

# Proyecto: CarPlace.



## CARPLACE

### **Miembros del equipo**

Iván Aguilera Calle

Patricia Díez García

Sergio Freire Fernández

Daniel García Moreno

Manuel Hidalgo Lorente

Miguel Jiménez Rodríguez

Verónica Morante Pindado

Isabel Núñez de la Torre

Alejandro Pidal Gallego

# Índice

<b>1. <u>Descripción Material</u></b>	<b>1</b>
<b>2. Javadoc</b>	<b>2</b>
<b>3. Pruebas de caja negra</b>	<b>3-6</b>
<b>4. Calculo de métricas</b>	<b>6-7</b>
<b>Anexo 1: Comentarios para mejorar la asignatura</b>	<b>7</b>

# **1. Descripción material**

El material consta de la Especificación de requisitos software (IEEE 830) en la que detallamos los requisitos que tiene que implementar el sistema y su funcionalidad. También está incluido el plan de proyecto de software (modelo Pressman) que contiene las estimaciones, gestión de riesgos, planificación temporal, recursos, organización del personal y los mecanismos de seguimiento y control de nuestro proyecto. La planificación temporal en formato Ms Project. Los diagramas del proyecto (casos de uso, actividades, secuenciales, clase y despliegue) en formato modelio y PDF, los diagramas están divididos en subsistemas y a su vez por funciones. Por último, nuestra aplicación, CarPlace, en lenguaje java completamente funcional y esta memoria en la que realizamos las pruebas de caja negra y el cálculo de métricas.

## **2. Javadoc**

A continuación se muestra el javadoc de nuestra aplicación, indicando los parámetros de entrada y salida de cada función con una breve descripción sobre su funcionamiento.

Enlace: [Javadoc](#)

## **3. Pruebas de caja negra**

En las pruebas de caja negra vamos a indicar si las funciones descritas en la SRS funcionan y observaciones de los requisitos requeridos. A continuación presentamos las funciones divididas en subsistemas.

### **Subsistema sesión**

#### **1. Función: Log-in**

- Funciona: Sí
- Observaciones: La función cumple con la especificación de la SRS.
  - Cualquier usuario registrado en la base de datos podrá iniciar sesión en el sistema, en caso de no estar registrado se le facilitará un enlace para poder realizar el registro.

#### **2. Función: Log-out**

- Funciona: Sí
- Observaciones: Es una función que se ejecuta como se describe en las SRS y los diagramas UML.
  - El usuario dispondrá de un botón en el menú que le permitirá salir de su sesión, para ello llamará de nuevo a la función anteriormente descrita (Log-in) de forma que desaparecerá la sesión actual y nos devolverá a la pantalla de inicio de sesión, además cambia el campo “logueado” para indicar que el usuario se ha desconectado.

## **Subsistema Usuario**

### **3. Función: Alta Usuario**

- Funciona: Sí.
- Observaciones: La función cumple con la funcionalidad descrita en la SRS.
  - Un usuario que no esté registrado en el sistema podrá darse de alta facilitando sus datos.

### **4. Función: Baja Usuario**

- Funciona: Si.
- Observaciones: La función cumple con la funcionalidad descrita en la SRS.
  - Solo los administradores podrán dar de baja a un usuario.

### **5. Función: Modificar usuario**

- Funciona: Sí.
- Observaciones: La función cumple con la funcionalidad descrita en la SRS.
  - El usuario logueado podrá modificar sus datos. Si el usuario es administrador podrá modificar el usuario que quiera incluso darle permisos de administrador a un usuario o quitárselos.

### **6. Función: Consultar usuarios**

- Funciona: Sí.
- Observaciones: La función cumple con la funcionalidad descrita en la SRS.
  - Solo los usuarios con permisos de administrador podrán consultar a todos los usuarios.

### **7. Función: Mostrar usuario**

- Funciona: Sí.
- Observaciones: La función cumple con la funcionalidad descrita en la SRS.
  - El usuario podrá visualizar sus datos en cualquier momento. Si el usuario tiene permisos de administrador, desde el panel de consultar usuarios, podrá elegir cualquier usuario para visualizar sus datos.
  -

## **Subsistema Alquiler**

### **8. Función: Alta alquiler**

- Funciona: Sí.

- Observaciones: La función cumple con la funcionalidad descrita en la SRS.
  - El usuario logueado en la aplicación puede registrarse varios alquileres.

#### **9. Función: Baja alquiler**

- Funciona: Sí.
- Observaciones: La función cumple con la funcionalidad descrita en la SRS.
  - El usuario logueado en la aplicación puede consultar su lista de alquileres y seleccionar un alquiler de la lista y eliminarlo. Quedando así, el vehículo del alquiler eliminado, disponible.

#### **10. Función: Consultar alquileres**

- Funciona: Sí.
- Observaciones: La función cumple con la funcionalidad descrita en la SRS.
  - El usuario logueado en la aplicación puede consultar su lista de alquileres. Si el usuario es administrador puede consultar los alquileres de otros usuarios desde el panel de consultar usuarios.

#### **11. Función: Mostrar alquiler**

- Funciona: Sí.
- Observaciones: La función cumple con la funcionalidad descrita en la SRS.
  - El usuario puede acceder a mostrar un alquiler seleccionándolo de la lista de alquileres desde el panel consultar alquileres y pulsando sobre el botón mostrar alquiler.

