

Plan de pruebas Mostrar todos los precios de combustible disponibles

Aitor Angulo Salas

Pruebas de aceptación

Prueba 00: Éxito

1. El usuario ingresa en la aplicación.
2. El usuario accede a los detalles de la gasolinera, donde aparece toda la información sobre la gasolinera.
3. El sistema le muestra todos los precios de los combustibles que tiene disponibles esa gasolinera ordenados alfabéticamente y con el formato concreto de "combustible: precio", donde los ':' de todos los combustibles deben aparecer alineados.

Prueba 01: Sin información de los precios

1. El usuario ingresa en la aplicación.
2. El usuario accede a los detalles de la gasolinera, donde aparece toda la información sobre la gasolinera.
3. Se verifica que aparece el apartado de precios vacío debido a que no se proporcionan la información relativa a los precios de la gasolinera en concreto.

Pruebas de interfaz de usuario

Se va a verificar el correcto funcionamiento de la aplicación al mostrar todos los precios disponibles de la gasolinera. Se usará la herramienta Espresso para validar la interfaz de la aplicación.

UI 01: MostrarTodosLosPreciosCompleto

- a. Muestra los precios de una gasolinera la cual tiene todos los combustibles disponibles.

UI 02: MostrarTodosLosPreciosParcial

- a. Muestra los precios de una gasolinera la cual tiene algunos combustibles, pero no todos.

UI 03: MostrarTodosLosPreciosVacios

- a. Muestra la lista de precios vacía ya que la gasolinera no contiene ninguno.

Nota: Se creará un JSON con las 3 gasolineras del cual la vista obtendrá la lista de gasolineras, ya una vez dentro de la vista de detalle los datos se obtendrán del objeto gasolinera.

Gasolinera1: Gasolinera que contiene todos los combustibles disponibles.

Gasolinera2: Gasolinera que solo contiene Biodiesel, Gas Natural, Gasoleo A, Gasoleo B, Gasoleo Premium, Hidrogeno.

Gasolinera3: Gasolinera que no contiene ningun tipo de combustible.

Identificador	Entrada	Resultado
IU1.a	Gasolinera1	<ul style="list-style-type: none">• “Biodiesel : 1,1€”• “Bioetanol : 1,2€”• “Gas Natural Comprimido : 1,3€”• “Gas Natural Licuado : 1,4€”• “Gases licuados del petróleo : 1,5€”• “Gasoleo A : 1,6€”• “Gasoleo B : 1,7€”• “Gasoleo Premium : 1,8€”• “Gasoleo 95 E10 : 1,9€”• “Gasoleo 95 E5 : 2,0€”• “Gasoleo 95 E5 Premium : 2,1€”• “Gasoleo 98 E10 : 2,2€”• “Gasoleo 98 E5 : 2,3€”• “Hidrogeno : 2,4€”
IU2.a	Gasolinera2	<ul style="list-style-type: none">• “Biodiesel : 1,1€”• “Gas Natural Comprimido : 1,3€”

		<ul style="list-style-type: none"> • “Gasoleo A : 1,6€” • “Gasoleo B : 1,7€” • “Gasoleo Premium : 1,8€” • “Hidrogeno : 2,4€”
IU3.a	Gasolinera3	Lista vacia sin ningún elemento

Clases y métodos añadidos para la historia de usuario

Para la implementación de esta historia de usuario, hemos agregado o modificado los siguientes métodos:

- Clase Utils: `rellenaListaCombustibles(Gasolinera gasolinera): List<GasolineraCombustible>`
- Clase CombustibleArrayAdapter
- Clase MainPresenter: `init(IMainContract.View view): Void`

Pruebas para los métodos

Prueba unitaria `rellenaListaCombustibles(Gasolinera gasolinera): List<GasolineraCombustible>`

El test está en la clase UtilsTest.

Argumentos:

- Gasolinera1 es una gasolinera con todos los tipos de combustibles disponibles.
- Gasolinera2 es una gasolinera con algunos de los combustibles disponibles, pero no todos.
- Gasolinera3 es una gasolinera con ningún combustible disponible.

Identificador	Entrada	Resultado
U1.a	Gasolinera gasolinera1	<ul style="list-style-type: none"> • “Biodiesel : 1,1€” • “Bioetanol : 1,2€” • “Gas Natural Comprimido : 1,3€” • “Gas Natural Licuado : 1,4€” • “Gases licuados del petróleo : 1,5€” • “Gasoleo A : 1,6€” • “Gasoleo B : 1,7€” • “Gasoleo Premium : 1,8€” • “Gasoleo 95 E10 : 1,9€”

		<ul style="list-style-type: none"> • “Gasoleo 95 E5 : 2,0€” • “Gasoleo 95 E5 Premium : 2,1€” • “Gasoleo 98 E10 : 2,2€” • “Gasoleo 98 E5 : 2,3€” • “Hidrogeno : 2,4€”
U1.b	Gasolinera gasolinera2	<ul style="list-style-type: none"> • “Biodiesel : 1,1€” • “Gas Natural Comprimido : 1,3€” • “Gasoleo A : 1,6€” • “Gasoleo B : 1,7€” • “Gasoleo Premium : 1,8€” • “Hidrogeno : 2,4€”
U1.c	Gasolinera gasolinera3	Lista vacia sin ningún elemento

Prueba unitaria init(IMainContract.View view): Void

El test está en la clase MainPresenterITest.

El test verifica que la aplicación se inicializa bien y la carga de gasolineras se hace de manera correcta.

Identificador	Entrada	Resultado
U1.a	List<Gasolinera> listaGasolineras	<ul style="list-style-type: none"> • Verify(mockView).init() • Verify(mockView). getGasolinerasRepository() • Verify(mockView). showStations(listaGasolineras) • Verify(mockView). showLoadCorrect(4) • Verify(mockView). getPuntosInteresDAO()

El método mockea el repositorio de gasolineras mediante la listaGasolineras, posteriormente verifica que se ejecuta el método init(), getGasolinerasRepository(), showStations(listaGasolineras), showLoadCorrect(4) y getPuntosInteresDAO().

Reporte final

La especificación del Plan de Pruebas ha sido realizada por Aitor Angulo Salas, y la correspondiente implementación de las pruebas las ha realizado Alejandro Arana Gómez. La implementación del test del método `init` lo ha implementado de manera conjunta Alejandro Arana Gómez y Aitor Angulo Salas. El plan de pruebas de Alejandro Arana Gómez (500634-OrdenarGasolinerasMasCercanas-TestPlan) ha sido modificado y listo para su recolección en el apartado de planes de prueba.