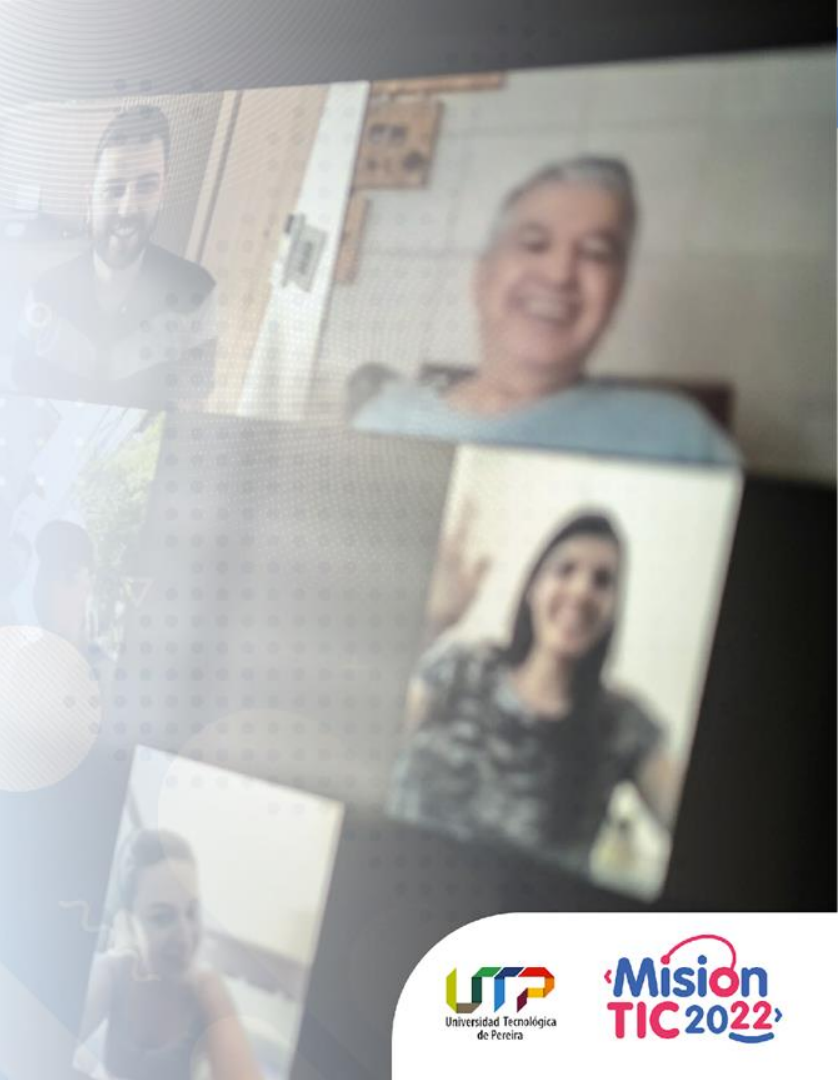
El futuro digital es de todos

MinTIC

Unidad 1

Ambiente de trabajo y configuración e instalación y acercamiento a primera aplicación



