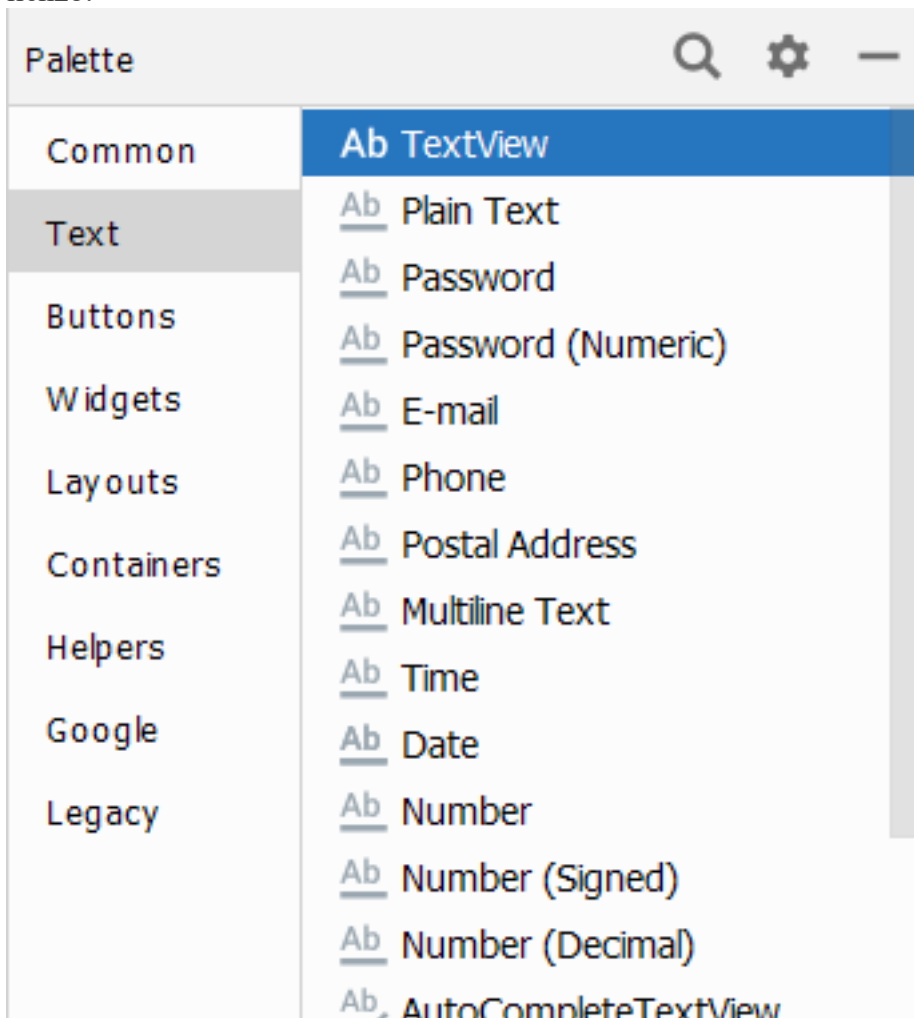


## EJERCICIO 9: TextView

TextView una de las directas descendientes de la clase View. Al ser la abstracción responsable de contener texto, varios de los elementos gráficos del paquete android.widget, como Button y EditText, reutilizan esta naturaleza.

Para agregar un TextView desde el editor de layout de Android Studio tan, solo dirígete a **Palette > Text** o **Palette > Common**. Selecciona la opción **TextView** y arrástralo al lienzo:



Esto creará una nueva etiqueta `<TextView>` en tu layout similar a esta:

```
<TextView
    android:id="@+id/hello_world_text"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello World!" />
```

**Asignar color:**

android:textColor="#3F51B5"

**Determinar tamaño:**

android:textSize="24sp"

**Estilo del texto:** normal, bold y italic, se combinan con |

android:textStyle="normal|bold"

**Tipo de fuente:** normal, sans serif y monospace

android:typeface="monospace"

**URLs:** aplicar android:autoLink con los siguientes valores

all: Todos los patrones

email: Direcciones de correo

none: Ninguno por defecto

phone: teléfonos

web: URL web

usar android:textColorLink para modificar el color del enlace

**Texto a mayúscula:**

android:textAllCaps="true"

**Familia tipo letra:** Debe ser declarada en el directorio res/font. Se debe dar clic derecho en res seleccionando AndroidResourceDirectory y seleccionar el tipo font y agregar desde Google Fonts la fuente deseada aplicándola como android:fontFamily="@font/[font-family]"

android:fontFamily="sans-serif-condensed-medium"

**Apariencia:** Las escalas de tipografía se encuentran definidas en Material Design.

android:textAppearance="@style/TextAppearance.MaterialComponents.HeadLine4"

## Ejercicio 10 EditText

### Objetivo:

Cambiar los tipos de entrada de un EditText  
Modificar atributos como hint, tamaño de texto, color de línea y más  
Manejar eventos de cambio de texto con la interfaz TextWatcher  
Cambiar el foco entre varios edit texts  
Personalizar el estilo  
Autocompletar un EditText  
Ubicar un TextView en un Layout con Android Studio

### La Clase EditText En Android

Es la expansión de TextView con capacidad de editar contenido para recibir texto por parte del usuario. Visualmente estos proyectan una línea inferior del color del acento del tema y un texto auxiliar llamado hint que representa el contenido asociado al view.

1. Crear un nuevo proyecto de nombre Ejercicio10.
2. Escribir el siguiente código en res->layout del archivo fragment\_first.xml
3. Abrir el archivo fragment\_second.xml e incluir el siguiente código:

```
<?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=".SecondFragment">

    <TextView
        android:id="@+id/textview_second"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintBottom_toTopOf="@id/button_second"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <EditText
        android:id="@+id/campo_busqueda"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:ems="10"
        android:hint="Buscar cliente"
        android:imeOptions="actionSearch"
        android:singleLine="true"
        tools:ignore="MissingConstraints" />
```

```

<Button
    android:id="@+id/button_imprime"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Imprime"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/textview_second" />

<Button
    android:id="@+id/button_second"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/previous"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/button_imprime" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

4. Retornar el valor de texto de un EditText con el método getText(). Modificar el layout para que el botón aparezca por debajo del editText. Asignar un manejador de clicks al botón con el atributo onClick.

5. En el SecondFragment colocar el siguiente código:

```

package com.proyectos.ejercicio10;

import android.os.Bundle;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.EditText;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.fragment.app.Fragment;
import androidx.navigation.fragment.NavHostFragment;

import com.proyectos.ejercicio10.databinding.FragmentSecondBinding;

import java.util.zip.Inflater;

public class SecondFragment extends Fragment {

    private FragmentSecondBinding binding;

    @Override
    public View onCreateView(
        LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState
    ) {

```

```

        binding = FragmentSecondBinding.inflate(inflater, container,
false);

        return binding.getRoot();

    }

    public void onCreateView(@NonNull View view, Bundle
savedInstanceState) {
        super.onCreateView(view, savedInstanceState);
        binding.buttonImprime.setOnClickListener(new
View.OnClickListener() {
            EditText et1;
            int n=8;
            @Override
            public void onClick(View view) {

                et1=(EditText)
getView().findViewById(R.id.campo_busqueda);

                Toast.makeText(view.getContext(), "mensaje con valores "
+ n + " " + et1.getText().toString(), Toast.LENGTH_SHORT).show();

            }
        });

        binding.buttonSecond.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {

                NavHostFragment.findNavController(SecondFragment.this)
.navigate(R.id.action_SecondFragment_to_FirstFragment);
            }

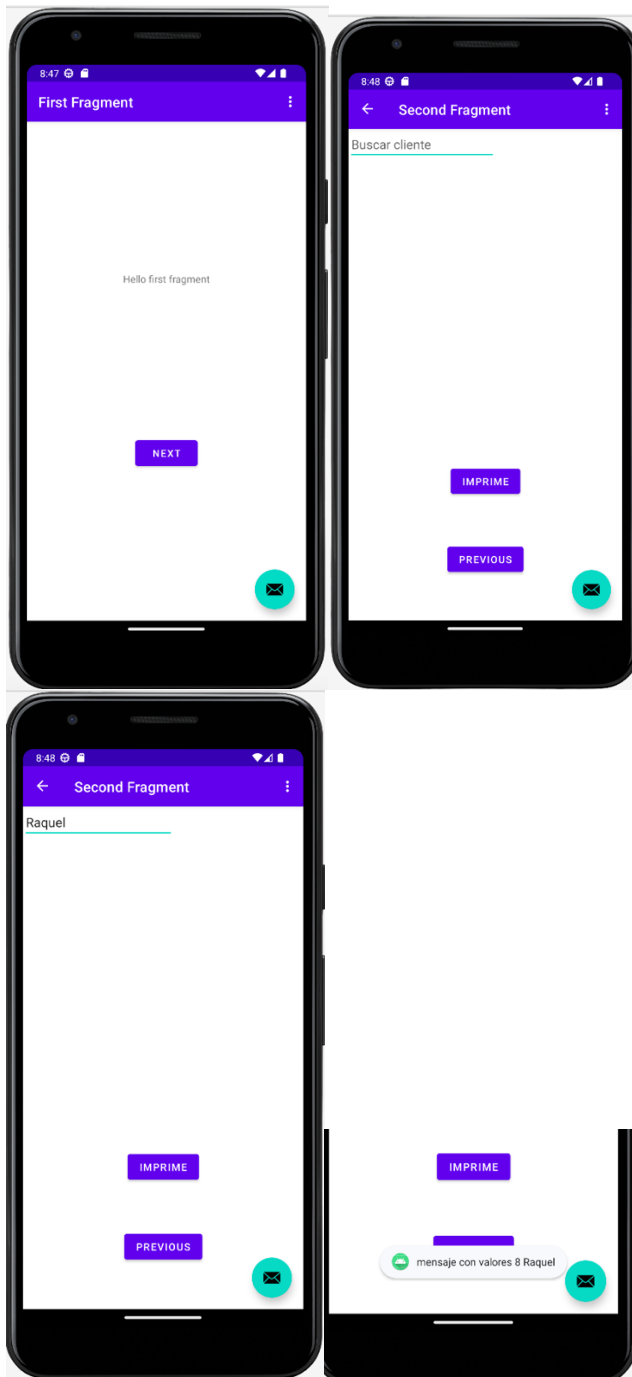
        });

    }

    @Override
    public void onDestroyView() {
        super.onDestroyView();
        binding = null;
    }
}

```

6. Ejecutar la aplicación que permite mostrar en un Toast el valor de la variable n y el contenido del EditText en el botón Imprimir.



Añadir la siguiente instrucción Toast en el MainActivity.java

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    binding = ActivityMainBinding.inflate(getLayoutInflater());
    setContentView(binding.getRoot());

    setSupportActionBar(binding.toolbar);
```

*Dra. Raquel Ochoa Ornelas*

```

NavController navController = Navigation.findNavController(this,
R.id.nav_host_fragment_content_main);
appBarConfiguration = new
AppBarConfiguration.Builder(navController.getGraph()).build();
NavigationUI.setupActionBarWithNavController(this, navController,
appBarConfiguration);

binding.fab.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Snackbar.make(view, "Replace with your own action",
Snackbar.LENGTH_LONG)
            .setAction("Action", null).show();
    }
});

Toast.makeText(this, "Bienvenidos... ", Toast.LENGTH_SHORT).show();
}

```

## Ejercicio 11. Edit Text validación

Diseñar una aplicación que lea la base y altura de un triángulo e imprima su área:

Objetivo: Conocer los distintos tipos de EditText

Existen valores para inputType:

Constante	Descripción
text	Recibe texto plano simple
textPersonName	Texto correspondiente al nombre de una persona
textPassword	Protege los caracteres que se van escribiendo con puntos
numberPassword	Contraseña de solo números enmascarada con puntos
textEmailAddress	Texto que será usado en un campo para emails
phone	Texto asociado a un número de teléfono
textPostalAddress	Para ingresar textos asociados a una dirección postal
textMultiLine	Permite múltiples líneas en el campo de texto
time	Texto para determinar la hora
date	Texto para determinar la fecha
number	Texto con caracteres numéricos
numberSigned	Permite números con signo
numberDecimal	Para ingresar números decimales

1. Diseñar el siguiente formulario en el SecondFragment.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.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=".SecondFragment">