### **Subsistema reservas**

#### **12. Función: Alta reserva**

- Funciona: Sí.
- Observaciones: La función cumple con la funcionalidad descrita en la SRS.
  - El usuario logueado en la aplicación puede registrarse varias reservas.

#### **13. Función: Baja reservas**

- Funciona: Sí.
- Observaciones: La función cumple con la funcionalidad descrita en la SRS.
  - El usuario logueado en la aplicación puede consultar su lista de reservas y seleccionar una reserva de la lista y eliminarla. Quedando así, el vehículo de la reserva eliminado, disponible.

#### **14. Función: Modificar reservas**

- Funciona: Sí.

- **Observaciones:** La función cumple con la funcionalidad descrita en la SRS.
  - El usuario logueado en la aplicación puede consultar su lista de reservas y seleccionar una reserva de la lista y presionar el botón mostrar reserva. Si en el panel mostrar reserva el usuario busca otro vehículo (solamente se habilita el botón de modificar reserva si selecciona otro vehículo, las fechas de la reserva no se pueden modificar) entonces se activa el botón modificar reserva, cambiando el vehículo de la reserva.

#### **15. Función: Consultar reservas**

- **Funciona:** Sí.
- **Observaciones:** La función cumple con la funcionalidad descrita en la SRS.
  - El usuario logueado en la aplicación puede consultar su lista de reservas. Si el usuario es administrador puede consultar las reservas de otros usuarios desde el panel consultar usuarios.

#### **16. Función: Mostrar reservas**

- **Funciona:** Sí.
- **Observaciones:** La función cumple con la funcionalidad descrita en la SRS.
  - El usuario puede acceder a mostrar una reserva seleccionándolo de la lista de reservas desde el panel consultar reservas y pulsando sobre el botón mostrar reserva.

### **Modulo vehículos**

#### **17. Función: Alta vehículo**

- **Funciona:** Sí
- **Observaciones:** la función cumple con la funcionalidad descrita en la SRS.
  - Si el vehículo nuevo tiene una matrícula ya existente no se podrá hacer el alta, tal y como se documenta en la SRS, así como si algún campo queda vacío.

#### **18. Función: Baja vehículo**

- **Funciona:** Sí
- **Observaciones:** la función cumple con la funcionalidad descrita en la SRS.
  - El usuario selecciona de la lista un vehículo y al darle al botón eliminar este queda eliminado de la BBDD.

#### **19. Función: Modificar vehículo**

- Funciona: Sí
- Observaciones: la función cumple con la funcionalidad descrita en la SRS.
  - El usuario, cuando ha seleccionado un vehículo y le ha dado a mostrar, desde aquí le puede haber dado al botón modificar que permite cambios en los campos de un vehículo y al darle guardados estos quedarán reflejados en la BBDD.

## **20. Función: Consultar vehículos**

- Funciona: Sí
- Observaciones: la función cumple con la funcionalidad descrita en la SRS.
  - El usuario, tras seleccionar unos filtros y darle a consultar se visualizará una lista de vehículos que cumplen con los filtros.

## **21. Función: Mostrar vehículo**

- Funciona: Sí
- Observaciones: la función cumple con la funcionalidad descrita en la SRS.
  - El usuario, tras seleccionar un vehículo de la lista y darle a mostrar saldrán todos los datos de un vehículo.

## **4. Cálculo de métricas**

Tras haber acabado el proyecto vamos a calcular las métricas, para las LDC vamos a contar todas las líneas excluyendo las líneas utilizadas para el javadoc. Las LDCJ vamos a contar las líneas utilizadas para generar el javadoc. Por último para las pagDoc vamos a contar el número de páginas de los diagramas UML, plan de proyecto y SRS.

### **LDC totales (sin comentarios para la documentación):**

- **Reservas: 2049**
- **Alquileres: 1570**
- **Sesión: 455**
- **Usuarios: 1246**
- **Vehículos: 2347**

**Total: 7667**

**LDCJ:**

- **Reservas: 265**
- **Alquileres: 214**
- **Sesión: 100**
- **Usuarios: 117**
- **Vehículos: 360**

**Total: 1056**

Para el cálculo de las métricas vamos a calcular LDC/pm, como hemos estado trabajando durante dos meses 9 personas  $7667/18=425,9$  LDC/pm.

El número de líneas utilizadas para el javadoc 1056 y LDC 7667, conlleva una métrica de  $1056/7667=0.14$  LDCJ/LDC.

**PagDoc:**

- **PP: 39 páginas.**
- **SRS:24 páginas**
- **UML: 35 páginas.**

**Total: 98 páginas.**

Por último, calculamos la métrica de pagDoc/KLDC, como KLDC es 7,6  $98/7,6=12,89$  pagDoc/KLDC.

## **Anexo 1: Comentarios para mejorar la asignatura**

- La asignatura debería estar en tercero, sobre todo para tener más conocimientos en bases de datos.
- Más ejemplos de diagramas en las diapositivas en vez de darlo casi todo como teoría
- Mayor realización de diagramas en clase, ya que la primera vez que nos enfrentamos a ellos es en el proyecto y no sabemos cómo son exactamente.