Configuración del entorno de desarrollo

En esta unidad de trabajo, se presentan algunos aspectos previos antes de empezar el desarrollo de la aplicación y se explica cómo configurar el entorno de trabajo para desarrollar aplicaciones Android.

Los objetivos particulares del módulo son que el alumno aprenda lo siguiente.

* Cuáles son los aspectos previos que hay que decidir antes de empezar a desarrollar una aplicación.
* Configurar un entorno de trabajo para desarrollar aplicaciones Android.

De esta manera, en el primer apartado se explican aspectos previos sobre el mercado y el plan de negocio de las aplicaciones móviles que hay que tener en cuenta antes de empezar el desarrollo de cualquier aplicación. En el segundo apartado, se explica cómo configurar el entorno de desarrollo. Observad que el entorno de desarrollo solo debe configurarse la primera vez que desarrollamos una aplicación.

1.1. Aspectos previos al desarrollo.

Antes de empezar el proceso de creación y desarrollo de una aplicación, es importante hacer un estudio de mercado y decidir el plan de negocio, ya que esto puede marcar el éxito o el fracaso de la aplicación. El mercado de las aplicaciones móviles es muy grande pero resulta importante tener claro el público objetivo al que va dirigida la aplicación y en qué versiones del sistema operativo se podrá usar la misma:

* *¿Cuál es el público objetivo de la aplicación?* El público objetivo es el destinatario final de la aplicación móvil. Definir y acotar a este público hace más fácil construir la aplicación más adecuada para los usuarios.
* Por ejemplo, no es lo mismo desarrollar una aplicación para un público que no es experto en tecnología que para otro que sí lo es. En el primer caso, la interfaz de usuario tiene que ser muy intuitiva y la aplicación simple, si no se corre el riesgo de que la gente lo considere demasiado complicado y no use la aplicación. Por el contrario, si el público es experto en tecnologías, es importante que la aplicación sea completa y aporte puntos de calidad respecto a su competencia.
* *¿Cuál es la versión de Android para la que queremos desarrollar? ¿Cuáles son las API que se usarán?* Una de las características principales del sistema operativo Android es su constante proceso de actualización. Las versiones del sistema operativo (SO) reciben el nombre de postres en inglés. En cada versión, el **postre** elegido empieza por una letra distinta siguiendo un orden alfabético. Las versiones del SO están relacionadas con un número de API. Cada vez que hay nueva versión del SO, aumenta el número de la versión de la API.

En el enlace siguiente hay un artículo donde se enumeran todas las [versiones](https://www.google.com/url?q=https%3A%2F%2Fen.wikipedia.org%2Fwiki%2FAndroid_version_history&sa=D&sntz=1&usg=AFQjCNH7-ZcTXf6Jnrme7B-BkESucvlxSA) [web, inglés] de Android hasta la fecha. Además del historial de versiones, también [se describen](http://www.google.com/url?q=http%3A%2F%2Fen.wikipedia.org%2Fwiki%2FAndroid_version_history&sa=D&sntz=1&usg=AFQjCNGWen96_mjdDNTrb5ncX3Nm0d70bA) las características específicas que diferencian diversas versiones del sistema operativo, información que puede ayudar al desarrollador a decidirse por una API u otra. En la [documentación oficial](https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels) [web, inglés] se presentan más detalladamente las novedades de cada nueva versión de API. Por último, el siguiente artículo ofrece un resumen visual de las principales [diferencias entre versiones](https://www.google.com/url?q=https%3A%2F%2Farstechnica.com%2Fgadgets%2F2016%2F10%2Fbuilding-android-a-40000-word-history-of-googles-mobile-os%2F&sa=D&sntz=1&usg=AFQjCNGMr_HS7gUOhlLu1wC8txU82XgfFQ) [web, inglés].

