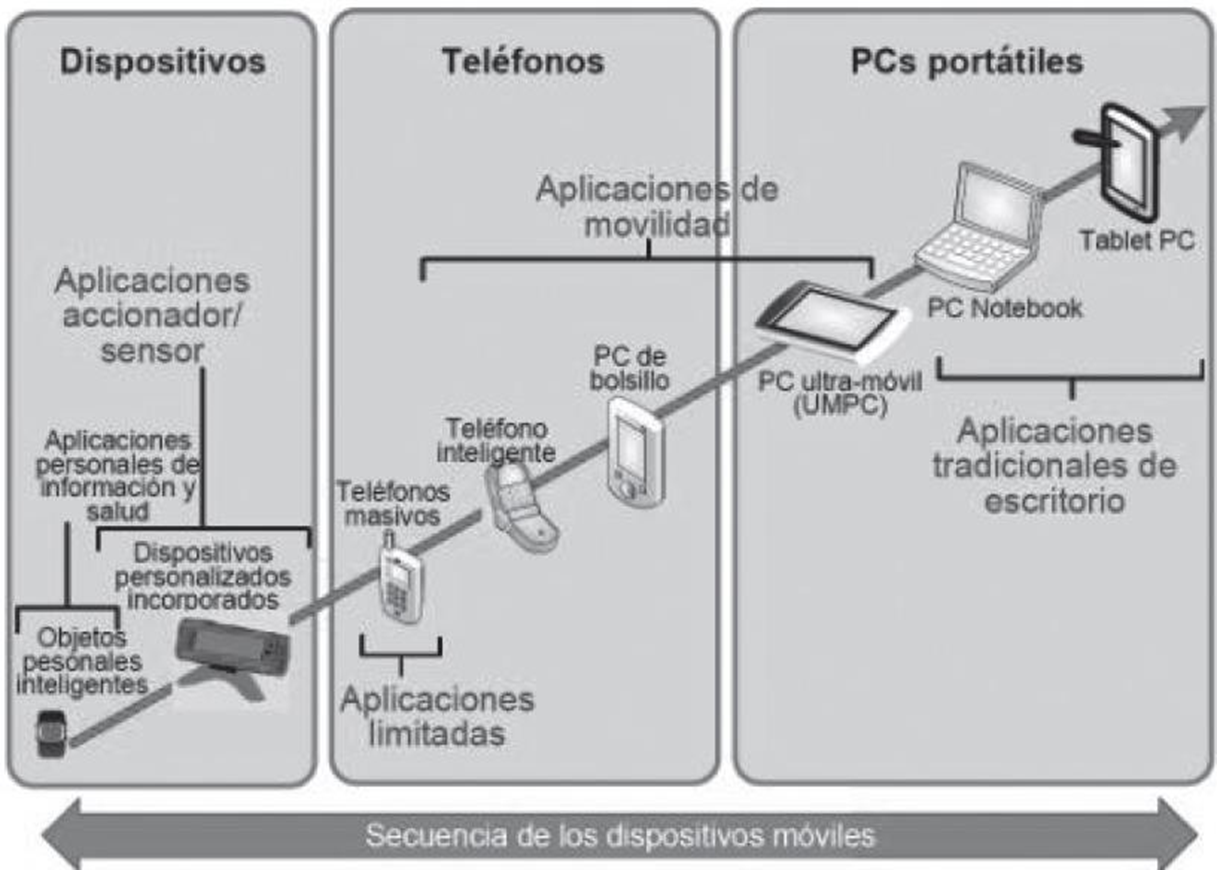
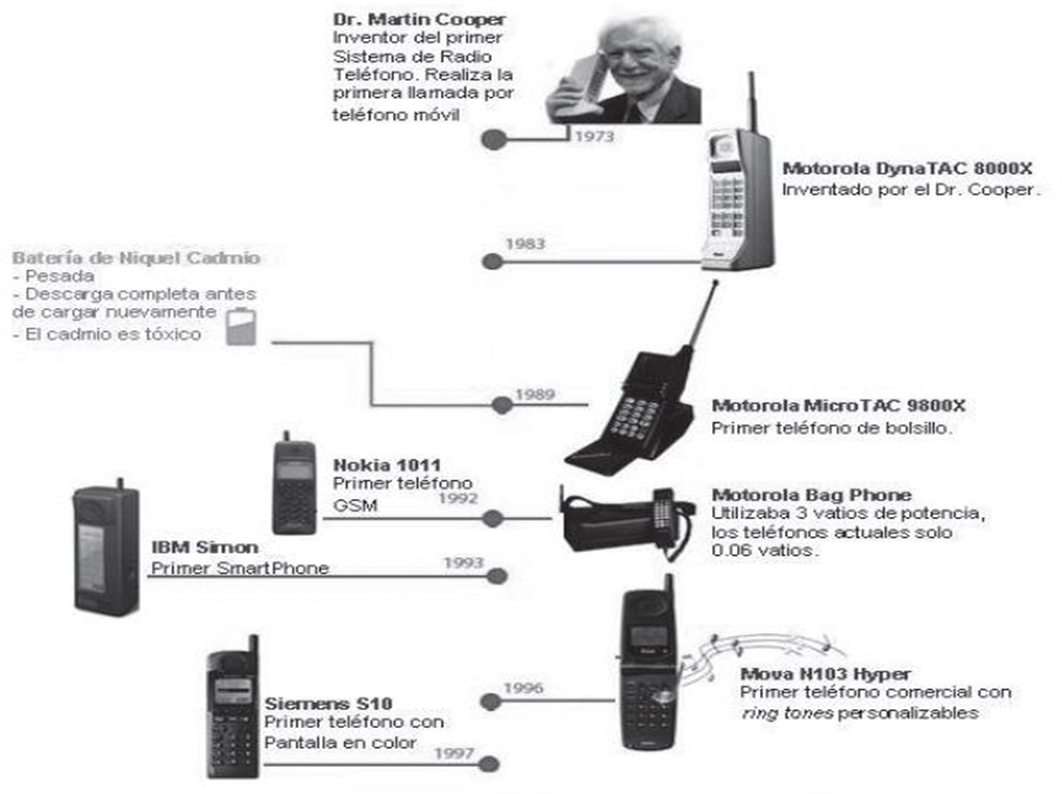
**Clase 1 teórica**

