

PASO DE VARIABLES ENTRE @COMPOSABLES.

La Calculadora maneja el paso de variables de estado mediante dos técnicas distintas, aunque hay tres. Vamos a ver el funcionamiento de las tres

Las variables son pasadas de una función a otra por valor cuando son datos simples o Strings o por referencia cuando se trata de objetos o arrays. En las funciones Composables pasa lo mismo.

Si tengo una variable String como pantalla, no puedo pasarla de un Composable a otro y modificarla directamente al ser pasada como valor. ¿Qué hacer entonces? Uno de estos tres métodos.

1. Usamos una función Lambda dentro de una función de orden superior
2. No pasamos la variable directamente sino el tipo `MutableState<String>` ya que se trata de una variable de estado. Ahora ya pasamos una referencia al estado.
3. Envolvemos el dato simple en un objeto que sirve de envoltura, este ya sí se puede pasar por referencia de una función a otra

1) MEDIANTE FUNCIONES LAMBDA

En el proyecto CalculadoraJPC observamos que el display de la pantalla es un Text y que maneja una variable de estado llamada pantalla que no es más que un String donde aparecerán los números y otros mensajes de texto.

Lo primero que podemos ver es que esta variable de estado está definida dentro del @Composable Calculadora, que es el que está en el nivel más alto de todos los @Composables.

```
var pantalla:String by remember{mutableStateOf("0")}
```

Cuando generamos la botonera (Botonera2) vemos que se trata de una función de orden superior a la cual pasamos un parámetro y una función lambda:

```
@Composable
fun Calculadora(...){
    var pantalla:String by remember{mutableStateOf("0")}
    Botonera2(calculadora) {nuevaPantalla -> pantalla=nuevaPantalla }
}
```

Botonera2 es la función, le pasamos como parámetro el objeto calculadora y una función anónima (Lambda) que lo que hace es tomar un valor en la entrada y pasárselo a pantalla. Esto se realiza allí donde está definida la función Botonera2:

```
fun Botonera2(calculadora:Calculadora, setPantalla:(String)->Unit) {
    .....
    for (j in 1..3) {
        val s=n[id]
        Button(onClick = {
            gestionBotones(s, calculadora){setPantalla(""+it)}
        })
        {Text(text = s,fontSize = 24.sp)}
```

```

    }
    id++
}
.....
}

```

setPantalla es el sitio por donde “se introducirá” la función Lambda: `{nuevaPantalla -> pantalla = nuevaPantalla }`. **it** es el valor que viene de gestiónBotones y se introduce en “nuevaPantalla”

De esta forma que estoy haciendo? Desde Botonera2 estoy cambiando el valor de la variable pantalla.

Y .. ¿Por qué no puedo cambiar desde dentro de Botonera2 el valor de pantalla?? ¿Porqué no la paso por argumento como hice con calculadora?? Algo así :

```

fun Botonera2(calculadora:Calculadora, pantalla:String){
    ....
    Button(onClick = {
        gestionBotones(s, calculadora){pantalla="" + it} NO FUNCIONA
    })
    {Text(text = s,fontSize = 24.sp)
    }
    ....
}

```

La razón es simplemente porque pantalla es una variable simple y cuando se intenta pasar por parámetro a una función, ésta se pasa **POR VALOR**. Por lo que no estamos modificando la variable pantalla realmente sino solo una copia de la misma.

2) PASAR DIRECTAMENTE MutableState<String>

Hay que cambiar el parámetro de entrada de la pantalla allí donde se necesite. Por ejemplo en:

```

fun Botonera2(calculadora:Calculadora, pantalla:MutableState<String>){
    ....
    Button(onClick = {
        //Se debe acceder al dato con .value
        gestionBotones(s, calculadora){pantalla.value="" + it}
    })
    {Text(text = s,fontSize = 24.sp)
    }
    ....
}

```

Para que esto funcione en el Composable Calculadora hay que hacer los siguientes cambios:

- **val** pantalla = remember { mutableStateOf("0") }
//Ahora pantalla no es de tipo String sino un mutableState
- En la etiqueta tb se accede al dato con .value `Text(text = pantalla.value...`

3) PASAR DATOS CON OBJETOS DE ENVOLTURA.

Si envolvemos los datos en un objeto, entonces los objetos se pasan por referencia y por tanto podremos manipular el original desde la función llamada.

1. Primero creamos la envoltura:

```
class EstadoPantalla() {  
    var pantalla by mutableStateOf("0")  
}
```

2. Creamos la variable de estado

```
var estadoPantalla = remember {EstadoPantalla() }
```

3. Ahora desde el Composable principal podemos pasar las variables como referencia ya que son objetos

```
Botonera3(calculadora, estadoPantalla)  
BotonesOperaciones3(calculadora,estadoPantalla)
```

4. Los parámetros son del tipo EstadoPantalla. Por ejemplo:

```
fun Botonera3(calculadora:Calculadora, estadoPantalla:EstadoPantalla) {..}
```

Una vez dentro se usa: estado.pantalla="dato"