    <TextView
        android:id="@+id/textview_second"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintBottom_toTopOf="@id/button_second"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <EditText
        android:layout_width="300dp"
        android:layout_height="wrap_content"
        android:id="@+id/editText1"
        android:hint="Primer valor"
        android:inputType="numberSigned"
        android:layout_marginLeft="15dp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textview_second"
        />

    <EditText
        android:layout_width="300dp"
        android:layout_height="wrap_content"
        android:id="@+id/editText2"
        android:hint="Segundo valor"
        android:layout_below="@+id/editText1"
        android:layout_centerHorizontal="true"
        android:inputType="numberSigned"
        android:layout_marginLeft="15dp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editText1"
        />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="SUMAR"
        android:id="@+id/button_Sumar"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editText2"
        />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```



```

        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Resultado es: "
        android:id="@+id/textViewResultado"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/button_Sumar"/>

<Button
    android:id="@+id/button_second"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/previous"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textViewResultado" />
</android.support.constraint.ConstraintLayout>

```

## 2. Escribir el siguiente código en SecondFragment.java

```

package com.example.ejercicio11;

import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

import android.support.annotation.NonNull;
import android.support.v4.app.Fragment;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import androidx.navigation.fragment.NavHostFragment;

import com.example.ejercicio11.databinding.FragmentSecondBinding;

public class SecondFragment extends Fragment {

    private FragmentSecondBinding binding;

    @Override
    public View onCreateView(
        LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState
    ) {

        binding = FragmentSecondBinding.inflate(inflater,
        container, false);
        return binding.getRoot();

    }

    public void onViewCreated(@NonNull View view, Bundle
    savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);
    }
}

```

```

        binding.buttonSumar.setOnClickListener(new
View.OnClickListener() {
            EditText et1, et2;

            @Override
            public void onClick(View view) {

                et1=(EditText)
getView().findViewById(R.id.editText1);
                et2=(EditText)
getView().findViewById(R.id.editText2);
                final TextView tvResultado = (TextView)
getView().findViewById(R.id.textViewResultado);

                int aux1 =
Integer.valueOf(et1.getText().toString());
                int aux2 =
Integer.valueOf(et2.getText().toString());
                int resultado = aux1 + aux2;
                tvResultado.setText(""+resultado);
                Toast.makeText(view.getContext(), "mensaje con
valores " + resultado + " " + et1.getText().toString() + " " +
et2.getText().toString(), Toast.LENGTH_SHORT).show();

            }
        });

```

```

        binding.buttonSecond.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {

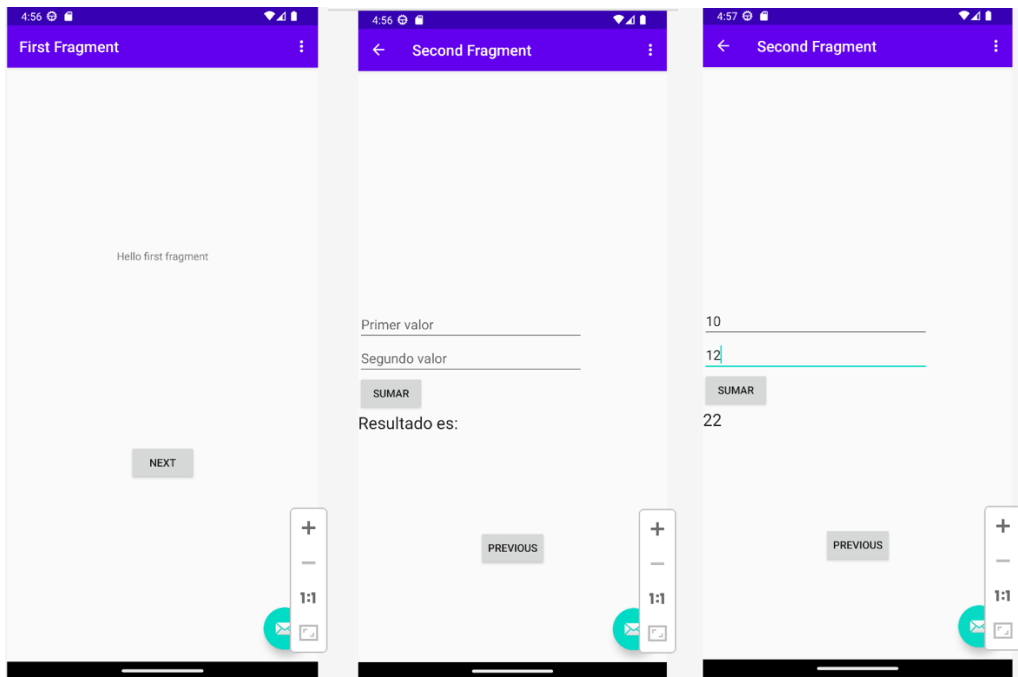
NavHostFragment.findNavController(SecondFragment.this)

.navigate(R.id.action_SecondFragment_to_FirstFragment);
            }
        });

        @Override
        public void onDestroyView() {
            super.onDestroyView();
            binding = null;
        }
    }
}

```

### 3. Ejecutar la aplicación:



## Ejercicio 12. Validaciones completas

Objetivo: Reconocer todos los inputType de los EditText

### 1. Escribir el siguiente código en fragment\_second.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.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=".SecondFragment">

    <TextView
        android:id="@+id/textview_second"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintBottom_toTopOf="@id/button_second"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <EditText
        android:layout_width="300dp"
        android:layout_height="wrap_content"
        android:id="@+id/editTextTel"
        android:hint="Teléfono"
```

```

        android:inputType="phone"
        android:layout_marginLeft="15dp"
        app:layout_constraintStart_toStartOf="parent"

    />

    <EditText
        android:layout_width="300dp"
        android:layout_height="wrap_content"
        android:id="@+id/editTextCorreo"
        android:hint="Email"
        android:layout_centerHorizontal="true"
        android:inputType="textEmailAddress"
        android:layout_marginLeft="15dp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editTextTel"
    />

    <EditText
        android:layout_width="300dp"
        android:layout_height="wrap_content"
        android:id="@+id/editTextPassword"
        android:hint="Password"
        android:layout_centerHorizontal="true"
        android:inputType="textPassword"
        android:layout_marginLeft="15dp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editTextCorreo"
    />

    <EditText
        android:layout_width="300dp"
        android:layout_height="wrap_content"
        android:id="@+id/editTextPasswordNum"
        android:hint="Password numérico"
        android:layout_centerHorizontal="true"
        android:inputType="numberPassword"
        android:layout_marginLeft="15dp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editTextPassword"
    />