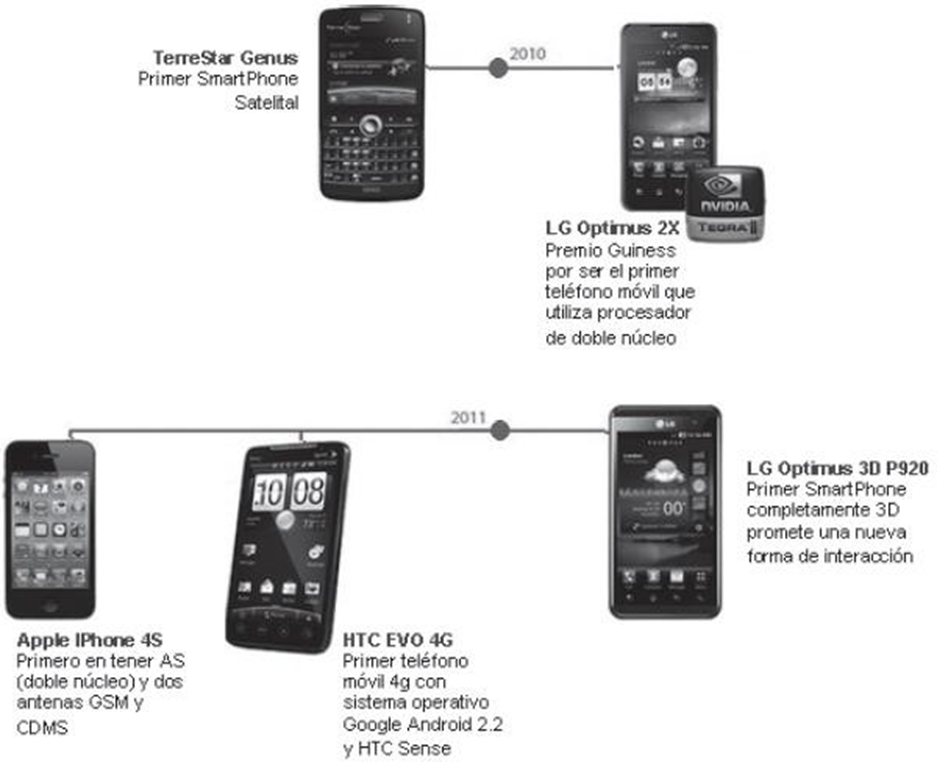
**Sistemas operativos para dispositivos móviles, historia, evolución**

La revolución de los dispositivos para comunicaciones móviles tiene tan solo 25 años. Durante este tiempo la tecnología ha evolucionado desde la voz a la información inalámbrica y el uso de dispositivos móviles se ha convertido en algo cotidiano. Gracias a ello, en la actualidad tenemos la posibilidad de comunicarnos con cualquier persona, en cualquier momento y desde casi cualquier lugar. La primera semilla para este gran desarrollo fue plantada por el célebre matemático escocés James Clerk Maxwell, quien formuló, en el año 1860, un par de ecuaciones cuya solución predijo la propagación de las ondas electromagnéticas a la velocidad de la luz. Se necesitaron 20 años para comprobar dicha predicción en un laboratorio y otros 20 años más para que se llevara a cabo la primera aplicación móvil.



