Proyecto: CarPlace.

Especificación de Requisitos Software (Formato IEEE Std. 830-1998)

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 Galleg

Índice

1. Introducción

Este documento es una Especificación de Requisitos Software (ERS) para el proyecto CarPlace.

El contenido ha sido estructurado siguiendo el modelo "IEEE Recommended Practice for Software Requirements Specification ANSI/IEEE 830 1998".

1.1 Propósito

El propósito de la creación de esta especificación de requisitos es la definición de manera clara y concisa de las funcionalidades y restricciones del proyecto.

Así mismo, entre los objetivos destacados de la creación de esta SRS son: la de establecer un acuerdo entre clientes y desarrolladores sobre qué debe hacer el programa, conseguir reducir el esfuerzo de desarrollo, aportar una base para conseguir realizar una estimación fiable de los costes y planificación, proporcionar una guía para la validación y verificación, conseguir facilitar la transferencia del programa o también servir como base para mejoras posteriores.

Nuestro proyecto se verá dirigido a diversos públicos, principalmente a aquellos que se vean en la necesidad de alquilar un vehículo por diversas razones, ya sean: el fallo de su propio vehículo o el alquiler de limusinas para fiestas, así mismo, va dirigido a aquellas personas extranjeras que no dispongan de vehículos y lo necesiten por un período de tiempo concreto.

Una vez realizada esta especificación con sus correspondientes revisiones, servirá como base para el posterior desarrollo del proyecto.

1.2 Ámbito del Sistema

Para la realización de este proyecto hemos decidido darle el nombre de CarPlace, debido a la simplicidad y a la fácil memorización.

Nuestro sistema realizará diversas acciones:

En primer lugar, tendremos la interfaz del administrador, el cual podrá realizar las acciones de compra-venta de vehículos. Por otro lado tenemos la interfaz del usuario, que realizará la acción de alquiler de los mismos.

Además se podrá optar por la selección de reservar o alquilar un coche, moto o limusina específico ya que dentro de los cuales se realizarán diversas filtraciones entre las cuales son destacables: el modelo, la marca del vehículo, tipo de carburante y número de plazas.

El objetivo principal de la realización de esta aplicación, será poder prestar unos servicios a aquellas personas que no se puedan permitir el mantenimiento continuado de un vehículo a largo plazo, o a aquellas personas que no tengan la disponibilidad de tener un vehículo de alta gama como puede ser una limusina.

Por otro lado, el beneficio de la realización de este proyecto será con fines académicos. Principalmente la meta de la creación de este proyecto será poder desarrollar los conocimientos necesarios para conseguir gestionar el desarrollo de una aplicación de forma eficaz.

Para poder proceder a la utilización de esta aplicación tanto el usuario como el administrador deberán tener un ordenador mediante el cual accederán a la interfaz gráfica de la aplicación, la cual permitirá la posibilidad de gestionar la utilización de los recursos proporcionados.

1.3 Definiciones, Acrónimos y Abreviaturas

1.3.1.- Definiciones

- Alquiler: Procedimiento por el cual un vehículo pasa a estar disponible para el uso de un usuario durante un tiempo predefinido.
- Reservar: Procedimiento por el cual un vehículo queda pedido para su posterior alquiler.
- Usuario: persona encargada en realizar la reserva o el alquiler de los vehículos.
- Administrador: persona cuyo encargo será el de realizar las tareas de compra y venta de vehículos, así como el proceso de transacción de alquiler a los clientes (usuarios).

1.3.2.- Acrónimos

ERS	Especificación de Requisitos Software
UML	Unified Modeling Language
GUI	Interfaz Gráfica de Usuario

1.4 Referencias

- IEEE Recommended Practice for Software Requirements Specification. ANSI/IEEE std. 830, 1998.
- Apuntes de la asignatura.

1.5 Visión General del Documento

Este documento se divide en tres partes:

A lo largo de la primera parte se desarrolla la introducción al ERS proporcionándose una visión general de los recursos del sistema.

Durante la segunda parte observamos una descripción general del proyecto, en el cual están incluidas las perspectivas y funciones del producto así como las características de los usuarios además de restricciones, supuestos y dependencias o requisitos futuros.

Finalmente en la tercera parte encontramos los detalles y requisitos que debe de satisfacer el proyecto.

2. Descripción General

En este punto se especificarán brevemente las perspectivas y funciones del producto, así como futuras funcionalidades y restricciones que pueda sufrir el mismo.

2.1. Perspectiva del Producto

El producto está destinado a fines únicamente académicos, por lo tanto es totalmente independiente de otros productos.

Por su condición independiente este producto no forma parte de otros mayores, por lo que será necesario definir sus propias interfaces y módulos.

2.2. Funciones del producto

Las funcionalidades solicitadas por el cliente están enfocadas al manejo y desarrollo de las distintas funcionalidades de la aplicación para facilitar el uso de esta al usuario. Entre las funciones más destacadas tenemos el registro y log-in del usuario con las cuales podrá acceder a las funcionalidades propias de la aplicación, funciones para filtrar y ordenar por la categoría que se desee, reservar o alquilar vehículos.

Así como un usuario estándar habrá un administrador que podrá gestionar las bases de datos de los vehículos de la empresa.

2.3. Características de los Usuarios

Esta aplicación está enfocada a los clientes de la empresa.

No necesitarán grandes conocimientos en informática, con conocimientos básicos en informática a nivel de usuario será suficiente, puesto que desarrollaremos una interfaz sencilla, intuitiva y fácil de manejar.

2.4. Restricciones

Nuestro proyecto es un proyecto académico y por tanto las restricciones son muy específicas. No se relacionará con otras aplicaciones por ser una aplicación independiente por lo que no tendrá restricciones en este aspecto.

Sin embargo, la principal restricción será el lenguaje de programación, ya que solo podremos utilizar JAVA, teniendo que usar polimorfismo. Además, también estará

restringido por los contenidos que debemos aplicar a nuestro proyecto y por las indicaciones del profesor.

Otra de las restricciones será aplicar técnicas de Ingeniería del Software, así como el uso de interfaces gráficas y otras directrices marcadas por el profesor como son el seguimiento de los patrones DAO, SA, fachada y transfer.

Así mismo, nos limitará el proyecto nuestros propios conocimientos, ya que al tener poca experiencia con proyectos de ingeniería del software y nuestros escasos conocimientos en programación JAVA y en BBDD pueden llegar a limitar la funcionalidad de nuestro programa.