    <EditText
        android:layout_width="300dp"
        android:layout_height="wrap_content"
        android:id="@+id/editTextNombre"
        android:hint="Nombre"
        android:layout_centerHorizontal="true"
        android:inputType="textPersonName"
        android:layout_marginLeft="15dp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editTextPasswordNum"
    />

    <EditText
        android:layout_width="300dp"

```

```

        android:layout_height="wrap_content"
        android:id="@+id/editTextPostal"
        android:hint="Dirección postal"
        android:layout_centerHorizontal="true"
        android:inputType="textPostalAddress"
        android:layout_marginLeft="15dp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editTextNombre"
    />

    <EditText
        android:layout_width="300dp"
        android:layout_height="wrap_content"
        android:id="@+id/editTextMultiline"
        android:hint="Texto multilínea"
        android:layout_centerHorizontal="true"
        android:inputType="textMultiLine"
        android:layout_marginLeft="15dp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editTextPostal"
    />

    <EditText
        android:layout_width="300dp"
        android:layout_height="wrap_content"
        android:id="@+id/editTextTime"
        android:hint="Time"
        android:layout_centerHorizontal="true"
        android:inputType="time"
        android:layout_marginLeft="15dp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editTextMultiline"
    />

    <EditText
        android:layout_width="300dp"
        android:layout_height="wrap_content"
        android:id="@+id/editTextDate"
        android:hint="Date"
        android:layout_centerHorizontal="true"
        android:inputType="date"
        android:layout_marginLeft="15dp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editTextTime"
    />

    <EditText
        android:layout_width="300dp"
        android:layout_height="wrap_content"
        android:id="@+id/editTextNumber"
        android:hint="Número"
        android:layout_centerHorizontal="true"
        android:inputType="number"
        android:layout_marginLeft="15dp"
        app:layout_constraintStart_toStartOf="parent"

```

```

        app:layout_constraintTop_toBottomOf="@+id/editTextDate"
    />

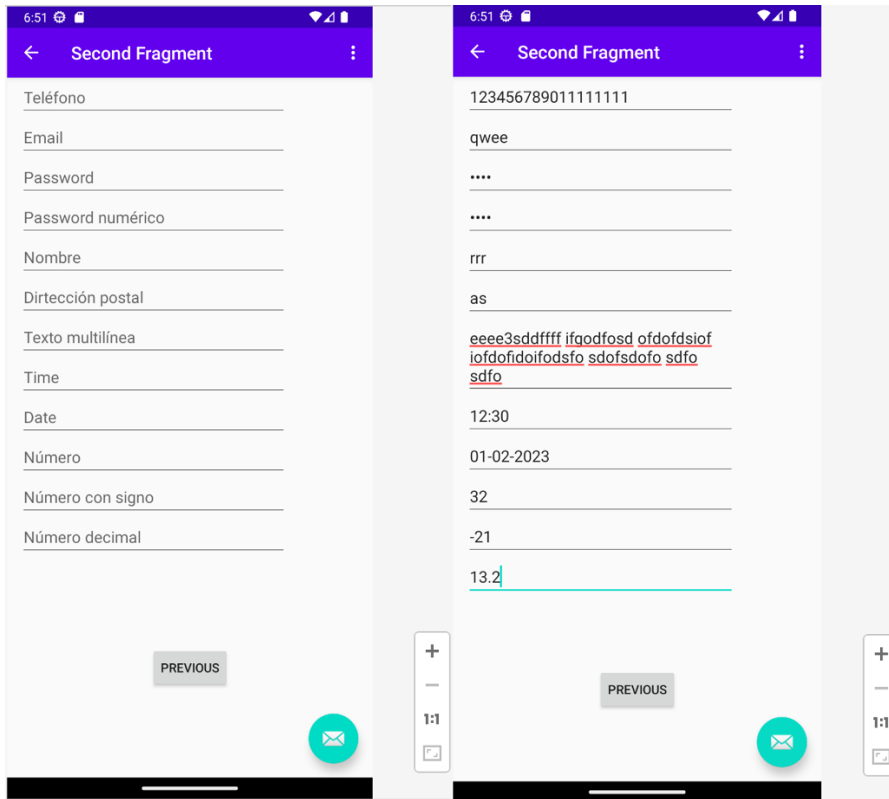
    <EditText
        android:layout_width="300dp"
        android:layout_height="wrap_content"
        android:id="@+id/editTextNumberSigned"
        android:hint="Número con signo"
        android:layout_centerHorizontal="true"
        android:inputType="numberSigned"
        android:layout_marginLeft="15dp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editTextNumber"
    />

    <EditText
        android:layout_width="300dp"
        android:layout_height="wrap_content"
        android:id="@+id/editTextNumberDecimal"
        android:hint="Número decimal"
        android:layout_centerHorizontal="true"
        android:inputType="numberDecimal"
        android:layout_marginLeft="15dp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editTextNumberSigned"
    />

