

## Plan de pruebas US 505739 Ordenar resultados por precio.

### PRUEBAS DE ACEPTACIÓN

Las pruebas de aceptación se encuentran definidas en la tarjeta de scrumdesk “505739 - Ordenar resultados por precio”.

### PRUEBAS DE UI

Son las mismas que las pruebas de aceptación, solo que en este caso se ha sintetizado la prueba UI.6 y UI.7 en UI.1 y UI.2, con excepción de que los resultados obtenidos se filtran en base a un fichero JSON llamado “gasolineras\_test\_505739”, y se automatizan a través de Espresso.

Una tabla resumen del fichero JSON mencionado se encuentra a continuación:

Rótulo (Identifica un objeto de gasolinera)	Precio Gasolina 95 E5	Precio Gasóleo A
CEPSA	1,679	1,509
REPSOL	1,669	1,569
PETRONOR	1,639	1,525
PETRONOR COPIA	1,639	1,525
REDETRANS	- (No tiene este combustible)	1,299
EROSKI	1,629	-
ALSA	-	-

Tabla 1: Resumen datos JSON.

Los casos de prueba de UI definidos según los datos anteriores son los siguientes:

ID	Entrada	Resultado
UI.1	Gasolina 95 E5, Ascendente (menor a mayor)	{ EROSKI, PETRONOR, PETRONOR COPIA, REPSOL, CEPSA, REDETRANS, ALSA }
UI.2	Gasóleo A, Descendente (mayor a menor)	{ REPSOL, PETRONOR, PETRONOR COPIA, CEPSA, REDETRANS, EROSKI, ALSA }
UI.3	Restablecer orden	{ CEPSA, REPSOL, PETRONOR, PETRONOR COPIA, REDETRANS, EROSKI, ALSA }
UI.4	Gasolina 95 E5, Descendente, Cancelar	{ CEPSA, REPSOL, PETRONOR, PETRONOR COPIA, REDETRANS, EROSKI, ALSA }
UI.5	OrderByPrice(Gasóleo A, Ascendente) FilterByType(Gasolina 95 E5)	Toast notificando colisión entre filtros.

Tabla 2: Casos de prueba UI.

### **LISTA DE CLASES CON MÉTODOS A PROBAR**

A continuación, se listan las clases con los métodos a probar por haber sido creados o modificados.

De la clase de dominio *OrderByPrice*, los métodos:

- `int compare(Gasolinera g1, Gasolinera g2)`

De la clase del presentador *MainPresenter*, los métodos:

- `void onOrderClicked();`
- `void onFuelTypeSelected(FuelTypeEnum type);`
- `void onMethodOrderSelected(OrderMethodsEnum orderMethod);`
- `void onOrderPopUpAcceptClicked();`
- `void onOrderPopUpCancelClicked();`
- `void onOrderPopUpClearClicked();`

### **ESPECIFICACIÓN DE PRUEBAS**

A continuación, se especifican casos de prueba unitaria o integración para dos métodos diferentes.

#### **Prueba unitaria**

De la clase de dominio *OrderByPrice*, empleando los datos de la Tabla 1, el método:

- `int compare(Gasolinera g1, Gasolinera g2)`

ID	Entrada	Valor esperado
UD1.a	<code>compare(REPSOL, CEPSA), fuelType= Gasolina 95 E5, ascending=true</code>	1 (param1 < param2)
UD1.b	<code>compare(CEPSA, REPSOL), fuelType= Gasolina 95 E5, ascending=true</code>	-1 (param1 > param2)
UD1.c	<code>compare(REPSOL, CEPSA), fuelType= Gasóleo A, ascending=false</code>	1 (param1 < param2)
UD1.d	<code>compare(CEPSA, REPSOL), fuelType= Gasóleo A, ascending=false</code>	-1 (param1 > param2)
UD1.e	<code>compare(PETRONOR, PETRONOR COPIA), fuelType= Gasolina 95 E5, ascending=false</code>	0
UD1.f	<code>compare(REDETRANS, EROSKI), fuelType= Gasolina 95 E5, ascending=false</code>	1 (param1 sin tipo combustible)
UD1.g	<code>compare(REDETRANS, EROSKI), fuelType= Gasóleo A, ascending=false</code>	-1 (param2 sin tipo combustible)
UD1.h	<code>compare(REDETRANS, ALSA), fuelType= Gasolina 95 E5, ascending=false</code>	0 (param1 & param2 sin tipo combustible)

Tabla 3: Casos de prueba unitaria `compare()`.

