

# Ejercicio de Navegación con menú lateral

[Descargar estos apuntes](#)

## Ejercicio

Lo primero que debemos hacer es descargar el [código](#) para añadir a un proyecto base. Como se puede ver, contiene:

- el paquete mocks con sus componentes y el repository, este no debe estar dentro de mocks, sino de deberás alojarlo en el lugar que le corresponde según la arquitectura que ya conocemos.
- el modelo
- y además de un paquete componets con los elementos que necesitaremos y que ya se han codificado en ejercicios anteriores.

El menú lateral y la lista se puede recuperar del ejercicio del tema 4.1 o volver a codificarlo, para reafirmar conocimientos.

Este ejercicio está relacionado con la última parte del tema 4.2 de *Integrando navegación y NavigationBar*, ya que deberemos crearnos un **ContactosScreen** principal que incluirá el Scaffold con ambas AppBars, la snackBar mientras que **en el slot del contenido principal implementará un NavHost que será el contenedor de ListadoScreen o de TrabajoScreen** según el caso. Para entender mejor el funcionamiento, puedes descargar el siguiente [vídeo](#).

En este caso, como queremos que la aplicación sea lo más real posible, se deberán alojar los estados en los viewModels correspondientes, también se pasan como [recurso](#) para agilizar la implementación.

Por tanto, para desarrollar quedaría las pantallas:

- ListadoScreen, que muestra la lista de todos los contactos o filtrada por el item pulsado en el menú lateral, le corresponde ListadoContactosViewModel.
- TrabajoScreen, que mostrará una pantalla con la imagen y los Taps del ejercicio del tema 4.1, y que pasará a ser el contenido del NavHost cuando se pulse un contacto **del filtro de trabajo**, el resto de filtros no tienen pantalla asociada (por lo que pulsar

en otro tipo de contactos no haría efecto). Observa que cuando se visualiza esta pantalla deja de mostrarse la barra inferior, ya que no tendría utilidad. Además el nombre del trabajador pulsado, se muestra en la parte superior de la imagen, habrá que pasarlo con un parámetro de navegación. A esta screen se le asocia TrabajoViewModel.

- ContactosScreen como hemos dicho antes, contiene toda la pantalla de aplicación, las AppBar, Menús y NavHost para el contenido principal. Tiene asociado también el ListadoContactosViewModel.

Al pulsar sobre la opción Salir Aplicación, del menú desplegable, se cerrará la aplicación. Para ello usaremos el siguiente código:

```
val activity = LocalContext.current as? Activity
activity?.finish()
//tendrás que situarlo como corresponda
```



Además como hemos visto en los apuntes, deberemos crear el paquete navigation para incluir todo lo referente a la navegación, se quedarán los archivos:

- ContactosNavHost
- ListadoRoute
- TrabajoRoute