

# PROGRAMACIÓN MULTIMEDIA Y DISPOSITIVOS MÓVILES

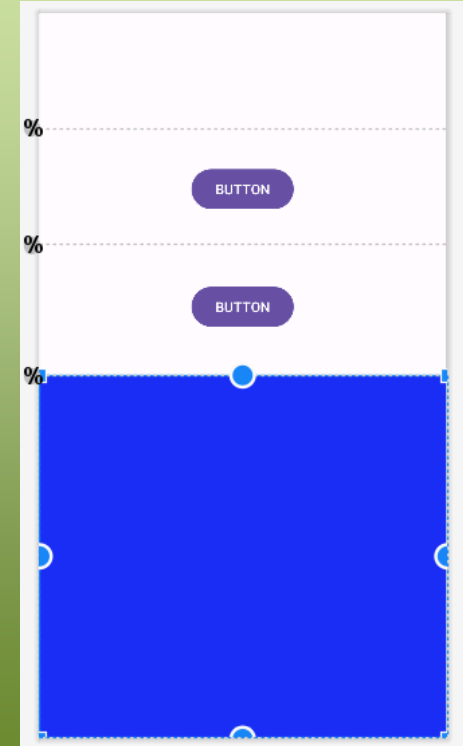
## ANEXO TEMA 5:

Cargar fragments en tiempo de ejecución.



## 2. Cargar Fragments en tiempo de ejecución

- Podemos también cargar distintos Fragments en tiempo de ejecución.
- **Ejemplo Transacciones Fragment:** Procedemos a crear un proyecto nuevo llamado *Transacciones Fragment*. A continuación le añadimos dos Fragments al nuevo proyecto (*AzulFragment* y *AmarilloFragment*) y diseñamos la actividad principal tal y como muestra la imagen.



## 2. Cargar Fragments en tiempo de ejecución

- Para añadir el *FragmentManager* (*Container* -> *FragmentManager*) sólo debemos arrastrarlo a nuestra activity.
- Para que se estire lo máximo posible en el Constraint Layout, debemos poner a *odp* los valores para el alto y el ancho del *FragmentManager*, como se muestra en la siguiente diapositiva.

## 2. Cargar Fragments en tiempo de ejecución

```
<androidx.fragment.app.FragmentContainerView
    android:id="@+id/fragmentContainerView"
    android:name="com.example.ejemplotransaccionesfragment.AzulFragment"
    android:layout_width="0dp"
    android:layout_height="0dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/guideline4"
    tools:layout="@layout/fragment_azul" />
```

## 2. Cargar Fragments en tiempo de ejecución

- En la clase *MainActivity.kt* debemos añadir lo siguiente:
  - **onCreate:** Implementación de la funcionalidad de cada botón (botón que carga el *AzulFragment* y botón que carga el *AmarilloFragment*).

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)

    val btnAzul: Button = findViewById(R.id.btnAzul)
    val btnAmarillo: Button = findViewById(R.id.btnAmarillo)

    //BOTÓN FRAGMENT AZUL
    btnAzul.setOnClickListener { it: View!
        var blueIsClicked: Boolean = true
        loadFragment(blueIsClicked)
    }

    //BOTÓN FRAGMENT AMARILLO
    btnAmarillo.setOnClickListener { it: View!
        var blueIsClicked: Boolean = false
        loadFragment(blueIsClicked)
    }
}
```

## 2. Cargar Fragments en tiempo de ejecución

- Función ***loadFragment*** que contiene la lógica para reemplazar un Fragment por otro.

```
//FUNCIÓN ENCARGADA DE CARGAR UN FRAGMENT
private fun loadFragment(blueIsClicked: Boolean) {

    //Clase para interactuar con un fragment dentro de un activity
    val fragmentManager: FragmentManager = supportFragmentManager
    //Clase para realizar operaciones con fragments (transacciones)
    val transaction: FragmentTransaction = fragmentManager.beginTransaction()

    // Reemplaza el contenido del FragmentContainerView con el nuevo Fragment,
    // en función del botón pulsado
    if(blueIsClicked) {
        transaction.replace(R.id.fragmentContainerView, AzulFragment())
    } else {
        transaction.replace(R.id.fragmentContainerView, AmarilloFragment())
    }

    // Agrega la transacción a la pila hacia atrás (opcional) -> Útil si queremos deshacer acciones
    // y no cerrar directamente la aplicación. Permite utilizar el botón de retroceso del dispositivo
    // para deshacer la última transacción de fragmento
    transaction.addToBackStack( name: null)

    // Commit la transacción (confirma los cambios realizados)
    transaction.commit()
}
```

# EJERCICIOS

- **Ejercicio 1:** Crea una app con las especificaciones del ejemplo anterior.

# EJERCICIOS

- **Ejercicio 2:** Crea una app con un Fragment superior y otro inferior. El inferior contendrá los botones *Inicio* y *Sumador*.
- Al pulsar *Sumador*, se cargará en la parte superior otro Fragment que permitiría sumar dos números que introduzca el usuario.
- Al pulsar en *Inicio*, se cargará en el Fragment superior la pantalla de Bienvenida.

