## Plan de pruebas US 500865 Filtrar por precio máximo

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 los criterios de aceptación definidos en la historia de usuario y se ejecutarán de forma manual.
* Pruebas unitarias. Se utilizará técnica de prueba de métodos y 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 y Mockito.
* Pruebas de integración. Se pretende probar la integración entre la capa de negocio y de la de persistencia. Para la definición de los casos de prueba se utilizará técnica de prueba de métodos y de caja negra, será necesario utilizar Junit.
* Pruebas de Interfaz gráfica. Se aplica la técnica basada en los criterios de aceptación, por lo tanto, se automatizarán las pruebas de aceptación mediante JUnit y Expresso.

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

**PRUEBAS DE ACEPTACIÓN**

En base a los criterios de aceptación se definen los siguientes escenarios (suponemos que la interfaz por construcción no permite dejar ningún campo requerido en blanco, para evitar casos de prueba referidos a información no disponible):

A1. CA: Caso de éxito

1. a. El cliente realiza una pulsación de selección (un click) sobre la opción de filtrar en el toolbar de la aplicación.
2. La aplicación muestra una ventana emergente con las opciones de filtrado, entre ellas la del filtrado por precio máximo.
3. El cliente selecciona en un deslizable el precio máximo que quiere establecer.
4. El cliente selecciona la opción de aplicar filtros.
5. La aplicación cierra la ventana emergente de selección de los filtros.
6. Se verifica que la aplicación muestre correctamente las gasolineras filtradas.
7. Se verifica que la aplicación muestre correctamente en un toast el número de gasolineras cargadas al aplicar el filtro.

A2. CA: Reestablecer filtros

1. El cliente realiza una pulsación de selección (un click) sobre la opción de filtrar en el toolbar de la aplicación.
2. La aplicación muestra la ventana emergente con las opciones de filtrado, entre ellas la de restablecer filtros.
3. El cliente selecciona la opción de restablecer filtros.
4. Se verifica que la aplicación muestre todas las gasolineras almacenadas en el sistema.
5. Se verifica que la aplicación muestre correctamente en un toast el número de gasolineras cargadas al restablecer los filtros.

A3. CA: Cerrar sin aplicar filtros

1. El cliente realiza una pulsación de selección (un click) sobre la opción de filtrar en el toolbar de la aplicación.
2. La aplicación muestra una ventana emergente con las opciones de filtrado, entre ellas la del filtrado por precio máximo.
3. El cliente selecciona en un deslizable el precio máximo que quiere establecer.
4. El cliente selecciona la opción de cerrar la ventana emergente.
5. La aplicación cierra la ventana emergente de selección de los filtros.
6. Se verifica que la aplicación muestre correctamente las gasolineras sin aplicar el filtro.

A4. CA: Aplicar filtros sin seleccionar opciones de filtrado

1. El cliente realiza una pulsación de selección (un click) sobre la opción de filtrar en el toolbar de la aplicación.
2. La aplicación muestra una ventana emergente con las opciones de filtrado, entre ellas la del filtrado por precio máximo, con la opción por defecto que selecciona sin precio máximo.
3. El cliente no selecciona en el deslizable ningún precio máximo.
4. El cliente selecciona la opción de aplicar filtros.
5. La aplicación cierra la ventana emergente de selección de los filtros.
6. Se verifica que la aplicación muestre correctamente las gasolineras sin aplicar el filtro.
7. Se verifica que la aplicación muestre correctamente en un toast el número de gasolineras cargadas sin aplicar el filtro.

A5. CA: No se encuentran coincidencias para el filtro aplicado

1. El cliente realiza una pulsación de selección (un click) sobre la opción de filtrar en el toolbar de la aplicación.
2. La aplicación muestra una ventana emergente con las opciones de filtrado, entre ellas la del filtrado por precio máximo.
3. El cliente selecciona en un deslizable el precio máximo que quiere establecer.
4. El cliente selecciona la opción de aplicar filtros.
5. La aplicación cierra la ventana emergente de selección de los filtros.
6. Se verifica que la aplicación no muestre ninguna gasolinera.
7. Se verifica que la aplicación muestre en un toast que no ha encontrado ninguna coincidencia.