Historia y evolución (1973-2011)







**Características y limitaciones en el desarrollo de aplicaciones para dispositivos móviles Las aplicaciones móviles son distintas a las hechas para pcs de escritorio por varias razones:**

* La capacidad para comunicarse desde cualquier lugar cambia la interacción del usuario con la aplicación.
* La interfaz de usuario para una pantalla y teclados pequeños difiere de forma significativa de la interfaz de una aplicación diseñada para una pc de escritorio.
* Los tipos de canales de comunicación son diferentes, los dispositivos móviles incorporan capacidades de voz, mensajería, información de geolocalización y vídeo conferencia.
* La naturaleza de las redes inalámbricas, aunque las redes ofrecen capacidades de datos de banda ancha, estas pueden variar, dependiendo de la calidad de la señal y de la disponibilidad de conexión de la red.

**Limitaciones en la ejecución de aplicaciones para dispositivos móviles**

Estas limitaciones están relacionadas, principalmente, con las características hardware y de conexión, asociadas a cada dispositivo móvil.

Las aplicaciones deben diseñarse evitando la sobrecarga de elementos multimedia, que exijan del dispositivo ciclos de procesamiento demasiado largos.

La distribución de los elementos en la interfaz marca la diferencia, lo fundamental es ofrecer interfaces a través de las cuales la entrada de los datos se realice de la forma más intuitiva y natural.

También hay que tener presente que cuando se trata del desarrollo de las aplicaciones para teléfonos móviles, la función de teléfono (es decir, realizar y/0 recibir llamadas), tiene ela prioridad mas alta. En cuanto a las conexiones, debemos tener presente que pueden fallar y de hecho es algo que sucede con relativa frecuencia, por ejemplo, por la falta de cobertura en determinadas áreas redes inalámbricas con claves.

**Enfoques para el desarrollo de aplicaciones móviles**

Aplicaciones Móviles Nativas: Si bien el desarrollo de aplicaciones para dispositivos móviles se remonta por lo menos a 12 años atrás, ha habido un crecimiento exponencial desde que se abrió la tienda de aplicaciones de iPhone en julio de 2008. Desde entonces, los fabricantes de dispositivos han creado tiendas de aplicaciones para otros dispositivos móviles, incluyendo Android, BlackBerry, Nokia Ovi, Windows Phone, entre otros [16].

Las aplicaciones nativas se desarrollan utilizando un entorno de desarrollo integrado (IDE) que proporciona las herramientas de desarrollo necesarias para la construcción y depuración de aplicaciones. El código fuente se compila para obtener código ejecutable, proceso similar que el utilizado para las tradicionales aplicaciones de escritorio.

**Clientes JME**: (*Java Platform Micro Edition*, Plataforma Java Micro Edición): las aplicaciones se escriben en Java y se compilan para ejecutarse contra una máquina virtual Java (JVM), diseñada específicamente para computadoras de mano y clientes móviles. Este enfoque proporciona dos beneficios principales: tiempo de desarrollo rápido y la posibilidad de utilizar el mismo código base en un gran número de dispositivos.

**Aplicaciones Móviles Híbridas** (PhoneGap, Sencha Touch, Ionic). son una combinación de tecnologías web como **HTML, CSS y JavaScript**, que no son ni aplicaciones móviles verdaderamente nativas, porque consisten en un WebView ejecutado dentro de un contenedor nativo, ni tampoco están basadas en Web, porque se empaquetan como aplicaciones para distribución y tienen acceso a las APIs nativas del dispositivo.

