



Plan de pruebas del caso “Buscar gasolinera por rango de precios”

Los niveles de prueba que se van a aplicar son los siguientes:

- Pruebas de aceptación. Las pruebas de aceptación se definirán siguiendo una estrategia basada en la historia de usuario y se ejecutarán de forma manual.
- Pruebas de integración. La estrategia para la definición del orden de las pruebas de integración será jerárquica. Se probará:
 - La integración entre la capa de negocio y la de persistencia. En este caso, para la definición de los casos de prueba se utilizarán técnica de métodos y caja negra y se utilizará Junit y Roboelectric.
- Pruebas unitarias. Se utilizará la técnica de prueba de métodos, usando técnicas de caja negra (partición equivalente y AVL) para la definición de los casos de prueba de cada método de cada clase o componente. Será necesaria la utilización de JUnit, Mockito y Espresso.
- Pruebas de interfaz. Las pruebas de interfaz se definirán siguiendo los criterios de las pruebas de aceptación. Será necesario utilizar JUnit y Mockito.

A continuación, se muestra una especificación detallada de los casos de prueba a aplicar en cada nivel mencionado anteriormente.

PRUEBAS DE ACEPTACIÓN

En base al caso de uso que se trata, se muestran las pruebas de aceptación correspondientes:

A1: CU: Buscar gasolinera por rango de precios

- Éxito. Filtro establecido
- Éxito. Filtro restablecido.
- Error. Búsqueda no válida (caída de red).
- Error. Búsqueda no válida (fallo en el repositorio de datos del ministerio).
- Error. Búsqueda no válida (inexistencia de gasolineras con los filtros seleccionados).

Los casos de prueba definidos para cada uno de estos escenarios, suponiendo que el estado inicial del sistema es el reflejado en el repositorio de datos del ministerio, son los que se muestran en la Tabla 1. Los casos de prueba deberían ser ejecutados en el orden indicado, para que el estado final del sistema sea igual al inicial.

Tabla 1. Casos de prueba de aceptación

Identificador	Entrada	Resultado
A1.a	El usuario introduce un valor acorde con el precio del carburante y selecciona aplicar filtros.	Se muestra una lista de gasolineras. CEPSA Carretera 6316 KM, 10,5 Gasolina: 1,859 Diesel: 1,999



		REPSOL CR N-629 79,7 Gasolina: 1,819 Diesel:2,009 ...
A1.b	El usuario selecciona la cruz que se encuentra en el menú ListaGasolineras	Se restablece el filtro de precio anteriormente establecido.
A1.c	El usuario introduce un valor acorde con el precio del carburante y selecciona aplicar filtros.	Se muestra el siguiente mensaje de error. "Conéctese a la red y vuelva a intentarlo más tarde."
A1.d	El usuario introduce un valor acorde con el precio del carburante y selecciona aplicar filtros.	Se muestra el siguiente mensaje de error. "No se puede conectar con la base de datos. Inténtelo de nuevo más tarde".
A1.e	El usuario introduce un valor demasiado bajo para el precio del carburante y selecciona aplicar filtros.	Se muestra el siguiente mensaje de error. "No se encuentra ninguna gasolinera con un precio menor al establecido".

PRUEBAS DE INTEGRACIÓN (Jesús Revuelta)

El orden de las pruebas y los casos de prueba a realizar serían los siguientes:

1. MainPresenter con GasolineraRepository. Estas pruebas coincidirán con las pruebas unitarias de dominio de la clase MainPresenter, definidas como UGIC.x en la sección unitarias, aquí renombrados como IGIC.x.
2. FiltrarPrecioPresenter con Prefs. Estas pruebas coincidirán con las pruebas unitarias de dominio de la clase FiltrarPrecioPresenter, definidas como UGIC.x en la sección unitarias, aquí renombrados como IGIC.x.

En este ejemplo, la prueba de la integración de los componentes DAO con la BBDD no tiene sentido pues se implementa con un fichero.

PRUEBAS UNITARIAS (Pedro Berrío)

Pruebas unitarias de la capa de negocio

Para poder llevar a cabo estas pruebas, será necesario el uso de objetos Mock para la interfaz IPrefs. Se aplica prueba de métodos, siendo los casos de prueba definidos para cada método los siguientes.

Se aplican pruebas unitarias de la capa de negocio para dos clases: FiltrarPrecioPresenter (nueva clase creada) y MainPresenter, ya existente pero en la que se han añadido distintos métodos.