En cuanto al software, contamos con la limitación de que debe ser un software de escritorio, no web, y debe contar con una interfaz gráfica de usuario (GUI).

Respecto a las restricciones de seguridad, cada usuario está registrado con su propia contraseña para evitar que pueda acceder a los datos de otros usuarios.

2.5. Suposiciones y Dependencias

Nuestro proyecto podía variar a lo largo del tiempo ya que al ser un proyecto académico está sujeto a cambios en los conocimientos, los contenidos o incluso al personal disponible para la realización del mismo.

Al utilizar JAVA para programar, un lenguaje multiplataforma, el sistema operativo debería ser independiente y no suponer un riesgo que implique un cambio en la aplicación.

Además el proyecto está sujeto a futuros cambios en los requisitos una vez empezado, debido a que nuestra inexperiencia puede llevarnos a cometer errores y eso supondría cambios tanto en el futuro programa como en la SRS o el Plan de Proyecto.

En resumen, los puntos principales que pueden variar a lo largo del desarrollo del proyecto y aquellos que podrían suponer cambios en el mismo son: la experiencia y conocimiento de los desarrolladores, el personal disponible para realizarlo y los propios cambios en el proyecto por errores no detectados a tiempo.

2.6. Requisitos Futuros

En este punto analizaremos las posibles mejoras de nuestra aplicación. Estas pueden ser muchas pero trataremos unas pocas:

Se podría mejorar la aplicación para adaptarla a su uso web. Con esto también se podría crear una versión para el móvil que permitiera a los usuarios alquilar un vehículo desde sus teléfonos.

Se podrían crear funciones para comparar coches, o funciones que muestren imágenes del vehículo.

Se podrían implementar funciones que premien con descuentos u otros elementos a los clientes que usen la aplicación, por ejemplo cada 4 alquileres el 5º tiene un 15% de descuento, etc.

Se podría adaptar a varios idiomas para el uso por usuarios de otros países o comunidades autónomas con idioma propio.

3. Requisitos específicos

Esta sección contiene todos los requisitos del sistema, siendo la parte más importante de la SRS. En ella se hace referencia a la información de los dos primeros puntos, siendo esta más detallada. Gracias a este apartado los diseñadores van a poder hacer una serie de pruebas para comprobar si la aplicación satisface los requisitos.

3.1 Interfaces externas

En este apartado vamos a presentar las GUI de la aplicación con sus correspondientes entradas y salidas y una breve descripción.

Subsistema sesión

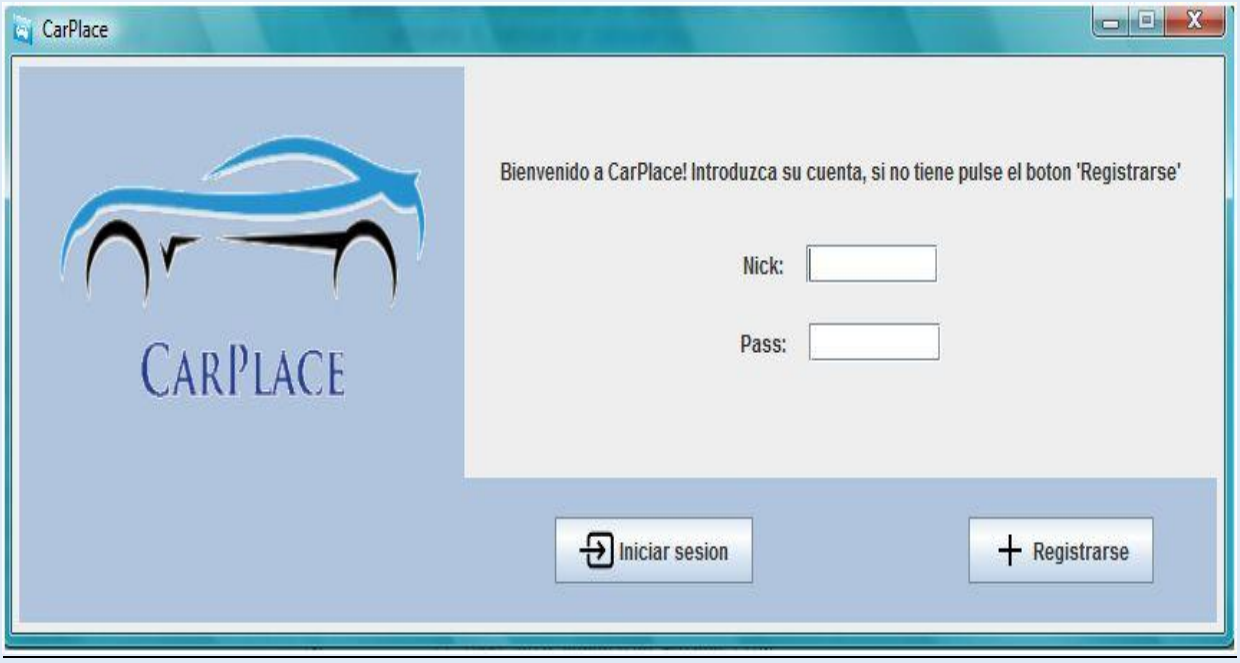

Figura1.Log-in

Permite a un usuario iniciar sesión introduciendo su contraseña y Nick, o también registrarse en la aplicación.

Figura 2.Menú principal

El usuario puede acceder a todas las opciones de la aplicación como alta reserva, alta alquiler, mostrar, modificar y consultar un alquiler o reserva. Si eres administrador puedes añadir, modificar y eliminar un usuario o un vehículo.