    <Button
        android:id="@+id/button_second"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/previous"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editTextNumberDecimal"
    />
</android.support.constraint.ConstraintLayout>

```

## 2. Ejecutar la aplicación



## Ejercicio 13 EditText editable, focos y longitud

**Objetivo:** Aplicar EditText no editable, focos de entrada y longitud máxima.

Escribir el siguiente código en el fichero fragment\_second.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.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=".SecondFragment">

    <TextView
        android:id="@+id/textview_second"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintBottom_toTopOf="@id/button_second"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
```

```

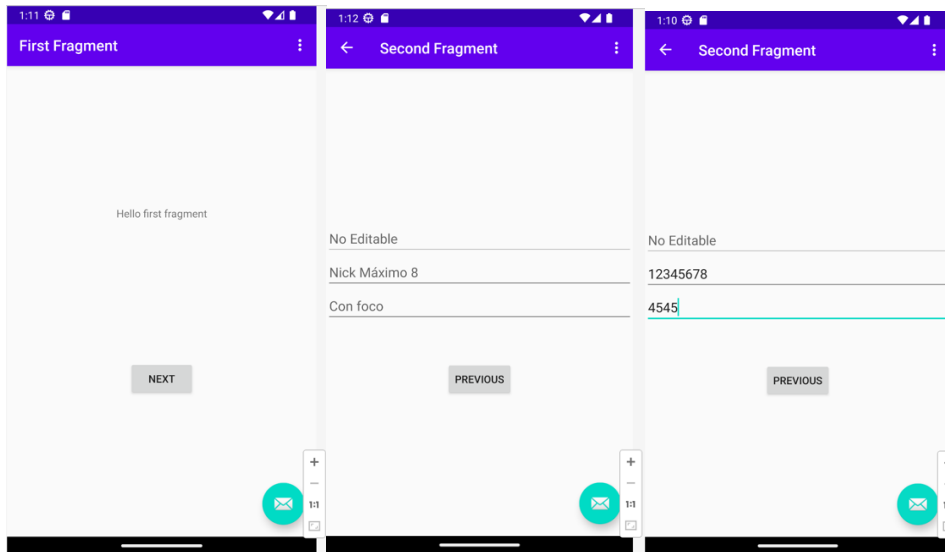
        android:id="@+id/campo_no_editable"
        android:enabled="false"
        android:hint="No Editable"
        app:layout_constraintTop_toBottomOf="@id/textview_second"/>

<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/nombre_usuario"
    android:inputType="text"
    android:hint="Nick Máximo 8"
    android:maxLength="8"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    app:layout_constraintTop_toBottomOf="@+id/campo_no_editable"/>
<EditText
    android:id="@+id/campo_sin_foco"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:focusable="true"
    android:hint="Con foco"
    app:layout_constraintTop_toBottomOf="@+id/nombre_usuario"/>
<Button
    android:id="@+id/button_second"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/previous"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/textview_second" />
</android.support.constraint.ConstraintLayout>

```

Ejecutar la aplicación





## Ejercicio 14. Casillas de Verificación

Las casillas de verificación permiten que el usuario seleccione una o más opciones de un conjunto. Por lo general, debes presentar cada opción de casilla de verificación en una lista vertical.

Para crear cada opción, crea un objeto `CheckBox` en tu diseño. Dado que un conjunto de opciones de casilla de verificación le permite al usuario seleccionar varios elementos, se administra cada casilla por separado y debes registrar un objeto de escucha de clics para cada una.

Una clase clave es la siguiente:

- `CheckBox`

### Cómo responder a eventos de clic

Cuando el usuario selecciona una casilla de verificación, el objeto `CheckBox` recibe un evento de clic.

A fin de definir el controlador de eventos de clic para una casilla de verificación, agrega el atributo `android:onClick` al elemento `<CheckBox>` en tu diseño XML. El valor de este atributo debe ser el nombre del método al que deseas llamar en respuesta a un evento de clic. La Activity que aloja el diseño debe implementar el método correspondiente.

1. Escribir el siguiente código en el archivo `fragment_second.xml`:

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.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=".SecondFragment">

    <TextView
        android:id="@+id/textview_second"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintBottom_toTopOf="@id/button_second"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <CheckBox
        android:id="@+id/checkBox"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:text="Recordarme" />

    <Button
        android:id="@+id/button_second"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/previous"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textview_second" />
</android.support.constraint.ConstraintLayout>

```

## 2. Escribir el siguiente Código en el archivoSecondFragment.java

```

package com.example.ejemplo14;

import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

import android.support.annotation.NonNull;
import android.support.v4.app.Fragment;
import android.widget.CheckBox;
import android.widget.Toast;

import androidx.navigation.fragment.NavHostFragment;

import com.example.ejemplo14.databinding.FragmentSecondBinding;

public class SecondFragment extends Fragment {

```

```

private FragmentSecondBinding binding;

@Override
public View onCreateView(
    LayoutInflater inflater, ViewGroup container,
    Bundle savedInstanceState
) {

    binding = FragmentSecondBinding.inflate(inflater, container,
false);
    return binding.getRoot();

}

public void onViewCreated(@NonNull View view, Bundle
savedInstanceState) {
    super.onViewCreated(view, savedInstanceState);

    binding.checkBox.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            CheckBox seleccionDireccion;
            seleccionDireccion=getView().findViewById(R.id.checkBox);
            String s = "Estado: " + (seleccionDireccion.isChecked() ?
"Marcado" : "No Marcado");
            //Toast.makeText(this, s, Toast.LENGTH_LONG).show();
            Toast.makeText(view.getContext(), "mensaje con valores "
+ s, Toast.LENGTH_SHORT).show();

        }
    });

    //_____

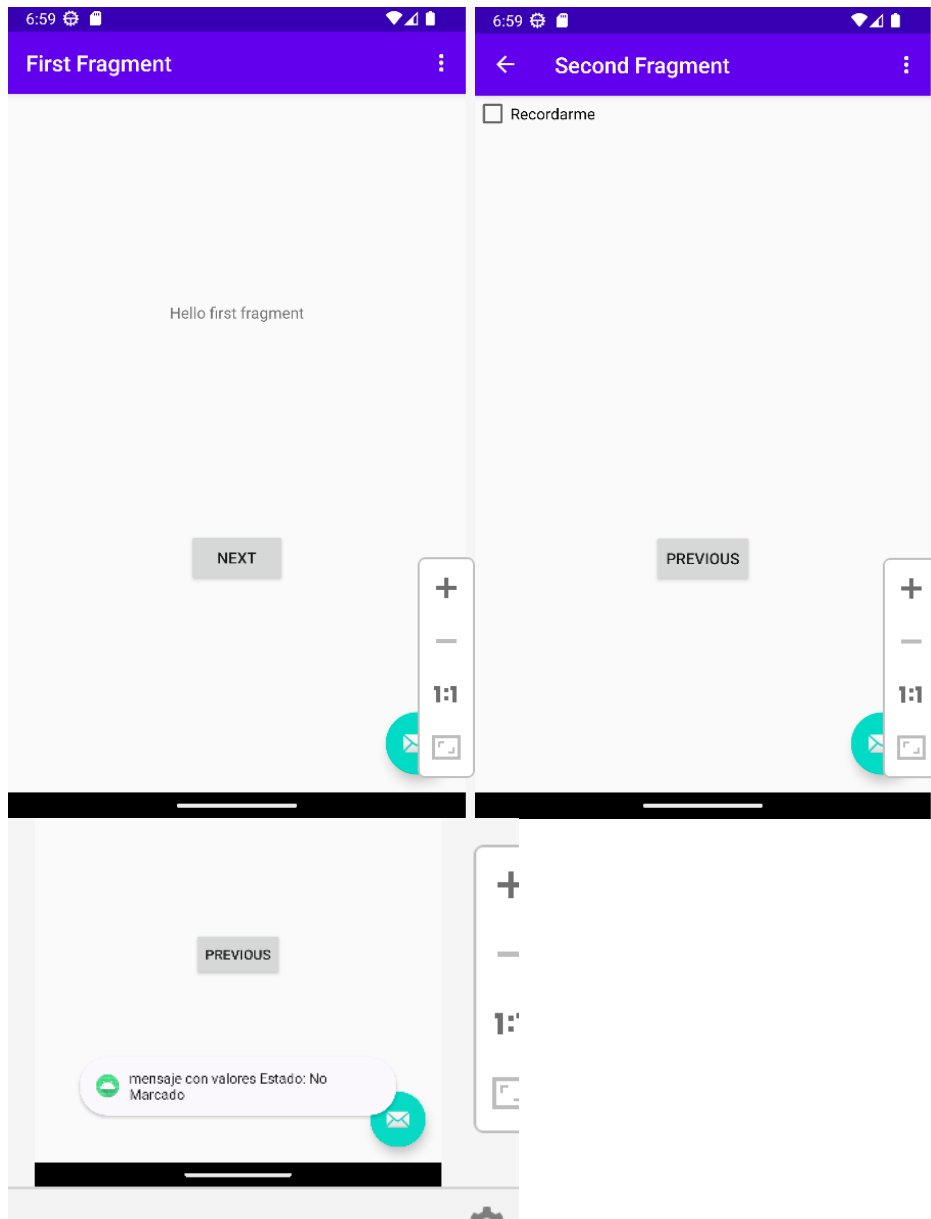
    binding.buttonSecond.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View view) {
            NavHostFragment.findNavController(SecondFragment.this)

