

## Plan de pruebas de la aplicación CarChargers

### Ordenar Puntos de Carga por Ubicación por Defecto

Los niveles de prueba que se van a aplicar son los siguientes:

- Pruebas de aceptación.
- Pruebas unitarias.
- Pruebas de Interfaz.

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

#### **MÉTODOS A PROBAR**

LocationComparator.compare()

calculaDistancia()

#### **1) PRUEBAS DE ACEPTACIÓN**

**Objetivo:** Verificar que, al abrir la aplicación, la lista de cargadores aparezca ordenada por cercanía a la ubicación del dispositivo del usuario.

A0: Éxito

1. El cliente abre la aplicación.
2. El sistema comprueba que tiene los permisos de ubicación
3. Si no los tiene el sistema muestra un pop-up que solicita el permiso de ubicación del cliente.
4. El sistema comprueba que tiene la ubicación activa
5. El sistema muestra la lista de cargadores ordenados en base a la cercanía a la ubicación actual del cliente.
6. Se verifica que la lista de puntos de carga se encuentra ordenada en base a la ubicación del cliente mostrándose en primer lugar el punto más cercano.

A1: Imposibilidad de acceder a la ubicación del cliente

1. El cliente abre la aplicación.
2. El sistema comprueba que tiene los permisos de ubicación
3. Si no los tiene el sistema muestra un pop-up que solicita el permiso de ubicación del cliente.
4. El sistema comprueba que tiene la ubicación activa.
5. El sistema no puede acceder a la ubicación por la mala cobertura.
6. El sistema muestra la lista sin ordenar, según se recupera del repositorio.
7. El sistema muestra un mensaje con la siguiente información: "Localización no disponible para ordenamiento".
8. Se verifica que la lista no está ordenada por la ubicación, pudiendo comprobar que los datos se han cargado directamente del repositorio.

**A2: Error de conexión a la red**

1. El cliente abre la aplicación.
2. El sistema comprueba que tiene los permisos de ubicación
3. Si no los tiene el sistema muestra un pop-up que solicita el permiso de ubicación del cliente.
4. El sistema comprueba que tiene la ubicación activa.
5. El sistema muestra una lista de cargadores vacía
6. El sistema muestra un pop-up que dice: "No se ha podido cargar la información."
7. Se verifica que la lista no contiene ningún punto de carga a la vista del cliente.

**A3: Error punto sin ubicación especificada**

1. El cliente abre la aplicación.
2. El sistema comprueba que tiene los permisos de ubicación
3. Si no los tiene el sistema muestra un pop-up que solicita el permiso de ubicación del cliente.
4. El sistema comprueba que tiene la ubicación activa.
5. El sistema muestra la lista de cargadores en el orden establecido y en los casos que no se haya podido determinar la ubicación, aparecerán al final de dicha lista.
6. Se verifica que la lista está ordenada por la ubicación y al final de ella aparecen los puntos de carga sin ubicación.

**Tabla 1. Casos de prueba de aceptación**

<b>Identificador</b>	<b>Entrada</b>	<b>Resultado</b>
A0	<ol style="list-style-type: none"> <li>1. Abrir aplicación</li> <li>2. Aceptar solicitud de permisos encaso de no tenerlos</li> </ol>	Lista de cargadores ordenados en base a la cercanía a la ubicación actual del cliente.
A1	<ol style="list-style-type: none"> <li>1. Abrir aplicación</li> <li>2. Aceptar solicitud de permisos encaso de no tenerlos</li> </ol>	Lista de cargadores ordenados según se recuperan del repositorio y mensaje: "localización no disponible para ordenamiento".
A2	<ol style="list-style-type: none"> <li>1. Abrir aplicación</li> <li>2. Aceptar solicitud de permisos encaso de no tenerlos</li> </ol>	Lista de cargadores vacía y mensaje: "no se ha podido acceder a los datos".
A3	<ol style="list-style-type: none"> <li>1. Abrir aplicación</li> <li>2. Aceptar solicitud de permisos encaso de no tenerlos</li> </ol>	Lista de cargadores ordenados en base a la cercanía a la ubicación actual del cliente con los cargadores sin ubicación al final.

## 2) PRUEBAS UNITARIAS

**Objetivo:** Verificar que las unidades de código individuales funcionen correctamente

### → Capa Presenter

1. Las pruebas se realizarán sobre unas coordenadas del usuario que son: Lat: -29.6866 Lon: -123.3046

- Método **LocationComparator.compare()**

Identificador	Entrada	Valor esperado
UGIC.1a	(Cargador1 (latitud:52.343197 y longitud:-0.170632), Cargador2 (latitud: -29.6866, longitud: -1.170632))	1
UGIC.1b	(Cargador1 (latitud: 53.343197 y longitud: -1.170632), Cargador2 (latitud: 52.343197, longitud: -0.170632))	-1
UGIC.1c	(Cargador1 (latitud: 52.343197 y longitud:-0.170632), Cargador2 (latitud: 52.343197, longitud:-0.170632))	0
UGIC.1d	(null, Cargador2 (latitud:53.343197, longitud: -1.170632))	-1
UGIC.1e	(Cargador1 (latitud:52.343197 y longitud:-0.170632), null)	1
UGIC.1f	(null, null)	0

- Método **calculaDistancia()**

Identificador	Entrada	Valor esperado
UGIC 2.a	(47.0616, 146.3375, -29.6866, -123.3046)	[12408570 * 0.99, 12408570 * 1.01]
UGIC 2.b	(0.0, 146.3375, -29.6866, -123.3046)	[10051790 * 0.99, 10051790 * 1.01]
UGIC 2.c	(47.0616, 0.0, -29.6866, -123.3046)	[14851500*0.99, 14851500 * 1.01]
UGIC 2.d	(47.0616, 146.3375, 0.0, -123.3046)	[10044320*0.99, 10044320 * 1.01]
UGIC 2.e	(47.0616, 146.3375, -29.6866, 0.0)	[16558930*0.99, 16558930 * 1.01]

## 3) PRUEBAS DE INTERFAZ

**Objetivo:** Verificar que la funcionalidad del código sobre la interfaz funciona correctamente

1. A0: Se verifica que los cargadores se encuentran correctamente ordenados en función a la cercanía a una latitud y longitud simulada

**DESARROLLO DE PRUEBAS:**

1. Diseño plan de Pruebas: Yago Nava
2. Implementación pruebas unitarias: Luis Álvarez (LocationComparatorTest)
3. Implementación prueba de interfaz: Yago Nava (UbicacionOkUITest)