

Calculadora de billetes de Renfe

Por Alberto Colmenar Casas, 2º DAM

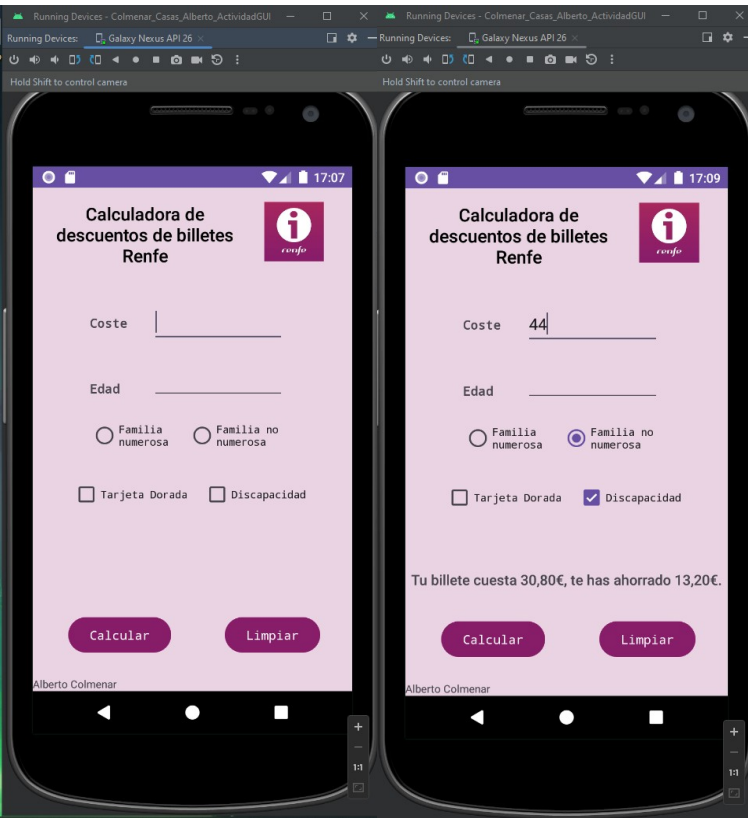


La aplicación tiene como objetivo ayudar a las personas a la hora de saber cuanto te costaría un billete de tren con las diversas variables que nos ofrece Renfe (ficción) en temas de descuentos.

Además de saber el precio final de un billete nos muestra el dinero que te ahorras al disfrutar de uno de sus descuentos.

Interfaz gráfica

Caso normal



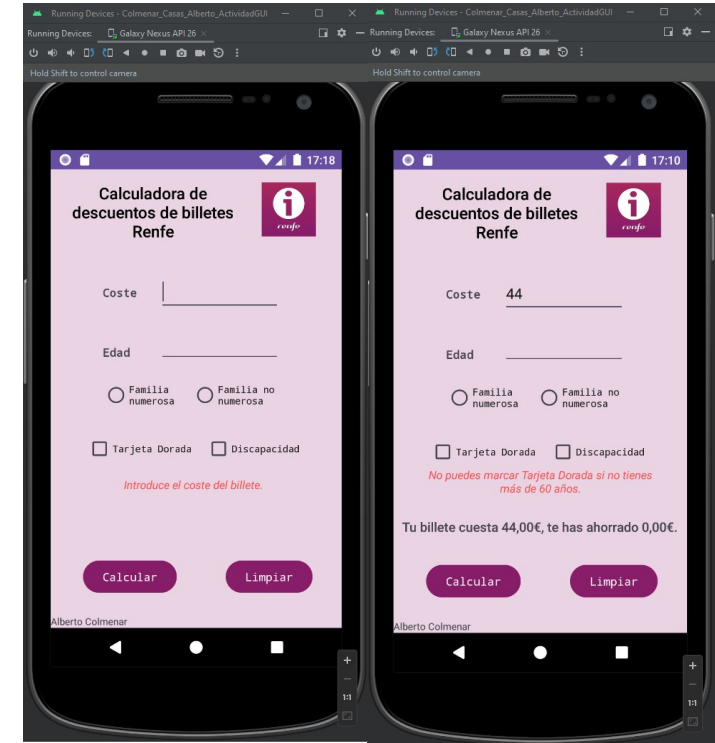
El campo de 'Coste' es obligatorio, debes introducir el coste del billete.

