Ejercicio 4

Descargar este enunciado pdf o html

Partiendo del **Ejercicio 3**, se pide realizar las siguientes tareas:

Especificaciones

1. Crea un BD con Room para almacenar todos los datos de nuestro modelo denominada "coches.db". La BD debe contener una única tabla denominada coches donde la clave primaria será el campo id de tipo Int que tienes en el modelo. Los campos de las tabla deben ir en snake_casing

Las operaciones sobre la tablas serán todas asíncronas y debes definir las siguientes:

- Operaciones básicas insert , delete , update y count
- get(): Devuelve todos los coches de la base de datos.
- get(Int): Devuelve un coche de la base de datos a partir de su id.
- getOertas(): Devuelve todos los coches de la base de datos que tengan un porcentaje de descuento mayor que 0. SELECT * FROM coches WHERE porcentaje_descuento > 0
- 2. Añade las anotaciones de Hilt en la aplicación y prepara los '*providers*' que consideres necesarios de las instancias de Room para la inyección de dependencias con **Hilt**. Teniendo en cuenta que se creará una **instancia única** de cada uno de ellos.
- 3. Modifica el 'repositorio' para usar los métodos definidos.
- 4. Realiza la primera carga de datos de la base de datos de Room a partir de los datos CocheDaoMock . Solo si la BD está vacía.

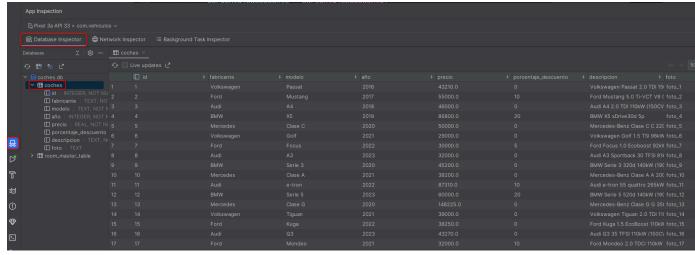
Para ello, realiza la carga de datos en la BD al iniciar la aplicación. Utilizando el método onCreate de la clase VehiculosApplication .



Nota

Recuerda que los métodos de Room deben ser asíncronos y que debes tener en cuenta que la carga de datos se realiza solo si la BD está vacía.

5. Comprueba con el App Inspection que la carga de datos se ha realizado correctamente.



6. Por último, deberás controlar que los datos que usas en los ViewModels de todas las screen de la aplicación son extraídos de Room, en ningún caso deberá hacer referencia a los datos mooks anteriores.