

# Trabajando con Scaffold

[Descargar estos apuntes](#)

## Ejercicio

En este ejercicio vamos a familiarizarnos con el componente que facilita la estructuración de nuestra aplicación, y a la vez utilizar los distintos elementos para la navegación y acciones que permite nuestra App.

Para facilitar el desarrollo, vamos a realizar todo el ejercicio usando las preview, sin preocuparnos de las elevaciones de estado a los ViewModel.

Tampoco vamos a implementar las funcionalidades de las acciones ni las de la navegación, solo nos centraremos en la parte de la composición visual.

## Scaffold, TopAppBar y BottomAppBar

Primero nos crearemos una función AppBarScreen que va a contener las dos AppBars tanto superior como inferior, y que estarán añadidas en los Slots correspondientes del componente Scaffold.

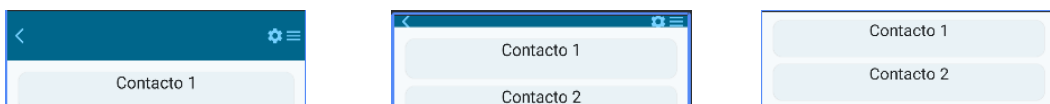
Los iconos usados en la **TopAppBar** son:

`ArrowBackIosNew`, `Settings` y `Menu`

Mientras que los usados en la **BottomAppBar** son: `Edit`, `Delete` y `Add` para el botón flotante

Como contenido principal tendremos un **Surface** con un **LazyColumn** que incluya un **ElevatedCard** con el texto como se ve en la imagen.

Además vamos a hacer que la **TopAppBar** interactúe con el scroll de la lista, de forma que cuando se haga scroll desaparezca, volviendo visible al volver al primer elemento. Deberás probar cual es el valor de **TopAppBarDefaults** que permite esta acción.



👉 Como vamos a reutilizar algunos de los componentes en otras partes de la aplicación, se recomienda crear el paquete components dentro de features y añadir aquí el código para la

TopAppBar y el de BottomAppBar.

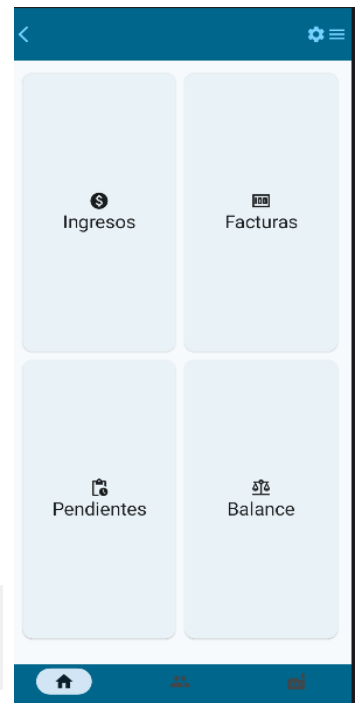
## Scaffold, NavigationBar

En el mismo proyecto nos crearemos otro fichero con una función `NavigationBarScreen` que va a contener la misma barra de aplicación superior del screen anterior (por eso lo de meter el código en el paquete `componets`), pero ahora la barra inferior será una **NavigationBar**. Los iconos son: `Home`, `People` y `Factory`

Como contenido principal del Scaffold usaremos un **Surface** con un **LazyVerticalGrid** que distribuya los **ElevatedCard**. Los iconos:

`Paid`, `Money`, `PendingActions`, `Balance`

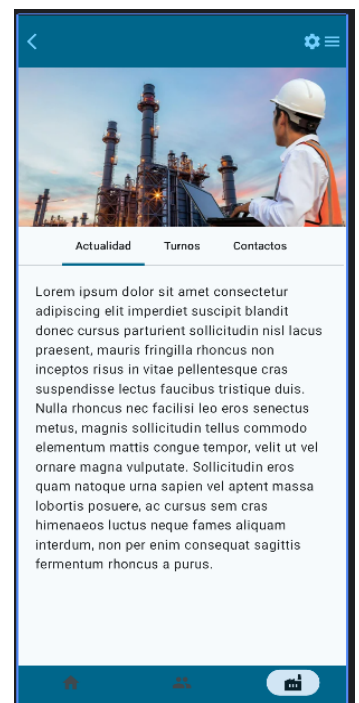
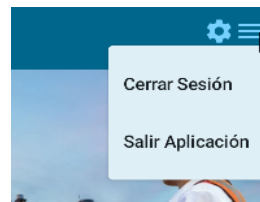
💡 Sería más optimo que crearas una colección para añadir el contenido de la Card, puedes crear una data class para ello.



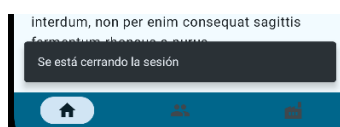
## Scaffold, Tabs, DropdownMenu y SnackBar

La siguiente pantalla va a incluir un Tab en el cuerpo principal del Scaffold, para ello copiaremos la función `NavigationBarScreen` en otro fichero y le cambiaremos el nombre a `TabScreen`. Debemos añadir un tab del tipo **SecondaryScrollableTabRow** dentro de un **Column** que contendrá como primer elemento un **Image**. El elemento que mostrará el contenido del Tab, será un Surface que por ahora tendrá un texto largo para mostrar las noticias de actualidad.

Ahora añadiremos un Menú desplegable **DropdownMenu**, a la opción de menú de la TopAppBar para que se muestre de la siguiente manera, controlaremos que se muestre y oculte de forma correcta.

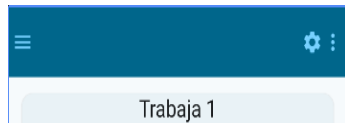


Para acabar con esta parte del ejercicio añadiremos al Scaffold un **SnackBar** con el mensaje "Saliendo de la aplicación" o "Se está cerrando la sesión", dependiendo de la opción del menú pulsada.



## Scaffold y Navigation Drawer

Para acabar vamos a copiar la parte del ejercicio que incluye las dos AppBar, que en un principio debería estar en la función AppBarScreen y lo vamos a renombrar a NavigationDrawerScreen. Como vamos a incluir un menú lateral, vamos a cambiar los iconos de la TopAppBar de la siguiente manera, ahora estará asignado a navigationIcon el `Menu` y a la action de la derecha `MoreVert` :



Crearemos la estructura necesaria para añadir el menú, recuerda que en este caso el elemento superior será **ModalNavigationDrawer** que contendrá **ModalDrawerSheet** en el Slot `drawerContent` y **Scaffold** con todo su contenido en la ranura `content` .

Por su parte, la imagen la podemos añadir en el content del ModalDrawerSheet, que tiene un scope ColumnScope, antes de incluir los **NavigationDrawerItem**.



💡 **Tips:** La línea se ha conseguido usando un HorizontalDivider.