**Jerarquía de un proyecto Android:**

El **MainActivity.java** es el primer código que se ejecuta con el método **onCreate :**

**Texto

Descripción generada automáticamente**

Dentro de la carpeta **> res (recursos de la aplicación) > layout,** el archivo **activityMain.xml** que tiene la distribución de los componentes que se añadan a la aplicación, hay 2 vistas

* **Design:** Es la vista diseño (arrastrar elemento, y cambiar las propiedades)
* **Text:** Se ve las propiedades de cada componente



En la carpeta **values** el fichero **strings.xml**, sirve para mostrar al usuario todos los mensajes de tipo texto

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

El archivo **AndroidManifest.xml,** nos dice cuál es la aplicación que se va a ejecutar, ytambién aplica permisos

Texto

Descripción generada automáticamente

El método **findViewById(R.id.button)** es un método heredado de AppCompacActivity (**Recupera un componente**), este método, devuelve un **View**, todos los componentes heredan de View

**COMPONENTES BASICOS**

Un **TextView** es un componente que almacena un texto.

Un **botón**, muestra una funcionalidad.

Para dar funcionalidad a un botón, se recupera el botón, y se utiliza **setonClickListener**(**new OnClickListener**), se crea un método onclick (View view) donde se puede poner.

El **Log.d()** es para depurar, los mensajes de error, aparecerán en una pantalla, llamada logCat

Una **Interfaz** es un fichero que define funcionalidades de una clase

**COMPONENTES BASICOS 2**

**Spinner**

Un **Spinner** es un desplegable que puede trabajar con datos de manera estática, lo que significa que estos datos no cambian durante la ejecución de la aplicación, a menos que los manipulemos en tiempo de ejecución.

Puedes llenarlo de dos formas diferentes:

Crear un array de cadenas en los recursos XML, como se muestra a continuación:

**xml**

**Copy code**

**<string-array name="ciudades">**

**<item>Barcelona</item>**

**<item>Madrid</item>**

**<item>Valencia</item>**

**</string-array>**

Luego, puedes enlazarlo al Spinner buscando la propiedad "entries" en la vista de diseño y seleccionando el array con @array/.

Llenarlo en tiempo de ejecución, generalmente dentro del método onCreate del MainActivity. Por lo general, se crea un método separado, como "inicializarSpinner". Primero, creas un array o ArrayList con los datos que deseas mostrar. Luego, obtienes una referencia al Spinner y creas un adaptador, como ArrayAdapter, que requiere el contexto, un diseño y el array que creaste. El código se vería así:

ArrayAdapter<String> adapter = new ArrayAdapter<>(this, android.R.layout.simple\_spinner\_item, array);

spinner.setAdapter(adapter);

Puedes personalizar la acción que ocurre cuando se selecciona un elemento del Spinner utilizando setOnItemSelectedListener(). Por ejemplo:

spinner.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {

@Override

public void onItemSelected(AdapterView<?> parentView, View selectedItemView, int position, long id) {

// Código que se ejecutará cuando se seleccione un elemento

}

@Override

public void onNothingSelected(AdapterView<?> parentView) {

// Código que se ejecutará cuando no se seleccione ningún elemento

}

});

**RadioGroup y RadioButton**

Un **RadioGroup** es un conjunto de botones de opción (RadioButton) donde solo se puede seleccionar uno a la vez. Los botones de opción se encuentran dentro del RadioGroup.

Puedes utilizar el método setOnCheckedChangeListener para saber cuál de los RadioButton ha sido seleccionado en el grupo. Un ejemplo sería:

RadioGroup radioGroup = findViewById(R.id.radioGroup);

radioGroup.setOnCheckedChangeListener(new RadioGroup.OnCheckedChangeListener() {

@Override

public void onCheckedChanged(RadioGroup group, int checkedId) {

RadioButton radioButton = findViewById(checkedId);

if (radioButton != null) {

String opcionSeleccionada = radioButton.getText().toString();

// Realiza acciones basadas en la opción seleccionada

}

}

});

**Toast**

Por último, mencionar el uso de Toast, que se utiliza para mostrar un mensaje breve en la pantalla. Puedes utilizarlo para proporcionar información o retroalimentación al usuario.

**CREACIÓN DE OBJETOS**

Si se ha copiado el misimo proyecto, le hemos cambiado el nombre, entonces, lo podemos cambiar el nombre del **app\_name** al nuevo nombre en el archivo **strings.xml**

**TODO**, sirve para “recordatorio para hacer el código en un futuro”

No solo hay el onclicklistener sino que hay mas métodos

Como ontouchlistener

El click no devolvía nada, pero el touch devuelve un valor booleano, como muchos métodos, esto esta hecho para si tenemos que propagar a eventos a paneles superiores

Si devolvemos true, si hemos gestionado el evento, por lo que no lo propagamos

En device manager hay dos opciones:

**Cold boot now**: que reinicia la maquina

**Wipse data**: borra todo de la maquina

**VIEW BILDING Y DATA BILDING**

El View Bilding, sirve para acceder más rápido a los objetos (te ahorras el findViewById())

Para facilitar el relleno de datos, tenemos el data binding, al activarlo en el build gradle >buildfeatures, y sincronizar, en el layout que tenemos los datos que se tenfan que rellenar creamos

Estoe eliminara unos atributos y creara una etiqueta llamada data y donde alamacenareos varibles (variable), donde creamos una que sea persona con atributo name y type usuario, después en las etiquetas de los edittextpodemos poner

android:text="@{persona.nombre}”

esto hará que estas ganen el valor que se les ponga

depues en el mainactivity, podemos ver como al hacer binding hay un método set y get persona que sirven para manejar esta variable

a veces surge un erro que para solucionarlo tendremos que ir al build gradle y poner en dependecies

implementation(platform("org.jetbrains.kotlin:kotlin-bom:1.8.0"))

Data bidnidng este objeto se asigne a la interfaz grafica para que se asignen

El viewBilding, evita el **NullPointerException**

**MENU CONTEXTO**

Para introducir un menu en contexto, haremos click derecho en la carpeta **res** > new > Android resources File > se le aplica el tipo menu

Para visualizar la barra de titulo , para cambiarlo es en la carpeta theme > themes.xml