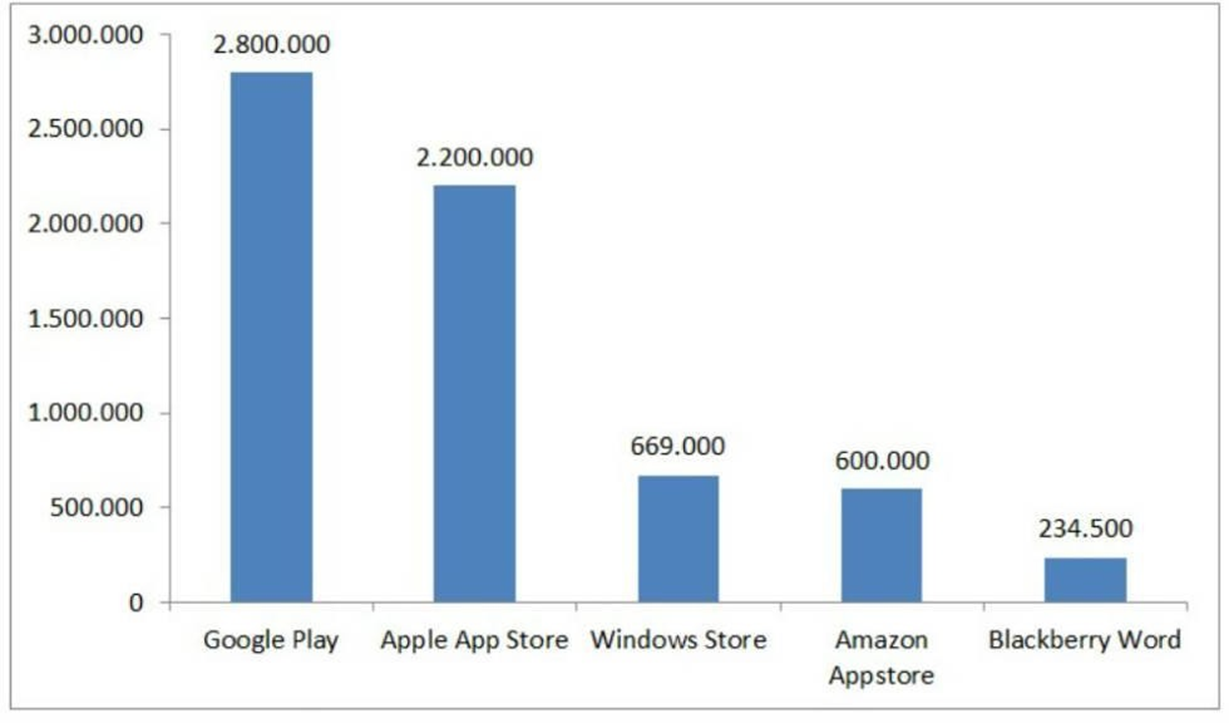
Desde una aplicación híbrida es posible acceder a las capacidades del dispositivo, a través de diversas API. las aplicaciones híbridas ofrecen grandes ventajas permitiendo la reutilización de código en las distintas plataformas, el acceso al hardware del dispositivo, y la distribución a través de las tiendas de aplicaciones. En contrapartida, se observan dos desventajas de las aplicaciones híbridas respecto del caso nativo:

1. la experiencia de usuario se ve perjudicada al no utilizar componentes nativos en la interfaz, y
2. la ejecución se ve ralentizada por la carga asociada al contenedor web.

**Predicciones sobre la distribución del Mercado de Móviles en Estados Unidos**

P

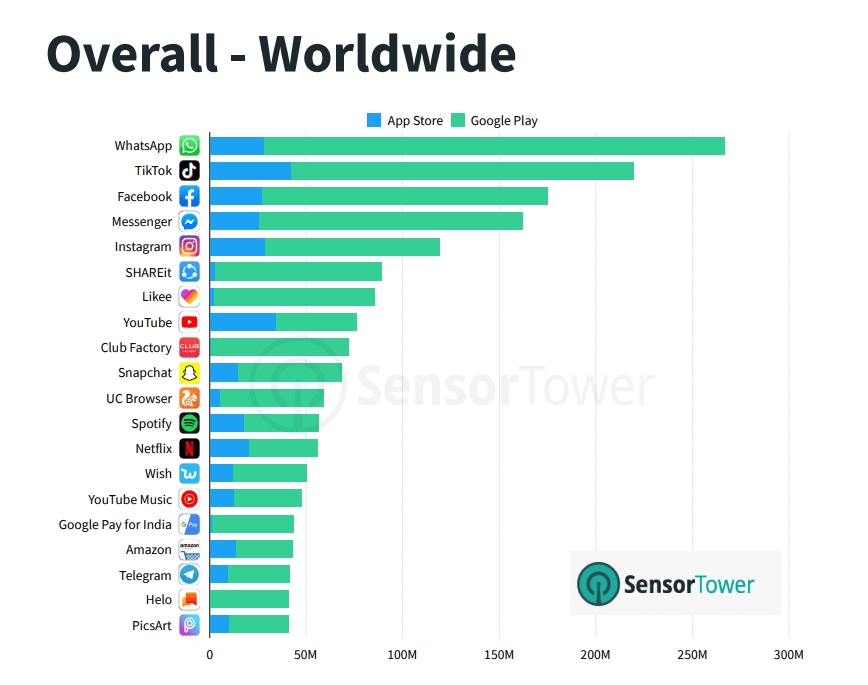
predicciones sobre la distribución del Mercado de Tiendas de Apps(2017)



