



Plan de pruebas #506177-Ordenar por precio

PRUEBAS DE ACEPTACIÓN

En base de los criterios de aceptación se identifican los siguientes escenarios:

A1. a CA: Caso de éxito: La aplicación muestra la lista ordenada de datos correctamente bajo el criterio de orden seleccionado (ASCENDENTE).

1. El usuario pulsa el botón de ordenar en la toolbar.
2. El usuario selecciona el tipo de carburante en el menú desplegable.
3. El usuario selecciona el orden por el cual organizar los precios.
4. El usuario pulsa el botón “ordenar”.
5. La aplicación muestra una lista con las gasolineras encontradas con el carburante seleccionado y ordenadas de manera ascendente.
6. La aplicación muestra un toast con el número de gasolineras encontradas.
7. Se verifica que la lista ha sido ordenada correctamente de manera ascendente.

A1. b CA: Caso de éxito: La aplicación muestra la lista ordenada de datos correctamente bajo el criterio de orden seleccionado (DESCENDENTE).

1. El usuario pulsa el botón de ordenar en la toolbar.
2. El usuario selecciona el tipo de carburante en el menú desplegable.
3. El usuario selecciona el orden por el cual organizar los precios.
4. El usuario pulsa el botón “ordenar”.
5. La aplicación muestra una lista con las gasolineras encontradas con el carburante seleccionado y ordenadas de manera descendente.
6. La aplicación muestra un toast con el número de gasolineras encontradas.
7. Se verifica que la lista ha sido ordenada correctamente de manera descendente.



A1. c CA: Caso de éxito: La aplicación muestra la lista ordenada de datos correctamente y las gasolineras con caracteres erróneos o nulos son enviados al final de la lista de datos

1. El usuario pulsa el botón de ordenar en la toolbar.
2. El usuario selecciona el tipo de carburante en el menú desplegable.
3. El usuario selecciona el orden por el cual organizar los precios.
4. El usuario pulsa el botón “ordenar”.
5. La aplicación muestra una lista con las gasolineras encontradas con el carburante seleccionado y ordenadas de manera ascendente o descendente en función de su selección, al existir gasolineras que carecen del carburante por el que se ordena ó su valor es nulo, éstas se muestran sin orden alguno al final de la página.
6. La aplicación muestra un toast con el número de gasolineras encontradas.
7. Se verifica que la lista ha sido ordenada correctamente y que las gasolineras con caracteres erróneos se encuentran al final de la misma.

A2. CA: Caso de error: Resultado con lista vacía.

1. El usuario pulsa el botón de ordenar en la toolbar.
2. El usuario selecciona el tipo de carburante en el menú desplegable.
3. El usuario selecciona el orden por el cual organizar los precios.
4. El usuario pulsa el botón “ordenar”.
5. La aplicación no encuentra gasolineras que puedan ser ordenadas bajo el criterio seleccionado.
6. La aplicación muestra un toast con 0 gasolineras encontradas.

En el caso de éste plan de pruebas se obvia el correcto funcionamiento de la funcionalidad de filtros. Al tener valores fijos en los selectores de orden dentro de la interfaz, no es posible que éstos produzcan un fallo de entrada (las combinaciones de ambos selectores son siempre válidas). Por lo tanto el único caso de “fallo” posible es el cual parta de una lista vacía por motivos ajenos a ésta funcionalidad.



PRUEBAS UNITARIAS

Se realizarán las pruebas unitarias de los métodos *ordenarGasolinerasPorPrecio* e *init* de la clase *mainPresenter*.

Método ***ordenarGasolinerasPorPrecio(Combustible combustible, Orden orden)***

- I. Combustible y orden válidos ascendente.
- II. Combustible y orden válidos descendente.
- III. Combustible y orden válidos pero lista de gasolineras vacía.
- IV. Combustible y orden válidos pero hay gasolineras sin datos específicos.

Tabla 2. Casos de prueba de unitaria

Identificador	Entrada*	Valor esperado
UP.I	("GASOLEOA","ASCENDENTE")	Llama a <i>getPrecioCombustible</i> de cada gasolinera y ordena de manera ascendente.
UP.II	("BIODIESEL","DESCENDENTE")	Llama a <i>getPrecioCombustible</i> de cada gasolinera y ordena de manera ascendente.
UP.III	("GASOLEOA","DESCENDENTE")	Llama a <i>getPrecioCombustible</i> de cada gasolinera, al no tener datos ordena una lista vacía.
UP.IV	("BIODIESEL","ASCENDENTE")	Llama a <i>getPrecioCombustible</i> de cada gasolinera, al existir gasolineras con valores inválidos de la especificación se mueven al final de la lista y el resto se ordenan como se determine por el parámetro de entrada.

Internamente se establece una lista local de gasolineras y se asegura su correcto orden en función de los parámetros de entrada.



- Método *init(View view)*

Para éste método valdrá con mockear correctamente la vista para después crearla empleando el main presenter en OrdenarTest.

Tabla 3. Casos de prueba unitaria

Identificador	Entrada	Valor esperado
UP2.I	(mockView)	Se muestran correctamente las gasolinera <i>mockeadas</i>

Resultados de las pruebas:

Pruebas unitarias:

Se comprueba con un determinado grupo de gasolineras que al realizar el orden éste es correcto y se muestran como se espera en función de su precio de combustible.

Pruebas de interfaz**:

Se comprueba el correcto funcionamiento de los componentes de orden en la aplicación.

No es posible comprobar a través de interfaz si el orden es correcto ya que todos los componentes de la vista en este caso comparten el mismo id y al tratar de clasificarlos provoca error en el test, sin embargo éste funcionamiento ya ha sido comprobado en los test unitarios.

*** Debido a una serie de errores con el entorno de desarrollo ya no se comprueba el mensaje de confirmación de carga (toast) en las pruebas de interfaz.*