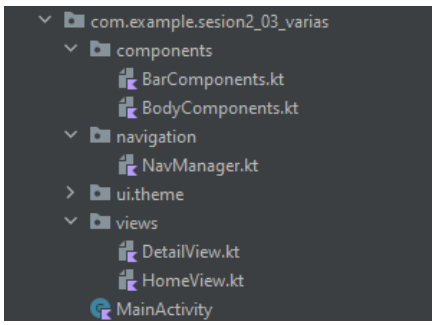
Sesión 2-03 Clase del 23 de octubre

En esta sesión aprenderemos a trabajar con el componente Scaffold que nos facilita crear una estructura de una pantalla y crearemos una organización adecuada de un proyecto en el que va a haber varias pantallas.

1.- Creamos la siguiente estructura de proyecto y explicamos:



2.- Preparamos código inicial en pantallas **Home** y **Detail.** Por ejemplo, para **Detail**:

```
@Composable
fun DetailView(){

}

@Composable
fun ContentDetailView(){

}
```

3.- En **ContentHomeView** escribe código para crear una columna que, de momento, sólo incluye un texto "**Zona de ContentHomeView**" con tamaño 20 y totalmente centrado en pantalla.

4.- Ahora, en **ContentView** añadimos un componente **Scaffold** para organizar la distribución de la pantalla en:

- Top Bar
- Container
- Bottom Bar
- Floating Action Button

```
@Composable
public fun Scaffold(
    modifier: Modifier = Modifier,
    topBar: @Composable () -> Unit = {},
    bottomBar: @Composable () -> Unit = {},
    snackbarHost: @Composable () -> Unit = {},
    floatingActionButton: @Composable () -> Unit = {},
    floatingActionButtonPosition: FabPosition = FabPosition.End,
    containerColor: Color = MaterialTheme.colorScheme.background,
    contentColor: Color = contentColorFor(containerColor),
    contentWindowInsets: WindowInsets = ScaffoldDefaults.contentWindowInsets,
    content: @Composable (PaddingValues) -> Unit
): Unit
```

```
fun HomeView(){
    Scaffold (

    ){

    }
}
```

De momento nos da un error de **paddingValues.** Estos valores garantizan que los contenidos de **Scaffold** estén correctamente alineados. De momento, no consideramos el error usando en la función composable la anotación:

```
@SuppressLint("UnusedMaterial3ScaffoldPaddingParameter")
```

5.- Ahora, iremos dando forma a cada una de las partes de **Scaffold**. En primer lugar, dentro del scope (container) de **Scaffold**, hacemos una llamada a que se pinte la función **ContentHomeVideo**. Probamos la funcionalidad.

Antes tendremos que cargar **HomeView** desde **MainActivity**.

6.- Dentro de las propiedades de **Scaffold**, asignaremos una lambda a **topBar** para dar forma a la **Top Bar** de la pantalla. Pintaremos, de momento un texto “TOP BAR” de tamaño 25.

```
topBar = {
    CenterAlignedTopAppBar(
       title={Text(text="TOP BAR",
            fontSize = 25.sp)})
    //Pueden usarse otras funciones de composición como SmallTopAppBar
 }
```

Al incluir este código, se pide que incluyamos una anotación por estar esto en fase experimental:

```
@OptIn(ExperimentalMaterial3Api::class)
```

Pero, esto hecho, para nada tiene apariencia de una **Top App Bar**.

Hacemos que el texto se pinte de blanco y el fondo de **Top Bar** en azul.

```
topBar = {
    CenterAlignedTopAppBar(
       title={Text(text="TOP BAR",
            fontSize = 25.sp, color = Color.White)},
        colors=TopAppBarDefaults.centerAlignedTopAppBarColors(
            containerColor =  Color.Blue))
    //Pueden usarse otras funciones de composición como SmallTopAppBar
 }
```

7.- Usamos colores del tema para la **top bar**.

```
topBar = {
    CenterAlignedTopAppBar(
       title={Text(text="TOP BAR",
            fontSize = 25.sp, color = MaterialTheme.colorScheme.primaryContainer)},
        colors=TopAppBarDefaults.centerAlignedTopAppBarColors(
            containerColor =  MaterialTheme.colorScheme.onPrimaryContainer))
    //Pueden usarse otras funciones de composición como SmallTopAppBar
 }
```

8.- Utilizamos la estructura creada de proyecto. En **BarComponents** agregamos un componente **BarTitle** para componer el texto de la **Top Bar**.

```
@Composable
fun TitleBar(texto:String){
    Text(text=texto,
        fontSize = 25.sp,
        color = MaterialTheme.colorScheme.primaryContainer)
}
```

Y sustituimos la composición del **Text** en **topBar** de **Scaffold** por:

```
topBar = {
     CenterAlignedTopAppBar(
        title={ TitleBar(texto = "TOP BAR")},
         colors=TopAppBarDefaults.centerAlignedTopAppBarColors(
             containerColor =  MaterialTheme.colorScheme.onPrimaryContainer))
   }
```