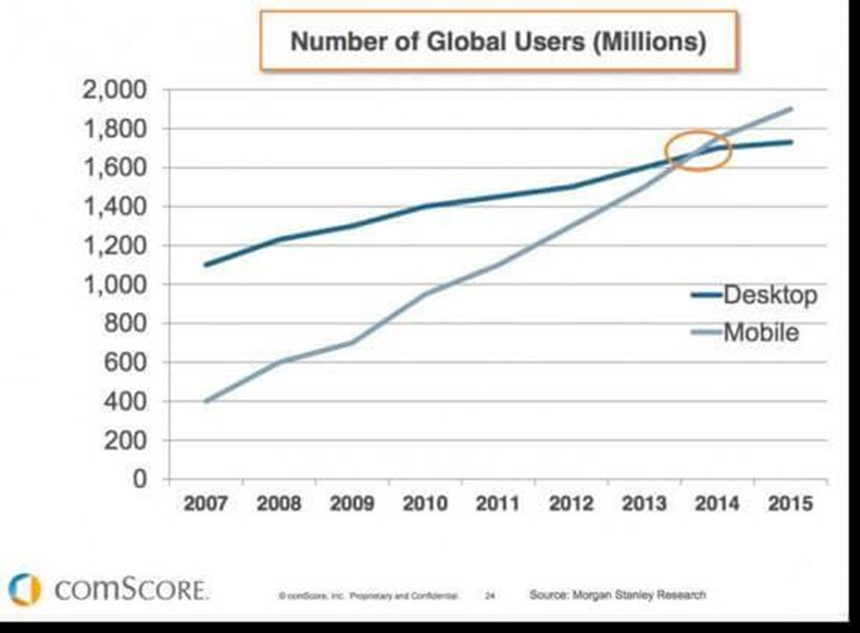
**Predicciones sobre la distribución del Mercado de Móviles en Estados Unidos**

**¿Como se gasta el tiempo en aplicaciones? (En que se entretienen los usuarios de apps)**

**Cuáles son las apps más utilizadas**

****

**Son más utilizados los dispositivos móviles (mobile) que los ordenadores (desktop)**

****

**¿Como se gasta el tiempo en aplicaciones? Mercado de Móviles en Estados Unidos Predicciones – 2017**

**(En que se entretienen los usuarios de apps)**

**Tecnologías disponibles**

* Android
* Android es una plataforma formada por un conjunto de software en estructura de pila (*software stack*) que incluye un sistema operativo, software para conectar aplicaciones (*middleware*) y aplicaciones base. El SDK (*Software Development Kit*, Kit de Desarrollo de Software) de Android proporciona varias herramientas y API (*Applications Programming Interface*, Interfaz de Programación Aplicaciones) que son necesarias desarrollar aplicaciones Android.
* Estas aplicaciones se desarrollan en lenguaje Java

Android está desarrollado por *Open Handset Alliance* (OHA), una agrupación de 78 compañías para desarrollar estándares abiertos para dispositivos móviles y que está liderada por Google. Inicialmente Android fue desarrollado por la compañía Android Inc., que fue comprada en el año 2005 por Google. El sistema operativo se anunció el 5 de noviembre de 2007. Google libera la mayoría del código Android bajo una licencia Apache (licencia libre y de código abierto). Desde su creación ha ido pasando por diferentes versiones, desde la versión primera (1.0) hasta la actual (12, denominada Tambien *API 33*).

**ARQUITECTURA ANDROID**

D

Clase 2

**Entornos de desarrollo para la creación de las aplicaciones para dispositivos móviles.**

Un entorno de desarrollo integrado (IDE) es un sistema de software para el diseño de aplicaciones que combina herramientas del desarrollador comunes en una sola interfaz gráfica de usuario (GUI).