Después tienes el grupo de radios que si marcas 'Familia numerosa' te descuenta un 20%

El checkbox de 'Discapacidad' te descuenta un 30% y el de 'Tarjeta Dorada' 5€ solo si tienes más de 60 años escritos en el campo 'Edad'.

Tanto si se da al botón 'Calcular' como a los distintos checks y radios se muestra el mensaje de tu coste añadidos todos los descuentos.

Otros casos



Si das al botón 'Calcular' sin haber introducido ningún Coste se produce un error (Imagen 1)

Si das al checkbox 'Tarjeta Dorada' y no has introducido Edad o es menor de 60 años no te calcula el descuento y muestra un error (Imagen 2)

Código actividad

Inicializaciones iniciales

Principal

```
tools  Git  Window  Help  Colmenar_Casas_Alberto_ActividadGUI - MainActivity.java [Colmenar_Casas_Alberto_ActividadGUI.app.main]
colmenar  casas  alberto  actividadgui  MainActivity
MainActivity.java  activity_main.xml

41 // Método que calcula el precio del billete con todos sus descuentos
42 5 usages  Alberto clase
43 public void calcular(View v) {
44     textoErrores.setText("");
45     textoCalculo.setText("");
46     if (cajaPrecio.getText().length() != 0) {
47         precioBillete = Double.parseDouble(cajaPrecio.getText().toString());
48         double descuentoProvisional = 0;
49         if (checkTarjeta.isChecked()) {
50             if (cajaEdad.getText().length() != 0) {
51                 edad = Integer.parseInt(cajaEdad.getText().toString());
52                 if (edad >= 60) {
53                     descuentoProvisional = descuentoProvisional + descTarjetaDorada;
54                 } else {
55                     textoErrores.setText("No puedes marcar Tarjeta Dorada si no tienes más de 60 ...");
56                     checkTarjeta.setChecked(false);
57                 }
58             } else {
59                 textoErrores.setText("No puedes marcar Tarjeta Dorada si no tienes más de 60 ...");
60                 checkTarjeta.setChecked(false);
61             }
62         }
63         if (checkDiscapacidad.isChecked()) {
64             descuentoProvisional = descuentoProvisional + (precioBillete * descDiscapacidad);
65         }
66         if (radioGroup.getCheckedRadioButtonId() == R.id.radioNumerosa) {
67             descuentoProvisional = descuentoProvisional + (precioBillete * descFamiliaNum);
68         }
69         precioFinal = precioBillete - descuentoProvisional;
70         Resources res = getResources();
71         String finalt = "Tu billete cuesta {precioFinal}€, te has ahorrado {desc...";
72         textoCalculo.setText(finalt);
73     } else {
74         textoErrores.setText("Introduce el coste del billete.");
75     }
76 }
```

El método calcular tiene la lógica de la app.
Comprueba que se ha introducido el coste del billete sino muestra error.

Y después va mirando los checkbox y radios para, si son marcados, calcular el descuento y lo va guardando en la variable 'descuentoProvisional'

A la hora de comprobar la 'Tarjeta Dorada' lee el campo 'Edad' para ver si le corresponde ese descuento por edad.

Al final se calcula el precio final restando el 'descuentoProvisional' con el coste introducido y se muestra por pantalla.

