



DESCRIPCIÓN DEL PROYECTO: CALCULADORA DE EDAD

Andrés Rita Rita

10° "A" ING. EN DESARROLLO Y GESTION DE SOFTWARE



El Diseño del activity

El diseño del activity de la aplicación **Calculadora de Edad** se ha creado para permitir al usuario ingresar dos fechas y calcular la diferencia en años entre ellas. La interfaz es sencilla y fácil de usar, con campos de entrada claramente etiquetados y un botón que inicia el cálculo. El diseño utiliza un `ConstraintLayout`, que permite organizar los elementos de manera eficiente y responsiva. Los componentes clave del diseño son:

- **Campo de entrada para la primera fecha:** Permite al usuario ingresar su primera fecha en el formato **dd/mm/aaaa**.
- **Campo de entrada para la segunda fecha:** Permite al usuario ingresar la segunda fecha en el mismo formato.
- **Botón "Calcular Años":** Al presionar este botón, se ejecuta la función que calculará la diferencia entre las dos fechas.
- **TextView para mostrar el resultado:** Muestra el resultado del cálculo, es decir, la diferencia en años entre las dos fechas.

El Código XML del activity diseñado

Descripción del Código XML

Declaración del Layout:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://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">
```

Se utiliza un `ConstraintLayout` como contenedor principal. Este tipo de layout permite una disposición flexible y responsiva de los elementos en la pantalla, utilizando restricciones para definir sus posiciones relativas.

1. Campo de Entrada para la Primera Fecha:

```
<EditText
    android:id="@+id/fechaInicio"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:hint="Introduce la primera fecha (dd/mm/aaaa)"
    android:inputType="date"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    android:layout_marginTop="16dp"
    android:layout_marginStart="16dp"
    android:layout_marginEnd="16dp" />
```

Funcionalidad: Este `EditText` permite al usuario ingresar la primera fecha.

Atributos:

- `id`: Identificador único para referirse al campo en el código Java.
- `hint`: Texto que aparece como sugerencia cuando el campo está vacío.
- `inputType`: Especifica que el campo está diseñado para aceptar fechas.
- `layout_width` y `layout_height`: Controlan el tamaño del campo. El ancho está configurado para ocupar el espacio disponible entre los márgenes.

- layout_margin: Espaciado alrededor del campo para una mejor estética.

2. Campo de Entrada para la Segunda Fecha:

```
<EditText
    android:id="@+id/fechaFin"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:hint="Introduce la segunda fecha (dd/mm/aaaa)"
    android:inputType="date"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/fechaInicio"
    android:layout_marginTop="16dp"
    android:layout_marginStart="16dp"
    android:layout_marginEnd="16dp" />
```

- **Funcionalidad:** Similar al primer campo, este EditText permite al usuario ingresar una segunda fecha.
- **Atributos:** Mantiene una estructura y funcionalidad similares al campo de entrada anterior, pero está diseñado para estar debajo del primer campo (layout_constraintTop_toBottomOf).

3. Botón para Calcular la Diferencia de Años:

```
<Button
    android:id="@+id/calculateButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Calcular Años"
    android:onClick="calcularAños"
    app:layout_constraintTop_toBottomOf="@id/fechaFin"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    android:layout_marginTop="16dp" />
```

- **Funcionalidad:** Este botón permite al usuario ejecutar el cálculo de la diferencia de años entre las dos fechas ingresadas.
- **Atributos:**

- id: Identificador único para referirse al botón en el código Java.
- text: Texto que se muestra en el botón.
- onClick: Especifica el nombre de la función Java (calcularAnios) que se ejecutará cuando se haga clic en el botón.
- layout_constraintTop_toBottomOf: Ubica el botón debajo del campo de fecha final.

4. TextView para Mostrar el Resultado:

```
<TextView
    android:id="@+id/resultTextView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="La diferencia se mostrará aquí"
    app:layout_constraintTop_toBottomOf="@id/calculateButton"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    android:layout_marginTop="16dp" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

- **Funcionalidad:** Este TextView se utiliza para mostrar el resultado del cálculo de años.
- **Atributos:**
 - id: Identificador único para referirse al TextView en el código Java.
 - text: Texto inicial que se muestra en la vista antes de que se calcule la diferencia.
 - layout_constraintTop_toBottomOf: Ubica el TextView debajo del botón de cálculo.

El Código de la clase java que manipula a los objetos del activity.

Descripción del Código Java

Importaciones Necesarias:

```
package com.example.actividad2;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;
```

- Se importan las clases necesarias para crear la actividad y manejar las entradas del usuario, así como para el manejo de fechas.

