

## Grupo 4. PineApple.

# PLAN DE PRUEBAS DEL TICKET “NO ADMITIR CARACTERES EXTRAÑOS AL INTRODUCIR MARCA Y MODELO EN LOS VEHÍCULOS”

### AUTORES

- Definición de las pruebas de aceptación, anhadirVehiculo: Eduardo Barañano y Roberto Matas.
- Definición de las pruebas de aceptación, no caracteres extraños: Sergio Varona
- Definición de las pruebas unitarias: Daniel Llovio y Víctor Pérez
- Definición de las pruebas de interfaz: Eduardo Barañano y Roberto Matas
- Implementación de las pruebas unitarias: Daniel Llovio y Víctor Pérez
- Implementación de las pruebas de interfaz: Eduardo Barañano y Roberto Matas
- Redacción del plan de pruebas: Eduardo Barañano, Roberto Matas, Víctor Pérez, Daniel Llovio y Sergio Varona

### ÍNDICE

- Introducción
- Pruebas de Aceptación
- Pruebas Unitarias
- Pruebas de interfaz

### INTRODUCCIÓN

Mediante el presente documento se va a definir el plan de pruebas multinivel para el ticket de mantenimiento “No admitir caracteres extraños al introducir marca y modelo en los vehículos” de la aplicación **Gasgutan**. Los niveles de prueba que se van a aplicar son los siguientes, que son los ya utilizados en la historia de usuario “Añadir vehículo” a partir de la cual se origina el ticket, más los casos nuevos que corresponden a la comprobación de marcas y modelos de vehículos que contengan caracteres extraños:

- **Pruebas de aceptación:** Se han especificado varios casos de prueba a realizar en el dispositivo seleccionado por el *Product Owner* una vez se haya finalizado la implementación total de la funcionalidad en cuestión.
- **Pruebas 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* y *Robolectric*.

- **Pruebas de interfaz:** Se utilizará *JUnit* y *Espresso* para implementar las pruebas de interfaz.

## **PRUEBAS DE ACEPTACIÓN**

Las pruebas de aceptación que se realizarán a la hora de que el *Product Owner* decida evaluar el funcionamiento real de la aplicación son las que a continuación se especifican:

### **AñadirVehiculo()**

1. Añadir un vehículo en condiciones normales, rellenando todos los campos de manera válida, es decir, todos los campos serán cadenas de texto menos el combustible que es un seleccionable (Nombre, Modelo de Vehículo, Combustible en el Spinner y Matrícula). Al darle a aceptar, habrá que comprobar que el vehículo aparezca en la lista de vehículos dados de alta, a primera vista y saliendo y volviendo a entrar de la aplicación.
2. Añadir un vehículo dejando alguno de los campos de datos sin rellenar, como todos han de ser obligatorios, deberá mostrarse un mensaje avisando al usuario de que todos los campos deben ser completados, y no se cerrará el formulario.
3. Añadir un vehículo solo rellenando algunos campos y posteriormente darle a cancelar. Comprobar que la operación se ha abortado correctamente y al volver al formulario de añadir vehículo, todos los campos se han reseteado, es decir, están vacíos (no hay restos de la operación anterior).
4. Añadir un vehículo en condiciones normales, rellenando todos los campos de manera válida (Nombre, Modelo de Vehículo, Combustible en el Spinner y Matrícula), pero utilizando una matrícula repetida (es decir, que ya se encuentre dada de alta en otro vehículo que esté en la base de datos de la aplicación). De este modo, se deberá avisar al usuario de que la matrícula introducida ya pertenece a otro vehículo que se encuentra en la app y se le solicitará introducir una diferente, sin cerrar el formulario.

Las acordadas con el *Product Owner* correspondientes al ticket de mantenimiento que se está tratando ahora son:

### **NoPermitirCaracteresExtraños()**

Para comprobar el correcto funcionamiento, se agregará un nuevo vehículo en el que rellenaremos los campos de nombre y modelo con caracteres extraños como guiones o interrogaciones que comprometerían el funcionamiento de la aplicación, pero que gracias a la implementación del ticket se avisará al usuario que se están introduciendo caracteres inadecuados y que no se puede agregar el vehículo hasta que se introduzcan nuevos valores.

## PRUEBAS UNITARIAS

### Método `anadirVehiculo()` en ticket `NoCaracteresExtranhosVehiculo`

Los casos de prueba para el método **`anadirVehiculo()`** en el ticket **`NoCaracteresExtranhosVehiculo`** de la clase **`VehiclesActivity`** serán los mostrados a continuación en la Tabla 1.

#### Casos de prueba válidos

Identificador	Entrada	Valor esperado
UT.1a	("Renault", "Clio", "1456WTS", "Gasolina95")	Vehículo añadido

Tabla 1. Casos de prueba válidos para el método `anadirVehiculo()` de la clase `PresenterVehiculos`

#### Casos de prueba no válidos

UT.1c	("Renault*/-+/", "Clio", "1234TPW", "GasoleoA")	Error carácter no válido
UT.1d	("Renault", "Cl/-o/o--", "7654PWL", "Gasolina95")	Error carácter no válido
UT.1e	("Rena--ult--", "Cl-i+..o", "3426PLS", "Gasolina95")	Error carácter no válido

Tabla 2. Casos de prueba no válidos para el método `anadirVehiculo()` de la clase `PresenterVehiculos`

## PRUEBAS DE INTERFAZ

Los casos para probar que se han implementado en la clase **`AnadirVehiculoUITest`** son los definidos en la tabla a continuación que se muestra seguidamente. Para establecer dichos casos, se han tenido como referencia los casos relacionados con las pruebas de aceptación establecidas con el *Product Owner*, de tal manera que sea posible comprobar todas las diferentes alternativas que se puedan dar a la hora de añadir un vehículo en la propia aplicación. Se muestra también el caso adicional propiciado por la creación del ticket, correspondiente a la no aceptación de caracteres extraños en la marca y el modelo de cualquier vehículo que se desee añadir:

#### Casos de prueba válidos

Identificador	Entrada	Valor esperado
UIT.1	Introducir Valores, Nombre: Ford, Modelo: Focus, Combustible: Gasolina, Matrícula: 1234BBC. A continuación, pulsar el botón de aceptar.	El vehículo se muestra a continuación en la lista de vehículos existentes.
UIT.2	Introducir Valores, Nombre: dejar vacío, Modelo: dejar vacío, Combustible: Gasolina, Matrícula: 5678BBC. A continuación, pulsar el botón de aceptar.	Se muestra un mensaje solicitando que todos los campos deben de estar completados para poder añadir el vehículo.

<b>UIT.3</b>	Introducir Valores, Nombre: dejar vacío, Modelo: dejar vacío, Combustible: Gasóleo, Matrícula: 1234BBC. A continuación, pulsar el botón de cancelar.	El vehículo no se muestra en la lista de vehículos existentes y el formulario se resetea.
<b>UIT.4</b>	Introducir Valores, Nombre: BMW, Modelo: X1, Combustible: Gasóleo, Matrícula: 1234BBC. A continuación, pulsar el botón de aceptar.	Se muestra un mensaje explicando que ya existe un vehículo con dicha matrícula en el sistema.
<b>UIT.5</b>	Introducir Valores, Nombre: Seat, Modelo: León, Combustible: Gasóleo, Matrícula: 1111VVA. A continuación, pulsar el botón de aceptar.	Se muestra un mensaje explicando que la matrícula introducida contiene vocales y no es considerada válida.
<b>UIT.6</b>	Introducir Valores, Nombre: Se@t, Modelo: !', Combustible: Gasóleo, Matrícula: 2222VVV. A continuación, pulsar el botón de aceptar.	Se muestra un mensaje explicando que o el modelo o la marca contienen caracteres extraños y no se consideran como válidos.
<b>UIT.7</b>	Introducir Valores, Nombre: Chevrolet, Modelo: ----, Combustible: Gasóleo, Matrícula: 3333VVV. A continuación, pulsar el botón de aceptar.	Se muestra un mensaje explicando que o el modelo o la marca contienen caracteres extraños y no se consideran como válidos.
<b>UIT.8</b>	Introducir Valores, Nombre: ::::, Modelo: Golf, Combustible: Gasóleo, Matrícula: 4444VVV. A continuación, pulsar el botón de aceptar.	Se muestra un mensaje explicando que o el modelo o la marca contienen caracteres extraños y no se consideran como válidos.