Desde el punto de vista del desarrollador, a la hora de decidirse por una versión u otra hay que tener en cuenta lo siguiente: las versiones más recientes del sistema operativo permiten usar las últimas funcionalidades de programación introducidas por Google, pero es posible que no todos los usuarios estén actualizados a la última versión. Por el contrario, las versiones más antiguas serán compatibles con la mayoría de los dispositivos, pero no ofrecerán las últimas funcionalidades.

Por lo tanto, es necesario encontrar un equilibrio entre novedad e índice de mercado. Esta es una razón por la que resulta importante definir las funcionalidades necesarias de la aplicación, ya que pueden condicionar la elección de la versión de la API y de la versión del SO. Por otra parte, Google ofrece datos actualizados del [índice de mercado](http://developer.android.com/about/dashboards/index.html) [Web, Inglés] de cada una de las distribuciones del SO Android y de las API, que pueden ayudar en la toma de decisiones.

También hay que tener en cuenta que a diferencia del sistema operativo iOS, las actualizaciones del sistema operativo en Android no se propagan automáticamente a todos los dispositivos. La actualización de los SO Android pasa por el fabricante del dispositivo móvil y, por lo tanto, no depende solo de Google. Solo los dispositivos creados por Google tienen las últimas actualizaciones al momento.

Respecto al plan de negocio, hay que tener en cuenta los aspectos siguientes.

* *¿Cuál es la estrategia de monetización?* Es importante definir nada más empezar qué mecanismos utilizará una aplicación para generar ingresos. En la siguiente [Infografia](http://www.google.com/url?q=http%3A%2F%2Fapplicantes.com%2Fwp-content%2Fuploads%2F2013%2F03%2Finfografia-como-monetizar-app-movil2.png&sa=D&sntz=1&usg=AFQjCNHEg43XqwGIViYXKxKN5VGXgD6jdg) [Imagen, Inglés] se presentan datos interesantes sobre la monetización de apps. En la infografía, se muestra que durante todo el año 2012 la descarga de aplicaciones gratuitas supuso un 89% del total. Por lo tanto, ¿cuál es la forma para ganar dinero con una aplicación? Hoy día, encontramos distintas posibilidades para la monetización de una aplicación en dispositivos móviles.
  + ***Freemium***: aplicaciones gratuitas que ofrecen actualizaciones o contenidos extra de pago. Este modelo supone el 75% de los beneficios en Google Play (el *market* de Google para Android). Este modelo solo funciona si la aplicación tiene un gran volumen de descargas.
  + ***Publicidad in-app*:** las aplicaciones contienen anuncios para generar ingresos. Según la infografía, los anuncios son 11,4 veces más efectivos que los *banners* tradicionales, pero hay que tener en cuenta que el 70% de los usuarios piensa que la publicidad en móviles es intrusiva y los dispositivos con pantalla pequeña pueden quedar llenos de publicidad.
  + *Pago por aplicación*: el usuario paga directamente para descargarse la aplicación. Es uno de los modelos que puede generar más beneficios, pero cada vez funciona menos, ya que los usuarios, especialmente los de Android, no están dispuestos a pagar directamente por la aplicación.
* Respecto a la monetización de las aplicaciones móviles, se recomienda leer un [artículo](http://www.google.com/url?q=http%3A%2F%2Fwww.forbes.com%2Fsites%2Ftristanlouis%2F2013%2F08%2F10%2Fhow-much-do-average-apps-make%2F&sa=D&sntz=1&usg=AFQjCNHO9Kb5fJxzGuDyuvXBqyhTtjDnNQ) [Web, Inglés] de la revista Forbes en el que se habla de cuánto gana un desarrollador medio con sus aplicaciones.
* *¿Cuál es el plan de marketing?* Una vez la aplicación está desarrollada, hay que conseguir que los usuarios la descarguen entre las miles de aplicaciones ya existentes. Para esto, hoy día ya encontramos servicios de posicionamiento y promoción de aplicaciones. Además, es interesante que la aplicación salga en reseñas de blogs importantes o poner publicidad de nuestra aplicación en otras aplicaciones. Otro punto que hay que tener en cuenta es definir si la aplicación usará funcionalidades que compartan contenido en las redes sociales, lo cual da mucha viralidad a esta aplicación.
* *¿Cuál es el presupuesto?* Antes de crear la aplicación es muy importante definir el presupuesto, ya que de esto dependerá el desarrollo de la aplicación y la estrategia de marketing. Los costes principales del desarrollo de una aplicación se centran en tres ámbitos:
  + El registro de desarrollador en el *market* Android vale 25 dólares y se paga una sola vez.
  + Los costes de desarrollo. Estos costes dependen de la complejidad de la aplicación, de los servicios externos con los que se quiera enlazar (por ejemplo, si se quiere utilizar una [plataforma *back-end* de pago](https://www.google.com/url?q=https%3A%2F%2Fen.wikipedia.org%2Fwiki%2FMobile_backend_as_a_service&sa=D&sntz=1&usg=AFQjCNGEq8IIb5sF0l-kotJsiUn_BmtS_A)) y de la rapidez con la que se quiera desarrollar la aplicación.
  + Los costes de marketing de los servicios de posicionamiento que se contraten y la publicidad.

