

# Ejercicio 2

Descargar este enunciado [pdf](#) o [html](#)

Partiendo del [Ejercicio 1](#), se pide realizar las siguientes tareas:

1. Basándote en el componente `ItemListaCoches.kt` proporcionado, define la pantalla con la lista de coches en `GaleriaCochesScreen.kt` y su correspondiente `GaleriaCochesViewModel.kt` que gestione el filtrado usando Room, así como manteniendo los estados y gestionando los eventos de `GaleriaCochesScreen` en el paquete `.galeriacoches`.



## Aviso

La lista **NO debe contener** ningún tipo de animación al añadir o borrar elemento de la misma.

	Volkswagen Golf 2021	29000 €	
	Ford Focus 2022	28500 € antes 30000 €	5%
	Audi A3 2023	32000 €	
	Ford Mondeo 2021	28800 € antes 32000 €	10%
	Mercedes Clase A 2021	38200 €	
	Ford Kuga 2022	38250 €	
	Volkswagen Tiguan 2021	39000 €	

2. La lista de coches debe estar **contenida en un Scaffold** con un `CenterAlignedTopAppBar` que **colapse al hacer scroll** en la lista, un `BottomAppBar` que comentaremos en el siguiente punto y un **FAB** con la imagen `Filled.getFilterListIcon()` que me permitirá filtrar la lista de coches.

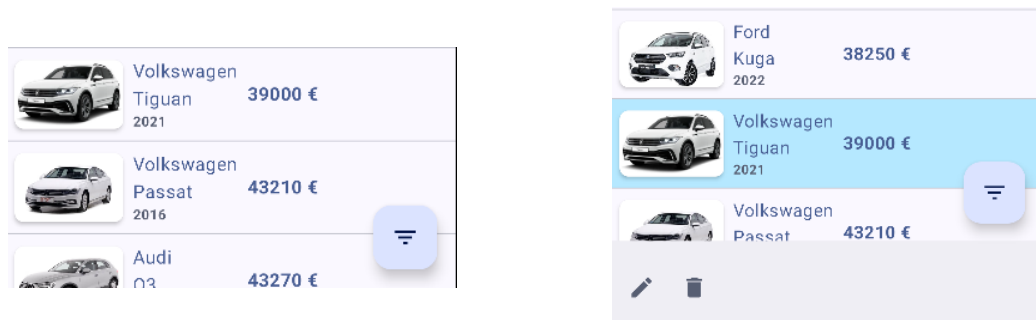
Como puedes ver en las imágenes de ejemplo abajo. Al pulsar el FAB se debe mostrar un `ModalBottomSheet` con:

- Un primer `SingleChoiceSegmentedButtonRow` para ordenar los coches por precio, modelo o año. Crea un `enum class Ordenacion` con los valores `PRECIO`, `MODELO` y `AÑO` para gestionar su visualización y guardar su estado en el VM. Puesto que, es visualizar los mismos coches en otro orden, no es necesario hacer una nueva petición al repositorio y utilizaremos el método `sortedBy` para cambiar el orden de la lista de coches.
- Un `HorizontalDivider`.
- Un `FilterChipWithIcon` para mostrar ver **solo las ofertas o todos los coches** que gestionaremos mediante una petición al repositorio.



3. Cuando seleccionemos una coche aparecerá una **BottomAppBar** en el Scaffold que debe ofrecer dos botones de acción:

- Uno para **ver la ficha del coche seleccionado** con el icono `painter = Filled.getEditIcon()`. De momento, no hará nada. Pero, en futuros ejercicios, nos llevará a la pantalla de edición de coche.
- Otro para **eliminar el coche seleccionado** con el icono `painter = Filled.getDeleteIcon()`.



4. Tanto las ordenaciones como el filtrado de ver solo ofertas se realizarán en el **GaleriaCochesViewModel.kt** llamando a los métodos correspondientes del repositorio que usarán las operaciones definidas en la BD con Room. Para ello deberás definir clase **GaleriaCochesEvent** que será un 'Sum-Type' con los siguientes eventos:

```
sealed interface GaleriaCochesEvent {
    data class OnSeleccionarCoche(val coche: CocheUiState) : GaleriaCochesEvent
    object OnVerSoloOfertas : GaleriaCochesEvent
    data class OnOrdenacion(val ordenacion: Ordenacion) : GaleriaCochesEvent
    object OnBorrarVehiculo : GaleriaCochesEvent
}
```



## Pista

Puedes definir el método privado ...

```
private suspend fun getCohes() : List<CocheUiState>
```

en el `GaleriaCochesViewModel.kt` que te devuelva la lista de coches ordenada y filtrada si fuera necesario usando los estados en el `ViewModel` .

```
var ordenacionState by mutableStateOf(Ordenacion.PRECIO)
    private set
var verSoloOfertas by mutableStateOf(false)
    private set
```