Subsistema usuario

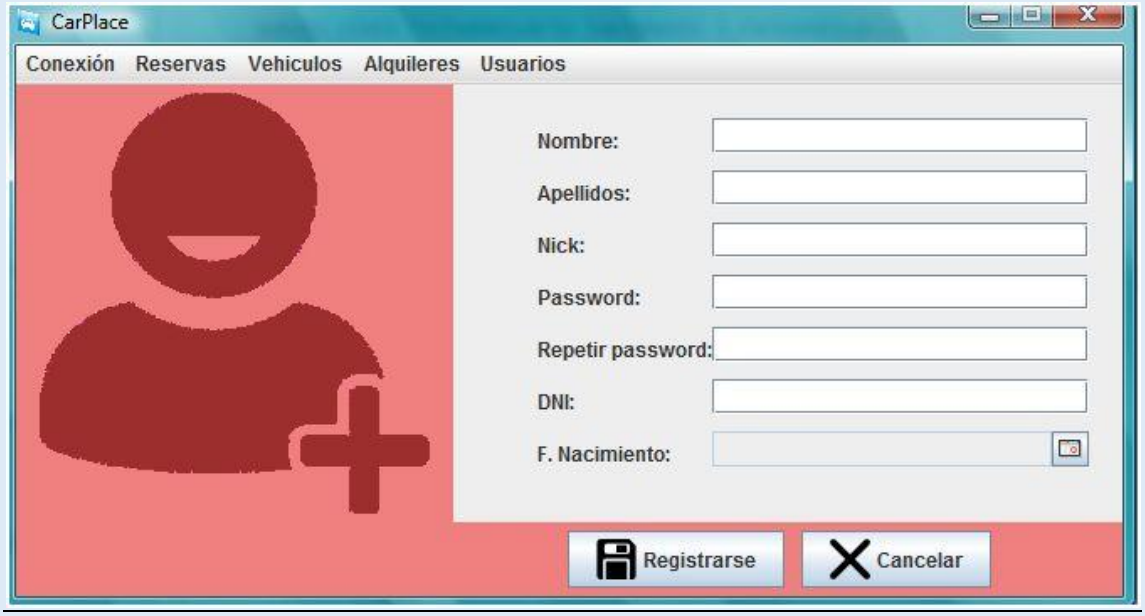

Figura 3.Alta usuario

Un usuario se puede dar de alta en la aplicación introduciendo su nombre, apellidos, Nick, password, DNI y su fecha de nacimiento.

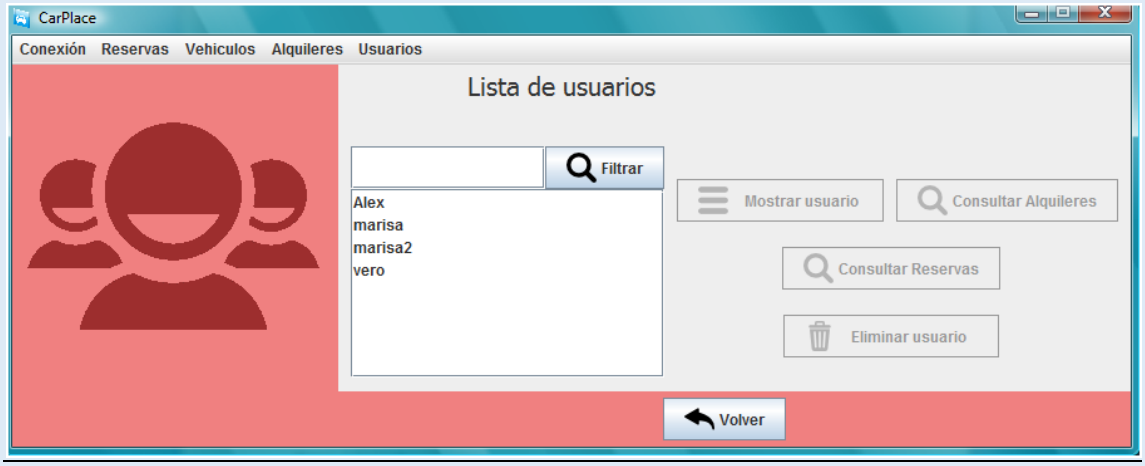

Figura 4.Consultar usuarios

Permite a un administrador mostrar, eliminar y modificar un usuario y mostrar sus alquileres y reservas.

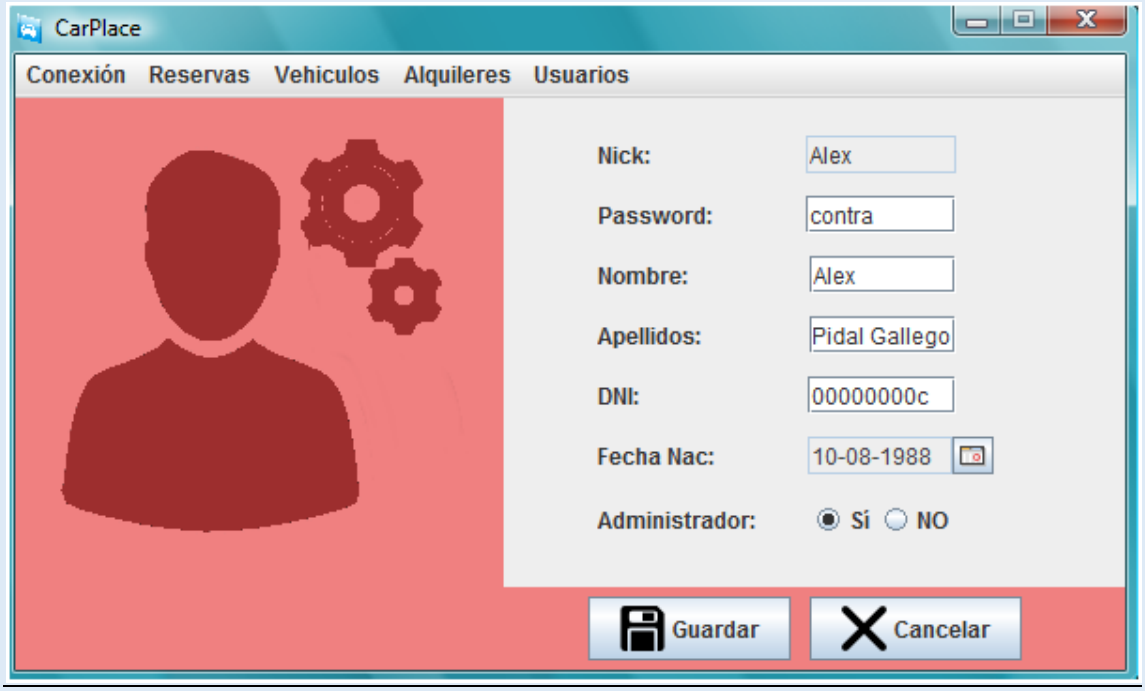

Figura 5.Mostrar usuario

Muestra la información de un usuario (Nick, nombre, apellidos DNI y fecha de nacimiento). El usuario puede modificar alguno de sus datos y guardarlos.