Tabla 1. Casos de prueba de aceptación

|  |  |  |
| --- | --- | --- |
| **Identificador** | **Entrada** | **Resultado** |
| A1 | 1.50 | Lista de gasolineras con precios no superior a 1.50 y Mensaje éxito |
| A2 |  | Lista con todas las gasolineras y Mensaje éxito |
| A3 |  | Lista con todas las gasolineras |
| A4 |  | Lista con todas las gasolineras y Mensaje éxito |
| A5 | 0.10 | Mensaje aviso gasolineras no encontradas |

**PRUEBAS UNITARIAS**

En esta historia de usuario no se modifica nada respecto al acceso y persistencia de datos, por lo tanto, las pruebas unitarias consistirán en pruebas en las clases de dominio, negocio y de presentación

**Pruebas unitarias de las clases de dominio**

Deberían probarse los métodos de la clase Filter.

* Método setMaxPrice( p : Double ) : IFilter

|  |  |  |
| --- | --- | --- |
| Identificador | Entrada | Valor esperado |
| UD1.a | 0.00 | Filter(0.00) |
| UD1.b | 2.00 | Filter(2.00) |
| UD1.c | -1.00 | Filter(0.00) |

* Método toFilter( g : List<Gasolinera> ) : List<Gasolinera>

|  |  |  |
| --- | --- | --- |
| Identificador | Entrada | Valor esperado |
| UD2.a | Filter(1.8)  [Gasolinera(Nombre, 1.8), Gasolinera(Nombre, 1.0), Gasolinera(Nombre, 1.9), Gasolinera(Nombre, NULL)] | [Gasolinera(Nombre, 1.8), Gasolinera(Nombre, 1.0)] |

**Pruebas unitarias de la capa de negocio**

Deberían probarse los métodos de la clase MainPresenter para ello seran necesarios mocks de IGasolinerasRepository, ICallBack, IFilter y IMainContract.View

* Método onFiltersPopUpFuelTypesOneSelected(int index, boolean value)
* Método onFiltersPopUpFuelTypesAccepted()
* Método onFiltersPopUpCancelClicked()
* Método onFiltersPopUpAcceptClicked()
* Método onFiltersPopUpClearFiltersClicked()

**PRUEBAS DE INTEGRACIÓN**

**Pruebas de integración de la capa de negocio**

Deberían probarse los métodos de la clase MainPresenter, para ello usaremos la clase GasolinerasRepository, Filter y el mockup de I MainConctract.View

* Método onFiltersPopUpFuelTypesOneSelected(int index, boolean value)
* Método onFiltersPopUpFuelTypesAccepted()
* Método onFiltersPopUpCancelClicked()
* Método onFiltersPopUpAcceptClicked()

**[Aquí meto los casos de prueba para este método pendiente de definir]**

* Método onFiltersPopUpClearFiltersClicked()

**PRUEBAS DE UI**

En este caso se aplica la técnica basada en casos de uso 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 UIT.X) pero automatizados a través de JUnit y Expresso.

|  |  |  |
| --- | --- | --- |
| **Identificador** | **Entrada** | **Resultado** |
| UIT.1 | 1.50 | Lista de gasolineras con precios no superior a 1.50 y Mensaje éxito indicando que obtiene X gasolineras |
| UIT.2 |  | Lista con todas las gasolineras y Mensaje éxito indicando que tiene 164 gasolineras |
| UIT.3 |  | Lista con todas las gasolineras |
| UIT.4 |  | Lista con todas las gasolineras y Mensaje éxito indicando que tiene 164 gasolineras |
| UIT.5 | 0.10 | Mensaje aviso gasolineras no encontradas |