

PLAN DE PRUEBA

Especificación de las pruebas de aceptación acordadas con el Product Owner

Prueba 00: Éxito (Orden ascendente)

1. El usuario selecciona el botón de ordenar.
2. El usuario selecciona el menú desplegable de *Criterio*.
3. El sistema muestra al usuario los posibles criterios de ordenación.
4. El usuario selecciona el criterio de ordenar por potencia.
5. El usuario selecciona el orden ascendente.
6. El usuario selecciona el botón de buscar.
7. Se verifica que el sistema muestra los puntos de carga en función de su potencia por orden ascendente.

Prueba 01: Éxito (Orden descendente)

1. El usuario selecciona el botón de ordenar.
2. El usuario selecciona el menú desplegable de *Criterio*.
3. El sistema muestra al usuario los posibles criterios de ordenación.
4. El usuario selecciona el criterio de ordenar por potencia.
5. El usuario selecciona el orden descendente.
6. El usuario selecciona el botón de buscar.
7. Se verifica que el sistema muestra los puntos de carga en función de su potencia por orden descendente.

Prueba 02: No existe ningún punto de recarga para ordenar

1. El usuario selecciona el botón de ordenar.
2. El usuario selecciona el menú desplegable de *Criterio*.
3. El sistema muestra al usuario los posibles criterios de ordenación.
4. El usuario selecciona el criterio de ordenar por potencia.
5. El usuario selecciona el orden ascendente o descendente.
6. El usuario selecciona el botón de buscar.
7. Se verifica que el sistema lanza un mensaje alertando de la existencia de ningún cargador para ordenar.

Prueba 03: No se selecciona ningún tipo de ordenación

1. El usuario selecciona el botón de ordenar.
2. El usuario selecciona el menú desplegable de *Criterio*.
3. El sistema muestra al usuario los posibles criterios de ordenación.
4. El usuario selecciona el criterio de ordenar por potencia.
5. El usuario selecciona el botón de buscar.
6. Se verifica que el sistema muestra un error alertando de la falta del tipo de ordenación (ascendente o descendente).

Prueba 04: No se selecciona ningún criterio de ordenación

1. El usuario selecciona el botón de ordenar.
2. El usuario selecciona el orden ascendente o descendente.

3. El usuario selecciona el botón de buscar.
4. Se verifica que el sistema muestra un error alertando de la falta del criterio para la ordenación.

Prueba 05: Varios cargadores tienen la misma potencia

1. El usuario selecciona el botón de ordenar.
2. El usuario selecciona el menú desplegable de *Criterio*.
3. El sistema muestra al usuario los posibles criterios de ordenación.
4. El usuario selecciona el criterio de ordenar por potencia.
5. El usuario selecciona el orden ascendente o descendente.
6. El usuario selecciona el botón de buscar.
7. Se verifica que el sistema muestra los puntos de carga en función de su potencia por orden. En caso de haber cargadores empatados en potencia, les muestra por orden alfabético.

Especificación de los casos de prueba unitarios

Prueba unitaria del método `onSortClicked(String criterio, Boolean ascendente)`

El método a implementar va a estar en la clase `MainPresenter` teniendo la siguiente estructura `onSortClicked(String criterio, Boolean ascendente)`.

El método tiene como entradas las pasadas como parámetros que son un string indicando el criterio para ordenar (actualmente solo está el criterio potencia) y el boolean que indica el orden, si es true es ascendente, si es false es descendente y si es Null dejara la lista como estaba. Además, tiene la lista existente de cargadores que cada uno tiene el valor del criterio que se va a comprobar (solo se ha implementado potencia, por lo tanto, aparece la potencia máxima en cada cargador). La lista está representada entre llaves y luego cada elemento de la lista es un cargador representado por corchetes y en su interior su potencia máxima y el nombre de la compañía del cargador.

Identificador	Entrada	Valor Esperado
UT.1a	("Potencia", true) {[600, "A"], [600.1, "A"], [0, "A"]}	{[0, "A"], [600, "A"], [600.1, "A"]}
UT.1b	("Potencia", false) {[600, "A"], [600.1, "A"], [0, "A"]}	{[600.1, "A"], [600, "A"], [0, "A"]}
UT.1c	("Potencia", null) {[600, "A"], [600.1, "A"], [600, "A"], [0]}	{[600, "A"], [600.1, "A"], [0, "A"]} Alerta al usuario
UT.1d	("Potencia", true) {[0, "A"], [600, "A"], [600.1, "A"]}	{[0, "A"], [600, "A"], [600.1, "A"]}
UT.1e	("Potencia", true) {[600, "A"]}	{[600, "A"]}
UT.1f	("Potencia", true) {[600.1, "A"], [600, "B"], [600, "A"], [0, "A"]}	{[0, "A"], [600, "A"], [600, "B"], [600.1]}
UT.1g	("Potencia", true) {}	{} Alerta al usuario
UT.1h	("Ninguno", true) {[600, "A"], [600.1, "A"], [0, "A"]}	{[600, "A"], [600.1, "A"], [0, "A"]} Alerta al usuario
UT.1i	(null, true) {[600, "A"]}	NullPointerException (No implementable)

Prueba unitaria del método potenciaMaxima()

El método a implementar es potenciaMaxima(): double de la clase Charger.

Como entrada se encuentra la lista de conexiones que tiene como atributo la clase Charger. En cada conexión se representa su potencia. La lista está representada entre llaves y luego cada elemento de la lista es una conexión representada por corchetes y en su interior su potencia.

Identificador	Entrada	Valor Esperado
UT.2a	{[600], [600.1], [0]}	600.1
UT.2b	{[600], [0], [600]}	600
UT.2b	{[2]}	2
UT.2c	{}	0

Reporte final

Gracias a las pruebas unitarias se encontró que había una línea en el código que no funcionaba siempre como queríamos la línea era "return (int) (ch1.maxPower() - ch2.maxPower());". Ya que si las potencias máximas son por ejemplo 600.1 y 600 este código va a indicar que son iguales al realizar la comparación, ya que la resta dará 0.1 y al hacer el cast a int dará 0.

Por otra parte, gracias a las pruebas de interfaz nos dimos cuenta de que no se había implementado que se ordenase por orden alfabético cuando haya empate en la potencia máxima de cargadores.