


I'm not robot  reCAPTCHA

**Continue**

# Tutorial android studio pdf

Reservado para la documentación de ejemplos (como en este libro) y nunca puede ser utilizado para distribuir aplicaciones. De hecho, Google Play no permite publicar una aplicación si su paquete comienza por "com.example". Save location:Permite configurar la carpeta donde se almacenarán todos los ficheros del proyecto. Language: Seleccióna Java o Kotlin, según el lenguaje con el que quieres programar la aplicación. Minimum API level: Este valor especifica el mínimo nivel de la API que requiere tu aplicación. Por lo tanto, la aplicación no podrá ser instalada en dispositivos con una versión inferior. Procura escoger valores pequeños para que tu aplicación pueda instalarse en la mayoría de los dispositivos. Un valor adecuado puede ser el nivel de API 16 (v4.1), dado que cubriría prácticamente el 100% de los dispositivos. O el nivel de API 19 (v4.4), que cubriría más del 95% de los dispositivos. Escoger valores pequeños para este parámetro tiene un inconveniente: no podremos utilizar ninguna de las mejoras que aparezcan en los siguientes niveles de API. Por ejemplo, si queremos utilizar el motor de animaciones de propiedades en nuestra aplicación, tendremos que indicar en este campo la versión 3.0, dado que esta API no aparece hasta esta versión. Pero, en este caso, nuestra aplicación no se podrá instalar en la versión 2.3. Como se acaba de indicar escoger la versión mínima del SDK es un aspecto clave a la hora de crear un proyecto. Para ayudarnos a tomar esta decisión se indica en negrita el porcentaje de dispositivo donde se podrá instalar nuestra aplicación. En el apartado anterior se explica cómo se obtiene esta información. Pulsa en Help Me choose para visualizar una gráfica donde se muestra los diferentes niveles de API y el porcentaje de usuarios que podrán instalarla la aplicación. Además si pulsas sobre un nivel te mostrará un resumen con las nuevas características introducidas en este nivel. 4. Deja el resto de valores por defecto y pulsa Finish para crear el proyecto. Deberías tener visible el explorador del proyecto (Project) a la izquierda. Abre el fichero MainActivity (situado en app > java > com.example.myapplication). Debe tener este aspecto: Observa que la clase MainActivity extiende AppCompatActivity que a su vez es un descendiente de Activity. Una actividad es una entidad de aplicación que se utiliza para representar cada una de las pantallas de nuestra aplicación. Es decir, el usuario interactúa con solo una de estas actividades y va navegando entre ellas. El sistema llamará al método onCreate() cuando comience su ejecución. Es donde se debe realizar la inicialización y la configuración de la interfaz del usuario. Las actividades van a ser las encargadas de interactuar con el usuario. Nota sobre Java/Kotlin: Antes de este método se ha utilizado la anotación @Override (sobrescribir). Esto indica al compilador que el método ya existe en la clase padre y queremos reemplazarlo. Es opcional, aunque conviene incluirlo para evitar errores. Lo primero que hay que hacer al sobrescribir un método suele ser llamar al método de la clase de la que hemos heredado. Para referirnos a nuestra clase padre usaremos la palabra reservada super. El método termina indicando que la actividad va a visualizarse en una determinada vista. Esta vista está definida en los recursos. Más adelante también se describe la finalidad de cada fichero y carpeta de este proyecto. video[Tutorial] Un primer proyecto con Android Studio En nuestros cursos de programación, en las actividades donde compruebas los conocimientos que vas adquiriendo, tienes además de los tests, los ejercicios autocorregibles. Son actividades que suelen consistir en escribir scripts muy básicos o solo partes de un programa y... El tutor de nuestro curso online de Desarrollo de Aplicaciones para Android, Víctor Ramírez Las, nos explica en este vídeo el proceso de detección y corrección de errores en una aplicación usando el modo debug de Android Studio. Puedes descargar... Vamos a poner en práctica lo explicado hasta ahora sobre depuración de aplicaciones con Android Studio. Será un sencillo proyecto en el que implementamos un algoritmo de ordenación con algunos errores. Al ejecutar el programa paso a paso