

6. Navegación II

Alberto Luengo



Navegación



Fragments: qué son y sus ciclos de vida

Un fragmento es un componente de Android que contiene parte del comportamiento y / o la interfaz de usuario de una actividad . Los fragmentos están vinculados a una sola actividad. En muchos sentidos, tienen una funcionalidad similar a la de las actividades.

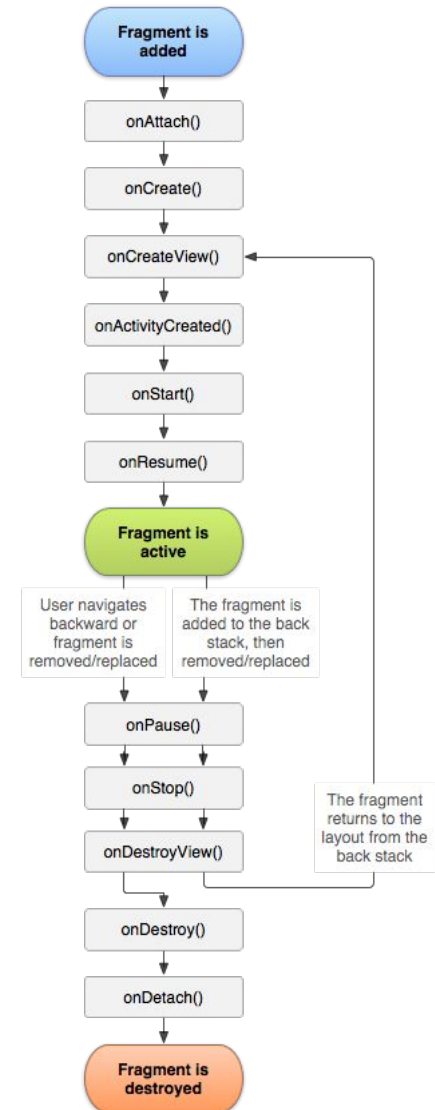
*Imagina por un momento que eres una actividad. Tienes mucho que hacer, por lo que puedes emplear a unos pocos **minions** para que te lleven la ropa y paguen impuestos a cambio de alojamiento y comida. Eso es algo así como la relación entre actividades y fragmentos.*

No es un requisito utilizar fragments, *porque tu mismo puedes lavar tu ropa y pagar tus impuestos*, pero si los usas bien pueden proporcionar:

- **Modularidad:** dividir código de actividad complejo entre fragmentos para una mejor organización y mantenimiento.
- **Reusabilidad:** colocar el comportamiento o las partes de la IU en fragmentos que pueden compartirse en múltiples actividades.
- **Adaptabilidad:** representa secciones de una interfaz de usuario como diferentes fragmentos y utiliza diferentes diseños según la orientación y el tamaño de la pantalla.

Tutorial para aprenderlo todo sobre los fragments:

