Informe de pruebas – US399114-OrdenarPorPrecio

PRUEBAS DE ACEPTACIÓN (Víctor Martínez Vila)

Cada una de las pruebas de aceptación definidas en el plan de pruebas de la historia de usuario "Ordenar por precio", a excepción del último escenario (fallo en el servidor remoto de datos), se llevaron a cabo en dos entornos de ejecución.

El primero fue el emulador de Android Studio, utilizando un medio virtual Pixel 2, mientras que el segundo fue un teléfono móvil particular modelo Xiaomi Redmi Note 4. En ambos las pruebas definidas se pasaron correctamente a la primera, y no hubo necesidad de retocar el código.

PRUEBAS UNITARIAS

- Clase PresenterGasolineras.java

  - Método cargaDatosRemotos(String direccion) -> (Víctor Martínez Vila)

Se llevaron a cabo los dos casos de prueba definidos, a saber, pasar una dirección nula al método de carga de datos y una dirección válida. En el primer caso, el test pasó correctamente, pero en el segundo hubo un error en el que se notificó que se debía utilizar un Mock, dado que se ejecutaba un método de la clase estática Log desconocida para la clase. Sin embargo, esto se debía a la no inclusión de una aserción, por lo que al introducirla el fallo se vio subsanado y el test pasó correctamente.

  - Método ordenarPorPrecioMenorAMayor(Filtro filtro) -> (Víctor Martínez Vila):

Inicialmente, no se definieron las dos excepciones propias de la clase, a saber, CombustiblesInvalidos y OrdenacionNoValida, que debían saltar cuando la lista de combustibles pasada como parámetro era incorrecta (es decir, no existían esos combustibles) y cuando se proporcionaba un string de ordenación no válido (no coincidía con los dos definidos, mayor a menor y menor a mayor), respectivamente.

Al principio se trató de solventar esta situación transformando el método a booleano, pero al existir un segundo caso de error, se consideró incorrecto que se retornara false para ambos casos, ya que así no se sabría qué ha fallado en el test.

Por tanto, se añadieron las dos excepciones y los test pasaron correctamente.

  - Método ordenarPorPrecioMayorAMenor(Filtro filtro) -> (Víctor Martínez Vila):

Dado que se comenzó a probar por el método anterior, al darnos cuenta de que eran necesarias dos excepciones propias para pulir los casos de prueba, se hizo uso de las mismas también para este método.

Como las pruebas definidas eran las mismas que para el método anterior, con la diferencia de que se cambiaba el string de ordenación y la comprobación del orden del array de gasolineras local (en el método anterior se comprobaba que estaban ordenadas de menor a mayor, en este se debía hacer la comprobación contraria), los test pasaron a la primera correctamente.

- Método ordenarPorPrecio(Filtro filtro) -> Víctor Martínez Vila

Dado que las pruebas de los dos métodos anteriores se implementaron antes que las de este, se tuvo en cuenta tanto las excepciones propias definidas como los casos de prueba, ya que este método es similar en cuanto a funcionamiento.

PRUEBAS DE INTERFAZ DE USUARIO (Roberto González Jiménez)

Las pruebas de interfaz de usuario se han realizado sobre la prueba de ordenar “Con un único combustible de mayor a menor” que se ha especificado previamente en el plan de pruebas, pero también se ha comprobado que lo puede ordenar de menor a mayor. Se ha programado en la clase OrdenarPorPrecioUITest, dentro del paquete llamado androidTest, donde se encuentran las distintas pruebas de interfaz de usuario. Para la realización de estas se ha empleado el framework de Google Espresso. Para que funcione correctamente se han añadido las dependencias correspondientes al fichero build.gradle.

En cuanto a la implementación, no he tenido muchas dificultades ya que me siento bastante cómodo con este software, pero sí que he tenido momentos en los que me he trabado un poco, por ejemplo a la hora de realizar el scroll para ver las gasolineras me falló por poner una id incorrecta, también a la hora de ver las gasolineras, no daba casi tiempo a mirar como estaban los precios, por lo que tuve que añadir el método Thread.sleep() para que hiciera una breve pausa, en este caso eran unas cuantas pausas de 2 segundos cada una. Por lo demás no hubo ningún problema.

Una vez resuelto, el test ejecutaba correctamente, por lo que la implementación de la prueba de interfaz de usuario ha resultado exitosa.