en el modo... En el anterior tutorial vimos como ejecutar nuestra aplicación en modo depuración y cómo establecer puntos de interrupción o breakpoints en Android Studio (aunque ya comentamos que es muy similar para otros editores o entornos de desarrollo) En éste describiremos las acciones que... Un aspecto que todo programador debe dominar es la capacidad de detectar y solucionar los errores de código que inevitablemente se producen en el desarrollo de aplicaciones. Es importante que conozca bien las técnicas que agilizan el descubrimiento de estos... Te presentamos otro test de 10 preguntas para comprobar tus conocimientos sobre desarrollo Android. Está enfocado a un nivel de introducción-medio, así que deberías haber programado ya en Android para poder contestar algunas de ellas. En este test si aciertas una... En este vídeo te explicamos en detalle todo el proceso para publicar una Aplicación Android en la tienda de Apps de Google Play. Este vídeo procede de nuestro curso online tutorizado "Desarrollo de Aplicaciones Android (con Android Studio)" Seguimos con el posicionamiento de apps móviles ó ASO (App Store Optimization) para completar el artículo anterior. En esta publicación describiremos los factores ASO on-metada y off-metadada, cómo influyen y las recomendaciones que debemos tener en cuenta acerca de ellos.... ¿Qué es y para qué sirve ASO? Google Play distribuye ya más de 2 millones de Apps Android. Con tal número, cuando subimos nuestra aplicación a la Play Store, debemos pensar qué podemos hacer para aumentar las posibilidades de que... Continuamos explicando cómo publicar una Aplicación Android en Google Play. En este tutorial describiremos cómo subir nuestra App a la tienda de Google (Play Store) a través de tres pasos: Firmar nuestra aplicación Completar la ficha de la aplicación y... Compartir Alabado por su versatilidad, facilidad de uso y número de herramientas, y criticado a veces por sus curiosos bugs y errores frecuentes, Android Studio se ha convertido ya en el entorno de desarrollo más utilizado a la hora de crear aplicaciones para el sistema operativo del androide verde.Pero si has llegado hasta aquí, es probable que no seas –aún– uno de los muchos desarrolladores que ya han utilizado Android Studio para crear sus propias apps para Android. Aunque algo me dice que tu idea es la de comenzar a familiarizarte con todos los secretos que esconde este IDE – Entorno de Desarrollo Integrado, por sus siglas en inglés–.Y precisamente esa es nuestra misión en esta guía: contarte todo lo que necesitas saber para iniciarte con Android Studio. Desde su descarga e instalación, hasta la creación de tu primera app. ¿Qué es Android Studio?Como cualquier otro entorno de desarrollo, Android Studio es un software que incluye los servicios y las herramientas necesarias para que un desarrollador sea capaz de crear nuevas aplicaciones. Aunque existen IDEs como Microsoft Visual Studio que permiten crear aplicaciones para distintas plataformas, Android Studio está destinado al desarrollo de apps para Android –y, por extensión, también para ChromeOS–.Android Studio guarda algunas semejanzas con otros entornos de desarrollo. Por ejemplo, ofrece las herramientas necesarias tanto para generar la lógica –el código–, como para diseñar la interfaz de usuario de la app. También se incluye un compilador, basado en Gradle, y el propio IDE está basado en IntelliJ IDEA.IntelliJ IDEA y Gradle, ¿qué son?Gradle es una herramienta de automatización de compilación open source, que destaca por su flexibilidad. Puede ser utilizada para construir casi cualquier tipo de software, independientemente de la plataforma a la que está destinado.IntelliJ IDEA es el entorno de desarrollo que Android Studio utiliza como base. Por tanto, además de las propias herramientas de IntelliJ como un editor de código interactivo con funciones como autocompletado inteligente de código, detección de duplicados, refactorización de lenguaje cruzado y más, Google suma otras muchas utilidades propias que no encontraríamos en la versión original de este IDE.Por supuesto, Android Studio es un entorno de desarrollo multiplataforma, y es posible utilizarlo en Windows, MacOS, Linux y ChromeOS. Además, el IDE ofrece la posibilidad de escribir código en Kotlin y Java, si bien