.navigate(R.id.action_SecondFragment_to_FirstFragment);
        }
    });
}

@Override
public void onDestroyView() {
    super.onDestroyView();
    binding = null;
}
}

```

### 3. Ejecutar la aplicación:



## Ejercicio 15. Ejercicio casillas de verificación

**Objetivo:** Aplicar un descuento si el total sobrepasa \$20,000 encendiendo la casilla de verificación.

1. Escribir el siguiente código en el archivo `fragment_second.xml`

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.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=".SecondFragment">

        <TextView
            android:id="@+id/textview_second"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            app:layout_constraintBottom_toTopOf="@id/button_second"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent" />

        <EditText
            android:id="@+id/campo_monto"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:ems="10"
            android:hint="Monto total"
            android:inputType="numberDecimal"
            android:singleLine="true"
            app:layout_constraintTop_toBottomOf="@+id/textview_second" />

        <CheckBox
            android:id="@+id/opcion_descuento"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Aplicar descuento"
            app:layout_constraintTop_toBottomOf="@+id/campo_monto"/>

        <Button
            android:id="@+id/button_second"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/previous"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toBottomOf="@+id/opcion_descuento" />
    </android.support.constraint.ConstraintLayout>

```

## 2. Escribir el siguiente Código en SecondFragment.java:

```

package com.example.ejemplo15;

import android.os.Bundle;
import android.text.Editable;
import android.text.TextUtils;
import android.text.TextWatcher;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

import android.support.annotation.NonNull;
import android.support.v4.app.Fragment;

```

```

import android.widget.CheckBox;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import androidx.navigation.fragment.NavHostFragment;

import com.example.ejemplo15.databinding.FragmentSecondBinding;

public class SecondFragment extends Fragment {

    private FragmentSecondBinding binding;

    @Override
    public View onCreateView(
        LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState
    ) {

        binding = FragmentSecondBinding.inflate(inflater, container,
false);
        return binding.getRoot();

    }

    public void onViewCreated(@NonNull View view, Bundle
savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);

        CheckBox opcionDescuento;
        TextView campoMonto;
        opcionDescuento = getView().findViewById(R.id.opcion_descuento);
        campoMonto = (EditText) getView().findViewById(R.id.campo_monto);

        campoMonto.addTextChangedListener(new TextWatcher() {

            @Override
            public void beforeTextChanged(CharSequence s, int start, int
count, int after) {

            }

            @Override
            public void onTextChanged(CharSequence s, int start, int
before, int count) {

            }

            @Override
            public void afterTextChanged(Editable s) {
                String str = s.toString();

                // Precondición
                if (TextUtils.isEmpty(str)) {
                    str = "0";
                }
            }
        });
    }
}

```

```

        float monto = Float.parseFloat(str);

        if (monto >= 2000) {
            opcionDescuento.setChecked(true);
        }
    }
});

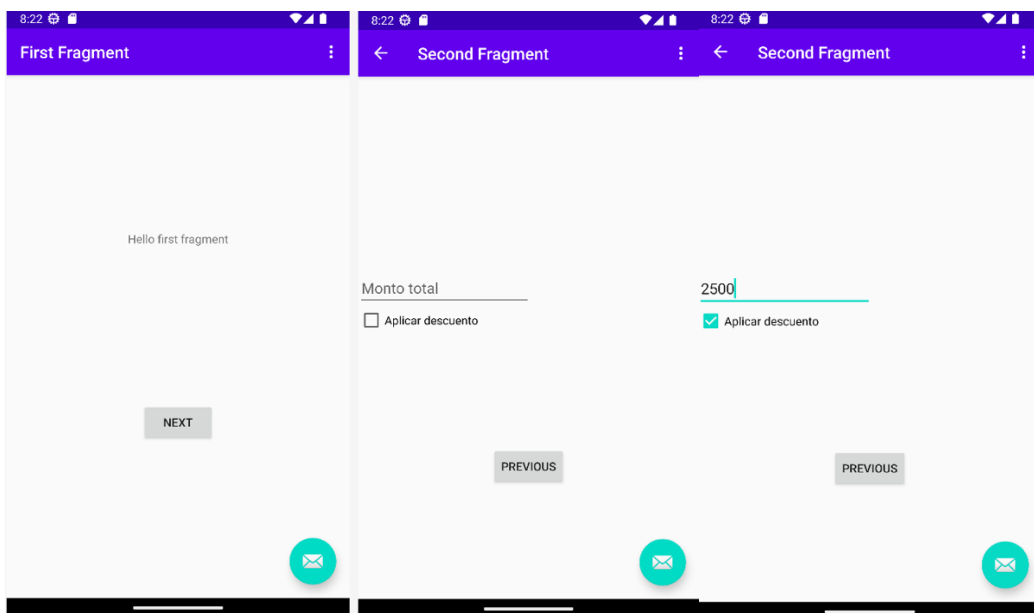
//-----

binding.buttonSecond.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View view) {
        NavHostFragment.findNavController(SecondFragment.this)
            .navigate(R.id.action_SecondFragment_to_FirstFragment);
    }
});
}

@Override
public void onDestroyView() {
    super.onDestroyView();
    binding = null;
}
}

