	Versión: 01	
	Código: GFPI-F-132	

GESTIÓN DE LA FORMACIÓN PROFESIONAL INTEGRAL

TALLER DE APRENDIZAJE

Denominación del programa o módulo de formación:	ANALISIS, DISEÑO Y DESARROLLO DE SOFTWARE	
Actividad técnica de aprendizaje - ATA:	Identificar los requerimientos del informe técnicos.	
Nombre del taller Apropiación del IDE Android Studio	Descripción:	N° 6
Objetivo de aprendizaje del taller:	Conocer el IDE y su funcionamiento	
Descripción del taller:		
Preparación (Recursos y medios):	Computador. Acceso a internet. Biblioteca SENA. Google scholar.	
Tiempo de ejecución	3 horas	
Bibliografía y webgrafía:		

DESARROLLO DEL TALLER

Actividad Individual:

- Ingresar al IDE de Android Studio e identificar en la interfaz del usuario de la ventana principal del IDE las siguientes partes (mediante su propia captura de pantalla ubicar cada una de las partes de la interfaz de usuario):

1. La **barra de herramientas** te permite realizar una gran variedad de acciones, como ejecutar tu app e iniciar las herramientas de Android.
2. La **barra de navegación** te ayuda a explorar tu proyecto y abrir archivos para editar. Proporciona una vista más compacta de la estructura visible en la ventana **Project**.
3. La **ventana del editor** es el área en la que puedes crear y modificar código. Según el tipo de actividad actual, el editor puede cambiar. Por ejemplo, cuando ves un archivo de diseño, el editor muestra el Editor de diseño.
4. La **barra de la ventana de herramientas** se encuentra afuera de la ventana del IDE y contiene los botones que te permiten expandir o contraer ventanas de herramientas individuales.

5. Las **ventanas de herramientas** te brindan acceso a tareas específicas, como la administración de proyectos, la búsqueda, el control de versiones, entre otras. Puedes expandirlas y contraerlas.
6. En la **barra de estado**, se muestra el estado de tu proyecto y el IDE, además de advertencias o mensajes.

- Investigar las combinaciones de teclas para algunas ventanas de herramientas útiles
- Investigar las combinaciones de teclas para completar código

Links:

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

Control de documento:	Nombre completo de quien elabora el cambio y centro de formación al que pertenece	Fecha
	Gustavo Adolfo Salazar Escobar	22/11/2021

		Versión: 01 Código: GFPI-F-132
GESTIÓN DE LA FORMACIÓN PROFESIONAL INTEGRAL		
TALLER DE APRENDIZAJE		
Denominación del programa o módulo de formación:	ANALISIS, DISEÑO Y DESARROLLO DE SOFTWARE	
Actividad técnica de aprendizaje - ATA:	Identificar los requerimientos del informe técnicos.	
Nombre del taller Apropiación del IDE Android Studio	Descripción:	N° 7
Objetivo de aprendizaje del taller:	Conocer el IDE y su funcionamiento	
Descripción del taller:		
Preparación (Recursos y medios):	Computador. Acceso a internet. Biblioteca SENA. Google scholar.	
Tiempo de ejecución	5 horas	
Bibliografía y webgrafía:		
DESARROLLO DEL TALLER		

Actividad Individual:

- Leer el documento “Aplicación Android-Hola Mundo” proporcionado por el instructor.
- Ingresar al IDE de Android Studio e ir realizando el paso a paso que se plantea allí para poder crear el proyecto y editar la vista de la aplicación.
- Emular la aplicación en su dispositivo móvil personal y tomar pantallazo del resultado de la ejecución.