1.2. Configuración del entorno de desarrollo

Este apartado explica las diferentes opciones para configurar el entorno de desarrollo que permitirá programar aplicaciones Android. Para esto, es necesario descargar y configurar una serie de herramientas y librerías que permitirán escribir, compilar, ejecutar y publicar las aplicaciones.

La configuración del entorno de desarrollo se divide en tres pasos:

1. Descarga de las herramientas de desarrollo.
2. Configuración del entorno de desarrollo integrado (IDE).
3. Configuración de los dispositivos en los que se va a depurar y testear la aplicación.

A continuación, podéis ver un tutorial que presenta brevemente los pasos que hay que dar en este proceso de instalación.

Cada uno de los siguientes subapartados explica con detalle los pasos que hay que dar en este proceso.

1.2.1. Herramientas de desarrollo.

El elemento básico para desarrollar aplicaciones Android es el *software development kit***(***SDK***)**. El SDK se puede utilizar desde la línea de comandos, descargándolo des de la página de Google [web, inglés]. De todas formas, la forma recomendada de utilizarlo es desde un entorno de desarrollo integrado (IDE) como Android Studio, que ya incorpora el SDK.

1.2.2. Entornos de desarrollo integrados (IDE)

El entorno de desarrollo de aplicaciones nativas Android soportado oficialmente por Google es [Android Studio](https://developer.android.com/studio/index.html) [web, inglés]. Por tanto, se recomienda instalar este IDE, ya que es el que utilizaremos a lo largo del curso.

A parte de Android Studio, hay otras herramientas que permitten desarrollar aplicaciones Android. Mencionaremos brevemente dos de estas herramientas: Eclipse y App Inventor.

1.2.2.1 Instalación de Android Studio.

Estas instrucciones son para instalaciones desde cero de Android Studio, es decir, para ordenadores que anteriormente no tenían Android Studio instalado. Dado que a partir de la versión 1, se han cambiado directorios de sitio, se recomienda realizar la desinstalación de Android Studio si se tiene una versión anterior.

Nota: Comprobar que se cumplen los [requisitos mínimos](http://developer.android.com/sdk/index.html#Requirements) [Web, Inglés]

Instalar Java

En primer lugar, comprobad si tenéis el Java Developer Kit (JDK) versión 6.0 o superior instalado (JRE por sí sola no es suficiente). Si no lo tenéis instalado o si su versión es inferior a 6.0, instalad el JDK de Java en la página [instalación de Java](http://www.google.com/url?q=http%3A%2F%2Fwww.oracle.com%2Ftechnetwork%2Fjava%2Fjavase%2Fdownloads%2Findex.html&sa=D&sntz=1&usg=AFQjCNGWCcKCIFm3bfDWtU41j6HJzekqNQ) [Web, Inglés] y selecciona la última JDK.