Subsistema vehículo

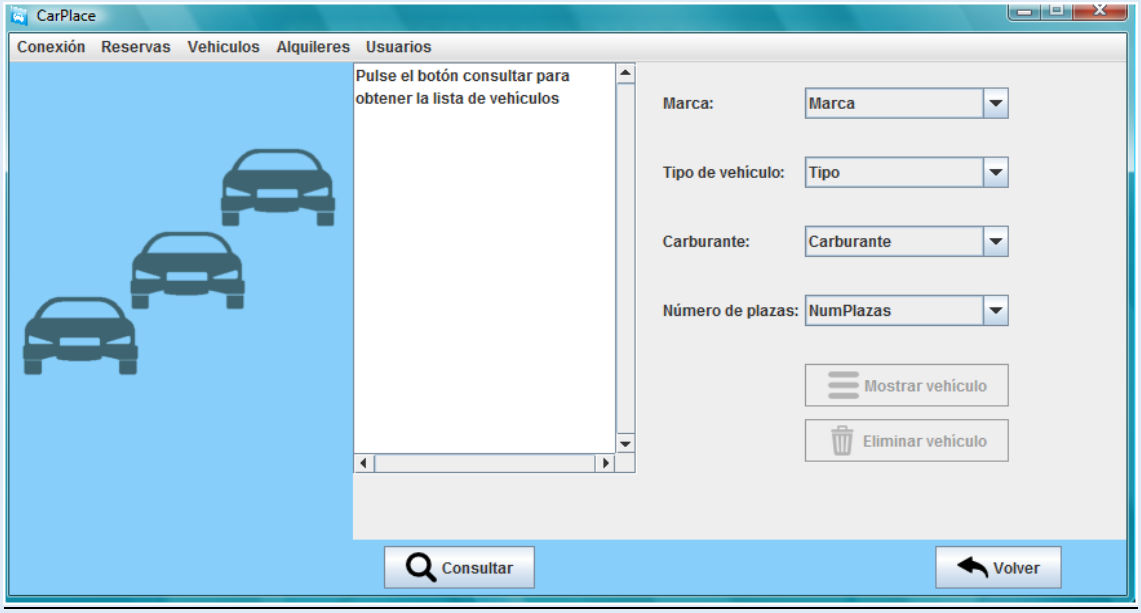

Figura 6.Consultar vehículos

Muestra una lista de vehículos con los filtros introducidos (marca, tipo de vehículo, carburante y número de plazas). Al seleccionar uno se puede eliminar, modificar y mostrar.

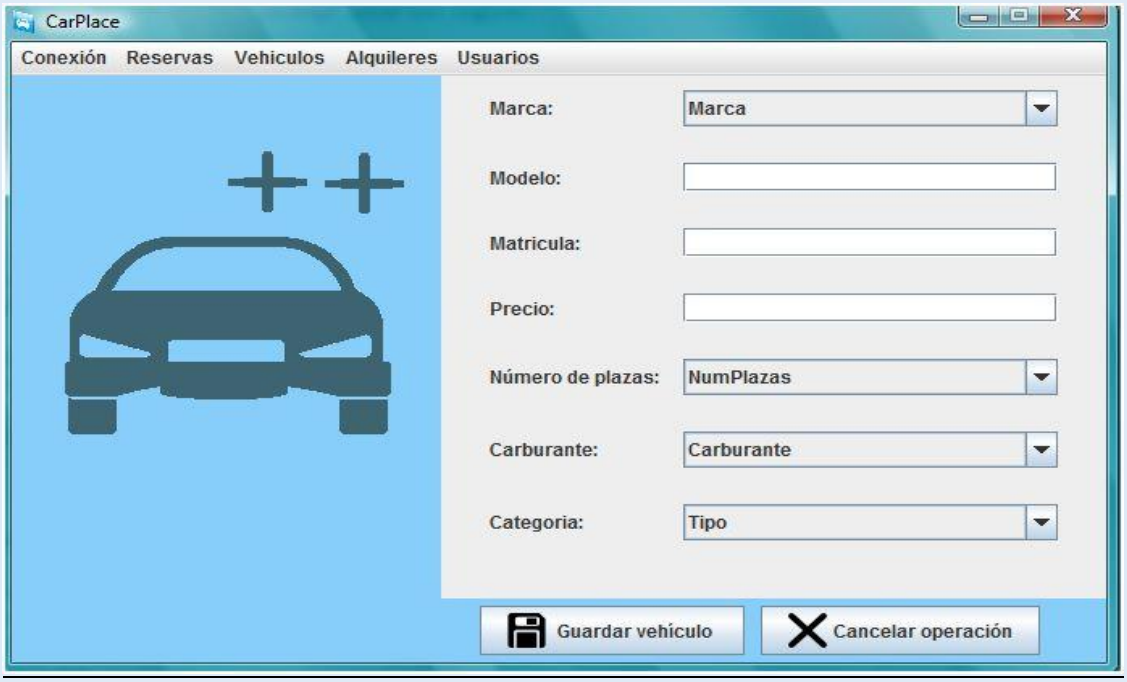

Figura 7.Alta vehículo

Ventana que permite dar de alta a un vehículo introduciendo su marca, modelo, matricula, precio, número de plazas, carburante y categoría.