**Características de un IDE**

**Editor de código fuente:** editor de texto que ayuda a escribir el código de software con funciones como el resaltado de la sintaxis con indicaciones visuales, el relleno automático específico del lenguaje y la comprobación de errores a medida que se escribe el código.

**Automatización de compilación local:** herramientas que automatizan tareas sencillas e iterativas como parte de la creación de una compilación local del software para su uso por parte del desarrollador, como la compilación del código fuente de la computadora en un código binario, el empaquetado del código binario y la ejecución de pruebas automatizadas.

**Depurador:** programa que sirve para probar otros programas y mostrar la ubicación de un error en el código original de forma gráfica.

**Razón de uso de un IDE**

Los IDE permiten que los desarrolladores comiencen a programar aplicaciones nuevas con rapidez, ya que no necesitan establecer ni integrar manualmente varias herramientas como parte del proceso de configuración. Tampoco es necesario que pasen horas aprendiendo a utilizar diferentes herramientas por separado, gracias a que todas están representadas en la misma área de trabajo. Esto resulta muy útil al incorporar desarrolladores nuevos, porque pueden confiar en un IDE para ponerse al día con los flujos de trabajo y las herramientas estándares de un equipo. De hecho, la mayoría de las características de los IDE están diseñadas para ahorrar tiempo, como el relleno inteligente y la generación automatizada del código, lo cual elimina la necesidad de escribir secuencias enteras de caracteres.

**Tipos de IDE**

**Cantidad de lenguajes compatibles:**algunos IDE son compatibles con un solo lenguaje, así que son mejores para un modelo de programación específico. Por ejemplo, IntelliJ es conocido principalmente como un IDE de Java. Otros IDE admiten una gran variedad de lenguajes de manera conjunta, como el IDE Eclipse, que admite Java, XML, Python, entre otros.

**Sistemas operativos compatibles:** el sistema operativo de un desarrollador determinará qué tipos de IDE son viables (salvo que el IDE esté basado en la nube), y estarán aún más limitados si la aplicación que se desarrolla está diseñada para el usuario final con un sistema operativo específico (como Android o iOS).

**Características de automatización:** si bien la mayoría de los IDE incluyen tres funciones fundamentales (el editor de texto, la automatización de compilación y el depurador), muchos admiten funciones adicionales, como la refactorización, la búsqueda de código y las herramientas de integración e implementación continuas (CI/CD).

**Impacto en el rendimiento del sistema:** es importante considerar el footprint del IDE en la memoria si el desarrollador desea ejecutar otras aplicaciones con uso intensivo de la memoria al mismo tiempo.

**Complementos y extensiones:** algunos IDE incluyen una función para personalizar los flujos de trabajo de forma que se adapten a las necesidades y preferencias del desarrollador.

**IDE para desarrollo de aplicaciones móviles**

La creciente popularidad de las aplicaciones diseñadas para teléfonos inteligentes y tabletas influye en casi todos los sectores, así que muchas empresas deben desarrollar aplicaciones móviles, además de las aplicaciones web tradicionales. Uno de los factores clave en el desarrollo de aplicaciones móviles es la selección de la plataforma. Por ejemplo, si se diseña una aplicación nueva para su uso en iOS, Android y una página web, será mejor comenzar con un IDE que brinde soporte en todas las plataformas para varios sistemas operativos.

**ANDROID STUDIO**

Android Studio es el entorno de desarrollo integrado oficial para la plataforma Android. Fue anunciado el 16 de mayo de 2013 en la conferencia Google I/O, y reemplazó a Eclipse como el IDE oficial para el desarrollo de aplicaciones para Android. La primera versión estable fue publicada en diciembre de 2014.

Android Studio es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de apps para Android y está basado en IntelliJ IDEA. Además del potente editor de códigos y las herramientas para desarrolladores de IntelliJ.

**Características:**

* Un sistema de compilación flexible basado en Gradle
* Un emulador rápido y cargado de funciones
* Un entorno unificado donde puedes desarrollar para todos los dispositivos Android
* Aplicación de cambios para insertar cambios de código y recursos a la app en ejecución sin reiniciarla
* Integración con GitHub y plantillas de código para ayudarte a compilar funciones de apps comunes y también importar código de muestra
* Variedad de marcos de trabajo y herramientas de prueba
* Compatibilidad con C++ y NDK
* Compatibilidad integrada con Google Cloud Platform, que facilita la integración con Google Cloud Messaging y App Engine