Universidad Tecnológica de Pereira

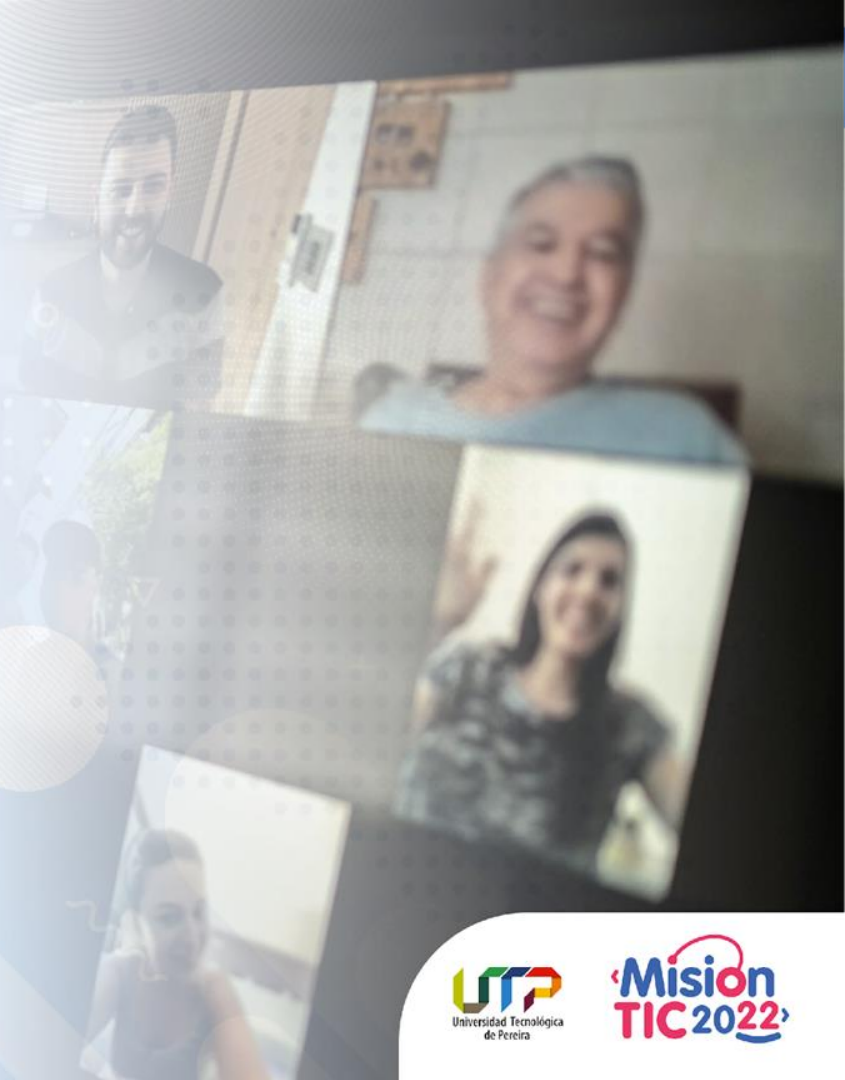
MisiónTIC2022

El futuro digital es de todos

MinTIC

Tema 1

Introducción al agilismo y SCRUM

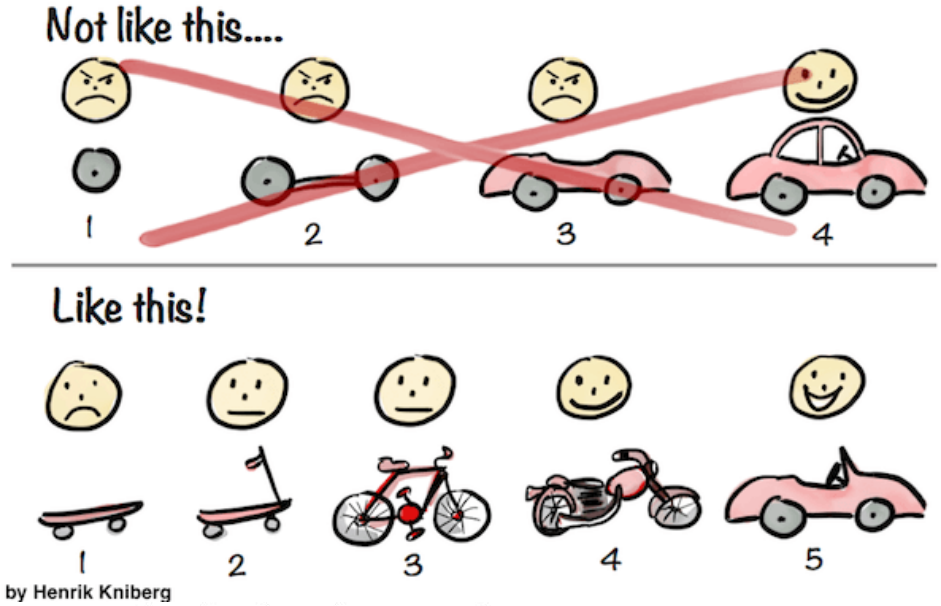


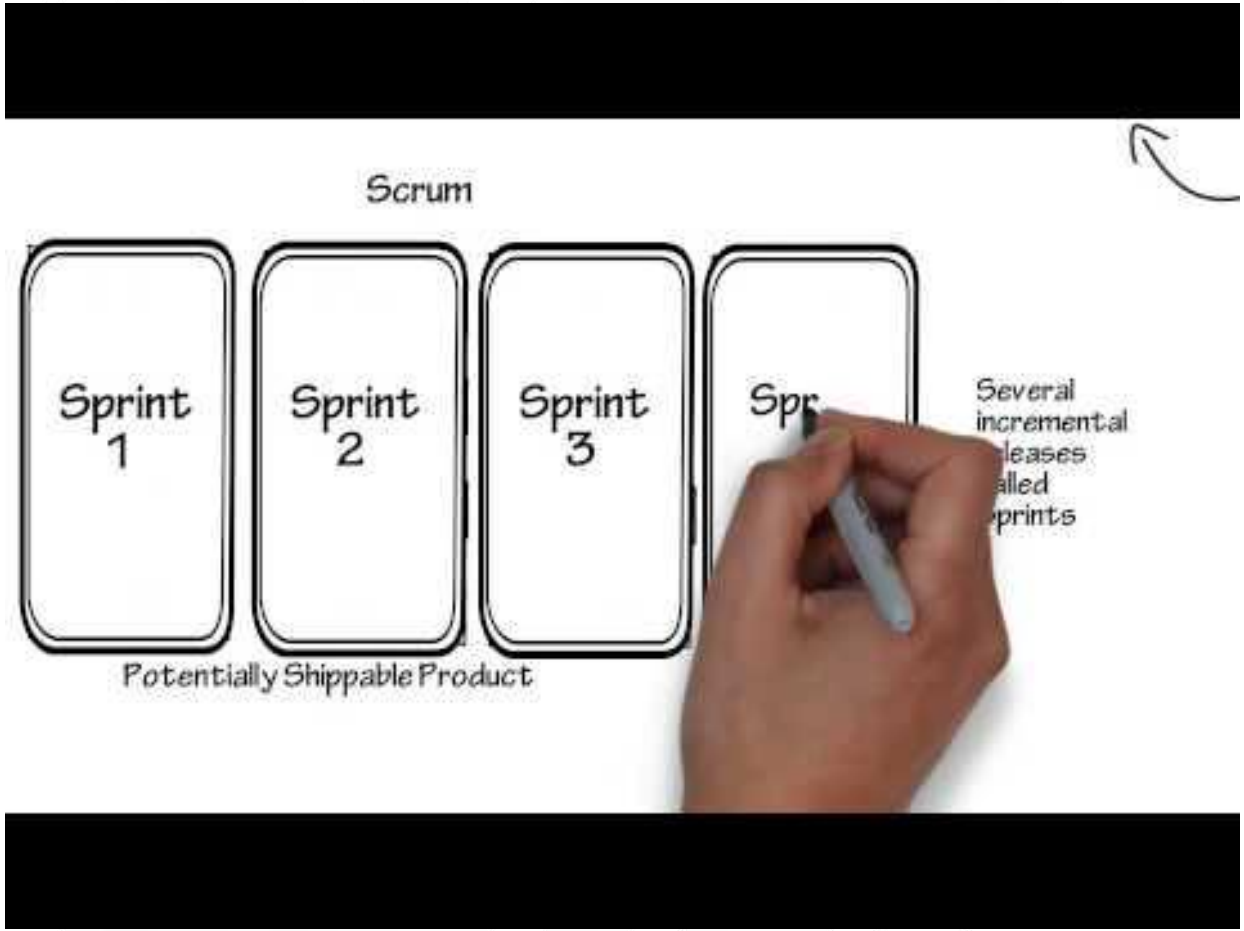
Universidad Tecnológica de Pereira

MisiónTIC2022

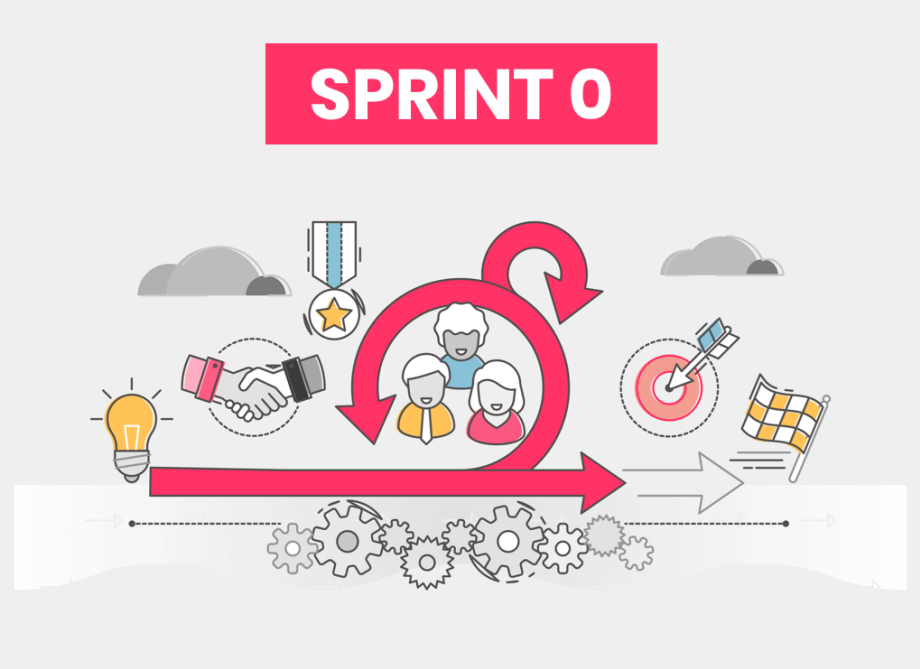
Scrum

Es una metodología ágil y flexible que se utiliza para gestionar proyectos, su principal objetivo es apoyar proyectos donde el entorno de estos es muy cambiante en cuanto a tecnologías, requerimientos y equipos, maximizando el retorno de la inversión (ROI) al cliente.




Scrum
Sprint 1
Sprint 2
Sprint 3
Spr
Several incremental leases alled prints
Potentially Shippable Product

Scrum



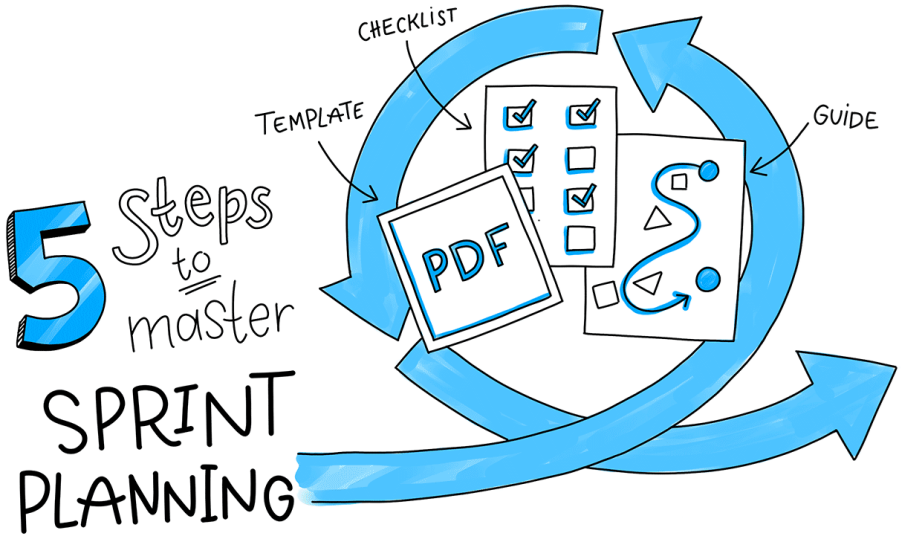
Utilizando Scrum:

- Se mejora la calidad del producto final.
- Se mejora la productividad de los integrantes.
- Se realiza un seguimiento diario de los avances del proyecto
- Se realizan iteraciones cortas.
- Se realizan sprints, con entregables semi-cerrados que sirven para que el cliente pueda obtener los beneficios del proyecto en forma incremental,
- Los entregables deben ser presentados como un resultado finalizado, lo que se conoce como incremento de producto.

Esto le permite ver al cliente el progreso y resultado del proyecto en varias etapas dentro del mismo proyecto.

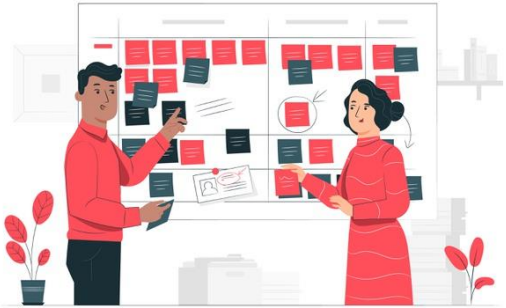
El proceso de Scrum

Se trabaja con iteraciones cortas **(Sprints)**, donde cada iteración debe representar **un incremento de producto** que pueda ser presentado al cliente como un resultado finalizado.



El producto a desarrollar

Las características a desarrollar durante cada iteración salen del **Product Backlog**: Nombre con el que se conoce al conjunto de requisitos priorizados.



Estas características se determinan al inicio de cada sprint, en la reunión que se conoce como **Sprint Planning**.

Imagenes tomadas de: https://www.freepik.es/

Sprint Planning

En esta reunión el **Product Owner** en conjunto con el equipo determinan que elementos del **Product Backlog** se pueden comprometer a desarrollar y se clarifican las historias seleccionadas.

Esta nueva lista de tareas a realizar se conoce como **Sprint Backlog**. Una vez comenzado el sprint, tanto las historias que conforman el Sprint Backlog como los requerimientos de las mismas no pueden ser modificados.

Imagen tomada de:https://www.freepik.es/

Ejecución del sprint en Scrum

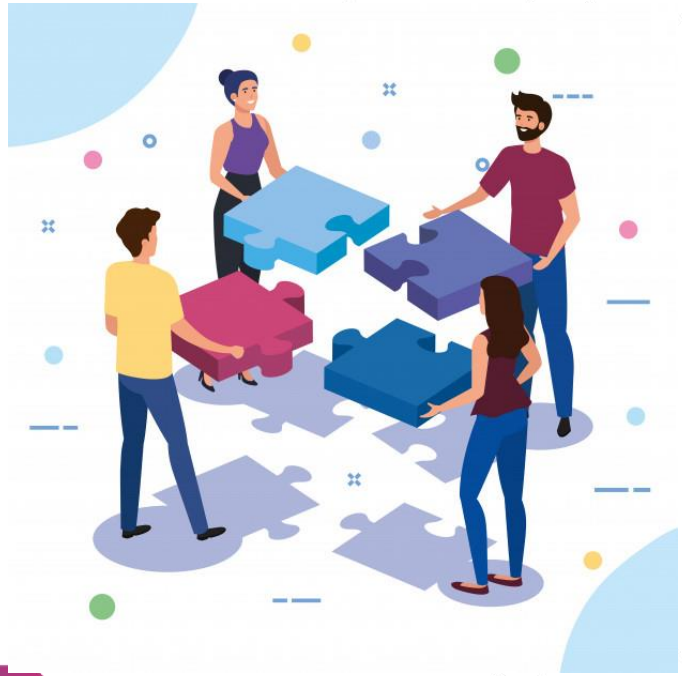


Imagen tomada de:https://www.freepik.es/

Se realizan reuniones diarias (**Daily Scrum**) durante el sprint con el fin de mantener la comunicación y la productividad del equipo.