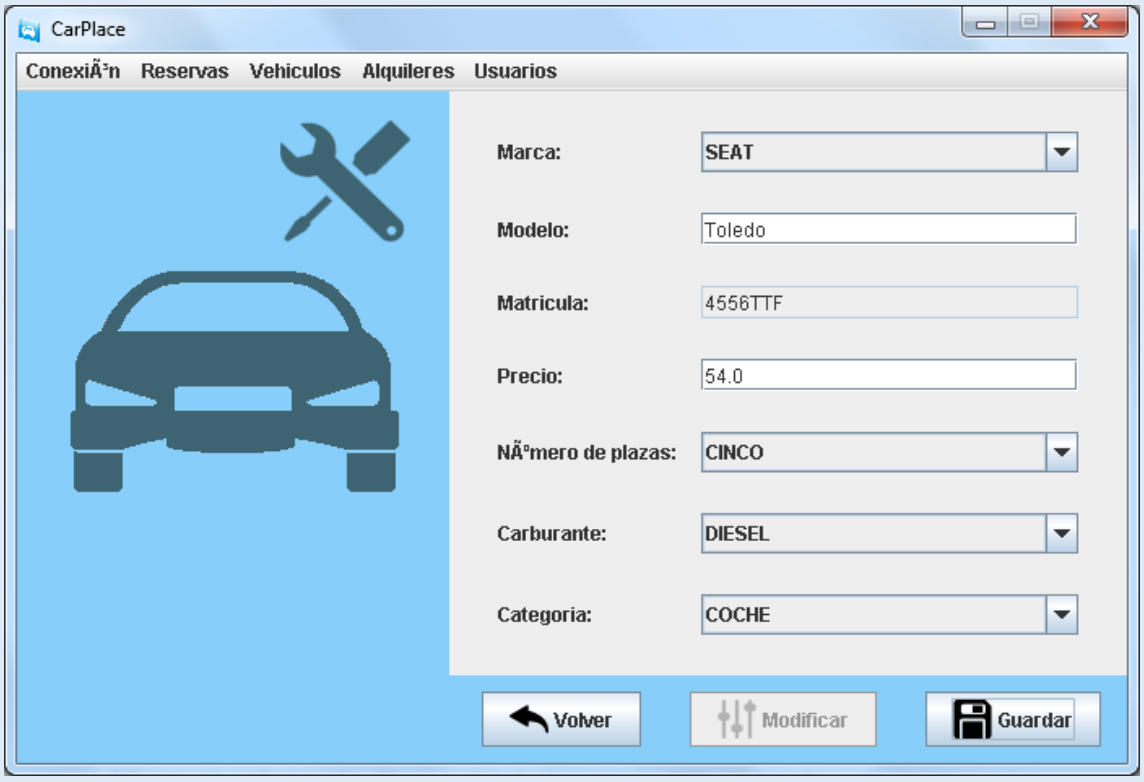

Figura 8.Mostrar vehículo

Muestra la información de un vehículo (marca, modelo, matricula, carburante, precio, número de plazas y categoría).El administrador puede modificar alguno de sus datos y guardarlos.

Subsistema alquiler

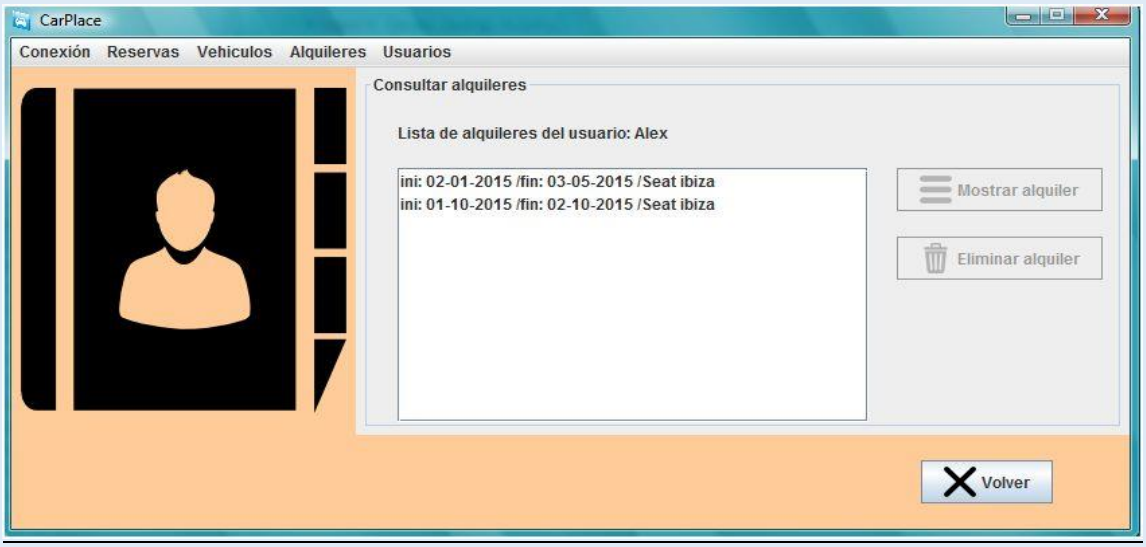

Figura 9.Consultar alquileres

Muestra la lista de alquileres de un usuario concreto, donde se puede mostrar y eliminar un alquiler.

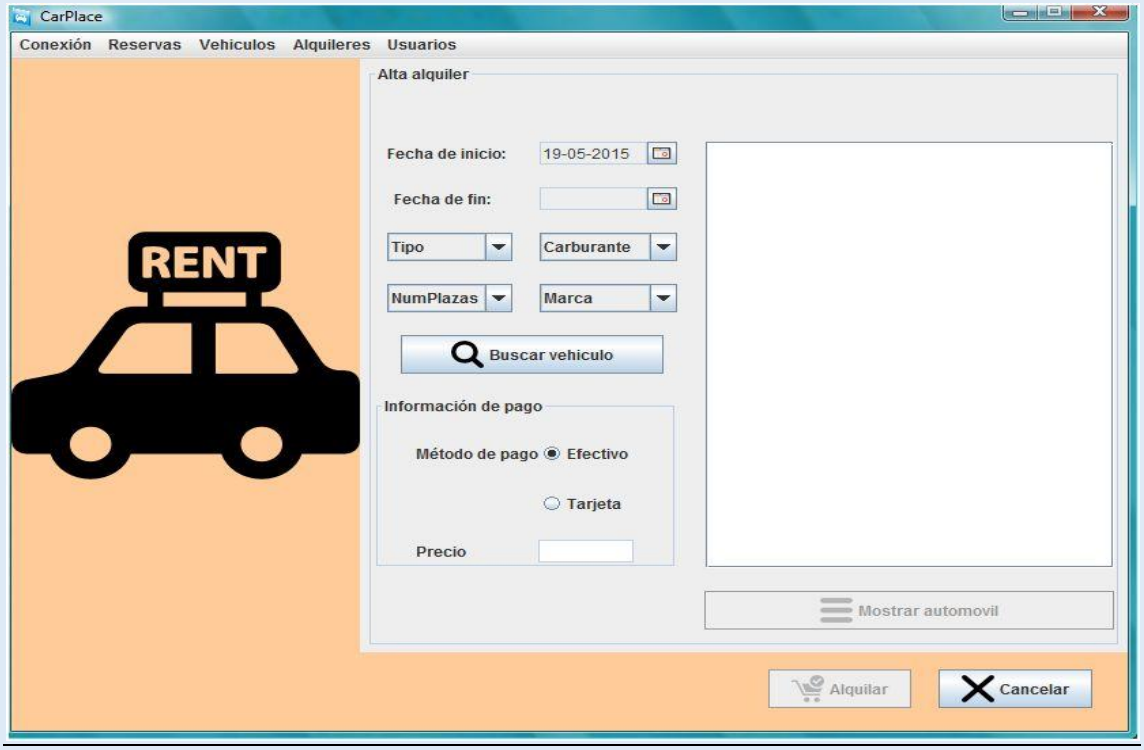

Figura 10.Alta alquiler

Permite dar de alta un alquiler, con su fecha de inicio y de fin, filtrar los vehículos disponibles, la información sobre el tipo de pago y el precio total de la operación.