<https://www.raywenderlich.com/169885/android-fragments-tutorial-introduction-2>



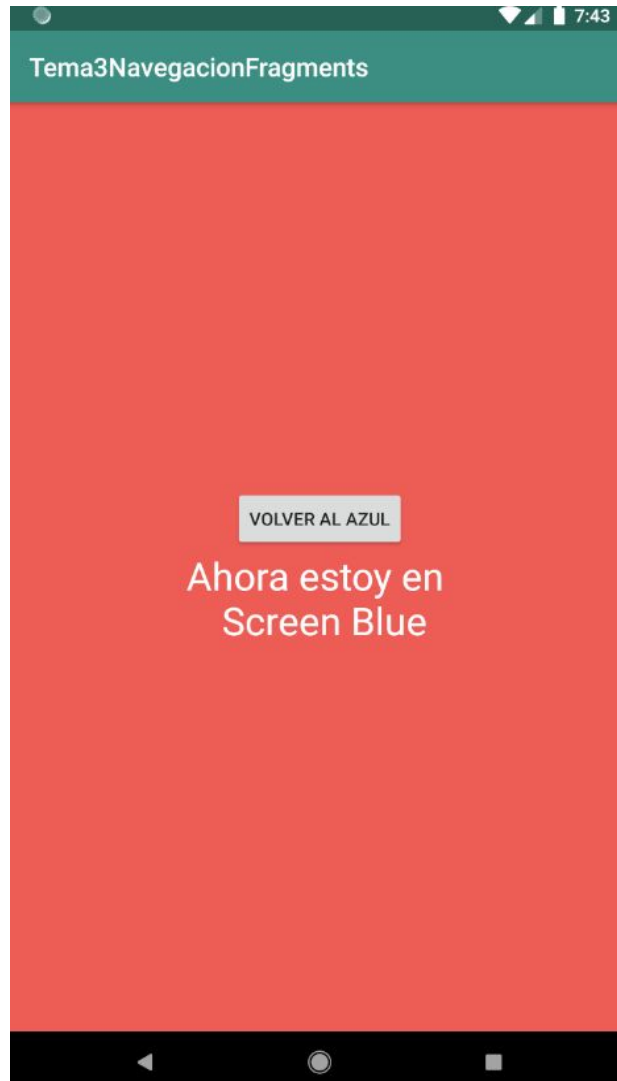
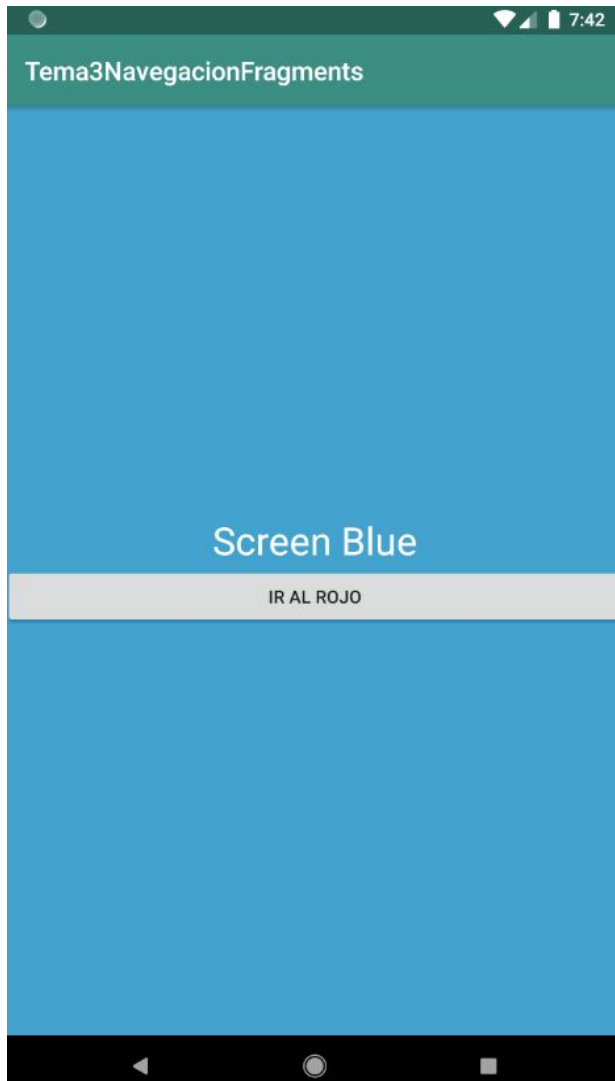
Vamos a crear DOS Fragment