```

### 3. Ejecutar la aplicación:



## Ejercicio 16. Verificar casillas de verificación

**Objetivos:** Ejercicio de Presentar al usuario un conjunto de gustos para seleccionar preferencias.

1. Cambiar el contenido del fragment\_second.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_column="0"
        android:layout_row="0"
        android:text="Elige tus intereses"
        android:textAppearance="@style/TextAppearance.AppCompat.Body1" />

    <CheckBox
        android:id="@+id/campo_belleza"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_column="0"
        android:layout_row="1"
        android:checked="false"
        android:text="Belleza" />

    <CheckBox
        android:id="@+id/campo_libros_literatura"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_column="1"
        android:layout_row="1"
        android:checked="false"
        android:text="Libros y Literatura" />

    <CheckBox
        android:id="@+id/campo_comida_bebidas"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_column="0"
        android:layout_row="2"
        android:checked="false"
        android:text="Comida y Bebidas" />

    <CheckBox
        android:id="@+id/campo_carreras" />
```



```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_column="1"
        android:layout_row="2"
        android:checked="false"
        android:text="Carreras" />

<CheckBox
    android:id="@+id/campo_videojuegos"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_column="0"
    android:layout_row="3"
    android:checked="false"
    android:text="Videojuegos" />

<CheckBox
    android:id="@+id/campo_educacion"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_column="1"
    android:layout_row="3"
    android:checked="false"
    android:text="Educación" />

<CheckBox
    android:id="@+id/campo_deportes"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_column="0"
    android:layout_row="4"
    android:checked="false"
    android:text="Deportes" />

<CheckBox
    android:id="@+id/campo_eventos"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_column="1"
    android:layout_row="4"
    android:checked="false"
    android:text="Eventos" />

<CheckBox
    android:id="@+id/campo_familia"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_column="0"
    android:layout_row="5"
    android:checked="false"
    android:text="Familia" />

<CheckBox
    android:id="@+id/campo_arte"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_column="1"

```

```

        android:layout_row="5"
        android:checked="false"
        android:text="Arte" />

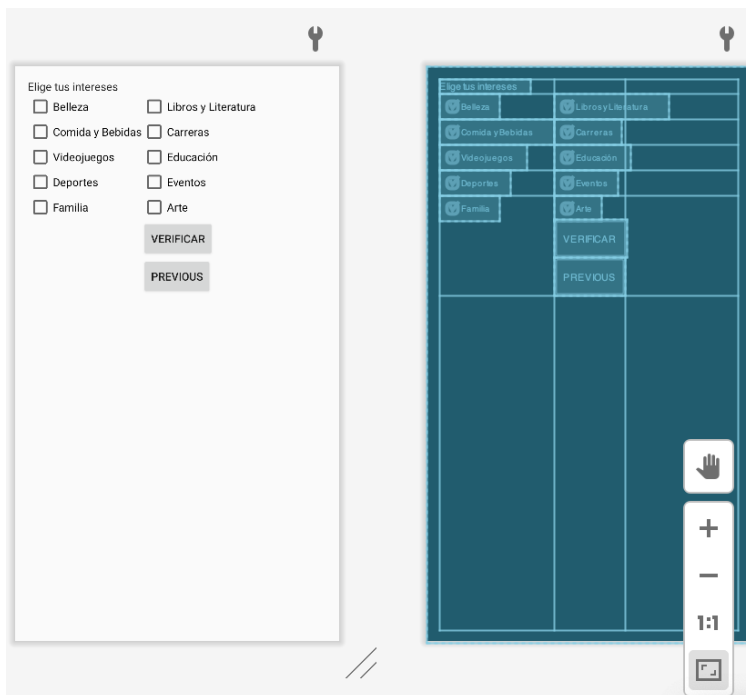
<Button
    android:id="@+id/button_verifica"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Verificar"
    />

<Button
    android:id="@+id/button_second"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/previous"
    />

</GridLayout>

```

## 2. Seleccionar la pestaña diseño



## 3. Ajustar el siguiente Código en el SecondFragment

```

import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

import android.support.annotation.NonNull;
import android.support.v4.app.Fragment;
import android.widget.CheckBox;
import android.widget.Toast;

```

```

import androidx.navigation.fragment.NavHostFragment;

import com.example.ejercicio16.databinding.FragmentSecondBinding;

public class SecondFragment extends Fragment {

    private FragmentSecondBinding binding;

    @Override
    public View onCreateView(
        LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState
    ) {

        binding = FragmentSecondBinding.inflate(inflater, container,
false);
        return binding.getRoot();

    }

    public void onViewCreated(@NonNull View view, Bundle
savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);

        binding.buttonVerifica.setOnClickListener(new
View.OnClickListener() {
            CheckBox c1,c2,c3,c4,c5,c6,c7,c8,c9,c10;

            @Override
            public void onClick(View view) {

                c1=getView().findViewById(R.id.campo_belleza);
                c2=getView().findViewById(R.id.campo_libros_literatura);
                c3=getView().findViewById(R.id.campo_comida_bebidas);
                c4=getView().findViewById(R.id.campo_carreras);
                c5=getView().findViewById(R.id.campo_videojuegos);
                c6=getView().findViewById(R.id.campo_educacion);
                c7=getView().findViewById(R.id.campo_deportes);
                c8=getView().findViewById(R.id.campo_eventos);
                c9=getView().findViewById(R.id.campo_familia);
                c10=getView().findViewById(R.id.campo_arte);
                String R1="Belleza: " + String.valueOf(c1.isChecked())+
"Libros: "+ String.valueOf(c2.isChecked())+ "Comida bebidas: "+
String.valueOf(c3.isChecked()) + "Careras: " +
String.valueOf(c4.isChecked()) + "Videojuegos: " +
String.valueOf(c5.isChecked());
                String R2="Educación: " + String.valueOf(c6.isChecked())
+ "Deportes: " + String.valueOf(c7.isChecked()) + "Ebventos: " +
String.valueOf(c8.isChecked()) + "Familia: " +
String.valueOf(c9.isChecked()) + "Arte: " +
String.valueOf(c10.isChecked());