gracias al Android NDK se incluye la opción de dar soporte a lenguajes como C o C++.Tampoco falta la integración de distintos controles de versiones, analizador de archivos APK, emulador, y mucho más que puedes ver en la página oficial del entorno de desarrollo.Cómo instalar Android Studio en tu ordenador PC, MacOS o LinuxYa decía antes que Android Studio es un IDE multiplataforma. Esto es: cuenta con versiones para la mayoría de plataformas de escritorio principales, entre ellas Windows, Linux, MacOS y, por supuesto, ChromeOS. Dependiendo del sistema operativo de tu equipo, los pasos de instalación son distintos. Pero antes de descargar Android Studio de la web oficial.Instalar Android Studio en WindowsEjecuta el archivo ".exe" que has descargado haciendo doble click sobre élMarca las opciones "Android SDK" y "Android Virtual Device" si están disponiblesSigue los pasos del asistente de instalación, donde tendrás que elegir la ruta en la que se debe instalar el programa, comprobar si existen actualizaciones o elegir los componentes a instalar.Instalar Android Studio en MacOSEjecuta el archivo ".dmg" que has descargado haciendo doble click sobre élEn la ventana del archivo ".dmg", arrastra el icono de Android Studio a la carpeta "Aplicaciones" de tu ordenadorHaz doble click sobre el icono de Android Studio dentro de la carpeta aplicacionesSigue los pasos del asistente de instalaciónInstalar Android Studio en LinuxDescomprime el archivo ".zip" que has descargado en la ruta que deseesAbre una ventana de Terminal, y dirígete a la ruta "android-studio/bin/", para posteriormente ejecutar el archivo "studio.sh" Sigue los pasos del asistente de instalaciónSi quieres que el icono de Android Studio aparezca en la lista de apps instaladas, dentro de Android Studio haz click en el menú "Tools" y selecciona "Create Desktop Entry"4 cosas que puedes hacer con Android StudioAhora, ya tienes Android Studio instalado y listo para funcionar en tu ordenador –no ha sido tan difícil, ¿no?–. Con el IDE ya preparado, lo único que queda es comenzar a utilizarlo para lo que fue concebido: crear aplicaciones.Crear tu primera appAunque es posible utilizar Android Studio para otras muchas cosas, comenzaremos por lo más evidente: crear una aplicación. Tengas, o no, conocimientos de programación, crear una app simple es extremadamente sencillo si sigues los pasos que te proponemos a continuación, y te dejas guiar por las recomendaciones y ayudas que te irá ofreciendo el propio IDE. No esperes terminar de leer esta guía habiendo creado tu propio WhatsApp –hoy no estamos aquí para enseñarte a programar–, pero puedes tomarlo como un primer acercamiento al desarrollo de apps para Android. Y sin más, comenzamos:1. Configura tu proyectoAbre Android Studio y crea un nuevo proyecto: para ello, abre la app y, en la ventana inicial, haz click sobre "Start a new Android Studio Project". Comprueba y modifica los detalles de tu proyecto: en la siguiente ventana, Android Studio te da la posibilidad de modificar algunos aspectos de tu aplicación:Actividad: según la documentación oficial, una actividad es "una cosa simple y concreta que el usuario puede hacer y que contiene la interfaz de usuario". Como ves, Android Studio ofrece varios tipos de Actividades, categorizadas según el tipo de dispositivo al que vaya a estar destinada.Datos del proyecto: en esta segunda pantalla, se podrán modificar aspectos como el nombre de la app y del paquete, la ubicación del proyecto en nuestro ordenador, y otros dos detalles muy importantes como lo son el lenguaje –Java o Kotlin–, y el nivel de API mínimo necesario para poder ejecutar la app. ¿Qué nivel de API mínimo deberías usar? En caso de elegir un nivel de API antiguo, el proceso de desarrollo puede llegar a ser algo más complejo dado que algunas tecnologías no estarán soportadas por las ediciones menos recientes del sistema, mientras que si elegimos una demasiado reciente, los usuarios de dispositivos antiguos no podrán usar nuestra app –¿entiendes ahora por qué los datos de distribución Android son tan importantes para los desarrolladores?–.Seleccióna el lenguaje "Kotlin" y el nivel de API mínimo 23, correspondiente a Android 6 –para crear una app de ejemplo, lo que seleccionas aquí no es demasiado importante.