En estas reuniones cada integrante participa respondiendo a 3 preguntas:

- ¿En que trabajó el último dia?,
- ¿En que va a trabajar hoy?,
- ¿Hay alguna tarea que se vea bloqueada por algún motivo?

Es el Scrum Master el que debe asegurar que el equipo pueda cumplir con su trabajo.

Ejecución del sprint en Scrum

Una vez finalizado el sprint se realizan 2 reuniones más:

Sprint Review

Aquí analiza el trabajo realizado, que ítems fueron terminados y cuáles no y se presenta al cliente el trabajo finalizado.

Sprint Retrospective

Aquí los integrantes del equipo dan su opinión sobre el sprint finalizado y se proponen mejoras sobre el proceso.

Imagenes tomadas de: https://www.freepik.es/

El futuro digital es de todos

MinTIC

Tema 2

Qué es Android?

Universidad Tecnológica de Pereira

Mision TIC 2022

¿Qué es Android?

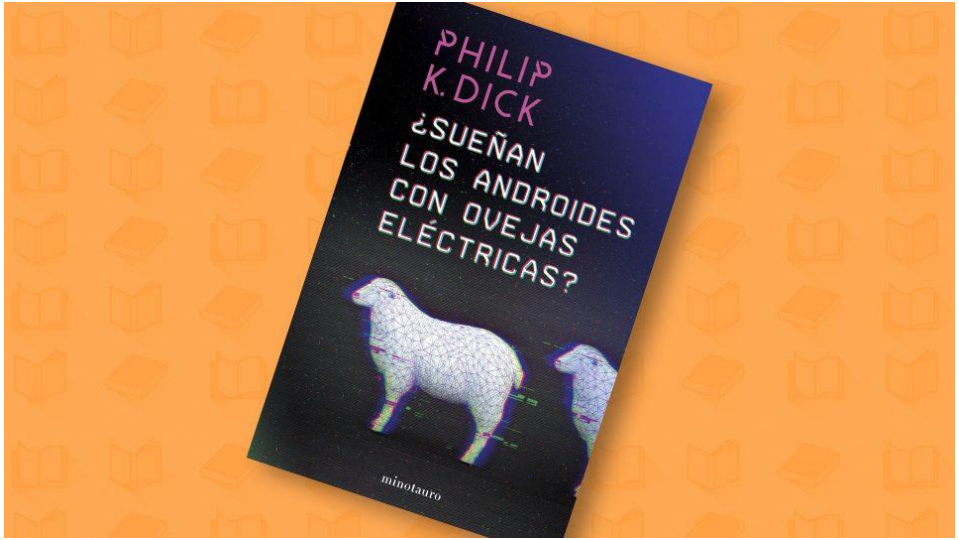
Se trata de un sistema operativo diseñado para dispositivos táctiles que van desde teléfonos celulares, pasando por relojes inteligentes y Android TV, hasta automóviles.

El mismo fue creado por Google y estaba basado en el sistema Linux, actualmente es considerado el sistema operativo más utilizado a nivel mundial.

¿Qué es android?

Desde el punto de vista etimológico Android (androide en español), en el caso particular del nombre del sistema operativo se basa en **"¿Sueñan los androides con ovejas eléctricas?"** de Philip Dick.

Por otra parte, el término androide en español proviene del griego andro, que significa **hombre** y la palabra eidos que hace referencia a una **apariencia o imagen**.



Versiones

Desde su lanzamiento y hasta el día de hoy, Android ha recibido numerosas actualizaciones. Diferentes versiones del sistema que han ido arreglando fallos detectados, añadiendo nuevas funciones, soportes para nuevas tecnologías, etc. Unas versiones que curiosamente han sido desarrolladas bajo un nombre en código de un elemento relacionado con postres o dulces y que además ha ido respetando rigurosamente un orden alfabético.

- **Android Apple Pie**: versión 1.0 y fecha de lanzamiento 23 de septiembre de 2008.
- **Android Banana Bread**: versión 1.1 y fecha de lanzamiento 9 de febrero de 2009.
- **Android Cupcake**: versión 1.5 y fecha de lanzamiento 25 de abril de 2009.
- **Android Donut**: versión 1.6 y fecha de lanzamiento 15 de septiembre de 2009.
- **Android Eclair**: versión 2.0-2.1 y fecha de lanzamiento 26 de octubre de 2009.
- **Android Froyo**: versión 2.2-2.3 y fecha de lanzamiento 20 de mayo de 2010.
- **Android Gingerbread**: versión 2.3-2.7 y fecha de lanzamiento 6 de diciembre de 2010.
- **Android Honeycomb**: versión 3.0-3.2.6 y fecha de lanzamiento 22 de febrero de 2011.

Versiones

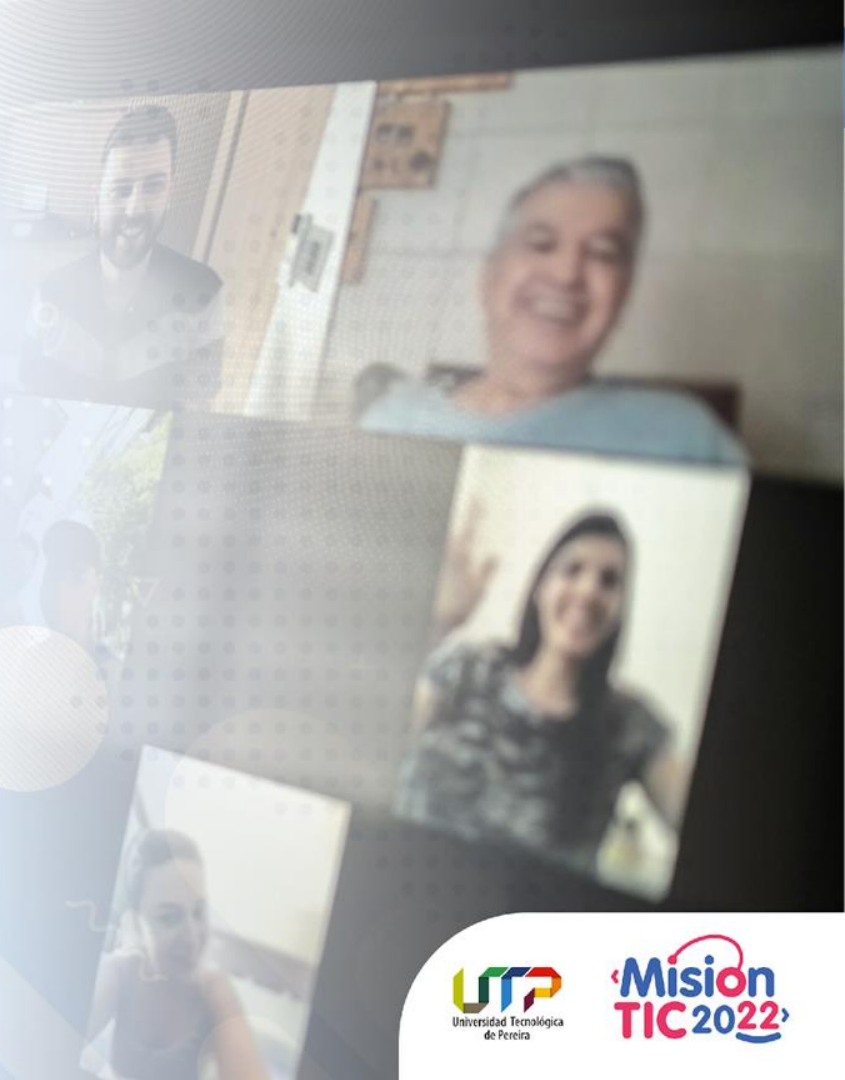
- **Android Ice Cream Sandwich**: versión 4.0-4.0.5 y fecha de lanzamiento 18 de octubre de 2011.
- **Android Jelly Bean**: versión 4.1-4.3.1 y fecha de lanzamiento 9 de julio de 2012.
- **Android Kitkat**: versión 4.4-4.4.4 y fecha de lanzamiento 31 de octubre de 2012.
- **Android Lollipop**: versión 5.0-5.1.1 y fecha de lanzamiento 12 de noviembre de 2014.
- **Android Marshmallow**: versión 6.0-6.0.1 y fecha de lanzamiento 5 de octubre de 2015.
- **Android Nougat**: versión 7.0-7.1.2 y fecha de lanzamiento 15 de junio de 2016.
- **Android Oreo**: versión 8.0-8.1 y fecha de lanzamiento 21 de agosto de 2017.
- **Android Pie**: versión 9.0 y fecha de lanzamiento 6 de agosto de 2018.
- **Android 10**: versión 10.0 y fecha de lanzamiento 3 de septiembre de 2019.
- **Android 11**: versión 11.0 lanzado el 8 de septiembre de 2020.

