

OBJETOS (VIEWS) DE UNA APLICACIÓN MÓVIL

1. Botón (Button)

Características:

- Se usa para ejecutar una acción al ser presionado.
- Soporta texto e íconos.
- Puede personalizarse con estilos, colores y formas.
- Implementa un **listener** para detectar clics (setOnClickListener).

Ejemplo xml:

```
<Button  
    android:id="@+id	btnEnviar"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Enviar" />
```

Ejemplo java:

```
Button btnEnviar = findViewById(R.id.btnEnviar);  
btnEnviar.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        Toast.makeText(MainActivity.this, "Botón presionado",  
        Toast.LENGTH_SHORT).show();  
    }  
});
```

2. Botón de Radio (RadioButton)

Características:

- Permite seleccionar **una sola opción** dentro de un grupo.
- Normalmente se agrupa con **RadioGroup**.
- Cada botón es excluyente respecto a los demás.

Ejemplo xml:

```
<RadioGroup  
    android:id="@+id/radioGroup"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content">  
  
<RadioButton  
    android:id="@+id/rbOpcion1"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Opción 1" />  
  
<RadioButton  
    android:id="@+id/rbOpcion2"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Opción 2" />  
</RadioGroup>
```

Ejemplo java:

```
RadioGroup radioGroup = findViewById(R.id.radioGroup);  
radioGroup.setOnCheckedChangeListener(new  
RadioGroup.OnCheckedChangeListener() {  
    @Override  
    public void onCheckedChanged(RadioGroup group, int checkedId) {  
        if (checkedId == R.id.rbOpcion1) {  
            Toast.makeText(MainActivity.this, "Seleccionaste Opción 1",  
Toast.LENGTH_SHORT).show();  
        } else if (checkedId == R.id.rbOpcion2) {  
            Toast.makeText(MainActivity.this, "Seleccionaste Opción 2",  
Toast.LENGTH_SHORT).show();  
        }  
    }  
});
```

3. CheckBox

Características:

- Permite seleccionar múltiples opciones a la vez.
- Cada CheckBox es independiente.
- Se puede verificar si está marcado con isChecked().

Ejemplo xml:

```
<CheckBox  
    android:id="@+id/chkAceptar"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Acepto términos y condiciones" />
```

Ejemplo java:

```
CheckBox chkAceptar = findViewById(R.id.chkAceptar);  
chkAceptar.setOnCheckedChangeListener(new  
CompoundButton.OnCheckedChangeListener() {  
    @Override  
    public void onCheckedChanged(CompoundButton buttonView, boolean  
isChecked) {  
        if (isChecked) {  
            Toast.makeText(MainActivity.this, "Aceptado",  
Toast.LENGTH_SHORT).show();  
        } else {  
            Toast.makeText(MainActivity.this, "No aceptado",  
Toast.LENGTH_SHORT).show();  
        }  
    }  
});
```

4. Cuadro de Texto de Edición (EditText)

Características:

- Permite al usuario **ingresar texto**.
- Puede configurarse para distintos tipos de entrada (texto, número, contraseña, email, etc.).
- Se obtiene el valor con `getText().toString()`.

Ejemplo xml:

```
<EditText  
    android:id="@+id/edtNombre"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:hint="Escribe tu nombre"  
    android:inputType="textPersonName" />
```

Ejemplo java:

```
EditText edtNombre = findViewById(R.id.edtNombre);  
Button btnMostrar = findViewById(R.id.btnEnviar);  
  
btnMostrar.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        String nombre = edtNombre.getText().toString();  
        Toast.makeText(MainActivity.this, "Hola " + nombre,  
        Toast.LENGTH_SHORT).show();  
    }  
});
```

5. TextView

Características:

- Se utiliza para mostrar texto en pantalla.
- No es editable (a diferencia de EditText).
- Permite personalizar tamaño, color, estilo, fuente, etc.

Ejemplo xml:

```
<TextView  
    android:id="@+id/txtMensaje"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Bienvenido a la app"  
    android:textSize="18sp"  
    android:textColor="@android:color/holo_blue_dark" />
```

Ejemplo java:

```
TextView txtMensaje = findViewById(R.id.txtMensaje);
txtMensaje.setText("Texto actualizado desde Java");
```

6. Caja Combo (Spinner)

Características:

- Despliega una lista desplegable de opciones.
- El usuario selecciona una opción de la lista.
- Funciona con un **ArrayAdapter**.

Ejemplo xml:

```
<Spinner
    android:id="@+id/spinnerOpciones"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
```

Ejemplo java:

```
Spinner spinner = findViewById(R.id.spinnerOpciones);

String[] opciones = {"Opción A", "Opción B", "Opción C"};
ArrayAdapter<String> adapter = new ArrayAdapter<>(this,
    android.R.layout.simple_spinner_item, opciones);
adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
spinner.setAdapter(adapter);

spinner.setOnItemSelectedListener(new
    AdapterView.OnItemSelectedListener() {
        @Override
        public void onItemSelected(AdapterView<?> parent, View view, int
        position, long id) {
            String seleccion = parent.getItemAtPosition(position).toString();
            Toast.makeText(MainActivity.this, "Seleccionaste: " + seleccion,
            Toast.LENGTH_SHORT).show();
        }

        @Override
        public void onNothingSelected(AdapterView<?> parent) { }
});
```