Establecer el JAVA\_HOME

Podemos ver cómo configurar la variable de entorno JAVA\_HOME [aquí](http://www.google.com/url?q=http%3A%2F%2Fdocs.oracle.com%2Fcd%2FE19509-01%2F820-3208%2Finst_cli_jdk_javahome_t%2Findex.html&sa=D&sntz=1&usg=AFQjCNF9y-CdflSIJuYmdil1UL2GoE_Vbg) [Web, Inglés].

Instalar Android Studio.

1.2.2.2 Eclipse.

Eclipse fué el IDE más utilizado para el desarrollo de aplicaciones de Android, pero desde que Android Studio apareció su popularidad como IDE de desarrollo de Android bajó hasta que en noviembre de 2016 Google anunció que ya no daba soporte oficial, ver [blog](http://www.google.com/url?q=http%3A%2F%2Fwww.androidpolice.com%2F2016%2F11%2F02%2Fgoogle-officially-ends-support-for-eclipse-android-developer-tools-in-favor-of-android-studio%2F&sa=D&sntz=1&usg=AFQjCNFtUrY3n3C-2LlARiM6X8MSKTju2w) [Web, Inglés].

1.2.2.3. App Inventor

El [App Inventor](http://www.google.com/url?q=http%3A%2F%2Fappinventor.mit.edu%2F&sa=D&sntz=1&usg=AFQjCNFXIQVAiGqFg73KCHFCSJExiK0tNA) [Web, Inglés] es una aplicación de Google Labs que permite la creación de aplicaciones para Android. Esta aplicación permite crear aplicaciones sin necesidad de tener conocimientos de programación. Las aplicaciones se crean de manera visual y luego se pueden publicar en Google Play. El problema es que la creación de aplicaciones está muy limitada, ya que no se puede usar código.

Quizá uno de los aspectos más interesantes del App Inventor es que crea aplicaciones nativas. El programa no crea *scripts* que requieren tiempo para ejecutarse, sino que se crean directamente los archivos compilados .apk que cualquier persona con un *smartphone* puede instalar.

1.2.3. Virtual devices

Una vez configurado el entorno de desarrollo, solo queda configurar los dispositivos con los que podemos testear y debugar la aplicación. Para esto, se pueden usar dispositivos virtuales o dispositivos físicos (en el caso de que se disponga de un teléfono móvil inteligente o tableta Android).

1.2.3.1. Android virtual devices (AVD)

Los dispositivos virtuales son emuladores de hardware y software que se ejecutan según las especificaciones que les da el usuario. En la [documentación de Google](https://developer.android.com/studio/run/managing-avds.html?hl=es-419) [Web, Español] se detalla cómo configurar los dispositivos virtuales (AVD) una vez instalado el IDE. En el caso de la última versión de Android Studio, los dispositivos virtuales se pueden configurar en el menu de "Tools -> Android -> AVD Manager".

Es importante tener en cuenta que un emulador nunca tendrá las mismas funcionalidades que un dispositivo físico. En la [documentación](https://developer.android.com/studio/run/emulator.html) [Web, Inglés] se explican con detalle las capacidades y las limitaciones de los emuladores. De este modo, por ejemplo, el emulador es más lento, no permite testear algunos de los sensores, no tiene soporte para Bluetooth, no puede determinar el estado de la red, no pude saber si existe tarjeta SD o no, etc.

1.2.3.2. Hardware devices

Los dispositivos físicos o *hardware devices*, como les llama Google, también se pueden usar para testear y depurar una aplicación. Para esto, es necesario conectar el dispositivo a través de un cable USB al ordenador y obtener el *driver* USB del fabricante del dispositivo. En la documentación de Google, se explica cómo [configurar el dispositivo móvil](https://developer.android.com/studio/run/device.html) [Web, Inglés] y cómo [instalar el driver USB](https://developer.android.com/studio/run/oem-usb.html) [Web, Inglés] según el fabricante del dispositivo móvil.