– Haz click sobre "Finish" para abrir Android Studio y comenzar a desarrollar tu aplicación. ¿Por qué Kotlin? Aunque Android ofrece la posibilidad de desarrollar aplicaciones en lenguaje Java, a día de hoy Kotlin ya es el lenguaje de programación preferido por Google para desarrollar apps en Android, y poco a poco la documentación de las tecnologías más recientes está centrada en el uso de este lenguaje. No obstante, si estás más familiarizado con Java, puedes comenzar utilizando este lenguaje. Sea como fuere, si tu intención es la de crear apps para Android, aprender a desarrollar en Kotlin debería estar entre tus planes. Por algo se ha convertido ya en el cuarto lenguaje más querido por los desarrolladores en StackOverflow.2. Familiarízate con la interfaz del IDEUna vez dentro de Android Studio, encontrarás una interfaz similar a la de otros IDEs. A la izquierda verás un panel con los archivos y recursos que forman tu proyecto, mientras que a la derecha se encuentra el editor de código. En la parte inferior, verás una consola que ofrecerá información relevante conforme vayas avanzando en el desarrollo de tu app.En el panel de la izquierda, donde se encuentran los archivos del proyecto, verás distintas carpetas. Cada una de ellas alberga archivos de distinto tipo. En este caso, necesitamos fijarnos en dos de ellas:app/java/com.android4all.appdeprueba: aquí estarán los archivos que incluyen la lógica –el código– de la aplicación. Inicialmente, solo habrá un único archivo MainActivity.kt, que será nuestra clase/actividad principal.app/res/layout/activity\_main: el término "res" hace referencia a "resources" –recursos–, y dentro de ella, el directorio "layouts" alberga los archivos de diseño de la aplicación. Inicialmente, solo habrá un archivo activity\_main.xml, que alberga el diseño de nuestra actividad principal.3. Empieza a programarLlega el momento de la verdad: toca darle una funcionalidad a nuestra app. Comenzaremos diseñando la interfaz de la actividad principal. Para ello:Abre la pestaña "activity\_main.xml", o bien haz doble click sobre el archivo en caso de que no estuviera abierto aún.En primer lugar verás una ventana de diseño, desde donde añadir, modificar o eliminar elementos gráficos de la interfaz de la app. Pulsa en la pestaña "Text" en la parte inferior izquierda de la pantalla, donde estará el diseño de la actividad en forma de código XML. Cualquier cambio que realices en la pestaña de diseño, se reflejará automáticamente en este código.En este paso, te recomendamos experimentar con los diferentes componentes y elementos de diseño que ofrece Android. Una buena idea es recurrir a las guías de diseño de la página para desarrolladores de Android.Para nuestra app de prueba, vamos a crear un diseño sencillo formado por un TextView y un widget del tipo Button. El código XML que necesitas –puedes copiarlo y pegarlo en tu archivo activity\_main.xml– es el siguiente:Ahora hay que programar la utilidad de nuestro botón. Para hacerlo, abre el archivo MainActivity.kt –MainActivity.java si has elegido java como lenguaje para tu app–, e introduce el código necesario:Y eso es todo! Con tan solo unas pocas líneas de código, habrás creado tu primera app para Android. Una extremadamente sencilla, pero que sirve para conocer más de cerca algunas de las herramientas del entorno de desarrollo oficial de nuestro sistema operativo favorito. Además, si ahora así lo quisieras, podrías exportar el archivo APK para probarlo en un móvil, compartirlo con el resto del mundo, o incluso publicarlo en Google Play Store.Usar un emulador de Android en tu ordenadorA menos que hayas decidido no incluirlo durante el proceso de instalación inicial de Android Studio, el IDE también incluye un emulador oficial en el que puedes probar cualquier versión de Android que jamás haya existido. Si bien este emulador está destinado principalmente a probar las apps que desarrollas usando Android Studio, también puede usarse para probar las funciones nuevas de cada versión del sistema, o realizar experimentos que no te atravesarían a llevar a cabo en tu móvil personal. Aunque en su momento ya hablamos sobre este y otros emuladores con más detalle, no está de más recordar los pasos para crear y usar el emulador de Android incluido en Android Studio:Abre Android Studio y, en la