                Toast.makeText(view.getContext(), "Resultados: " + R1 ,
Toast.LENGTH_SHORT).show();
            }
        });
    }
}

```

```

        Toast.makeText(view.getContext(), "Resultados: " + R2 ,
        Toast.LENGTH_SHORT).show();
    }

    });

    //-----

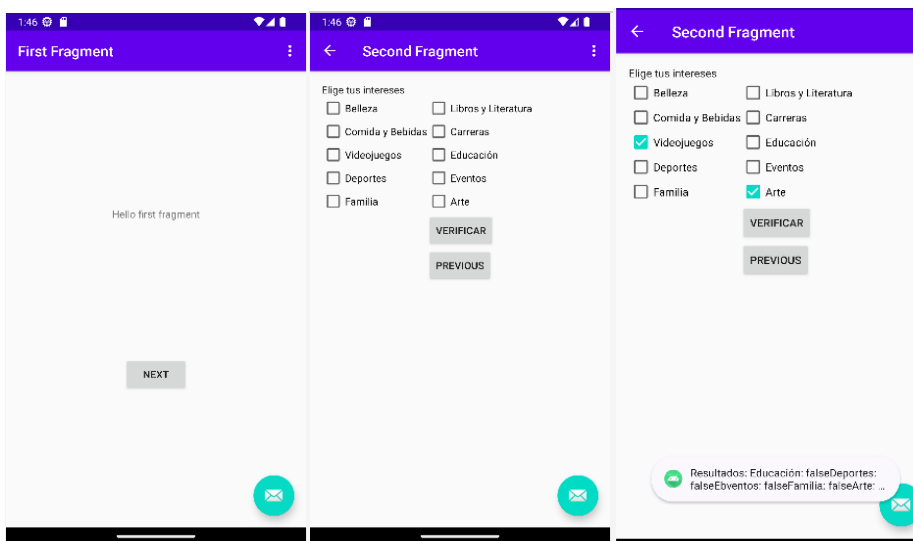
    binding.buttonSecond.setOnClickListener(new
    View.OnClickListener() {
        @Override
        public void onClick(View view) {
            NavHostFragment.findNavController(SecondFragment.this)

            .navigate(R.id.action_SecondFragment_to_FirstFragment);
        }
    });

    @Override
    public void onDestroyView() {
        super.onDestroyView();
        binding = null;
    }
}
}

```

#### 4. Ejecutar la aplicación



## Ejercicio 17. Botones de radio

**Objetivo:** Proporcionan exclusion mutua entre las opciones de un conjunto. La exclusión se logra con el componente RadioGroup, que acumula los radio buttons para permitir la selección de uno solamente.

RadioGoup extiende de LinarLayout, por lo que se puede utilizar el atributo android:orientation para alinear los radio buttons.

Un Radio Button puede atravesar por 5 estados al igual que los otros controles de selección.

- **hover:** el radio button está a la espera de eventos
- **focused:** el sistema de navegación de android enfocó al radio button
- **pressed:** el usuario mantiene presionado el radio button
- **disabled:** el radio button no está habilitado para interacción
- **disabled-focused:** combinación de **disabled** + **focused**.

El color que usa el radio button al estar activo se deriva del atributo `android:colorAccent` que se declara en el tema principal de la app.

### Obtener Valor De RadioButton

La obtención del estado de un radio button se realiza con el método `isChecked()` de la superclase `CompoundButton`. El retorno es booleano, donde `true` representa el estado «On» y `false` el estado «Off».

1. Escribir el siguiente código en el archivo `fragment_second.xml`:

```
<?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=".SecondFragment">

    <TextView
        android:id="@+id/textview_second"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintBottom_toTopOf="@id/button_second"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/etiqueta_opciones"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```

        android:text="Género seleccionado:"
        app:layout_constraintBottom_toTopOf="@id/button_second"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
    />

    <RadioGroup
        android:id="@+id/opciones_sexo"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center"
        android:orientation="horizontal">

        <RadioButton
            android:id="@+id/radio_masculino"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginRight="16dp"
            android:checked="true"
            android:text="Masculino"
        />

        <RadioButton
            android:id="@+id/radio_femenino"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:checked="false"
            android:text="Femenino" />

    </RadioGroup>

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"

        android:text="Mostrar"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/etiqueta_opciones"/>

    <Button
        android:id="@+id/button_second"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/previous"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/button" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

## 2. Escribir el siguiente Código en SecondFragment.java:

```
package com.example.ejercicio17;

import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.RadioButton;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.fragment.app.Fragment;
import androidx.navigation.fragment.NavHostFragment;

import com.example.ejercicio17.databinding.FragmentSecondBinding;

public class SecondFragment extends Fragment {

    private FragmentSecondBinding binding;

    @Override
    public View onCreateView(
        LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState
    ) {

        binding = FragmentSecondBinding.inflate(inflater, container,
false);
        return binding.getRoot();

    }

    public void onViewCreated(@NonNull View view, Bundle
savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);

        binding.button.setOnClickListener(new View.OnClickListener() {

            RadioButton b1, b2;
            String texto="";

            @Override
            public void onClick(View view) {

                b1=(RadioButton)
getView().findViewById(R.id.radio_masculino);
                b2=(RadioButton)
getView().findViewById(R.id.radio_femenino);
                final TextView tvResultado = (TextView)
getView().findViewById(R.id.etiqueta_opciones);

                if (b1.isChecked())
```

```

        texto="Seleccionó masculino";

    else
        texto="Seleccionó femenino";

    tvResultado.setText(texto);
    Toast.makeText(view.getContext(), texto,
Toast.LENGTH_SHORT).show();

    });

    binding.buttonSecond.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View view) {
        NavHostFragment.findNavController(SecondFragment.this)
.navigate(R.id.action_SecondFragment_to_FirstFragment);
    }
});

    @Override
    public void onDestroyView() {
        super.onDestroyView();
        binding = null;
    }
}
}

```

### 3. Ejecutar la aplicación:



