

## **Plan de pruebas US 500767-Mostrar gasolineras en punto de interés**

Autor: Sergio Salas

### **PRUEBAS DE ACEPTACIÓN**

Las pruebas de aceptación están definidas en la tarjeta correspondiente a la historia de usuario 500767 en ScrumDesk.

### **PRUEBAS UI**

En este caso se aplica la técnica basada en los criterios de aceptación 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.

Para estas pruebas se parte de que todos los puntos de interés están creados en la DAO.

<b>Identificador</b>	<b>Entrada</b>	<b>Valor esperado</b>
UIT.1	<a href="#"><u>Punto de interés con gasolineras.</u></a>	Se comprueba que se muestran las gasolineras. Se comprueba que se muestra la información correcta del punto de interés (Nombre, radio, número de gasolineras).
UIT.2	<a href="#"><u>Punto de interés sin gasolineras.</u></a>	Se comprueba que se muestra aviso no hay gasolineras. Se comprueba que se muestra la información correcta del punto de interés (Nombre, radio, número de gasolineras).
UIT.3	<a href="#"><u>Punto de interés con gasolineras que falla.</u></a> Utilizamos un punto de interés que por coordenadas debería tener varias gasolineras, pero utilizamos un JSON donde algunas de esas gasolineras tengan valores de latitud y longitud incorrectos.	Se comprueba que no se muestran todas las gasolineras que deberían mostrarse.

### **Pruebas unitarias de dominio**

Deberían probarse los métodos de la clase InterestPoint:

-getLocation(): Location

-isGasStationInRadius (Gasolinera g): Boolean

### **Método isGasStationInRadius (Gasolinera g): Boolean**

Se tiene un punto de interés con los siguientes datos:

Nombre: "Punto prueba"

Latitud: 43.4188

Longitud: -3.8407

Radio: 5 km

Se le pasa al método una gasolinera y se comprueba si esa gasolinera está dentro del rango del punto de interés.

Identificador	Entrada	Valor esperado
UD.1.a	Gasolinera dentro del rango del radio.  Gasolinera REPSOL Calle Boo, 52 Astillero. Latitud: 43,412222 Longitud: -3,843500	True
UD.1.b	Gasolinera fuera del rango del radio.  Gasolinera BALLENOIL Calle La Mar, 47. Latitud: 43,393556 Longitud: -3,463167	False
UD.1.c	Null	False
UD.1.d	Gasolinera con valores de latitud y longitud a null. Gasolinera CEPESA Avenida Cantabria, 99. Latitud: 0 Longitud: 0	False

## **Pruebas unitarias**

Deben probarse los métodos de la clase MainPresenter mediante el uso de mocks al View y el repositorio de gasolineras.

Deberían de probarse los siguientes métodos de la clase MainPresenter:

-init (IMainContract.View view) : void

Se va a implementar la prueba unitaria del método init (IMainContract.View view) : void:

Toda la parte de errores a la hora de obtener los datos se gestionan en la historia de usuario del modo sin conexión.

En nuestro caso, contamos que el load nunca tiene un error.

Identificador	Entrada	Valor esperado
UD.2.a	<a href="#">Punto de interés con gasolineras.</a> <a href="#">Lista con gasolineras.</a>	Se llama al init del view.  Se llama al showInterest PointInfo(puntoInteres, 1);  Se llama a show Stations(listaGasolineras)  Como la lista de gasolineras no está vacía: Se llama a showLoad Correct(1)
UD.2.b	<a href="#">Punto de interés con gasolineras.</a> <a href="#">Lista sin gasolineras.</a>	Se llama al init del view. Se llama al showInterest PointInfo(puntoInteres, 3);  Se llama a show Stations(listaGasolineras)  Como la lista de gasolineras está vacía:  Se llama a showLoadError();

## **Tablas de datos**

Para poder realizar pruebas necesitamos tener puntos de interés guardados.

Punto de interés con gasolineras:

<b>Id</b>	<b>name</b>	<b>color</b>	<b>Latitude</b>	<b>Longitude</b>	<b>radius</b>
1	Zona 1	AZUL	43.3192	-4.2987	2

1 gasolinera

Punto de interés sin gasolineras:

<b>id</b>	<b>Name</b>	<b>Color</b>	<b>Latitude</b>	<b>Longitude</b>	<b>radius</b>
2	Zona 2	VERDE	42.8881	-4.0025	0.5

0 gasolineras

Punto de interés con gasolineras que falla:

En el Json cambiaremos los valores de latitud y longitud de una gasolinera que debería pertenecer al punto de interés para que no aparezca.

<b>id</b>	<b>name</b>	<b>Color</b>	<b>Latitude</b>	<b>Longitude</b>	<b>radius</b>
3	Zona 3	GRIS	43.3846	-3.7385	4

2 gasolineras

Lista con gasolineras: 1 gasolinera

-E.S. TRECEÑO S.L. Latitud: 43,319556 Longitud: -4,309000

-REPSOL Latitud: 43,003861 Longitud: -4,131667

-CEPSA Latitud: 43,007139 Longitud: -4,128750

Lista sin gasolineras: 0 gasolineras

## **IMPLEMENTACIÓN DE LAS PRUEBAS**

### **- Prueba unitaria de la clase Main Presenter método init (IMainContract.View view): void**

Test realizado por Lucía Sañudo.

### **- Prueba unitaria de la clase InterestPoint método isGasStationInRadius (Gasolinera g): Boolean**

Como se utiliza Location y Color que necesitan del contexto de Android, he tenido que mockearlos. Para ello, he tratado de mockear el constructor, pero no funcionaba de forma correcta, por lo que he tenido que crear un método externo que reciba el Location para que no tenga que crear la clase y poderlo mockear correctamente.

Test realizado por Pablo Landeras.

### **-Prueba de UI**

Test realizado por Sergio Salas.