BlueFragment

RedFragment

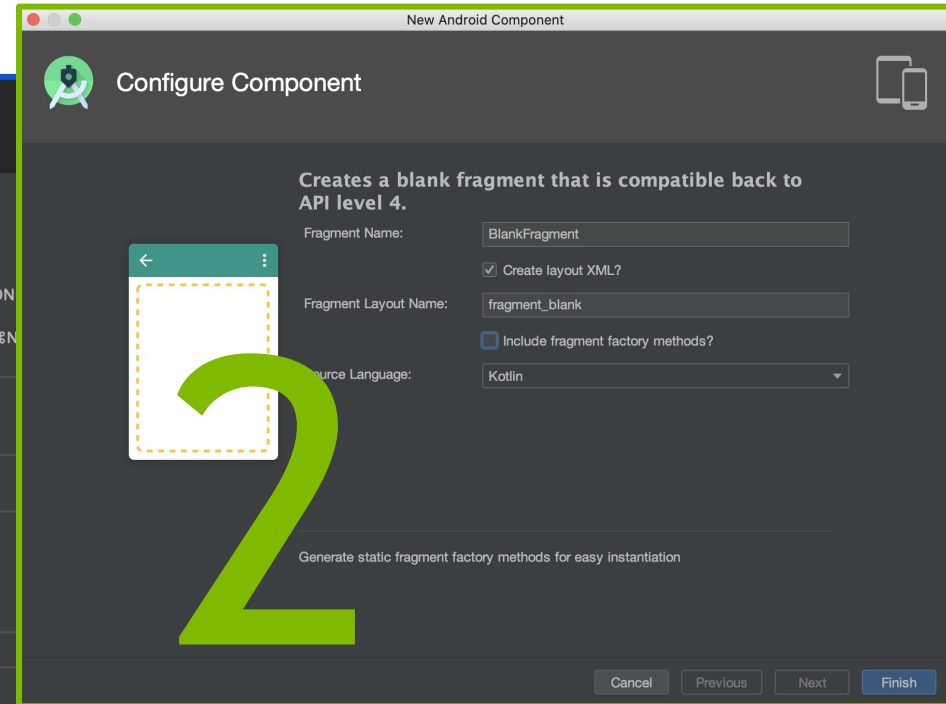
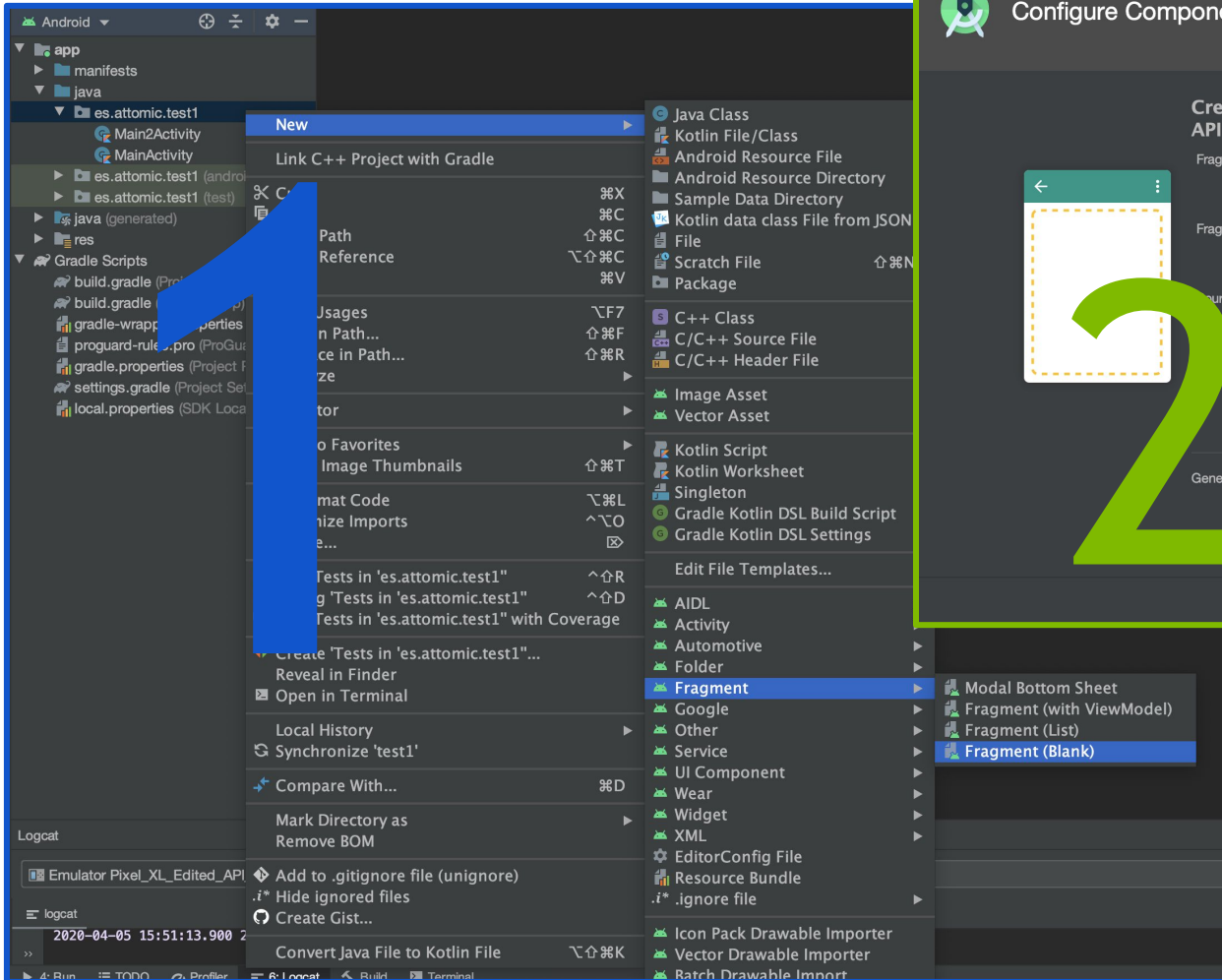
Nos tiene que quedar asi

```
class BlueFragment : Fragment() {  
  
    override fun onCreateView(  
        inflater: LayoutInflater, container: ViewGroup?,  
        savedInstanceState: Bundle?  
    ): View? {  
        // Inflate the layout for this fragment  
        return inflater.inflate(R.layout.fragment_blue, container, false)  
    }  
}
```



Navegación con Fragments

¿Cómo crear un fragment?



