

Plan de prueba: feature/479501-VerInformacion

Pruebas de Aceptación:

Prueba 00: Éxito

1. El usuario selecciona el punto de carga del que desea obtener información.
2. El sistema muestra la información detallada del punto de carga al usuario.

Prueba 02: Campos sin información.

1. El usuario selecciona el punto de carga del que desea obtener información.
2. Se verifica que el sistema muestra en los campos sin información el símbolo "-".

Pruebas de Interfaz:

Identificador	Entrada	Salida
A 00	Click en punto de carga 0.	Info general pto carga 0: Título = "Zunder", Info = "https://www.zunder.com/en/", Provincia = "Región de Murcia" Ciudad = "Balsicas - Centro Cívico (Zunder)" Precio = "0,39€/kWh AC" Dispo. = "Disponible"
A 02	Click en punto de carga vacío.	Campos sin info con "-"

Métodos susceptibles a pruebas

Clase MainView:

- showLoadErrorDialog(string error): En caso de haber error mostrando los cargadores, se mostrará este alert dialog.
- showLoadError(String error): por tanto, este método también es susceptible a pruebas.

Clase Charger:

- comprobarDisponibilidad(): Obtiene si el punto de carga está disponible o no.
- listarTiposConector(): Lista los tipos de conector de un punto de carga.

Pruebas Unitarias:

Método onChargerClicked(int index):

Identificador	Entrada	Valor esperado
U1.a	List<Charger> shownChargers ≠ null, Índice válido	Llamada a los métodos: shownChargers.get(index); showChargerDetails(charger);
U1.b	List<Charger> shownChargers = null, Índice válido	No realiza llamada a ningún método para mostrar los detalles del cargador.
U1.c	List<Charger> shownChargers = null, Índice inválido < 0	IndexOutOfBoundsException exception y no realiza llamada a ningún método para mostrar los detalles del cargador.
U1.d	List<Charger> shownChargers = null, Índice inválido > shownChargers.size()	No realiza llamada a ningún método para mostrar los detalles del cargador.
U1.e	List<Charger> shownChargers ≠ null, Índice inválido	No realiza llamada a ningún método para mostrar los detalles del cargador.

Pruebas Integración:

Método init(IMainContract.View view):

Identificador	Entrada	Valor esperado
I1.a	Conexión a internet	Llamada a los métodos: init(IMainContract.View view), showChargers(MainPresenter.this.shownChargers), showLoadCorrect(MainPresenter.this.shownChargers.size()),
I1.b	Sin conexión a internet	Llamada a los métodos: init(IMainContract.View view), showLoadError(error), (error = "El sistema no pudo conectarse a la red")

Pruebas captores:

Método showChargers(MainPresenter.this.shownChargers):

Identificador	Entrada	Valor Esperado
I1.a.1	Lista vacia	Lista vacia
I1.a.2	Lista 1 elto	Lista 1 elto
I1.a.3	Lista 4 eltos	Lista 4 eltos

Método `showLoadCorrect(MainPresenter.this.shownChargers.size())`:

Identificador	Entrada	Valor Esperado
I4.a	Lista vacia (integer 0)	Integer 0
I4.b	Lista 1 elto(integer 1)	Integer 1
I4.c	Lista 4 eltos(integer 4)	Integer 4

Reporte Final

En la ejecución de los test de interfaz, unitarios y de integración, a efectos mayores no se encontraron ningunos errores a lo largo del desarrollo de la historia de usuario de ver información detallada de un punto de carga.

Para la realización de las pruebas de interfaz, el entender cómo funcionan la ejecución de las mismas con el fichero json de la carpeta raw fue crucial, por el cual tuve que familiarizarme y usar como prueba un cargador definido en este fichero.

Por otra parte, las pruebas unitarias no resultaron muy complejas una vez entendidos bien los captosres y como aplicar los mocks a las vistas, ya que en un principio es un concepto que me costó comprender y me retrasó en la realización de las pruebas.

Finalmente, las pruebas de integración fueron las más complejas, ya que para ellas tuve que tener un entendimiento del funcionamiento del conjunto de la aplicación completo que en un inicio se me hizo algo pesado, pero una vez entendí el uso de los repositorios y como aplicarlos en las pruebas mediante los métodos "getFake" todo fue más rápido.