Definición de la Clase:

```
public class MainActivity extends AppCompatActivity {
```

- MainActivity extiende de AppCompatActivity, lo que le permite ser utilizada como una actividad en Android.

Declaración de Variables:

```
private EditText fechaInicio, fechaFin; 2 usages
private TextView resultTextView; 3 usages
```

- Se declaran dos variables EditText para las fechas de inicio y fin, y un TextView para mostrar el resultado.

Método onCreate:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    // Vincular los elementos de la interfaz con las variables Java
    fechaInicio = findViewById(R.id.fechaInicio);
    fechaFin = findViewById(R.id.fechaFin);
    resultTextView = findViewById(R.id.resultTextView);
}
```

- Este método se ejecuta cuando la actividad se crea. Configura la vista utilizando el layout definido en activity_main.xml y vincula los elementos de la interfaz a las variables Java.

Método calcularAnios:

```
public void calcularAnios(View view) {
    try {
        // Obtener las fechas de los campos de entrada
        String fechaInicioStr = fechaInicio.getText().toString();
        String fechaFinStr = fechaFin.getText().toString();

        SimpleDateFormat formato = new SimpleDateFormat(pattern: "dd/MM/yyyy");
        Date fechaInicioDate = formato.parse(fechaInicioStr);
        Date fechaFinDate = formato.parse(fechaFinStr);
    }
}
```

- Este método se llama cuando el usuario hace clic en el botón correspondiente.
- Se obtienen las cadenas de texto de las fechas ingresadas, se define un formato para parsear las fechas, y se convierten las cadenas en objetos Date.

Cálculo de la Diferencia de Años:

```
// Obtener los calendarios para las dos fechas
Calendar inicio = Calendar.getInstance();
inicio.setTime(fechaInicioDate);

Calendar fin = Calendar.getInstance();
fin.setTime(fechaFinDate);

// Calcular la diferencia de años
int aniosDiferencia = fin.get(Calendar.YEAR) - inicio.get(Calendar.YEAR);

// Ajustar si la fecha de fin es anterior al cumpleaños en el año
if (fin.get(Calendar.DAY_OF_YEAR) < inicio.get(Calendar.DAY_OF_YEAR)) {
    aniosDiferencia--;
}
```

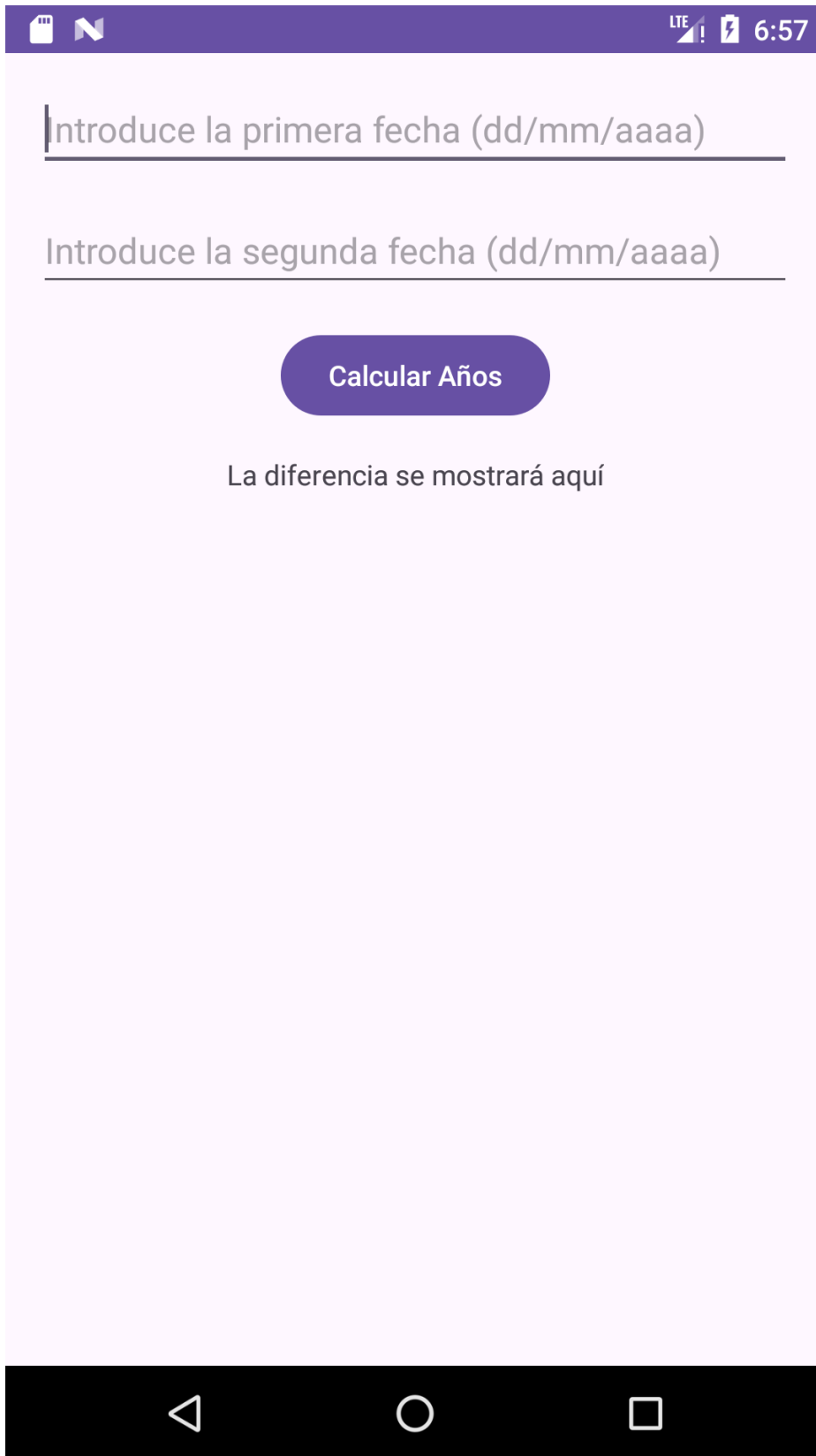
- Se crean instancias de Calendar para las fechas de inicio y fin.
- Se calcula la diferencia de años basándose en el año de cada fecha.
- Se ajusta la diferencia si la fecha de fin aún no ha alcanzado el día y mes de la fecha de inicio en ese año.

Mostrar el Resultado:

```
// Mostrar el resultado
resultTextView.setText("La diferencia es: " + aniosDiferencia + " años.");
} catch (Exception e) {
    // Si hay un error en la entrada, mostrar mensaje de error
    resultTextView.setText("Por favor, introduce fechas válidas (dd/mm/aaaa).");
}
}
```

- Si todo va bien, se muestra la diferencia de años en el TextView.
- Si hay un error (por ejemplo, si las fechas ingresadas son inválidas), se muestra un mensaje de error correspondiente.

La corrida y demostración del funcionamiento de la app.



The screenshot shows a mobile application interface with a purple header bar. The header contains a status bar with icons for signal strength, LTE, battery, and the time 6:57. Below the header, the app has a light purple background. It features two text input fields with placeholder text: "Introduce la primera fecha (dd/mm/aaaa)" and "Introduce la segunda fecha (dd/mm/aaaa)". Below these fields is a purple button with the text "Calcular Años". At the bottom of the main content area, there is a text label "La diferencia se mostrará aquí". The bottom of the screen shows the standard Android navigation bar with back, home, and recent apps icons.

Introduce la primera fecha (dd/mm/aaaa)

Introduce la segunda fecha (dd/mm/aaaa)

Calcular Años

La diferencia se mostrará aquí



LTE 6:56

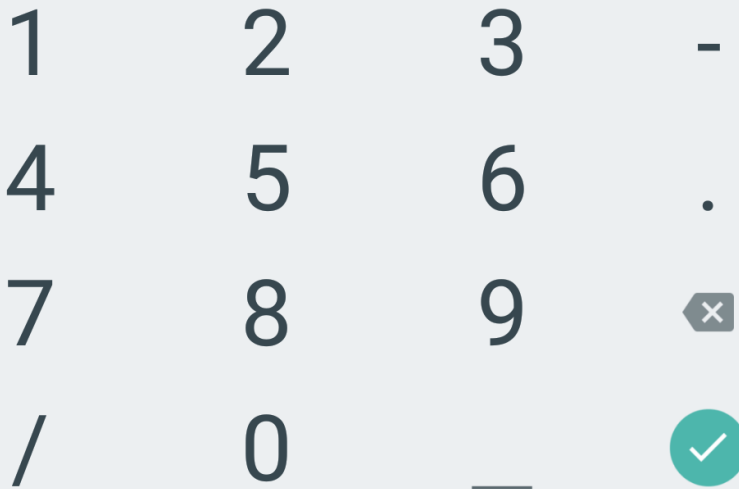
Introduce la primera fecha (dd/mm/aaaa)

Introduce la segunda fecha (dd/mm/aaaa)

Calcular Años

Por favor, introduce fechas válidas (dd/mm/aaaa).

Suggest contact names? Touch for info.





LTE 6:54

01/01/2000

11/10/2024

Calcular Años

La diferencia es: 24 años.

Suggest contact names? Touch for info.

1	2	3	-
4	5	6	.
7	8	9	×
/	0	_	>