Navegación con Fragments

1. Necesitamos crear un `FragmentManager` en una Activity como “contenedor” donde meter nuestro `Fragment`

```
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout 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"  
    android:orientation="vertical"  
    tools:context=".Main2Activity">  
  
    <androidx.fragment.app.FragmentContainerView  
        android:id="@+id/mainContainer"  
        android:layout_width="match_parent"  
        android:layout_height="match_parent" />  
  
</LinearLayout>
```

Navegación con Fragments

2. En el activity, le decimos que en el **FragmentManager** tiene que estar nuestro Fragment

```
class MainActivity : AppCompatActivity() {  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main2)  
  
        fragmentManager.beginTransaction().replace(R.id.mainContainer, BlueFragment()).commit()  
  
    }  
}
```



Volver atrás

Si necesitamos volver atrás entre fragments, antes de lanzarlo hay que añadirlo a una “pila” que se llama “backStack”. De esa manera se guardará que en algún momento queremos volver atrás.

Estos serian los pasos:

1. Lanzar el primer fragment desde la activity, a este no queremos volver a trás, si no que será el principal


```
class MainActivity : AppCompatActivity() {  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main2)  
  
        supportFragmentManager.  
            beginTransaction().  
            replace(R.id.mainContainer, BlueFragment())  
            .addToBackStack("BlueFragment").commit()  
    }  
}
```



Volver atrás

2. Desde el fragment “BlueFragment” llamaremos al “RedFragment” al pulsar un botón y le diremos que nos guarde el fragment en la pila

```
class BlueFragment : Fragment() {  
  
    override fun onCreateView(inflater: LayoutInflater, container: ViewGroup?, savedInstanceState: Bundle?): View? {  
        // Inflate the layout for this fragment  
        return inflater.inflate(R.layout.fragment_blue, container, false)  
    }  
  
    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {  
        super.onViewCreated(view, savedInstanceState)  
  
        btIrRojo.setOnClickListener {  
            activity!!.supportFragmentManager.beginTransaction().replace(R.id.mainContainer, RedFragment()).addToBackStack("RedFragment").commit()  
        }  
    }  
}
```



Siempre escribiremos el
codigo en
onViewCreated

Pasar parametros entre fragments

Desde el fragment “BlueFragment” al instanciar “RedFragment” iniciaremos los arguments y le pasaremos los datos que necesitemos para leer en el otro fragment

```
class BlueFragment : Fragment() {  
    override fun onCreateView(  
        inflater: LayoutInflater, container: ViewGroup?,  
        savedInstanceState: Bundle?  
    ): View? {  
        return inflater.inflate(R.layout.fragment_blue, container, false)  
    }  
  
    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {  
        super.onViewCreated(view, savedInstanceState)  
        btIrRojo.setOnClickListener {  
            activity?.let {  
                val fragment = RedFragment()  
                fragment.arguments = Bundle().apply {  
                    putString("nombre", "Charles")  
                }  
                it.supportFragmentManager.beginTransaction().replace(R.id.container, fragment).addToBackStack("").commit()  
            }  
        }  
    }  
}
```

Leer parametros entre fragments

Desde el fragment "RedFragment" leeremos nuestro parametro usando el atributo arguments y getString con el nombre que le dimos a nuestro argumento

```
override fun onCreateView(view: View, savedInstanceState: Bundle?) {  
    super.onCreateView(view, savedInstanceState)
```

```
    val nombre = arguments?.getString("nombre")
```

```
    Log.i("RedFragment", nombre)
```

```
}
```