Para las pruebas de la clase FiltrarPrecioPresenter, se utilizarán mocks de la interfaz IPrefs. Los casos de prueba definidos son los siguientes.

- Método estableceRango()

Identificador	Entrada	Valor esperado
UGIC.1a	"2,04"	-

- Método subePrecio()

Identificador	Entrada	Valor esperado
UGIC.2a	"2,04"	"2,05"

- Método bajaPrecio()

Identificador	Entrada	Valor esperado
UGIC.3a	"2,04"	"2,03"
UGIC.3b	"0,00"	"0,00"

Para las pruebas de MainPresenter será necesario utilizar mocks de la clase IGasolineraRepository. Los casos de prueba definidos son los siguientes.

- Método filtraPrecio()

Identificador	Entrada	Valor esperado
UGIC.4a	Lista de gasolineras con elementos [{"CEPSA","39526","CARRETERA 6316 KM 10,5","Alfoz de Lloredo","L-D: 08:00-21:00","1,999","1,859","06"}, ...] "2,04"	Lista de gasolineras con todos los elementos [{"CEPSA","39526","CARRETERA 6316 KM 10,5","Alfoz de Lloredo","L-D: 08:00- 21:00","1,999","1,859","06"}, ...]
UGIC.4b	Lista de gasolineras con elementos [{"CEPSA","39526","CARRETERA 6316 KM 10,5","Alfoz de Lloredo","L-D: 08:00-21:00","1,999","1,859","06"}, ...] "2,00"	Lista con algunos elementos Lista de gasolineras con elementos [{"CEPSA","39526","CARRETERA 6316 KM 10,5","Alfoz de Lloredo","L-D: 08:00- 21:00","1,999","1,859","06"}, ...]
UGIC.4c	[{"CEPSA","39526","CARRETERA 6316 KM 10,5","Alfoz de Lloredo","L-D: 08:00-21:00","1,999","1,859","06"}, ...] "" "-2,04"	Lista de gasolineras vacía []



UGIC.4d	Lista de gasolineras vacía [] "2,04"	Lista de gasolineras vacía []
---------	--	----------------------------------

- Método maximoEntreTodas()

Identificador	Entrada	Valor esperado
UGIC.5a	Lista de gasolineras con elementos [{"CEPSA","39526","CARRETERA 6316 KM 10,5","Alfoz de Lloredo","L-D: 08:00- 21:00","1,999","1,859","06"}, ...]	"2,02"
UGIC.5b	Lista de gasolineras vacía []	"0,00"

Pruebas unitarias de la capa de presentación

Para poder llevar a cabo estas pruebas, será necesario el uso de objetos Mocks para las interfaces IPrefs e IGasolinerasRepository.

En este caso se aplica la técnica basada en la historia de usuario para la definición de las pruebas a realizar. Los casos de prueba definidos serán los mismos que los de las pruebas de aceptación (renombrados como UVF.X) pero automatizados a través de JUnit y Espresso.

Pruebas de interfaz (Mario Ingelmo)

Se aplica la técnica basada en la historia de usuario para la definición de las pruebas a realizar. Los casos de prueba definidos serán los mismos que los de las pruebas de aceptación (renombrados como UVF.X) pero automatizados a través de JUnit y Espresso.

Tabla 2. Casos de prueba de interfaz

Identificador	Entrada	Resultado
A1.a	1,90 Selecciona aplicar filtros	Se muestra una lista de gasolineras. CEPSA Carretera 6316 KM, 10,5 Gasolina: 1,859 Diesel: 1,999 REPSOL CR N-629 79,7 Gasolina: 1,819 Diesel:2,009 ...



A1.b	El usuario selecciona la cruz que se encuentra en el menú ListaGasolineras	Se muestran todas las gasolineras sin importar el precio.
A1.c	1,90 Selecciona aplicar filtros	Se muestra el siguiente mensaje de error. "Conéctese a la red y vuelva a intentarlo más tarde."
A1.d	1,90 Selecciona aplicar filtros	Se muestra el siguiente mensaje de error. "No se puede conectar con la base de datos. Inténtelo de nuevo más tarde".
A1.e	0,10 Selecciona aplicar filtros	Se muestra el siguiente mensaje de error. "No se encuentra ninguna gasolinera con un precio menor al establecido".

Ángel Castanedo Grande