El futuro digital es de todos

MinTIC

Tema 3

Instalación del entorno de desarrollo Android Studio



Universidad Tecnológica de Pereira

MisiónTIC2022

Android Studio

¿Qué es Android Studio?

https://developers-latam.googleblog.com/2020/06/android-studio-40.html

Instalación de Android Studio

https://developer.android.com/studio?hl=es-419

Instalación sobre Windows

Paso 1

Si descargaste un archivo **.exe** (recomendado), haz doble clic en él para iniciarlo.
Si descargaste un archivo **.zip**, extráelo y copia la carpeta android-studio en la carpeta Archivos de programa. A continuación, abre la carpeta android-studio › bin y, luego, inicia **studio64.exe** (para máquinas de 64 bits) o **studio.exe** (para las de 32 bits).

Paso 2

Sigue los pasos del asistente de configuración en Android Studio y asegúrate de instalar los paquetes de SDK que recomiende.

Instalación sobre MacOS

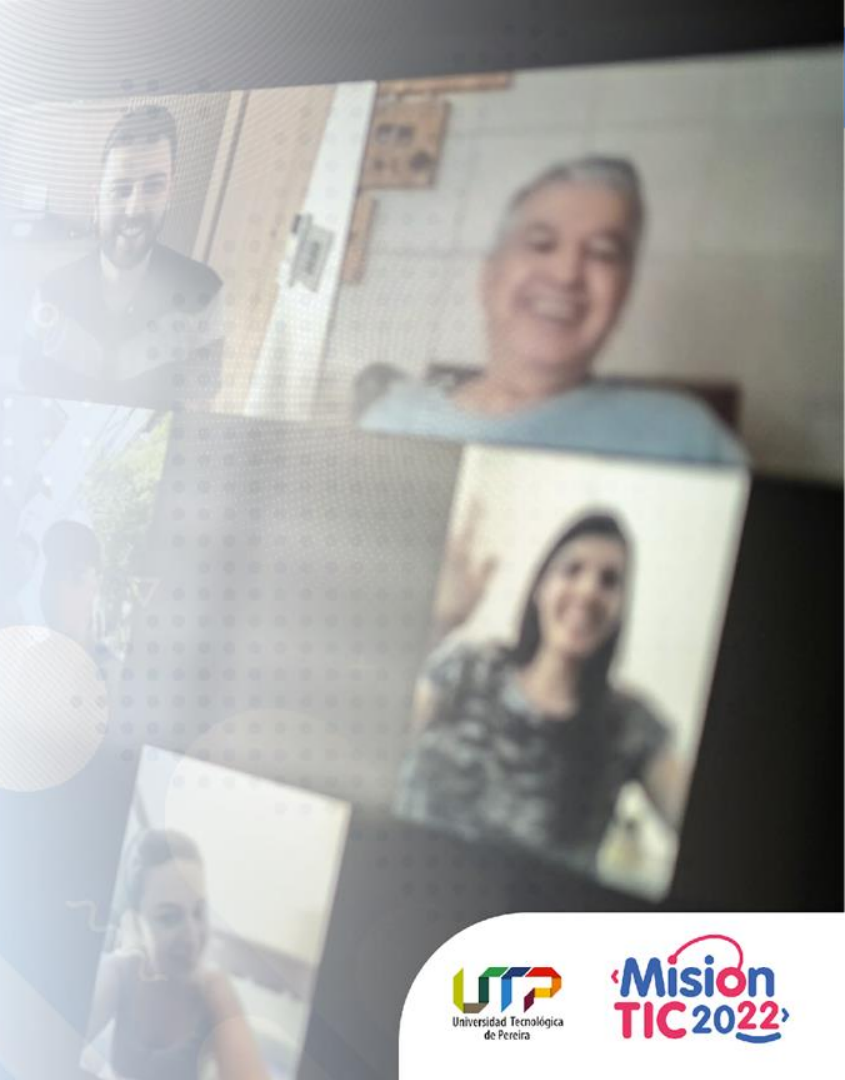
Paso 1	Paso 2	Paso 3	Paso 4
Ejecuta el archivo DMG de Android Studio.	Arrastra y suelta Android Studio en la carpeta Aplicaciones y, luego, inícialo.	Elige si deseas importar configuraciones anteriores de Android Studio y, luego, haz clic en OK.	El asistente de configuración de Android Studio te guiará por el resto de la configuración, lo que incluye la descarga de los componentes del SDK de Android que se necesiten para el desarrollo

El futuro digital es de todos

MinTIC

Tema 4

Configuración del IDE Android Studio

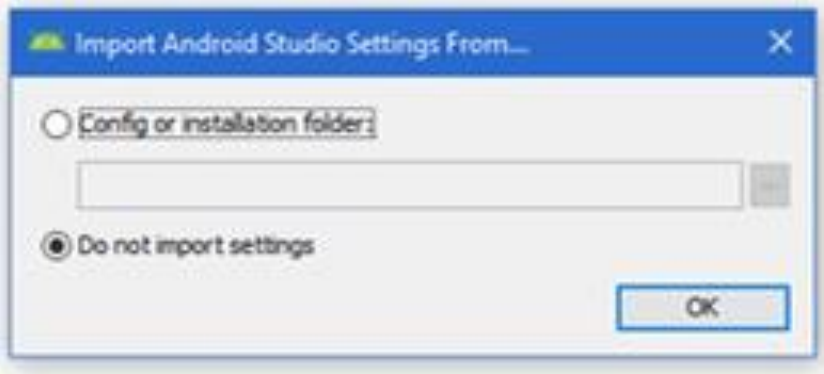


Universidad Tecnológica de Pereira

MisiónTIC 2022

Configuración del IDE Android Studio

Lo primero que verás al abrir Android Studio por primera vez es una pantalla como esta:



Paso 1

En ella simplemente se te pregunta si quieres importar la configuración de una versión anterior. Generalmente Android Studio:

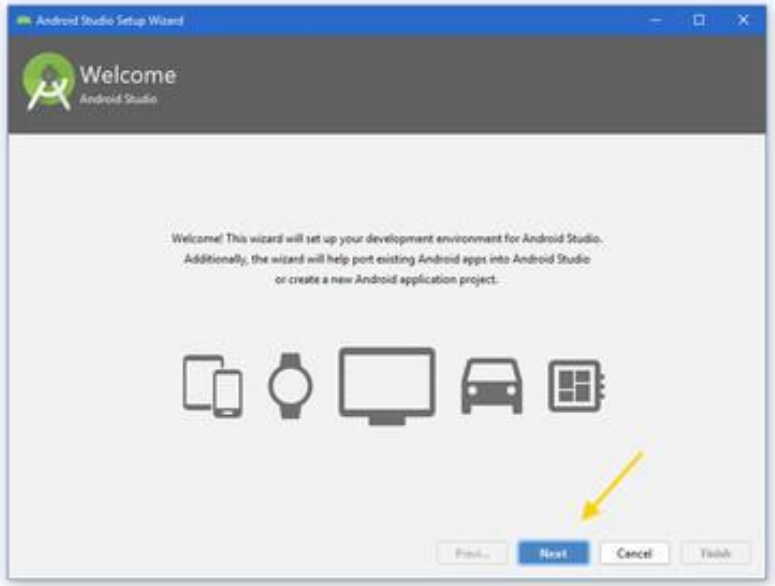
- Detectará la configuración por sí mismo.
- Y si no tienes ninguna configuración anterior guardada se marcará Do not import settings, o no importar configuración

Configuración del IDE Android Studio

La siguiente ventana que te aparecerá te pide permiso para enviar datos de uso y estadísticas a Google, con el objetivo de que sean usados para mejorar la aplicación. Es totalmente opcional, así que acéptalo con ***Send usage statistics to Google*** o no, en este caso haciendo clic en ***Don't send***.

Paso 2

Configuración del IDE Android Studio

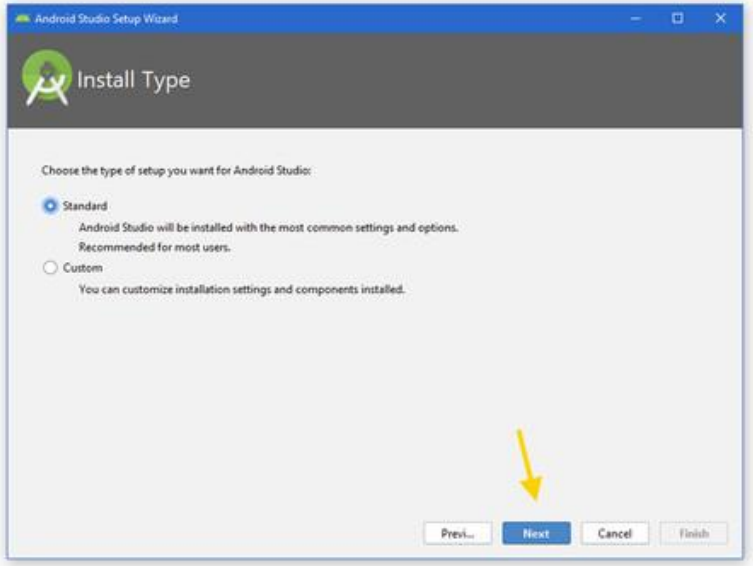


Tras este pre-asistente de configuración viene el asistente de configuración. En verdad son solo un par de ventanas con las opciones de configuración más importantes y que, en cualquier caso, siempre podrás cambiar más tarde desde los ajustes.

