# Plan de pruebas de la historia de usuario establecer tipo de combustible

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

* Pruebas de aceptación. Las pruebas de aceptación se van a definir siguiendo las indicaciones recibidas del Product Owner.
* Pruebas de unitarias. Se utilizará la técnica de prueba de métodos, usando técnicas de caja negra para la definición de los casos de prueba de cada método. Será necesaria la utilización de Junit.
* Pruebas de interfaz de usuario. Estas pruebas serán las equivalentes a las pruebas de integración que requieran el uso de una determinada interfaz de usuario, así como también el uso de otros elementos de la aplicación, como lo es el Presenter o el Model.

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

**PRUEBAS DE ACEPTACIÓN**

En base a la reunión que se ha mantenido con el Product Owner durante el Sprint Planning Meeting I, se han definido las siguientes pruebas de aceptación.

ESTABCOMB.A: Éxito

1. El usuario selecciona la opción de Establecer Combustible por defecto

2. El sistema despliega una ventana con los tipos de combustible disponibles

3. El usuario selecciona el combustible

4. El usuario selecciona la opción Aplicar

5. El sistema establece por defecto el tipo de combustible seleccionado y muestra las gasolineras con el tipo de combustible seleccionado.

6. El sistema muestra un mensaje de éxito.

7. Se verifica que el sistema muestra un mensaje de éxito.

ESTABCOMB.B: Éxito alternativa

1. El usuario selecciona la opción de Establecer Combustible por defecto

2. El sistema despliega una ventana con los tipos de combustible disponibles

3. El usuario selecciona el combustible

4. El usuario selecciona la opción Cancelar

5. El sistema muestra un mensaje de que no se ha aplicado ningún cambio.

6. Se verifica que el sistema muestra un mensaje de aviso.

ESTABCOMB.C: Sin conexión a internet

1. El usuario selecciona la opción de Establecer Combustible por defecto

2. El sistema despliega una ventana con los tipos de combustible disponibles

3. El usuario selecciona el combustible

4. El usuario selecciona la opción Aplicar

5. El sistema no logra conectarse a internet.

6. El sistema muestra la lista de gasolineras vacía.

7. El sistema muestra un mensaje de error informando al usuario de que no dispone de conexión a internet.

8. Se verifica que el sistema muestra el mensaje de error al usuario.

ESTABCOMB.D: Error de conexión a la base de datos

1. El usuario selecciona la opción de Establecer Combustible por defecto

2. El sistema despliega una ventana con los tipos de combustible disponibles

3. El usuario selecciona el combustible

4. El usuario selecciona la opción Aplicar

5. El sistema no logra establecer comunicación con la base de datos.

6. El sistema muestra la lista de gasolineras vacía.

7. El sistema muestra un mensaje de error informando al usuario de que no ha logrado comunicarse con la base de datos.

8. Se verifica que el sistema muestra el mensaje de error al usuario.

ESTABCOMB.E: Error genérico

1. El usuario selecciona la opción de Establecer Combustible por defecto

2. El sistema despliega una ventana con los tipos de combustible disponibles

3. El usuario selecciona el combustible

4. El usuario selecciona la opción Aplicar

5. El sistema detecta algún error no contemplado durante el flujo de la ejecución.

6. El sistema muestra la lista de gasolineras vacía.

7. Se verifica que el sistema muestra un mensaje al usuario de que ha ocurrido un error.

Tabla 1. Casos de prueba de aceptación

|  |  |  |
| --- | --- | --- |
| **Identificador** | **Entrada** | **Resultado** |
| ESTABCOMB.A | El tipo de combustible seleccionado | Se guarda el combustible seleccionado como combustible por defecto y se actualiza la lista de gasolineras mostrando el tipo de combustible indicado y su precio. |
| ESTABCOMB.B | El tipo de combustible seleccionado | Se guarda el combustible seleccionado como combustible por defecto y se actualiza la lista de gasolineras mostrando el tipo de combustible indicado y su precio. |
| ESTABCOMB.C | El tipo de combustible seleccionado | La lista de gasolineras se queda vacía y se muestra un mensaje de error informando al usuario de que no dispone de conexión a internet. |
| ESTABCOMB.D | El tipo de combustible seleccionado | La lista de gasolineras se queda vacía y se muestra un mensaje de error informando al usuario de que no ha logrado comunicarse con la base de datos. |
| ESTABCOMB.E | El tipo de combustible seleccionado | La lista de gasolineras se queda vacía y se muestra un mensaje informando al usuario de que ocurrido un error. |

**PRUEBAS UNITARIAS**

* **Pruebas unitarias de las clases de modelo**

**Clase Gasolinera:** Para esta clase se harán las pruebas para los métodos getLongitud y getLatitud.

* **Pruebas unitarias de las clases presenter**

**Clase PresenterGasolinera**:En esta clase se probarán los métodos *lecturaCombustiblePorDefecto* y *escrituraCombustiblePorDefecto*.

* **Pruebas unitarias de las clases utilities**

**Clase ParserJSONGasolineras**: Para esta clase se probarán los métodos *readGasolinera, readArrayGasolineras, leerCombustiblePorDefecto, escribirCombustiblePorDefecto*.

Tabla 2. Casos de prueba validos para el método readGasolinera

|  |  |  |
| --- | --- | --- |
| Identificador | Entrada | Valor esperado |
| UT.1a | Fichero JSON con todos los datos de una gasolinera. | Gasolinera creada correctamente con los valores necesarios leídos del fichero JSON. |
| UT.1b | Fichero JSON con un campo no existente que es necesario para inicializar la gasolinera. | Gasolinera creada correctamente, en el caso de los campos que no existan en el JSON se inicializan a “” si es un campo String, o a 0 si es un campo double. |
| UT.1c | Fichero JSON con campos mal etiquetados. | Gasolina creada con los valores correctamente leídos, salvo el o los campos mal etiquetados. |
| UT.1d | Fichero JSON Vacío | Se crea una gasolinera con todos los campos inicializados por defecto. |

Tabla 2. Casos de prueba no válidos para el método readGasolinera

|  |  |  |
| --- | --- | --- |
| Identificador | Entrada | Valor esperado |
| UT.1e | Fichero con formato no JSON | IOException |
| UT.1f | Fichero Vacío | IOException |

Tabla 2. Casos de prueba válidos para el método readArrayGasolineras

|  |  |  |
| --- | --- | --- |
| Identificador | Entrada | Valor esperado |
| UT.2a | Fichero JSON con datos de varias gasolineras. | Gasolineras creadas correctamente con los valores necesarios leídos del fichero JSON. |
| UT.2b | Fichero JSON con un campo no existente que es necesario para inicializar la gasolinera. | Gasolinera creada correctamente, en el caso de los campos que no existan en el JSON se inicializan a “” si es un campo String, o a 0 si es un campo double. |
| UT.2c | Fichero JSON con campos mal etiquetados. | Gasolina creada con los valores correctamente leídos, salvo el o los campos mal etiquetados. |
| UT.2d | Fichero JSON Vacío | Se crea la lista de gasolineras con solamente una gasolinera inicializada con todos los campos por defecto. |

Tabla 2. Casos de prueba válidos para el meotdo readArrayGasolineras

|  |  |  |
| --- | --- | --- |
| Identificador | Entrada | Valor esperado |
| UT.2e | Fichero con formato no JSON | IOException |
| UT.2f | Fichero Vacío | IOException |