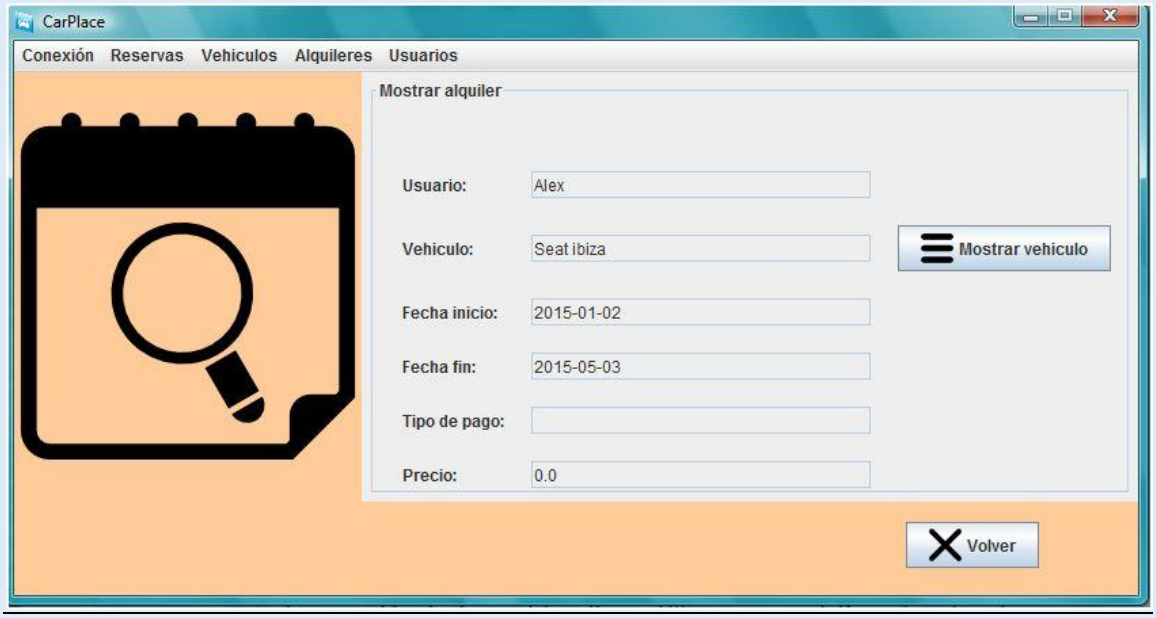

Figura 11.Mostrar alquiler

Ventana que muestra la información de un alquiler con el nombre de usuario al que pertenece y permite ver el vehículo alquilado.

Subsistema reservas

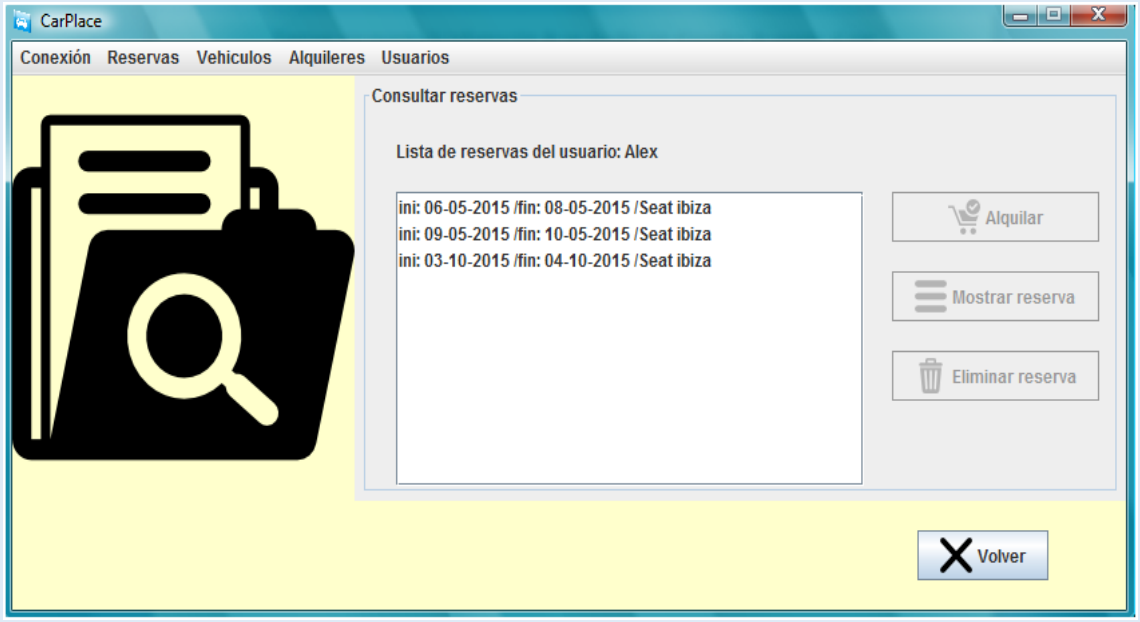

Figura 12.Consultar reservas

Muestra las reservas de un usuario concreto, pudiéndola eliminar, modificar, mostrar o confirmar dicha reserva dando a alquilar.