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

Control de documento:	Nombre completo de quien elabora el cambio y centro de formación al que pertenece	Fecha

	Gustavo Adolfo Salazar Escobar	1/12/2021

		Versión: 01 Código: GFPI-F-132
GESTIÓN DE LA FORMACIÓN PROFESIONAL INTEGRAL		
TALLER DE APRENDIZAJE		
Denominación del programa o módulo de formación:	ANALISIS, DISEÑO Y DESARROLLO DE SOFTWARE	
Actividad técnica de aprendizaje - ATA:	Identificar los requerimientos del informe técnicos.	
Nombre del taller Apropiación del IDE Android Studio	Descripción:	N° 8
Objetivo de aprendizaje del taller:	Conocer el IDE y su funcionamiento	
Descripción del taller:		
Preparación (Recursos y medios):	Computador. Acceso a internet. Biblioteca SENA. Google scholar.	
Tiempo de ejecución	8 horas	
Bibliografía y webgrafía:		
DESARROLLO DEL TALLER		

Actividad Individual:

Atender las explicaciones del instructor sobre los conceptos de POO en JAVA, posteriormente realizar los ejercicios planteados en el taller "ejercicios-basicos-poo-java" en cualquier compilador online o en un IDE de JAVA.

Links:

EJERCICIOS DE CLASES Y OBJETOS – Java a tu alcance (wordpress.com)

Control de documento:	Nombre completo de quien elabora el cambio y centro de formación al que pertenece	Fecha
	Gustavo Adolfo Salazar Escobar	5/12/2021

Aplicación Android "Hola mundo"

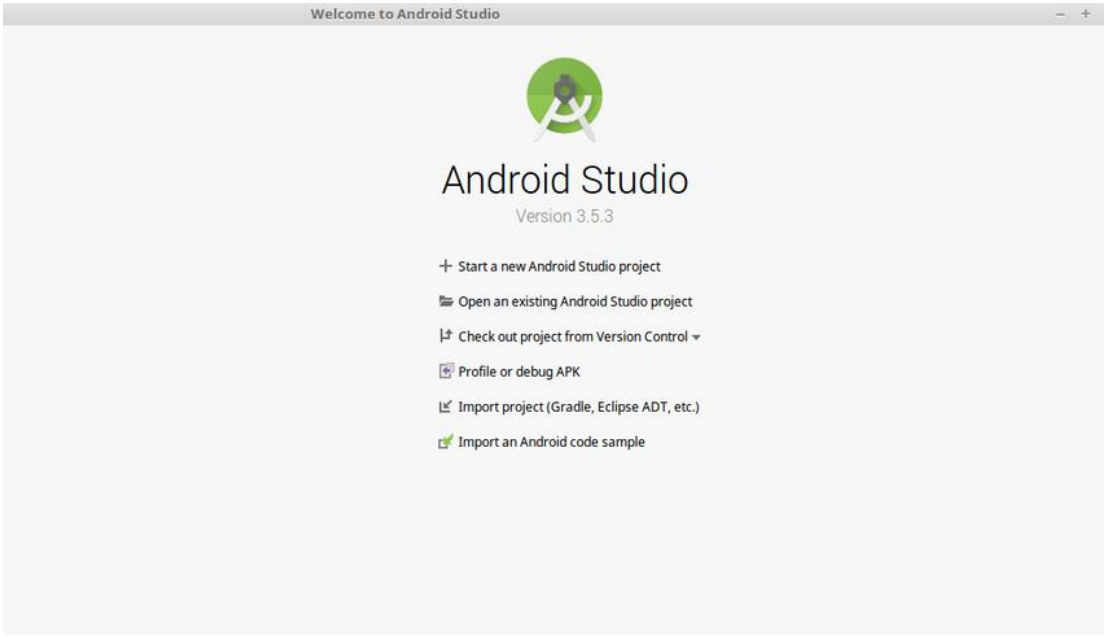
Primero echemos un vistazo a lo que estaremos creando. La aplicación Android "Hola mundo" terminada se verá así:



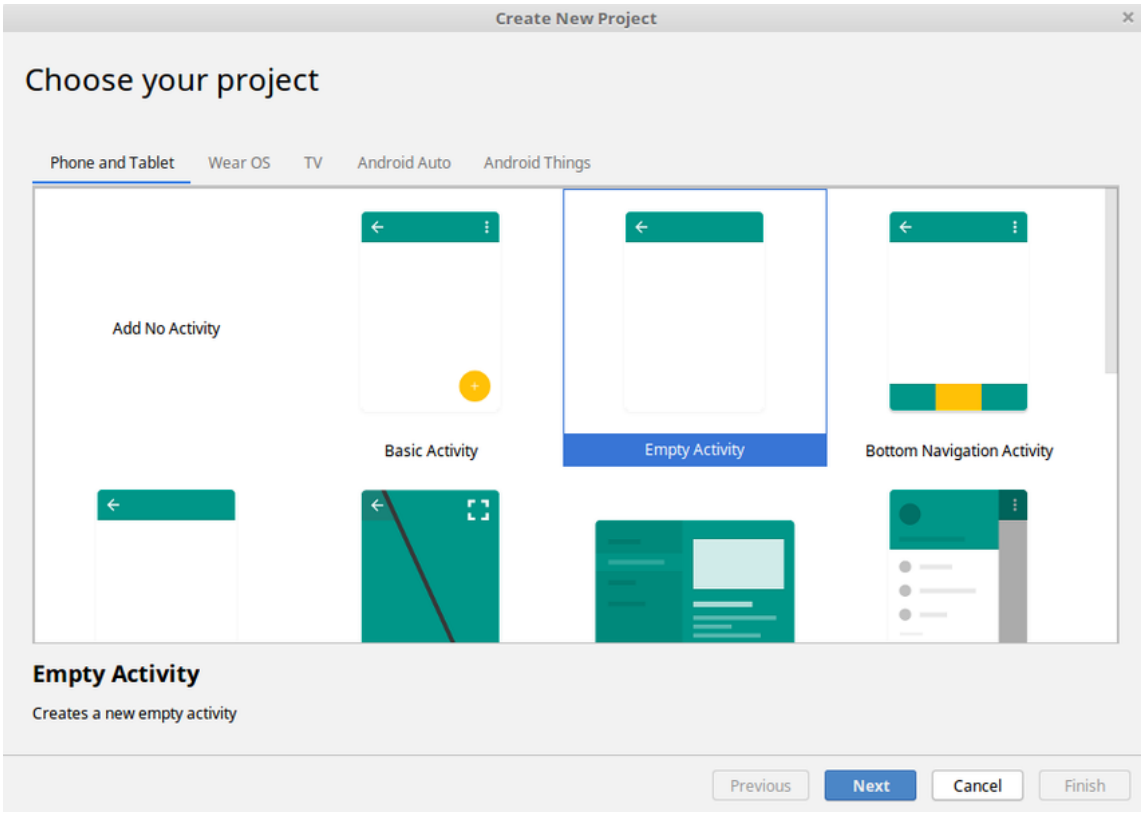
Para crear esta simple aplicación Android, solamente sigue los pasos de este tutorial.

Crea el proyecto de la aplicación

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



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, como se muestra a continuación.



Crea una actividad "Hola mundo"

Android Studio proporciona plantillas de actividades para ayudarte a comenzar. Para el proyecto "Hola mundo" elige **Empty Activity** (Actividad vacía) y haz clic en **Next** (Siguiente).

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. Cada actividad tiene una ventana en la que puede desplegar su interfaz de usuario. La ventana normalmente llena la pantalla, pero puede ser más pequeña que la pantalla y flotar por encima de otras ventanas.