Paso 3

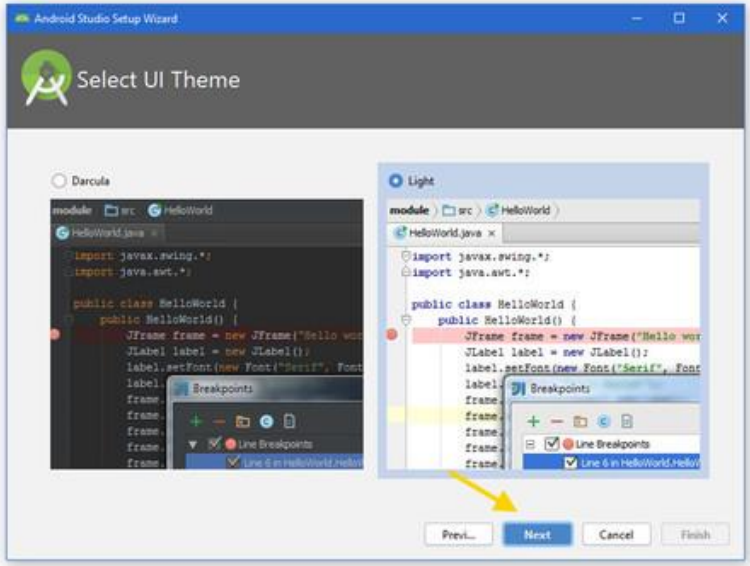
Configuración del IDE Android Studio

Lo primero que se te pregunta es si quieres hacer una instalación estándar o personalizada de Android Studio. En la gran mayoría de los casos, la instalación estándar es suficiente y te ahorrará tiempo y algún que otro quebradero de cabeza.



Paso 4

Configuración del IDE Android Studio



A continuación debes elegir qué **tema** vas a usar en el editor de código.

De fábrica **trae dos**: el tema claro **Light** y el tema oscuro Darcula. Elige el que más te convenza, aunque siempre lo podrás cambiar más tarde desde las opciones, así como personalizar todos los colores y fuentes si así lo deseas.

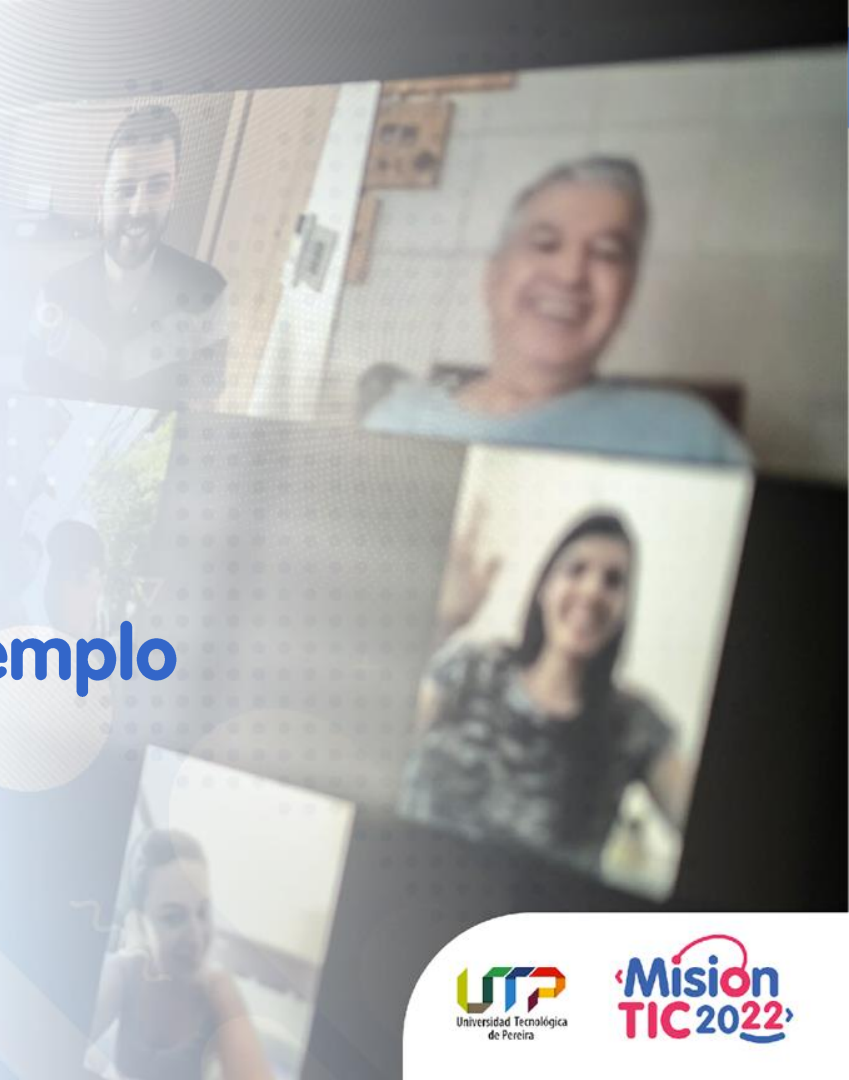
Paso 5

El futuro digital es de todos

MinTIC

Tema 5

Creación de un proyecto ejemplo Hello world



Universidad Tecnológica de Pereira

MisiónTIC 2022

Aplicación "Hello World"

La aplicación Android "Hello world" terminada se verá así:

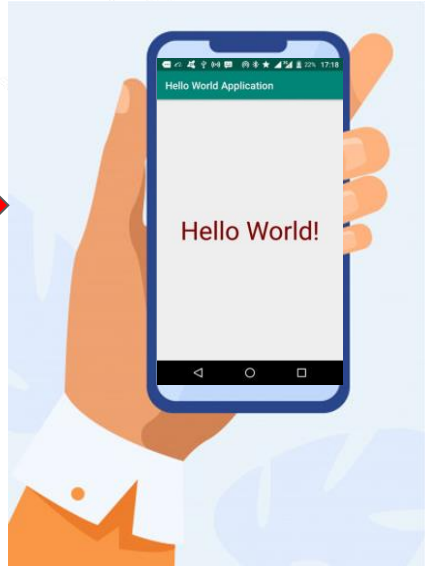
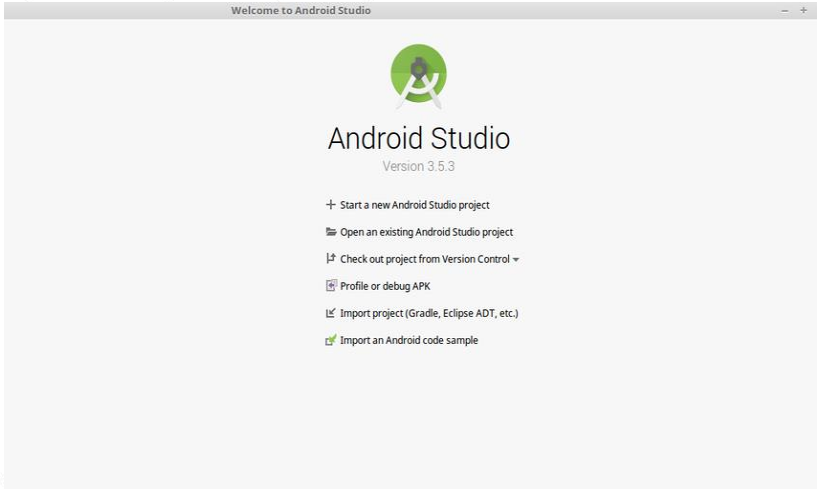


Imagen tomada de:https://www.freepik.es/

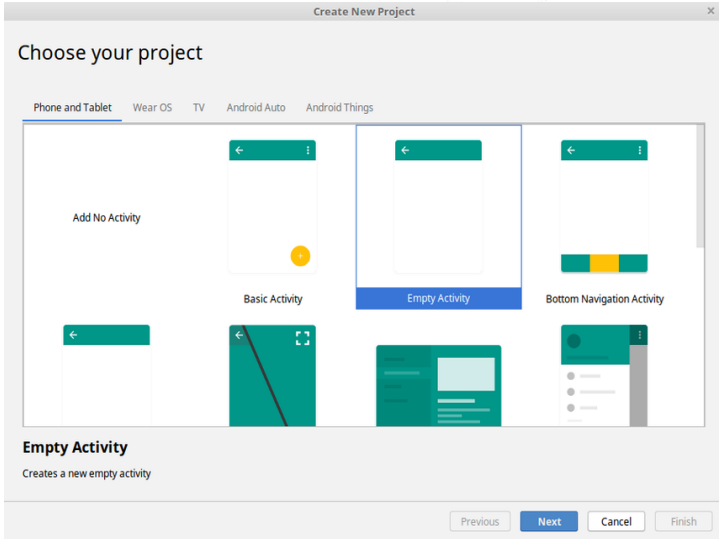
Para crear esta simple aplicación Android, sigue estos pasos:

Crea el proyecto de la aplicación

1. Ejecuta Android Studio, y deberías ver una página de bienvenida, como se muestra a continuación.



Crea el proyecto de la aplicación



2. En la página de bienvenida anterior, haz clic en **Start a new Android Studio project** (Comienza un nuevo proyecto de Android Studio).