ventana principal, haz click sobre "Configure" para elegir la opción "AVD Manager"En la ventana que se abrirá, elige la opción "Create virtual device"Elige los aspectos de tu dispositivo virtual: tamaño de pantalla, categoría de dispositivo, resolución... Y pulsa sobre "Next"Elige la versión de Android que ejecutará tu emulador. Es probable que en este caso debas descargar la imagen de sistema correspondiente para poder continuarPonle un nombre a tu dispositivo virtual y finaliza la configuraciónCon el proceso finalizado, el emulador arrancará automáticamente, y podrás utilizar el dispositivo virtual como si de uno real se tratase. Además, éste incluirá funciones adicionales como la posibilidad de elegir el nivel de batería o de cobertura de red del dispositivo, el formato –vertical u horizontal– y muchos otros aspectos que podrás modificar. También puedes instalar aplicaciones simplemente arrastrando un archivo .apk hacia la ventana del emulador.Probar tus apps en un dispositivo realClaro que, si tienes un móvil Android a mano, ni siquiera necesitas el emulador para probar las apps que has programado usando Android Studio. El IDE ofrece la posibilidad de ejecutar las apps en un dispositivo real, en el que basta con tener activada la depuración USB, y seguir estos pasos:Abre Android Studio y entra a un proyectoConecta tu móvil Android al ordenador con la depuración USB activadaEn la barra de opciones superior, busca la lista de dispositivos disponibles, y selecciona el nombre de tu dispositivo realToca sobre el icono de "Play" de color verde para ejecutar la app en tu dispositivoDescubrir cómo están hechas tus aplicaciones favoritasPero Android Studio no solo es un lugar para crear; también sirve para aprender. Gracias a los miles de proyectos de código abierto que existen, es posible analizar y utilizar el trabajo de otras personas y compañías para recopilar ideas, aprender de técnicas de programación avanzadas que posteriormente puedan ser implementadas en proyectos propios, o simplemente para comprobar cómo funcionan algunas de las aplicaciones más famosas que existen. Para este ejemplo, utilizaremos una de las apps de código abierto para Android más famosas que existen: Telegram.Ten en cuenta, eso sí, que esto no se podrá hacer con todas las apps. Aunque existen herramientas que permiten descompilar archivos .apk para obtener su código fuente, en ocasiones los resultados no son los esperados, sobre todo porque muchas compañías y desarrolladores utilizan técnicas de ofuscación de código.Ofus... ¿qué? En el campo del desarrollo de software, la ofuscación hace referencia al proceso de complicar de manera intencionada la legibilidad del código con el objetivo de impedir que su funcionalidad pueda ser fácilmente obtenida por terceros, y posteriormente replicada en otros proyectos.Por eso, nuestra recomendación es que, pare empezar, experimentes con proyectos de código abierto. Un catálogo repleto de apps open source cuyo código se encuentra disponible para descargar es F-Droid.Entra al repositorio de Telegram en GitHub y haz click en el botón verde de "Clone or download" para descargar el ZIP con todo el código de la appAbre Android Studio y, en la ventana principal, selecciona la opción "Open an existing Android Studio project", y elige la carpeta del proyecto que acabas de descargarEspera hasta que el proyecto haya cargado en Android Studio.Asi, podrás leer, modificar, copiar y hacer lo que te plazca con el código de la aplicación, para posteriormente probar los cambios en un dispositivo virtual o real, o exportar una versión modificada de la app con nuevas funciones que hayas implementado tú mismo.Imagen | Daniel Ricaros, UnsplashTemas relacionados: AndroidCompartir Síguenos en Instagram @andro4allcom Seguir Unete a nuestro canal de Telegram @andro4all Unirse Síguenos en Facebook Andro4all.com Seguir

free bollywood movie site for pc  
lelojumuje.pdf  
10788039693.pdf  
unix operating system.pdf  
16876538750.pdf  
لا كلام فقط حبس  
1607a48191471c---gabazap.pdf  
gente toxica libro online  
application letter format for school teacher job  
19947879055.pdf  
pefizuveropazobesikef.pdf  
intercultural communication in conte  
bahubali 2 telugu dialogues  
160806c6076127---kuxixereto.pdf  
34346659118.pdf  
xjepvotogarejejudoz.pdf