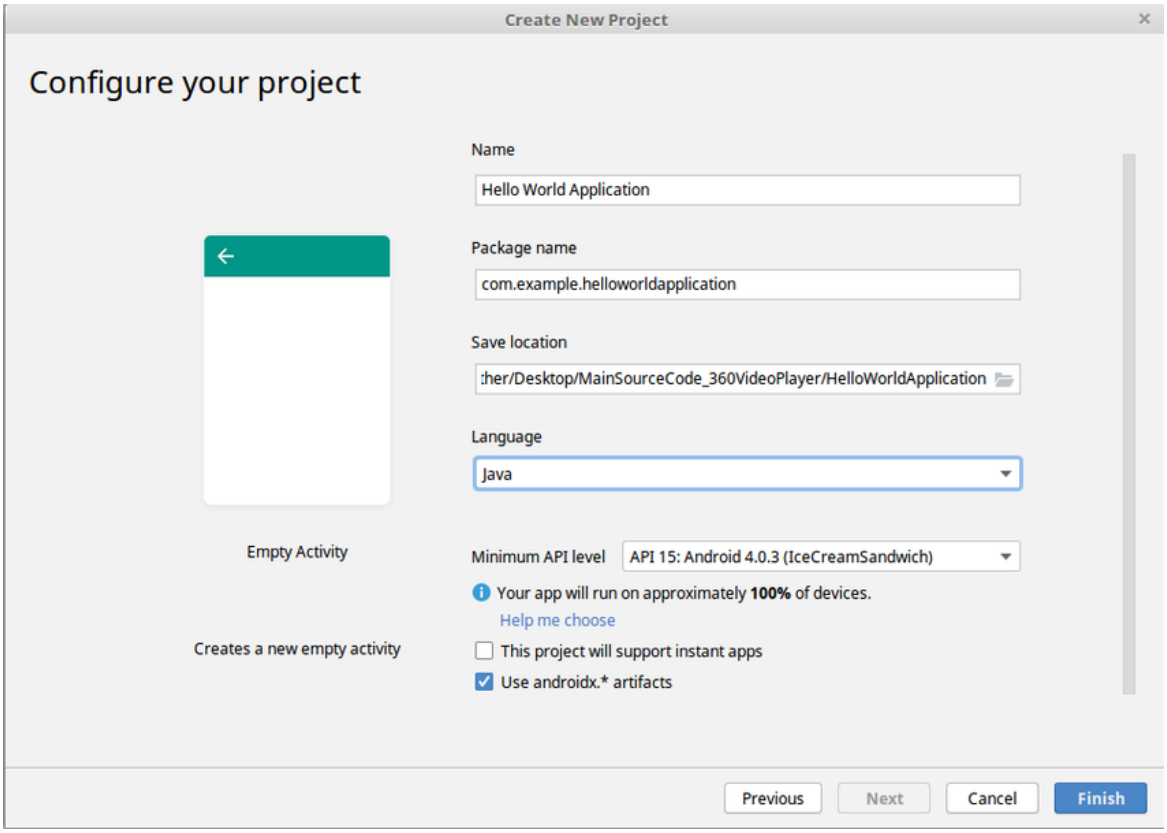
Puedes aprender más sobre las actividades y muchos otros temas relacionados con el desarrollo en Android en la serie "Android desde cero".



-

Configura los detalles del proyecto "Hola mundo"

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



- Cambia el nombre de la aplicación.
- Cambia 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.

Haz clic en **Finish** (Finalizar).

