac.

Para desarrollar en iOS necesitamos lo siguiente:

- Un equipo Windows, Mac o Linux para
- Un editor de texto o IDE, Visual Studio Code o IntelliJ son buenas opciones.
- Instalar Expo CLI en nuestra computadora Windows, Mac o Linux.

## APLICACIONES BÁSICAS

A continuación, citamos algunas de las aplicaciones básicas para iPhone que ya vienen incorporadas por defecto:

- App Store.
- Aplicación para conocer el tiempo.
- Calculadora.
- Calendario.
- Reloj, alarmas, cronómetros.
- Aplicación de notas.
- Recordatorios.
- Cámara.

A continuación, citamos algunas de las aplicaciones básicas para iPhone que se pueden instalar:

- WhatsApp.
- TikTok.
- YouTube.



# PUBLICACIÓN Y LOS MARKETS

## ESTUDIO INICIAL DEL MERCADO

Se trata de una serie de estudios que te ayudarán a conocer a tu público. El proceso de investigación del mercado de aplicaciones móviles te permitirá conocer y comprender las tendencias en la actualidad. Además, hacer un estudio inicial te dará acceso a datos más interesantes, como los intereses y lo que más valoran tus posibles clientes.

Antes de iniciar el estudio de mercado, hay que tener en cuenta estos puntos:

- ¿Es el momento adecuado para lanzar la aplicación?
- ¿La aplicación responde a las necesidades del cliente?

Las siguientes son algunas de las estrategias que deberás tener en cuenta:

- Conocer al público objetivo.
- Reunir y organizar los datos.
- Vigilar las redes sociales.

## APLICACIONES SIMILARES Y DIFERENCIACIÓN DE NUESTRO PRODUCTO

Existen miles de aplicaciones en los markets que hacen lo mismo, pero no todas tienen el mismo éxito. Por ejemplo, podemos encontrar multitud de apps para contar los pasos o los kilómetros que recorremos, pero cada una cuenta con sus propios números.

Una buena forma de intentar diferenciarnos de los demás es leyendo las reseñas de estas apps, seguro que muchas malas reseñas se repiten en temas que tú puedes mejorar, y lo mismo con las cosas buenas, seguro que las puedes mantener, e incluso mejorar.

La diferenciación de nuestro producto es lo que hace que nuestro producto sea diferente y atractivo a los ojos de los clientes. La diferenciación de producto es lo que te ofrece una ventaja competitiva en tu mercado.

Centrarse en tus clientes es un buen comienzo para conseguir una diferenciación de producto exitosa.

- ¿Qué necesidades tienen?
- ¿Qué no les ofrece nadie más?
- ¿Qué les frustra?

Si eres capaz de detectar los puntos anteriores y solucionar los problemas existentes, tendrás mucho adelantado.

## DESPLIEGUE, VERSIONES Y ACTUALIZACIONES

El despliegue de una aplicación no es más que la publicación y posteriores actualizaciones en su correspondiente market place.

Las versiones de una App las gestionamos nosotros, normalmente empezando por la 0.0.1 y a partir de ahí la vamos incrementando.

## VERSIONADO SEMÁNTICO

Existe algo en el mundo del software llamado versionado semántico. Consiste en un par de convenciones para asignar números de versión para tu software.

Básicamente, la idea es la siguiente:

Dado un número de versión PRINCIPAL.MENOR.PARCHE (MAJOR.MINOR.PATCH), incrementa la:

- Versión PRINCIPAL (MAJOR version) cuando realizas cambios no compatibles en la API.
- Versión MENOR (MINOR version) cuando agregas una funcionalidad en una manera compatible al API ya existente.
- Versión del PARCHE (PATCH version) cuando realizas correcciones de 'bugs' compatibles con el API ya existente.

Utilizando versionado semántico tendremos la capacidad de actualizar y versionar nuestras aplicaciones de forma ordenada entendiendo el motivo de cada actualización.

