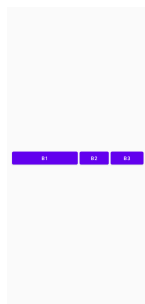


# Ejercicio resuelto 3 Botones

[Descargar estos apuntes](#)

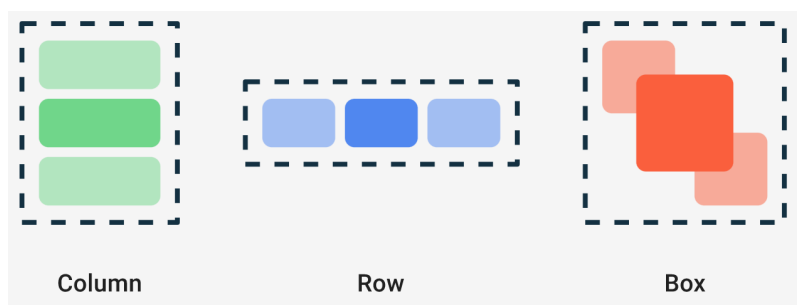
## Creando el proyecto

Queremo definir una función que nos permita visualizar tres botones en nuestra pantalla. Tal y como se ve en la siguiente imagen.



Como podemos observar existen tres botones, situados en el centro de la pantalla. El primero tiene un tamaño que es la mitad de la pantalla y los otros dos son la mitad que el primero, o ocupan cada uno la mitad del resto del tamaño de pantalla que queda libre.

Para poder crear una vista como la que se nos pide, es necesario comprender y saber utilizar la colección de contenedores de diseño básicos que ofrece **Compose** para organizar y controlar el diseño de tus elementos en pantalla: *Column*, *Row* y *Box*.



**Column** y **Row** son contenedores que organizan automáticamente sus elementos hijos en una dirección vertical u horizontal, mientras que **Box** es un contenedor que permite distribuir a los hijos en cualquier posición dentro de la pantalla.

¿Qué contenedor utilizaremos nosotros para nuestro ejemplo?

```

@Composable
fun TresBotones() {
    Row(
        modifier = Modifier
            .fillMaxSize()
            .padding(16.dp),
        verticalAlignment = Alignment.CenterVertically

    ) {
        //Vista
    }
}

```

#### Aclaraciones:

- *.fillMaxSize* ajusta el tamaño de **Row** al ancho y alto del contenedor padre, que es nuestra pantalla.
- *.padding* aplica un **padding** a la fila, desplazando su contenido respecto al borde del contenedor, el número de píxeles indicado.
- *verticalAlignment*, permite definir la alineación vertical de **Row** respecto a su contenedor, con el valor *Alignment.CenterVertically* alineamos verticalmente los elementos dentro de la fila en el centro de la pantalla

Añadimos el primer botón, para ello utilizaremos la función **Button** que nos ofrece

#### Jetpack compose

```

@Composable
fun TresBotones() {
    Row(
        modifier = Modifier
            .fillMaxSize()
            .padding(16.dp),
        verticalAlignment = Alignment.CenterVertically

    ) {
        Button(
            onClick = { /* acción al hacer clic */ },
            modifier = Modifier.fillMaxWidth(0.5f)
        ) {
            Text(text = "B1")
        }
    }
}

```

#### Aclaraciones:

- `.fillMaxWidth(0.5f)` ajusta el tamaño de **Button** para que su ancho sea la mitad del tamaño disponible.
- El parámetro **onClick** se utiliza para definir la acción que se realizará cuando se haga clic en el botón.
- Añadimos el contenido del botón entre las llaves (función lambda), podemos usar `Text(),Image()...`

Añadimos el segundo **Button**

```
@Composable
fun TresBotones() {
    Row(
        modifier = Modifier
            .fillMaxSize()
            .padding(16.dp),
        verticalAlignment = Alignment.CenterVertically
    ) {
        ...
        Button(
            onClick = { /* acción al hacer clic */ },
            modifier = Modifier
                .fillMaxWidth(0.5f)
                .padding(5.dp)
        ) {
            Text(text = "B2")
        }
    }
}
```

#### Aclaraciones:


- `.fillMaxWidth(0.5f)` ajusta el tamaño de **Button** para que su ancho sea la mitad del tamaño disponible, es decir, de la mitad del espacio que queda, coge la mitad.
- Colocamos un **padding** para separar los botones.

Finalmente definimos el tercer **Button** :

```

@Composable
fun TresBotones() {
    Row(
        modifier = Modifier
            .fillMaxSize()
            .padding(16.dp),
        verticalAlignment = Alignment.CenterVertically
    ) {
        ...
        Button(
            onClick = { /* acción al hacer clic */ },
            modifier = Modifier.fillMaxWidth(1.0f)
        ) {
            Text(text = "B3")
        }
    }
}

```

 **NOTA:** Estos contenedores se pueden combinar para lograr diseños más complejos y personalizados. Veremos más adelante ejemplos de **Column**, **Box** y combinaciones entre ellos.