

# EL COMPONENTE SURFACE

# EL COMPONENTE **SURFACE**

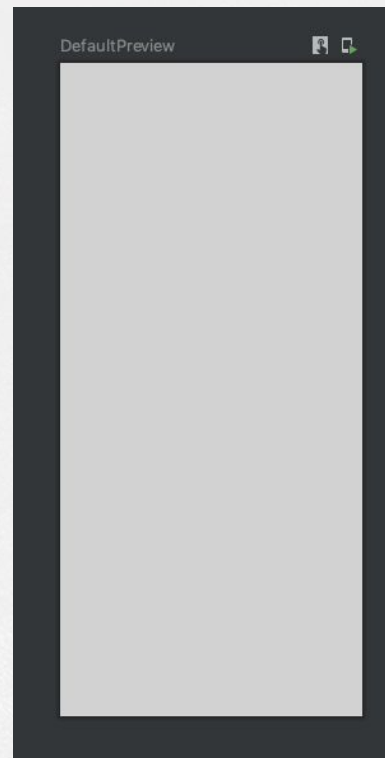
- **Surface** es uno de los componentes más sencillos de Jetpack Compose.
- Es un bloque de UI que podemos añadir a nuestra interfaz y que puede tener **color** y contener otros componentes.
- Podemos comparar el componente **Surface** con la clase **View** por simplicidad.

# EL COMPONENTE **SURFACE**

- **MainScreen:**
  - Nuevo componente **@Composable**.
  - Acepta **color** como argumento.
  - Si no le aplicamos modificadores no tendrá dimensiones y no podrá verse en la pantalla.
  - Solo acepta un componente hijo.

# EL COMPONENTE **SURFACE**

```
@Composable
fun MainScreen() {
    Surface(
        color = Color.LightGray,
        modifier = Modifier.fillMaxSize()
    ) {
    }
}
```



# EL COMPONENTE **SURFACE**

- Podemos añadir otros componentes dentro de **Surface**.
- **Text** utiliza **wrapContentSize** como modificador que indica que use solo el espacio necesario.

```
@Composable
fun MainScreen() {
    Surface(
        color = Color.LightGray,
        modifier = Modifier.fillMaxSize()
    ) {
        Text(
            text = "Hi Openwebinars!",
            style = MaterialTheme.typography.h5,
            modifier = Modifier.wrapContentSize()
        )
    }
}
```



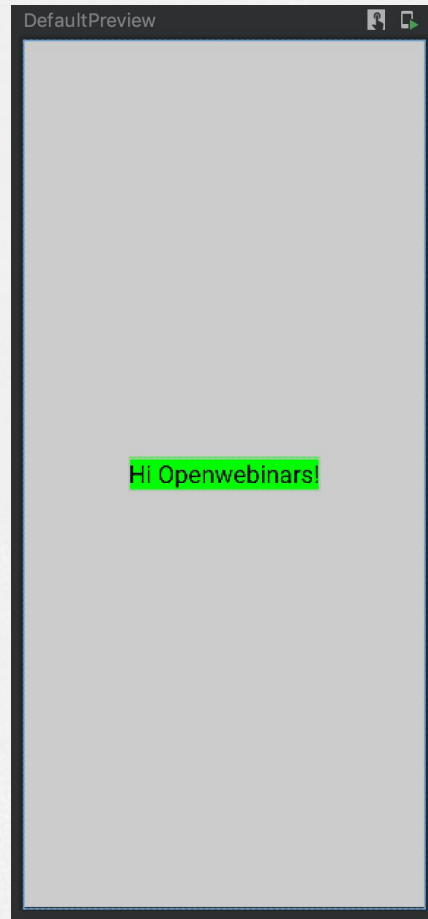
# EL COMPONENTE **SURFACE**

- También podemos anidar componentes **Surface**.
- De esta forma se puede establecer un background para el componente **Text**.
- **wrapContentSize** puede moverse ahora al componente padre.

```
@Composable
fun MainScreen() {
    Surface(
        color = Color.LightGray,
        modifier = Modifier.fillMaxSize()
    ) {
        Surface(
            color = Color.Green,
            modifier = Modifier.wrapContentSize()
        ) {
            Text(
                text = "Hi Openwebinars!",
                style = MaterialTheme.typography.h5
            )
        }
    }
}
```

# EL COMPONENTE **SURFACE**

- **Text** es situado en el centro de la pantalla de forma automática.
- **wrapContentSize** está aplicando una alineación automática: **Alignment.Center**



## EL COMPONENTE **SURFACE**

- La clase **Alignment** tiene multitud de valores para posicionar un componente dentro de su componente padre.
- Para este caso solo podremos elegir las que sean de clase **Alignment**.

▼ BottomCenter	Alignment
▼ BottomEnd	Alignment
▼ Center	Alignment
▼ BottomStart	Alignment
▼ Bottom	Alignment.Vertical
▼ CenterEnd	Alignment
▼ CenterStart	Alignment
▼ TopCenter	Alignment
▼ Top	Alignment.Vertical
▼ TopEnd	Alignment
▼ TopStart	Alignment
▼ CenterHorizontally	Alignment.Horizontal
▼ CenterVertically	Alignment.Vertical
▼ End	Alignment.Horizontal
▼ Start	Alignment.Horizontal

^↓ and ^↑ will move caret down and up in the editor [Next Tip](#)



# EL COMPONENTE **SURFACE**

```
@Composable
fun MainScreen() {
    Surface(
        color = Color.LightGray,
        modifier = Modifier.fillMaxSize()
    ) {
        Surface(
            color = Color.Green,
            modifier = Modifier.wrapContentSize(Alignment.TopEnd)
        ) {
            Text(
                text = "Hi Openwebinars!",
                style = MaterialTheme.typography.h5
            )
        }
    }
}
```



# EL COMPONENTE **SURFACE**

```
@Composable
fun MainScreen() {
    Surface(
        color = Color.LightGray,
        modifier = Modifier.fillMaxSize()
    ) {
        Surface(
            color = Color.Green,
            modifier = Modifier.wrapContentSize(Alignment.BottomStart)
        ) {
            Text(
                text = "Hi Openwebinars!",
                style = MaterialTheme.typography.h5
            )
        }
    }
}
```