## MODELOS DE NEGOCIO

Normalmente nos centramos en dos modelos de negocio cuando desarrollamos nuestras Apps.

- Premium.
- Freemium.

El modelo Premium en la mayoría de los casos se caracteriza por ofrecer una suscripción a nuestro público para que puedan acceder a los contenidos de nuestra App.

Un caso típico de un modelo Premium sería Amazon Prime Video, es cierto que podemos realizar una prueba para ver si sus servicios nos interesan, pero si finalmente no nos suscribimos, dejaremos de poder acceder a su contenido.

El modelo Freemium en cambio cambia este paradigma ofreciendo aparte de un modelo Premium, un modelo Free.

Un caso típico de modelo Freemium sería Spotify, puedes seguir escuchando música sin pagar para siempre, pero también puedes suscribirte y tener una serie de ventajas que los Free no tienen.

## GESTIÓN DE PROYECTOS MÓVILES

### METODOLOGÍAS TRADICIONALES

La metodología tradicional se caracteriza por seguir estrictamente una planificación marcada, que no deja lugar a la improvisación.

A primera vista, esto puede parecer beneficioso, pero no lo es tanto si tenemos en cuenta que nos movemos en un entorno dinámico en el que todo puede cambiar en cuestión de segundos y debemos estar preparados para afrontar estas situaciones.

Las principales características pueden resumirse en los siguientes puntos:

- Estabilidad del entorno.
- Predictivas.
- Rígidas.

## METODOLOGÍAS ÁGILES

El desarrollo ágil de software envuelve un enfoque para la toma de decisiones en los proyectos de software, que se refiere a métodos de ingeniería del software basados en el desarrollo iterativo e incremental, donde los requisitos y soluciones evolucionan con el tiempo según la necesidad del proyecto. Así el trabajo es realizado mediante la colaboración de equipos autoorganizados y multidisciplinarios, inmersos en un proceso compartido de toma de decisiones a corto plazo.

Cada iteración del ciclo de vida incluye planificación, análisis de requisitos, diseño, codificación, pruebas y documentación. Adquiere una gran importancia el concepto de "finalizado" (done), ya que el objetivo de cada iteración no es agregar toda la funcionalidad para justificar el lanzamiento del producto al mercado, sino incrementar el valor por medio de "software que funciona" (sin errores).

Las principales características pueden resumirse en los siguientes puntos:

- Incrementales, ya no somos rígidos.
- Impredecibles, pueden cambiar en cualquier momento.
- Flexibles, deben estar previstas para cambiar en cualquier momento.

## SELECCIÓN DE PLATAFORMA Y PÚBLICO

Hoy en día prácticamente el 100% de las aplicaciones se publican en Android y en iOS, el motivo es que en ambas plataformas existen miles de clientes potenciales que nadie desea dejar escapar.

Existen muchos casos donde debido a la dificultad y el coste que conlleva desarrollar para iOS, únicamente se publica en Android, pero siempre que se pueda la mejor opción es publicar en ambas plataformas.

Para realizar un estudio de tu público, simplemente revisa el apartado anterior Estudio inicial del mercado, ahí explicamos las partes más importantes de este punto.

## SEGUIMIENTO Y MANTENIMIENTO

El mantenimiento de una App consiste en que el producto esté siempre actualizado y listo por si hay que retomar el desarrollo para hacer mejoras. Por lo tanto, habrá que tenerlo adaptado a la versión más actual de los sistemas operativos, actualizar librerías, realizar pruebas y un sinfín de tareas para que al momento de añadir alguna nueva característica no se convierta en un drama.

Una de las premisas en el seguimiento y mantenimiento de las apps es adaptar las aplicaciones a las guías de estilo de los nuevos sistemas operativos. El motivo es que no dejan de aparecer nuevos tipos de dispositivo casi a diario, si no los tenemos en cuenta, podremos perder clientes por el camino.