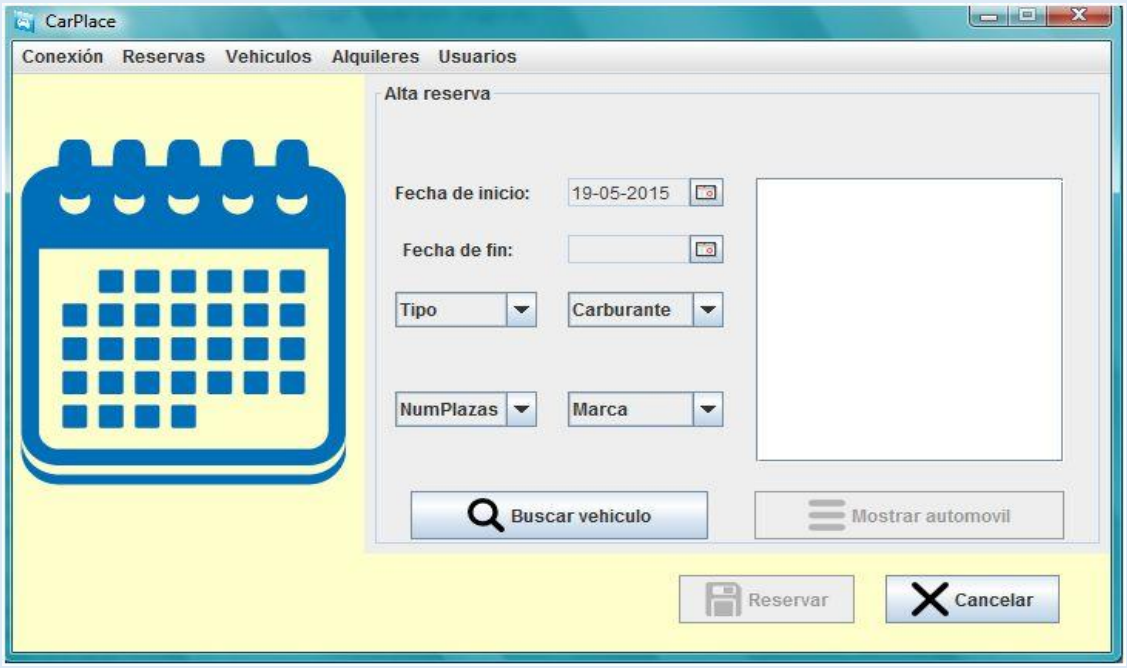

Figura 13.Alta reserva

Permite dar de alta una reserva introduciendo su fecha de inicio y fin, filtrar un vehículo de los disponibles y que se muestren en una lista para poder elegir uno.

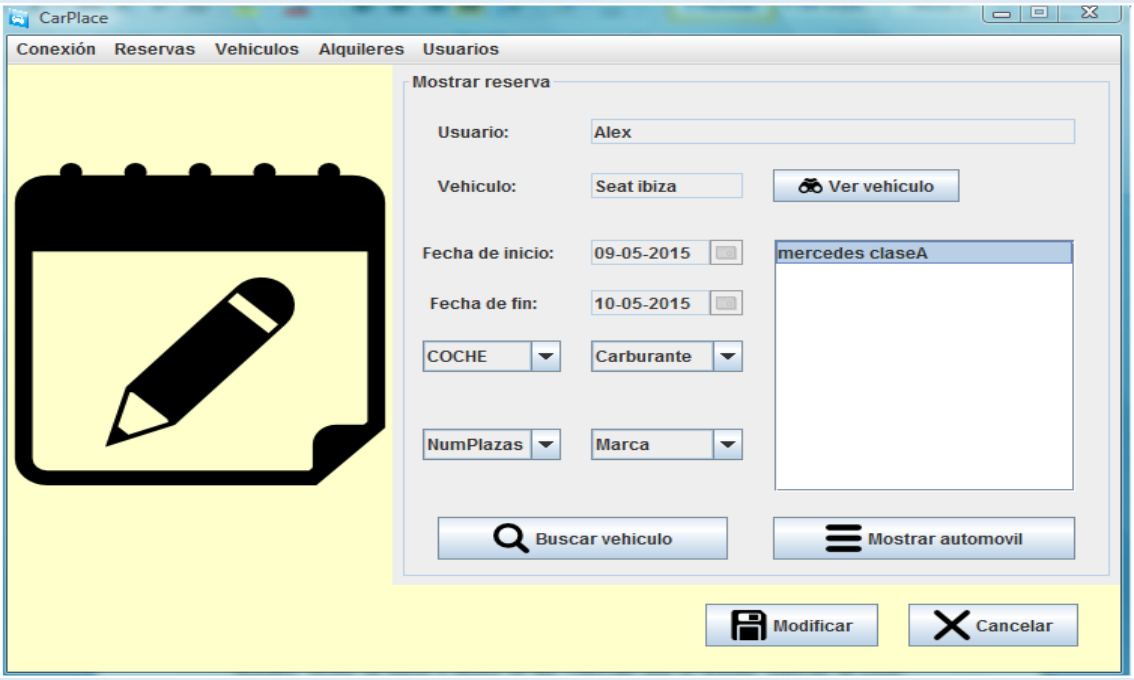

Figura 14.Mostrar reserva

Muestra la información de una reserva con el usuario de la reserva, el vehículo y la fecha de inicio y fin de la misma. El usuario puede modificar los datos de la reserva y si lo desea guardarlos.

Entradas:

Usuario: password, repassword, nombre, apellidos, fecha de nacimiento, Nick, DNI como campos obligatorios para registrarse.

Vehículo: categoría (coche, moto y limusina), marca (BMW, Renault, Opel, Peugeot, Seat), modelo, matrícula, carburante (diesel y gasolina), número de plazas (1, 2,3, 4, 5,+5) y precio.

Alquiler: fecha de inicio y fecha de fin, vehículo que se alquila, método de pago, usuario que realiza el alquiler y el precio total del alquiler.

Reservas: fecha de inicio y fin, vehículo reservado y Nick del usuario que realiza la reserva.

Salidas:

-Si el usuario no introduce un campo obligatorio para el registro o el administrador al añadir un coche “Por favor, rellene todos los campos”.

-Si al intentar acceder no eres válido “Su nombre o contraseña no son correctas”.

-Si introduces un nombre de usuario ya existente al registrarse “El nombre de usuario ya existe, intente otro”.

-El usuario al introducir una contraseña y al repetirla no coinciden “Las contraseñas no coinciden”.

3.2 Funciones

En esta sección se especifican las funciones que vamos a usar en nuestra aplicación, agrupándolas por distintas funcionalidades (sesión, usuarios, administrador…) y describimos varios apartados de cada función de la aplicación.

a) Sesión:

1. Función: Log-in

- Prioridad: Alta.
- Estabilidad: Alta.
- Descripción: Permite identificarte como usuario o administrador en la aplicación.
- Entrada: Nick del usuario y su respectiva contraseña.
- Salida: Ventana emergente de bienvenida a la aplicación.
- Precondición: Usuario no registrado, Nick y contraseña válidos.
- Post condición: El usuario pasa a estar logueado en la aplicación.
- Efectos laterales: Se muestra la ventana de la aplicación.
- Usuarios: Usuario y administrador

2. Función: Log-out

- Prioridad: Alta.
- Estabilidad: Alta.
- Descripción: Permite cerrar la sesión del usuario actual.
- Entrada: -
- Salida: Ventana emergente de salida de la aplicación.
- Precondición: Tiene que haber un usuario logueado.
- Post condición: El usuario logueado deja de estar logueado.
- Efectos laterales: Se muestra la ventana de inicio de sesión.
- Usuario: Usuario y administrador.

b) Funciones usuario:

3. Función: Alta Usuario

- Prioridad: Alta.
- Estabilidad: Alta.
- Descripción: Permite dar de alta un usuario en la aplicación.
- Entrada: Usuario (detallado en la sección 3.1).
- Salida: Ventana emergente de confirmación del registro.
- Precondición: Datos correctos (Nick entre 4 y 16 caracteres y que no exista en la base de datos, además el DNI tiene que ser uno auténtico).
- Post condición: Aplicación con el usuario nuevo registrado.
- Efectos laterales: Vuelve a la ventana del menú principal.
- Usuarios: Administrador y usuario.