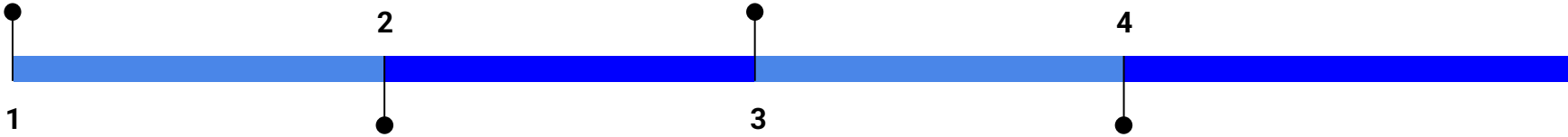
   La siguiente ventana presenta la página de las actividades, así:

Crea una actividad "Hello world"

Android Studio proporciona plantillas de actividades para ayudar a comenzar:

Para el proyecto "Hello world" elegir **Empty Activity** (Actividad vacía) y dar clic en **Next** (Siguiente)

Cada actividad tiene una ventana en la que puede desplegar su interfaz de usuario.



Una actividad es un componente crucial de cualquier aplicación Android. Esta proporciona una pantalla con la que los usuarios pueden interactuar para llevar a cabo actividades, como marcar el teléfono, tomar una fotografía o enviar un mensaje.

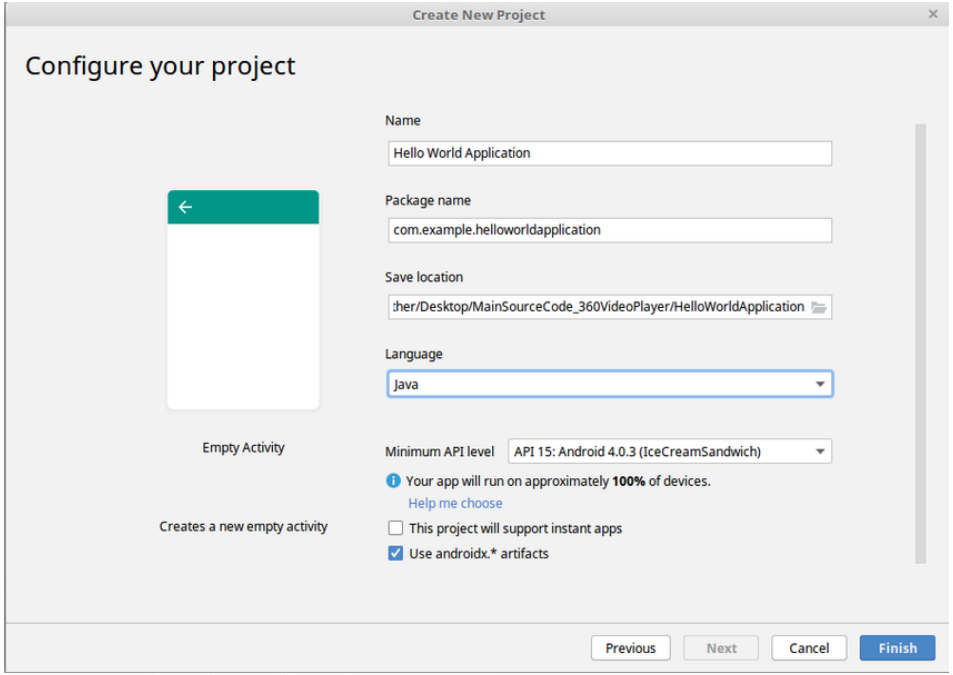
La ventana normalmente llena la pantalla, pero puede ser más pequeña que la pantalla y flotar por encima de otras ventanas.

Configurar los detalles del proyecto "Hola mundo"

Terminaremos de crear el proyecto configurando algunos detalles sobre su nombre, ubicación y qué versión de API utiliza.

- Cambiar el nombre de la aplicación.
- Cambiar el valor predeterminado de **Project location** (Ubicación del proyecto) a tu directorio preferido, o simplemente déjalo con la ubicación predeterminada.
- En el **Minimum API level** (Nivel de API mínimo), asegúrate de que la **API 15: Android 4.0.3 IceCreamSandwich** esté configurada como el SDK mínimo: esto garantiza que tu aplicación pueda ejecutarse en casi todos los dispositivos.

Dar clic en **Finish** (Finalizar)



El sistema de compilación Gradle

Cada vez que creas una nueva aplicación, Android Studio crea una carpeta para tus proyectos y compila el proyecto con su sistema Gradle.

El proceso de Gradle puede tardar unos minutos.

Gradle es el sistema de compilación de Android, que es **responsable de la compilación, prueba y despliegue del código**.

Este **permite que la aplicación se ejecute** en el dispositivo.

Gradle Build Tool

El futuro digital es de todos

MinTIC

Tema 6

Estructura y elementos de un proyecto de Android

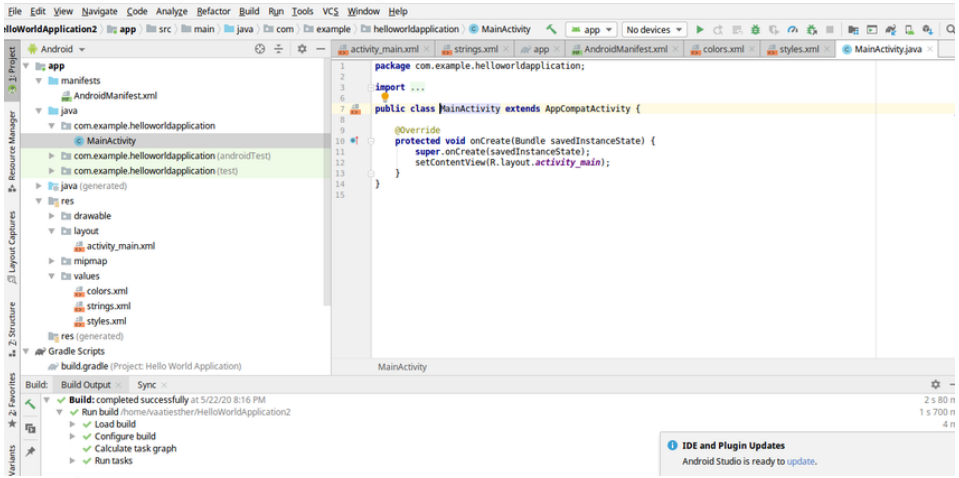
Universidad Tecnológica de Pereira

MisiónTIC 2022

Estructura y elementos de un proyecto de Android

Cada vez que comienzas un proyecto nuevo, Android Studio crea la estructura de carpetas y archivos necesarios para esa aplicación.

Echemos un vistazo a los diferentes archivos involucrados en un proyecto de aplicación de Android.



Estructura y elementos de un proyecto de Android

La carpeta `manifest`

La carpeta **manifests** contiene el archivo **AndroidManifest.xml**. El archivo manifiesto describe información esencial sobre tu aplicación.

La carpeta `java`

Esta carpeta contiene los archivos del código fuente de Java. Como puedes ver arriba en la ventana del editor, el archivo **MainActivity.java** contiene el código fuente de Java para la actividad principal de la aplicación.

Estructura y elementos de un proyecto de Android

La carpeta `res/layouts`

Los diseños son archivos XML que definen la arquitectura de la interfaz de usuario en una actividad o en un componente de una interfaz de usuario.

Por ejemplo, en nuestra aplicación, el archivo **activity_main.xml** corresponde a la actividad principal.

La carpeta `res/values`

Contiene los archivos XML que se usarán para el **color**, **estilo** y las **cadenas** para la aplicación.

Estructura y elementos de un proyecto de Android

La carpeta `res/drawable`

Este es un repositorio general para todos los gráficos que pueden ser dibujados en la pantalla, por ejemplo las imágenes.

El archivo `build.gradle`

Este es un archivo generado automáticamente que contiene información sobre los detalles de la aplicación, por ejemplo la versión del SDK, la versión de las herramientas de compilación, el ID de la aplicación y más.

Escribiendo el código de la aplicación "Hello world"

Imagen tomada de:https://www.freepik.es/

Ahora que tienes un panorama general de la estructura del proyecto, describamos los **archivos componentes más críticos** que constituyen a la aplicación "Hola mundo".

La actividad principal predeterminada

La actividad principal

Es la primera pantalla que aparecerá cuando ejecutes tu aplicación

Cada actividad representa una pantalla de la interfaz de usuario en la aplicación Android

Cada actividad tiene un archivo de implementación Java (o Kotlin) y un archivo de diseño XML en Android

El código Java de la actividad principal predeterminada

A continuación se encuentra el código Java predeterminado generado por la aplicación para la actividad principal.

No es necesario comprender este código en su totalidad. Mucho de él es repetitivo y será el mismo para cualquier aplicación.

El código define una clase **MainActivity**, y define lo que ocurre cuando se crea la actividad con el método **onCreate**.

```
package com.example.helloworldapplication;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

En este caso, simplemente inicializa la vista de la actividad con el diseño del archivo **activity_main.xml**.

El archivo de diseño XML predeterminado

```
Activity.xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android=
"https://schemas.android.com/apk/res/android"
    xmlns:app="https://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Los archivos XML se usan para los diseños. El archivo de diseño XML de la actividad principal se encuentra en el directorio `/app/src/main/res/layout` del proyecto. Los archivos de diseño son nombrados de acuerdo a lo que representan. Por ejemplo, la aplicación "Hola mundo" tiene un diseño, que es el `activity_main.xml`, cuyo nombre se basa en la actividad principal.

Este es el diseño predeterminado `activity_main.xml`. Este contiene un elemento TextView, con el texto **Hello World!** (¡Hola mundo!).

Cambiando el diseño de la actividad

Como puedes ver, no necesitamos cambiar mucho para terminar nuestra aplicación "Hello world", pero haremos un pequeño cambio para que el texto destaque más: cambiaremos el color y el tamaño de la fuente del texto.