Volver atrás por código

3. Y ahora, volveremos atrás desde el “RedFragment” al “BlueFragment”

```
class RedFragment : Fragment() {  
  
    override fun onCreateView(inflater: LayoutInflater, container: ViewGroup?, savedInstanceState: Bundle?): View? {  
        // Inflate the layout for this fragment  
        return inflater.inflate(R.layout.fragment_red, container, false)  
    }  
  
    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {  
        super.onViewCreated(view, savedInstanceState)  
  
        btnBack.setOnClickListener {  
            activity?.supportFragmentManager?.popBackStack()  
        }  
    }  
}
```

Para más información ver:

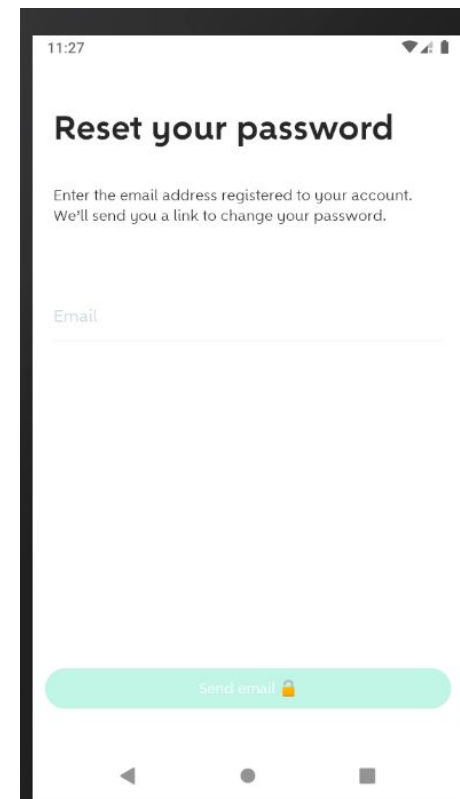
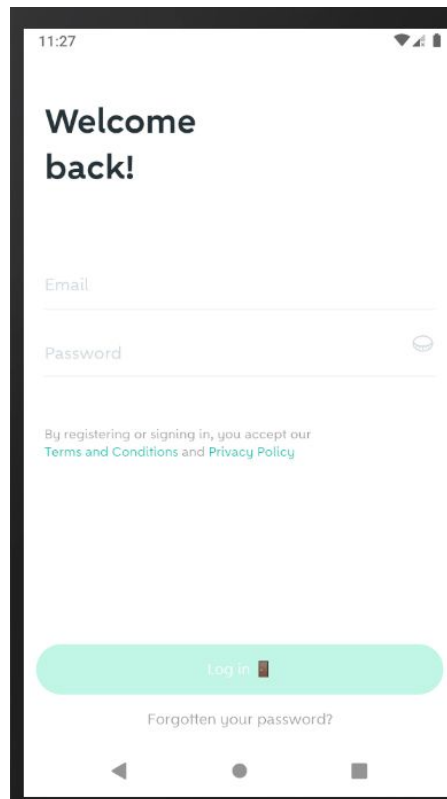
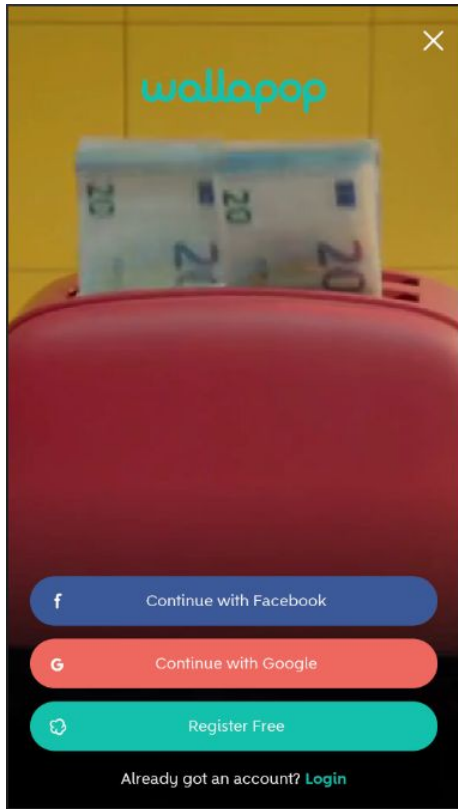
https://developer.android.com/guide/components/fragments?hl=es_419

Ejercicios

Clonar 3 pantallas de Wallapop

Pistas

1. Usar la web de [material buttons](#)
2. Usar la web de [text fields](#)
3. La pantalla 2 requiere un textview con enlaces a una web (usar HTML y Google)
4. La pantalla 1 requiere un degradado para mostrar correctamente el texto de abajo



Gracias.