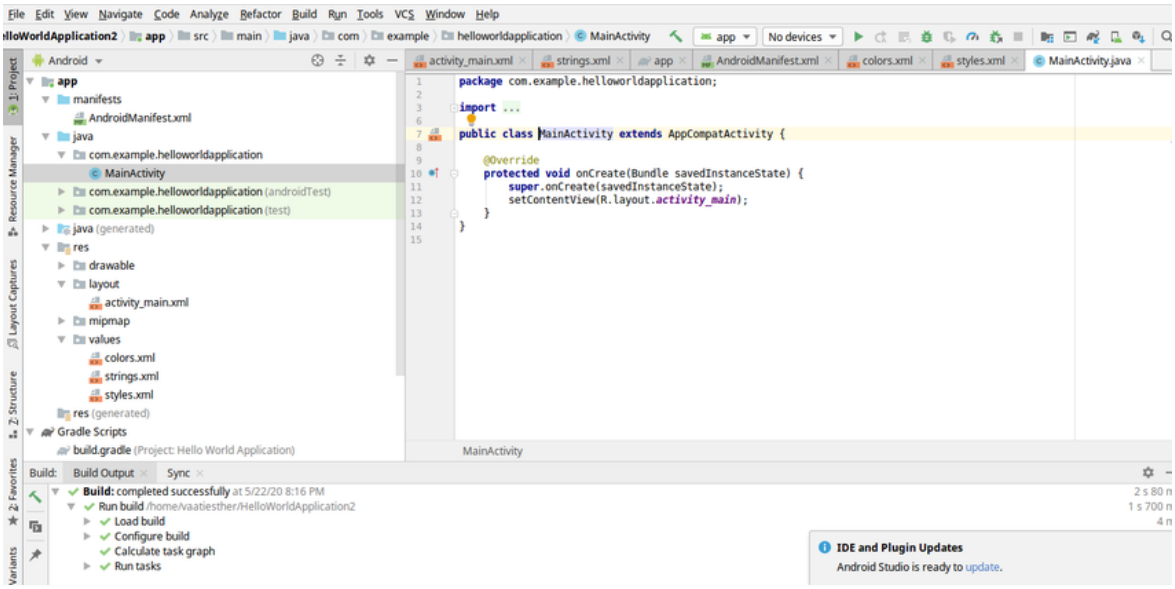
El sistema de compilación Gradle

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.

Advertisement

Explicando los archivos de un proyecto de aplicación 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.



La carpeta manifests

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.

La carpeta res

Esta carpeta incluye todos los recursos que no son de código, por ejemplo:

- Diseños (**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.
- Valores (**values**): contiene el color, estilo y los archivos XML de cadenas para la aplicación.
- Dibujables (**drawable**): este es un repositorio general para todos los gráficos que pueden ser dibujados en la pantalla, por ejemplo, las imágenes.
- **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 "Hola mundo"

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

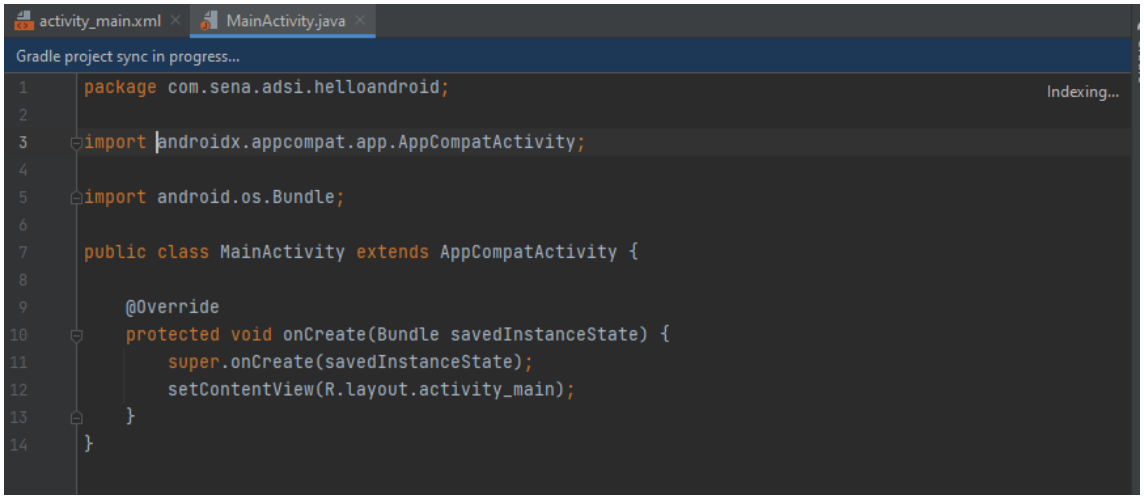
La actividad principal predeterminada

La actividad principal es la primera pantalla que aparecerá cuando ejecutes tu aplicación.

Cada actividad representa una pantalla de la interfaz de usuario de la aplicación Android. Cada actividad tiene un archivo de implementación Java (o Kotlin) y un archivo de diseño XML.

El código Java de la actividad principal predeterminada

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

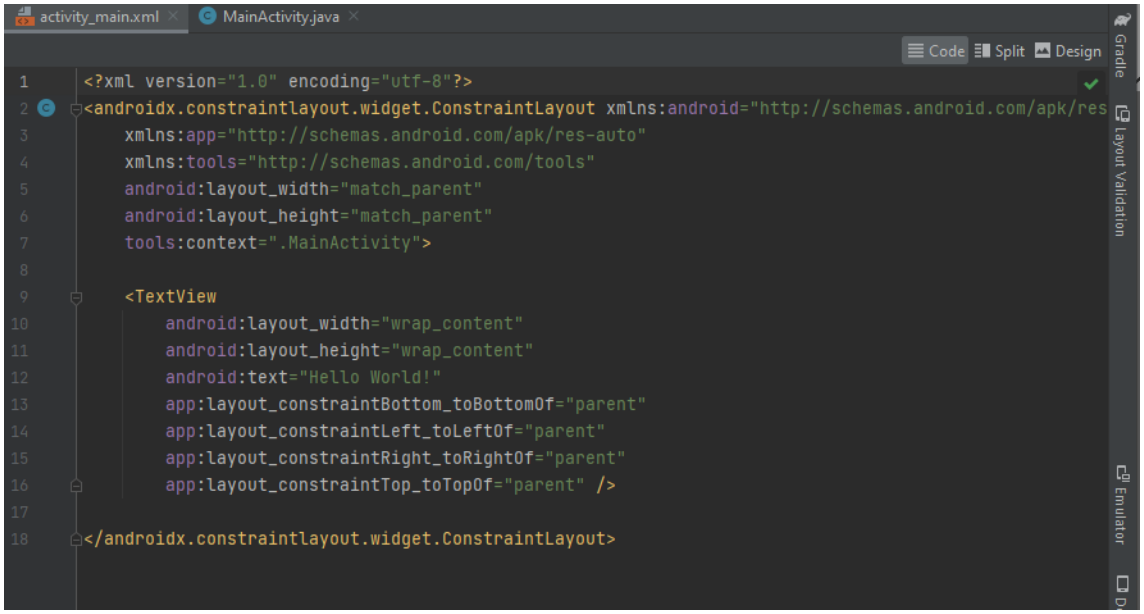


El código define una clase `MainActivity`, y define lo que ocurre cuando se crea la actividad con el método `onCreate`. En este caso, simplemente inicializa la vista de la actividad con el diseño del archivo `activity_main`.

El archivo de diseño XML predeterminado

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 text view, con el texto **Hello World!** (¡Hola mundo!).