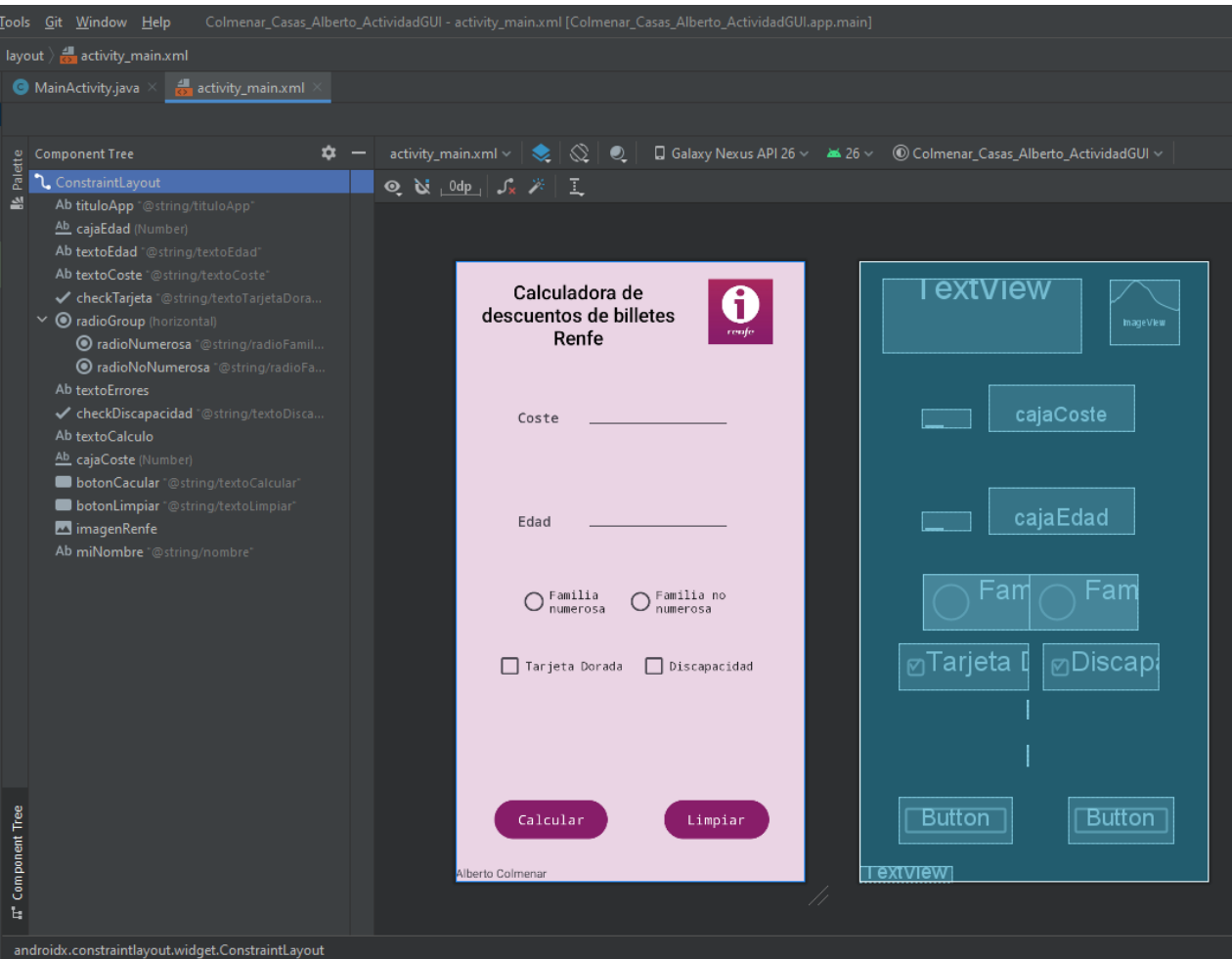
```
tools  Git  Window  Help  Colmenar_Casas_Alberto_ActividadGUI - MainActivity.java [Colmenar_Casas_Alberto_ActividadGUI.app.main]
colmenar  casas  alberto  actividadgui  MainActivity
MainActivity.java  activity_main.xml

42 public class MainActivity extends AppCompatActivity {
43
44     private final double descFamiliaNum = 0.2;
45     private final double descDiscapacidad = 0.3;
46     private final int descTarjetaDorada = 5;
47     private double precioBillete;
48     private int edad;
49     private double precioFinal;
50     EditText cajaPrecio;
51     EditText cajaEdad;
52     RadioGroup radioGroup;
53     CheckBox checkTarjeta;
54     CheckBox checkDiscapacidad;
55     TextView textoErrores;
56     TextView textoCalculo;
57
58     @Override
59     protected void onCreate(Bundle savedInstanceState) {
60         super.onCreate(savedInstanceState);
61         setContentView(R.layout.activity_main);
62         cajaPrecio = (EditText) findViewById(R.id.cajaCoste);
63         cajaEdad = (EditText) findViewById(R.id.cajaEdad);
64         radioGroup = (RadioGroup) findViewById(R.id.radioGroup);
65         checkTarjeta = (CheckBox) findViewById(R.id.checkTarjeta);
66         checkDiscapacidad = (CheckBox) findViewById(R.id.checkDiscapacidad);
67         textoErrores = (TextView) findViewById(R.id.textoErrores);
68         textoCalculo = (TextView) findViewById(R.id.textoCalculo);
69     }
70 }
```