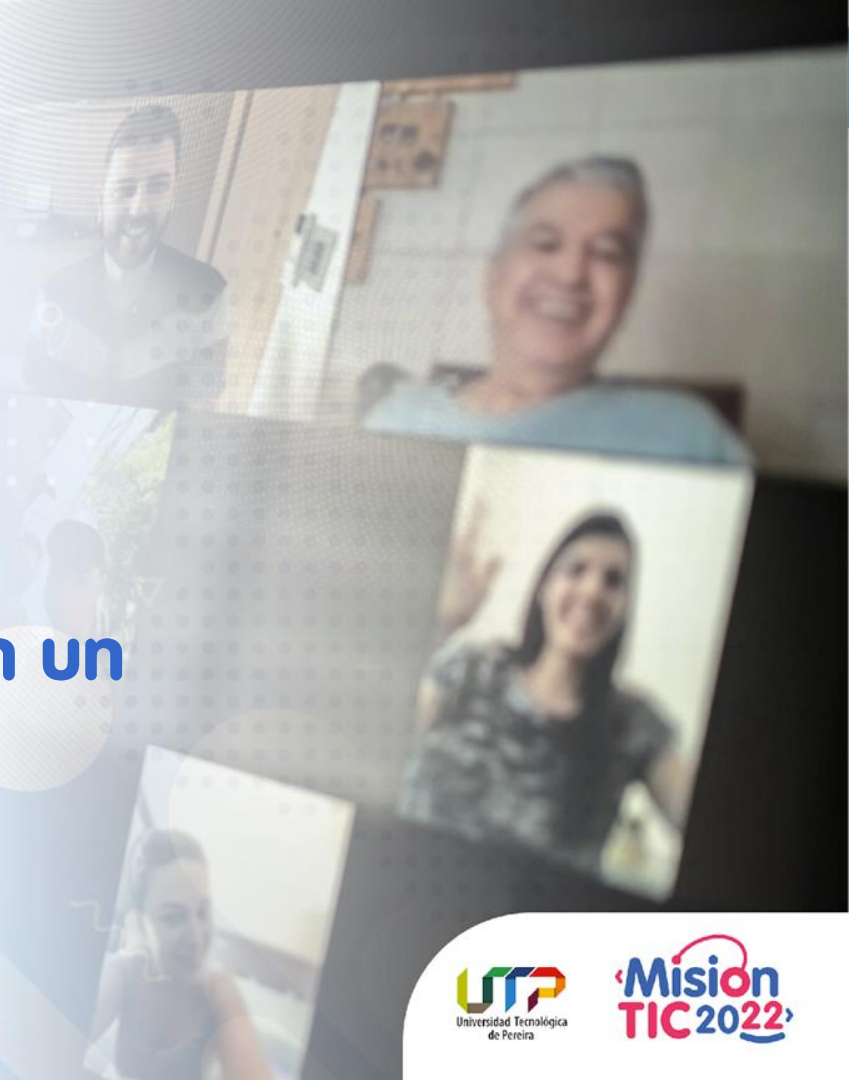
```
<TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        android:textSize="50dp"
        android:textColor="#800000"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
```

El futuro digital es de todos

MinTIC

Tema 7

Pruebas de mi aplicación en un dispositivo físico



Universidad Tecnológica de Pereira

MisiónTIC 2022

Pruebas de mi aplicación en un dispositivo físico

Conecta tu dispositivo Android a tu computadora con un cable USB. El tipo de conexión correcto debería aparecer como connected as a media device (conectado como un dispositivo de medios).

Imagen tomada de: https://www.movilzona.es/2020/04/30/modo-depuracion-usb-android/

También necesitarás tener habilitadas las opciones de desarrollador en tu dispositivo. Si esto aún no está habilitado, sigue estos pasos (esto funcionará en la mayoría de los dispositivos Android):

Paso 1

Abre el menú **Settings** (Configuraciones) en tu teléfono Android y desplázate hasta la parte inferior.

Paso 2

Pulsa en **About phone** (Acerca del teléfono) y desplázate hacia abajo de nuevo hasta que veas la opción **Build number** (Número de compilación).

Paso 3

Pulsa en el número de compilación varias veces. Pronto deberías ver una ventana emergente que dice algo similar a **You are five taps away from being a developer** (Estás a cinco pulsaciones de ser un desarrollador).

Paso 4

Sigue pulsando hasta que la ventana emergente diga que eres un desarrollador.

Paso 5

Regresa al menú principal **Settings › System › Advanced** (Configuraciones › Sistema › Avanzado). Las opciones de desarrollador deberían ser la penúltima opción. Activa las opciones de desarrollador.

Paso 6

En Android Studio navega hasta el menú superior y selecciona **Run 'app'** (Ejecutar 'aplicación'). Android Studio mostrará un cuadro de diálogo en el que puedes elegir en qué dispositivo ejecutar tu aplicación Android. Elige el dispositivo que conectaste y haz clic en el botón **OK**

Pruebas de mi aplicación en un dispositivo físico

La aplicación "Hola mundo" ahora debería estar ejecutándose en tu teléfono. Desde aquí puedes modificar tu aplicación según lo que desees y agregar más funciones.

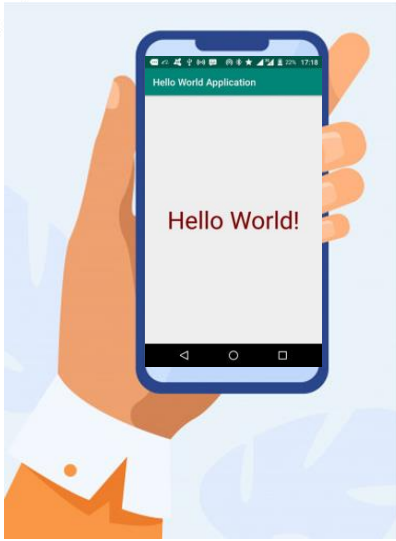


Imagen tomada de:https://www.freepik.es/

El futuro digital es de todos

MinTIC

Tema 8

Pruebas de mi aplicación en un dispositivo virtual

Universidad Tecnológica de Pereira

MisiónTIC2022

Android Emulator

Android Emulator

Imagen tomada de:https://www.freepik.es/

- Simula dispositivos Android en una computadora para que puedas probar tu app en diferentes dispositivos y niveles de API de Android sin necesidad de contar con los dispositivos físicos.
- Proporciona casi todas las funciones de un dispositivo Android real.
- Puedes simular llamadas y mensajes de texto entrantes, especificar la ubicación del dispositivo, utilizar diferentes velocidades de red, probar sensores de rotación y otros sensores de hardware, acceder a Google Play Store y mucho más.

Android Emulator

- En algunos casos, probar tu app en el emulador es más rápido y fácil que hacerlo en un dispositivo físico. Por ejemplo, puedes transferir datos con mayor velocidad al emulador que a un dispositivo conectado mediante USB.
- El emulador incluye configuraciones predefinidas para varios teléfonos y tablets Android, Wear OS y dispositivos Android TV.

Imagen tomada de:https://www.freepik.es/

Requisitos y recomendaciones

Imagen tomada de:https://www.freepik.es/

Además de los requisitos de sistema básicos para Android Studio, Android Emulator tiene los siguientes requisitos adicionales:

- Herramientas del SDK 26.1.1 o versiones posteriores
- Procesador de 64 bits
- Windows: CPU con compatibilidad para UG (invitado no restringido)
- HAXM 6.2.1 o posterior (se recomienda HAXM 7.2.0 o posterior)

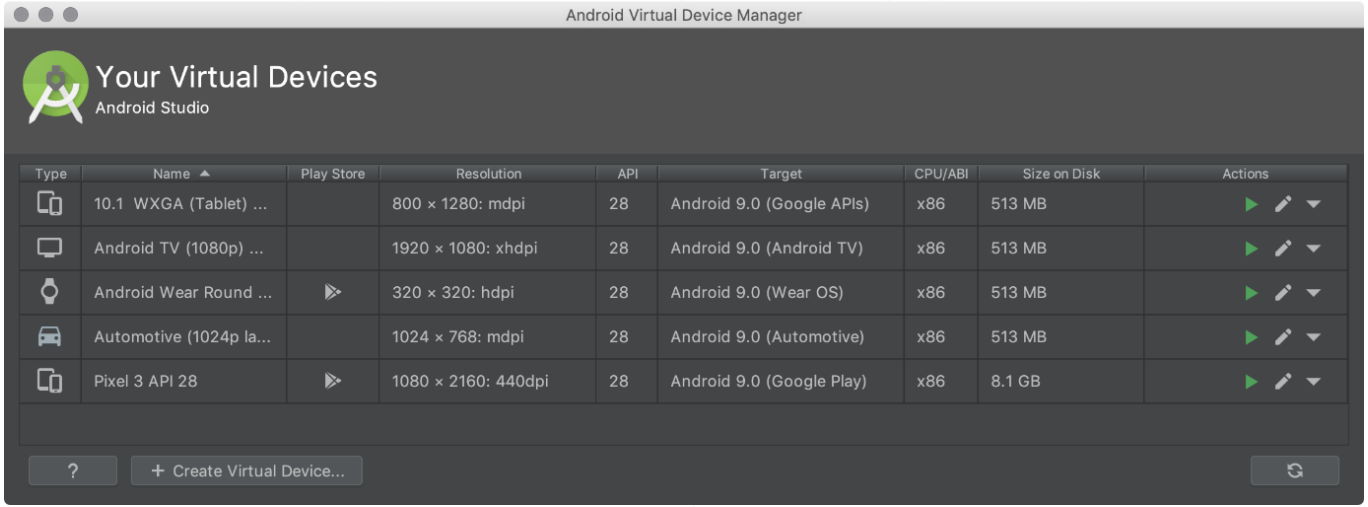
Requisitos y recomendaciones

El uso de la aceleración de hardware implica requisitos adicionales en Windows y Linux:

- **Procesador Intel en Windows o Linux**: procesador Intel compatible con Intel VT-x, Intel EM64T (Intel 64 Bit) y funcionalidad Execute Disable (XD)
- **Procesador AMD en Linux**: procesador AMD compatible con AMD Virtualization (AMD-V) y extensiones complementarias de transmisión SIMD 3 (SSSE3)
- **Procesador AMD en Windows**: Android Studio 3.2 o posterior, y Windows 10 lanzado después de abril de 2018 para Windows Hypervisor Platform (WHPX)

Imagen tomada de:https://www.freepik.es/

Dispositivos virtuales de Android (AVD)



Dispositivos virtuales de Android (AVD)

Imagen tomada de: https://blogsaverroes.juntadeandalucia.es

Cada instancia de Android Emulator usa un dispositivo virtual de Android (AVD) **para especificar la versión de Android y las características del hardware del dispositivo simulado.**

Si quieres probar tu app de manera eficaz, deberás crear un AVD que modele cada dispositivo en el que se ejecutará la app según su diseño. Para crear y administrar AVD, usa el Administrador de AVD.

Dispositivos virtuales de Android (AVD)

Imagen tomada de: https://blogsaverroes.juntadeandalucia.es

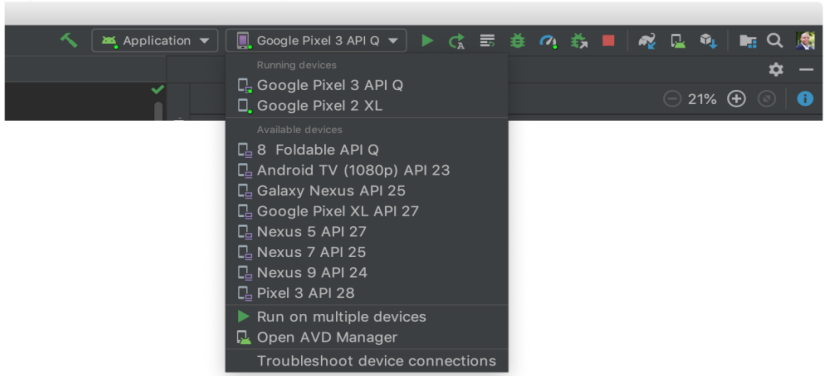
- Cada AVD funciona como un dispositivo independiente, con su propio almacenamiento privado para datos del usuario, tarjetas SD, etc.
- De forma predeterminada, el emulador almacena los datos del usuario, el contenido de la tarjeta SD y la memoria caché en un directorio específico para ese AVD.
- Cuando inicies el emulador, este cargará los datos del usuario y el contenido de la tarjeta SD desde el directorio del AVD.

Cómo ejecutar una app en Android Emulator

Puedes ejecutar una app desde un proyecto de Android Studio o ejecutar una que se haya instalado en Android Emulator de la misma manera en que lo harías con cualquier app en un dispositivo.

Para iniciar el emulador y ejecutar una app desde tu proyecto, haz lo siguiente:

1. En Android Studio, crea un dispositivo virtual de Android (AVD) que el emulador pueda usar para instalar y ejecutar tu app.
2. En la barra de herramientas, selecciona el AVD en el que deseas ejecutar la app desde el menú desplegable del dispositivo de destino.
3. Haz click en Run

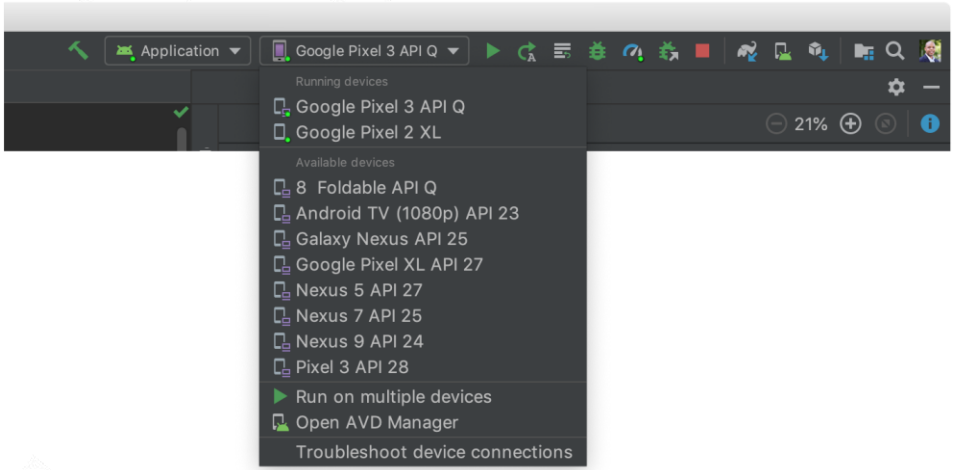


Cómo ejecutar una app en Android Emulator

Si recibes un mensaje de error o de advertencia en la parte superior del diálogo, haz clic en el vínculo para obtener más información o corregir el problema.

Debes corregir algunos errores para poder continuar, como ciertos problemas del Hardware Accelerated Execution Manager (HAXM de Intel).

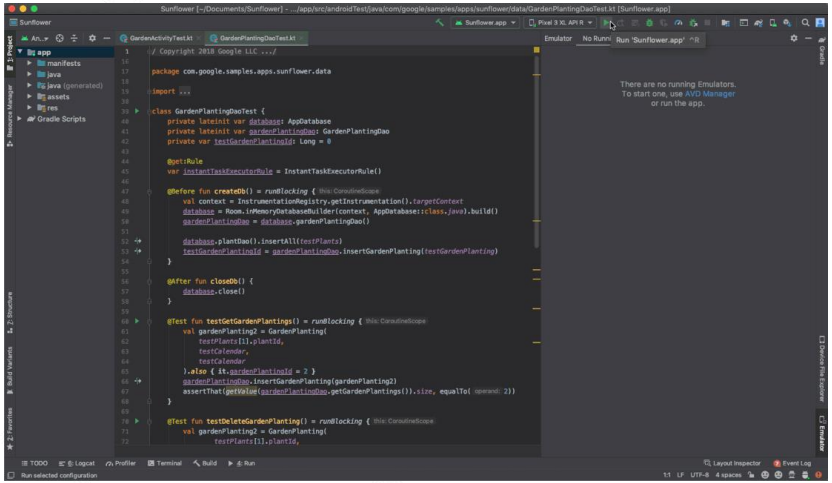
En Mac, si aparece el error `Warning: No DNS servers` found cuando inicias el emulador, verifica si tienes el archivo `/etc/resolv.conf`. Si no lo tienes, ingresa el siguiente comando en una ventana de terminal: `ln -s /private/var/run/resolv.conf /etc/resolv.conf`



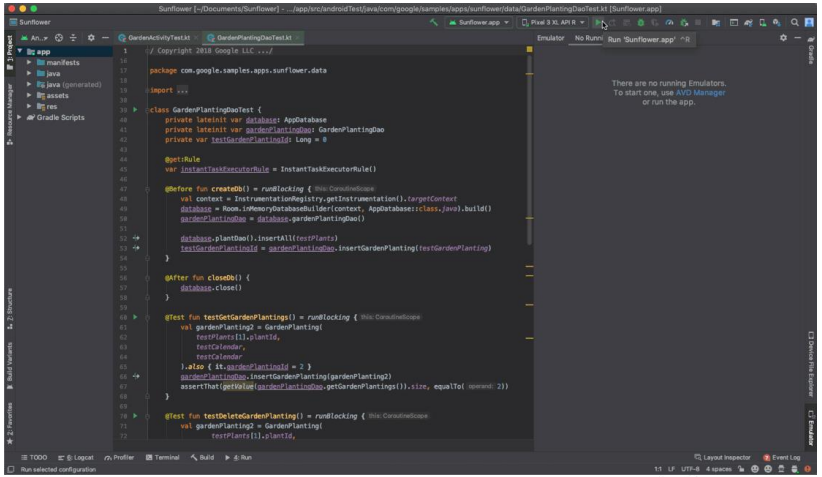
Cómo ejecutar Android Emulator directamente en Android Studio

Ejecuta Android Emulator directamente en Android Studio para:

- Conservar el estado real de la pantalla.
- Navegar rápidamente entre el emulador y la ventana del editor con teclas de acceso rápido.
- Y organizar el flujo de trabajo del IDE y el emulador en una ventana de aplicación única.



Cómo ejecutar Android Emulator directamente en Android Studio



Para ejecutarlo en Android Studio, asegúrate de usar Android Studio 4.1 o superior con la versión 30.0.10 o una versión posterior del emulador, y seguir estos pasos:

1. Haz clic en **File › Settings › Tools › Emulator** (o **Android Studio › Preferences › Tools › Emulator** en macOS), selecciona **Launch in a tool window** y haz clic en **OK**.
2. Si la ventana del emulador no aparece automáticamente, haz clic en **View › Tool Windows › Emulator**.
3. Para iniciar el dispositivo virtual, utiliza el **Administrador de AVD** o establécelo como destino cuando ejecutes tu app.