

APUNTES: USO DE UN Navigation Drawer

Antes llamado Navigation Drawer.

LIBRERÍAS: MATERIAL 3

OBJETIVOS: Enlazar una navegación simple entre pantallas con el Side Sheet, pero también reutilizando un componente TopAppBar (Barra de menú)

El funcionamiento del componente está en tres lugares:

1. El componente ModalNavigationDrawer. Es el componente que se abre y cierra como una persiana y que está definido en el Activity. Consta de varias partes:
 1. Variables de estado necesarias para el funcionamiento correcto incluída también la variable de estado para la navegación.
 2. El componente ModalNavigationDrawer que es llamado.
 1. Parte de configuración:
 1. drawerState que almacena si está abierto/cerrado
 2. drawerContent. Está definido en un archivo aparte.
 1. Scope.launch es una corrutina que es la que se encarga de abrir o cerrar el componente.
 2. Tiene una función lambda con un parámetro de entrada **route** para la navegación en navigationController.
 2. Parte de diseño:
 1. con el Composable Scaffold admite una topBar y dentro llamamos a nuestra TopBar personalizada
 3. Parte de navegación: Se define el NavHost para ser usado en la navegación entre pantallas.
 2. El DrawerContent. Es el que define el contenido.
 3. En la barra de menú hay un icono de menú que al ser pulsado desplegará el componente.

1º En el Activity:

```
class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()

        setContent {
            val navigationController = rememberNavController()
            val drawerState = rememberDrawerState(initialValue = DrawerValue.Closed)
            val scope = rememberCoroutineScope()

            ModalNavigationDrawer(
                drawerState = drawerState,
                drawerContent = {
                    DrawerContent {
                        route -> scope.launch { drawerState.close() }
                        navigationController.navigate(route)
                    }
                }
            ) {
                Scaffold(
                    topBar = {
                        MiTopAppBar(onMenuClick = { scope.launch {
                            drawerState.open() } })
                    }
                )
            }
        }
    }
}
```


En este archivo se define un `NavigationItem` para usarlo en una lista y cargarlo en el `ModalDrawerSheet`

El `DrawerContent` tiene definida una función Lambda: `onDestinationClicked: (String) -> Unit` es llamada en `onClick = { onDestinationClicked(item.route) }` de `NavigationDrawerItem`. Pasa el parametro `route` de `NavigationItem` a la función lambda que lo colocará en `navigationController.navigate(route)` desde el `Activity`.

2° `MiTopAppBar`.

No es parte del `Drawer` pero se usará el botón de menú para abrirlo por lo que comentamos aquí:

```
@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun MiTopAppBar(onMenuClick: () -> Unit) {
    TopAppBar(
        navigationIcon = {
            IconButton(onClick = onMenuClick ) {
                Icon(imageVector = Icons.Rounded.Menu, contentDescription =
null)
            }
        },
        title = { Text(text = "Gestión de alimentos") },
        actions = {
            IconButton(onClick = { /*TODO*/ } ) {
                Icon(
                    imageVector = Icons.Rounded.Search,
                    contentDescription = null
                )
            }
            IconButton(onClick = { /*TODO*/ }) {
                Icon(
                    imageVector = Icons.Outlined.AccountCircle,
                    contentDescription = null
                )
            }
        }
    )
}
```

En `fun MiTopAppBar(onMenuClick: () -> Unit)` Se define una función Lambda que se usa luego aquí:

```
IconButton(onClick = onMenuClick )
```

Al ser pulsado el icono se ejecuta `onClick` que llama a `onMenuClick`. Esta función Lambda está definida en:

```
topBar = { MiTopAppBar(onMenuClick = { scope.launch { drawerState.open() } })
IconButton(onClick = onMenuClick ) {
    Icon(imageVector = Icons.Rounded.Menu, contentDescription = null)
}
}
```

En el `Activity`. En esa función se implementa que el `Drawer` se abrirá o cerrará