Método del botón 'Limpiar' que borra toda la app

```
77 public void limpiar(View v) {
78     textoErrores.setText("");
79     textoCalculo.setText("");
80     cajaPrecio.setText("");
81     cajaEdad.setText("");
82     checkTarjeta.setChecked(false);
83     checkDiscapacidad.setChecked(false);
84     radioGroup.clearCheck();
85 }
```

Diseño de la app



El layout es un `ConstraintLayout` en disposición vertical con los distintos items organizados en filas.

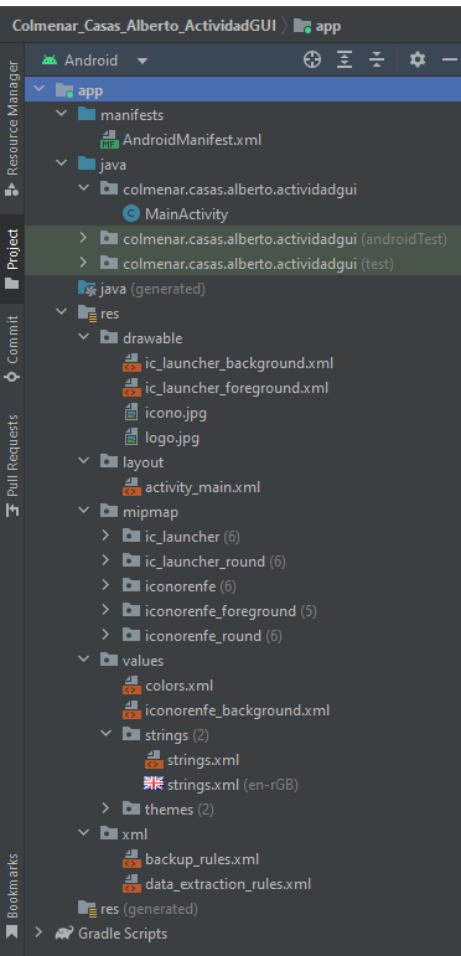
Tenemos el título y logo corporativo en la parte superior.

Después la parte de introducción de datos por parte del usuario y los distintos `checkbox` y `radiobutton`s.

En el espacio entre los `check` y los botones vendrían los textos de error y calculo del billete.

Y en la parte inferior los dos botones que tienen la lógica de la aplicación.

Recursos empleados



En drawable tenemos el logo.jpg utilizado en el diseño y el icono.jpg que es la imagen de referencia para crear el icono de la aplicación.

También tenemos el archivo strings.xml en inglés para poder utilizar la aplicación en dicho idioma.