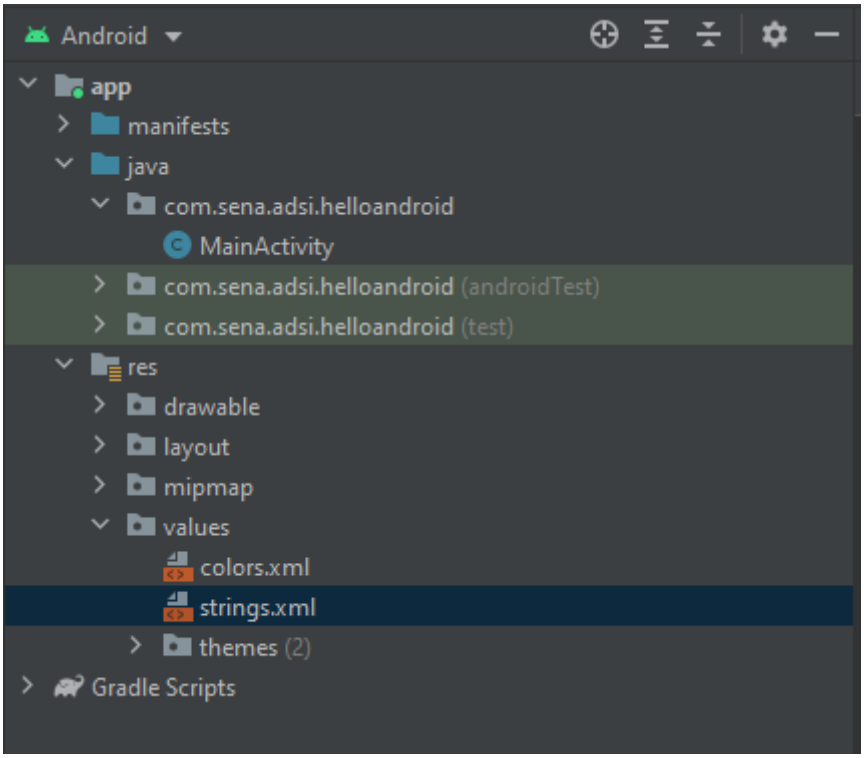
Cambiando el diseño de la actividad

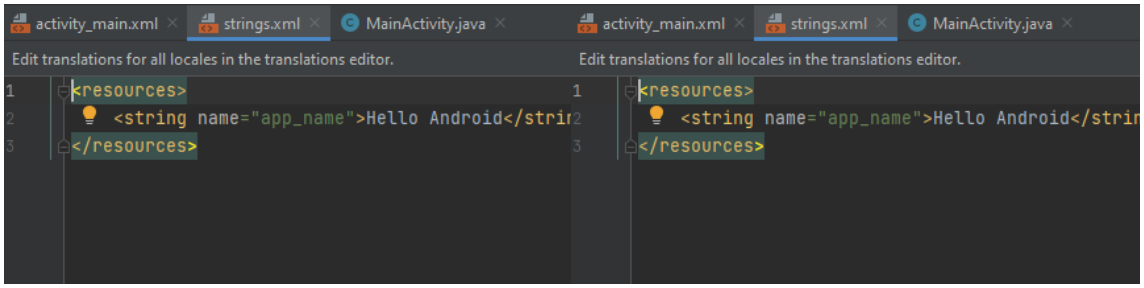
Como puedes ver, no necesitamos cambiar mucho para terminar nuestra aplicación "Hola mundo", pero haremos un pequeño cambio para que el texto destaque más: cambiaremos el color y el tamaño de la fuente del texto.



El archivo strings (cadenas)

El archivo **strings.xml** proporciona cadenas de texto para tu aplicación. Por ejemplo, un archivo de cadenas predeterminado se ve así:





Si quieres cambiar el nombre de tu aplicación, puedes hacerlo aquí.

Ejecutando la aplicación

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).

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):

- Abre el menú **Settings** (Configuraciones) en tu teléfono Android y desplázate hasta la parte inferior.
- Pulsa en **About phone** (Acerca del teléfono) y desplázate hacia abajo de nuevo hasta que veas la opción **Build number** (Número de compilación).
- 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).

- Sigue pulsando hasta que la ventana emergente diga que eres un desarrollador.
- Regresa al menú principal **Settings > System > Advanced** (Configuraciones > Sistema > Avanzado). Las opciones de desarrollador deberían ser la penúltima opción. Activa las opciones de desarrollador.

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**.

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.



EJERCICIOS BÁSICOS RESUELTOS POO JAVA

https://javaparajavatos.wordpress.com/category/ejercicios-de-clases-y-objetos/

Ejercicio 1: Constructores

Intenta escribir una clase que represente un reloj que señale la hora el minuto, y el segundo. La clase dispondrá de dos constructores, uno sin parámetros, que pone el reloj a 0:0:0, y otro al que se le pasa la hora, los minutos y los segundos.