#### **Prueba de integración**

Debemos probar la integración entre la nueva clase de dominio `OrderByType` y el Presenter modificado. Para ello, será necesario el uso de objetos Mock para la interfaz `IMainContract#View`, probando así la interacción entre `OrderByType` y el Presenter.

De la clase del presentador *MainPresenter*, empleando los datos de la Tabla 1, el método:

- `void onOrderPopUpAcceptClicked();`

ID	Entrada	Valor esperado
ID1.a	<code>fuelType=Gasolina 95 E5, ascending=true</code>	{ EROSKI, PETRONOR, PETRONOR COPIA, REPSOL, CEPSA, REDETRANS, ALSA }
ID1.b	<code>fuelType= Gasóleo A, ascending=false</code>	{ REPSOL, PETRONOR, PETRONOR COPIA, CEPSA, REDETRANS, EROSKI, ALSA }
ID1.c	<code>fuelType=null, ascending=false</code>	{ CEPSA, REPSOL, PETRONOR, PETRONOR COPIA, REDETRANS, EROSKI, ALSA }
ID1.d	<code>fuelType=Gasolina 95 E5, ascending=null</code>	{ CEPSA, REPSOL, PETRONOR, PETRONOR COPIA, REDETRANS, EROSKI, ALSA }
ID1.e	<code>fuelType=null, ascending=null</code>	{ CEPSA, REPSOL, PETRONOR, PETRONOR COPIA, REDETRANS, EROSKI, ALSA }

Tabla 4: Casos de prueba integración `onOrderPopUpAcceptClicked()`.

Nótese que los apartados de este documento han sido redactados siguiendo las [Normas de Evaluación para Procesos de Ingeniería Software - Planes de Prueba](#).

Plan de pruebas realizado por: Adrián Del Río

## IMPLEMENTACION DE LAS PRUEBAS:

### PRUEBA DE UI PARA EL CASO UI.1: Lucía Fernández Mancebo

Para la realización de estos test de UI, se ha decidido probar bajo el análisis de valores límite que la salida proporcionada en el UI.1 sea correcta solamente para el primer, tercer y último elemento mostrados en la lista de gasolineras que se obtienen como resultado. Asumiendo así, que, si dichos valores son correctos, entonces la ordenación se ha llevado a cabo con éxito.

Además, debido a que en la vista detallada con información de las gasolineras, no existe ningún campo que contenga el identificador concreto de la gasolinera, que se pueda comprobar con los identificadores proporcionados por el JSON, se ha decidido utilizar los campos de “Rótulo” y de “Precio Gasolina 95 E5” para determinar que las gasolineras estén ordenadas correctamente

bajo la salida proporcionada en el campo de Resultado de UI.1, pudiendo diferenciar así las gasolineras que puedan llegar a contener el mismo rótulo (en el JSON no se da el caso).

No se han detectado problemas de implementación a la hora de ejecutar los test. Sin embargo, ha habido algún que otro problema a la hora de implementar los test de UI, generado una serie de fallos, entre los que se encuentran:

- Al acceder a la vista detalle de la gasolinera concreta a probar, no se volvía una vez comprobados sus valores, a la vista que mostraba el ListView de las gasolineras. Por ende, al tratar de acceder a la vista detalle de la siguiente gasolinera, las pruebas finalizaban bajo error al no encontrar el elemento con el id proporcionado.
- En la vista detalle, los precios se redondean con 2 decimales (superior), de manera que, al comprobar los precios introduje en un inicio, el precio que figuraba en el JSON, y por ende, no coincidían los precios con los mostrados en la vista.

#### **PRUEBA DE UNITARIA PARA EL CASO UD.1: Miguel Monje Velarde**

Se han programado las pruebas según los datos indicados en el plan de pruebas, en este caso mediante la creación de las gasolineras en el propio caso de pruebas con ejecución satisfactoria.

#### **PRUEBA DE INTEGRACIÓN PARA EL CASO ID.1: Miguel Monje Velarde**

Se han programado las pruebas según los datos indicados en el plan de pruebas, durante el desarrollo no se han podido implementar los casos de prueba para c,d y e debido a que los filtros no pueden ser nulos y en caso de que se puse aceptar se introducen por defecto. No se ha detectado ningún fallo.