Además, habrá que realizar los siguientes métodos:

Uno que da la hora, los minutos y los segundos, separados por el carácter ":" , en una cadena.

Otro que también da la hora, pero en formato 24 horas (como el anterior) o en formato de 12 horas, en cuyo caso debe distinguir entre "am" (para horas de 0 a 11) o "pm", para horas de 12 a 23, también en una cadena.

Como puede hacerse

Los atributos serán tres enteros: para la hora, los minutos y los segundos. Habría que comprobar que los valores son correctos, tanto en el constructor como en los métodos que ponen la hora. Si son incorrectos, se debería lanzar una excepción descriptiva del problema. Como hay que hacer la comprobación en varios sitios, es conveniente escribir un método que lo haga. A este método se le puede llamar compruebaHora()

Métodos

Existirán dos métodos que dan la hora, que se llamarán dameHora(); el primero, sin parámetros, la devolverá como un objeto String. El segundo necesita un parámetro para determinar el formato de la hora. Para ello, se proporciona un enumerado con los dos posibles formatos, el de 24 y el de 12. En el caso de que el formato sea de 12 horas, y la hora sea de tarde o noche (13 a 23), habrá que restar 12 al valor de la hora y añadir la cadena "pm". En el mismo formato, pero con la hora entre 0 y 11 habrá que poner "am".

Parámetros
La hora, los minutos y los segundos serán representados en todos los casos con valores del tipo int. No obstante, se comprobarán los rangos adecuados.

Valor de retorno
El valor devuelto por los métodos dameHora() será un objeto de la clase String con la hora en el formato adecuado.

Ejercicio 2 Métodos

Vamos a calcular la distancia recorrida por un móvil. Se sabe que el móvil sigue movimientos rectilíneos entre una serie de puntos. Se pide escribir un método que reciba una serie de puntos y determine la distancia recorrida por el móvil. Para probar el método, intenta escribir un programa que calcule la distancia entre los puntos (0,0),

(2,4), (4,5), (4,6), (3,4) y (0,0) (es decir, que vuelve al origen)

Como puede hacerse

Este ejercicio es ya de cierto nivel (en mi opinión). Te voy a dar las siguientes pistas: define una clase, llamada Trayecto, que contendrá el método llamado recorrido. La clase Trayecto contendrá un método main() donde se generan los puntos que os pido en el ejercicio, y luego se llama al método recorrido() con dichos puntos.

Cuando un método escribe un número arbitrario de argumentos del mismo tipo, se puede especificar con la notación "…", que significa que se espera un número variable de argumentos. Con un bucle for se pueden recorrer todos los puntos de la secuencia. Para calcular el recorrido, se considera un punto cada vez. Entonces se calcula la distancia de un punto al anterior, añadiendo el resultado a una variable que acumula el recorrido total.

Ejercicio 3

Crea una clase llamada Libro que guarde la información de cada uno de los libros de una biblioteca. La clase debe guardar el título del libro, autor, número de ejemplares del libro y número de ejemplares prestados. La clase contendrá los siguientes métodos:
Constructor por defecto.

Constructor con parámetros.

Métodos Setters/getters

Método *préstamo* que incremente el atributo correspondiente cada vez que se realice un préstamo del libro. No se podrán prestar libros de los que no queden ejemplares disponibles para prestar. Devuelve true si se ha podido realizar la operación y false en caso contrario.

Método *devolución* que decremente el atributo correspondiente cuando se produzca la devolución de un libro. No se podrán devolver libros que no se hayan prestado. Devuelve true si se ha podido realizar la operación y false en caso contrario.

Método *toString* para mostrar los datos de los libros. Este método se heredada de Object y lo debemos modificar (override) para adaptarlo a la clase Libro.

Escribe un programa para probar el funcionamiento de la clase Libro.

NOTA: Ya sé que algunos conceptos aún no se han visto, pero si te interesa programar, es recomendable que investigues algunas cosas.