4. Función: Baja Usuario

- Prioridad: Alta.
- Estabilidad: Alta.
- Descripción: Permite dar de baja a un usuario en la aplicación.
- Entrada: Usuario a eliminar de la aplicación.
- Salida: Ventana emergente confirmación de la baja.
- Precondición: Usuario existente en la aplicación.
- Post condición: Aplicación sin el usuario eliminado.
- Efectos laterales: Vuelve a la ventana de mostrar usuario.
- Usuarios: Administrador.

5. Función: Modificar usuario

- Prioridad: Alta.
- Estabilidad: Media.
- Descripción: Permite modificar sus datos.
- Entrada: Usuario a modificar.
- Salida: Ventana emergente de confirmación de la modificación realizada.
- Precondición: Usuario logueado.
- Post condición: Usuario actualizado en la base de datos.
- Efectos laterales: Vuelve a la ventana del menú principal.

- Usuarios: Administrador y usuario.

6. Función: Consultar usuario

- Prioridad: Alta.
- Estabilidad: Media.
- Descripción: Permite realizar una búsqueda de usuarios mediante un filtro de nombres.
- Entrada: Filtro de búsqueda.
- Salida: Lista de usuarios cuyos datos coinciden con los filtros establecidos.
- Precondición: -
- Post condición: -
- Efectos laterales: -
- Usuarios: Administrador.

7. Función: Mostrar usuario

- Prioridad: Media.
- Estabilidad: Media.
- Descripción: Muestra los datos del usuario.
- Entrada: Usuario.
- Salida: La aplicación muestra los datos del usuario.
- Precondición: Usuario existente.
- Post condición: -
- Efectos laterales: -
- Usuarios: Usuario y administrador.

c) Funciones alquiler:

8. Función: Alta alquiler

- Prioridad: Alta.
- Estabilidad: Alta.
- Descripción: Permite añadir un nuevo alquiler en la aplicación.
- Entrada: Datos del alquiler a incorporar a la aplicación (Descritos en la sección 3.1).
- Salida: Ventana emergente confirmación del registro del alquiler.

- Precondición: Datos del alquiler correctos.
- Post condición: Aplicación con los datos del alquiler incorporados al sistema.
- Efectos laterales: -
- Usuarios: Usuario y administrador.

9. Función: Baja alquiler

- Prioridad: Alta.
- Estabilidad: Alta.
- Descripción: Permite dar de baja un alquiler en la aplicación.
- Entrada: Identificador del alquiler a eliminar de la aplicación.
- Salida: Ventana emergente confirmación de la eliminación.
- Precondición: Datos del alquiler correctos.
- Post condición: Aplicación sin el alquiler eliminado.
- Efectos laterales: -
- Usuarios: Usuario y administradores.

10. Función: Consultar alquiler

- Prioridad: Alta.
- Estabilidad: Media.
- Descripción: Permite realizar una búsqueda de alquileres mediante un Nick.
- Entrada: Identificador de usuario con el cual filtrar los alquileres.
- Salida: Lista de alquileres cuyos datos coinciden con los filtros establecidos.
- Precondición: -
- Post condición: Alquileres cuyos datos coinciden con los filtros establecidos.
- Efectos laterales: -
- Usuarios: Administrador y usuario.

11. Función: Mostrar alquiler

- Prioridad: Alta.
- Estabilidad: Alta.
- Descripción: Muestra la información del alquiler.
- Entrada: Identificador del alquiler.
- Salida: La aplicación muestra los datos del alquiler.

- Precondición: Alquiler existente.
- Post condición: -
- Efectos laterales: -
- Usuarios: Usuario y administrador.

d) Funciones reservas:

12. Función: Alta reservas

- Prioridad: Media.
- Estabilidad: Alta.
- Descripción: Permite hacer una nueva reserva.
- Entrada: Datos de la reserva a incorporar a la aplicación (Descritos en la sección 3.1).
- Salida: Ventana emergente de confirmación de la reserva realizada.
- Precondición: Los datos la reserva son correctos.
- Post condición: Aplicación con los datos de la reserva incorporados al sistema.
- Efectos laterales: -
- Usuarios: Usuario y administrador.

13. Función: Baja reservas

- Prioridad: Media.
- Estabilidad: Alta.
- Descripción: Permite dar de baja una reserva en la aplicación.
- Entrada: Identificador de la reserva a eliminar de la aplicación.
- Salida: Ventana emergente de confirmación de la baja de la reserva realizada.
- Precondición: Datos de la reserva correctos.
- Post condición: Aplicación con la reserva eliminada.
- Efectos laterales: -
- Usuarios: Usuario y administrador.

14. Función: Modificar reservas

- Prioridad: Media.
- Estabilidad: Alta.
- Descripción: Permite modificar una reserva.

- Entrada: Reserva a modificar.
- Salida: Ventana emergente de confirmación de la modificación de la reserva realizada.
- Precondición: Reserva existente.
- Post condición: Aplicación con la reserva actualizada.
- Efectos laterales: -
- Usuarios: Usuario y administrador.

15. Función: Consultar reservas

- Prioridad: Media.
- Estabilidad: Alta.
- Descripción: Permite realizar una búsqueda de reservas que filtra las reservas por un identificador de usuario.
- Entrada: Identificador del usuario por el que buscar las reservas.
- Salida: Lista de reservas cuyos datos coinciden con el identificador de usuario.
- Precondición: -
- Post condición: Reservas cuyos datos coinciden con el identificador de usuario.
- Efectos laterales: -
- Usuarios: Usuario y administrador.

16. Función: Mostrar reservas

- Prioridad: Media.
- Estabilidad: Alta.
- Descripción: Muestra los datos de la reserva.
- Entrada: Identificador de la reserva.
- Salida: La aplicación muestra los datos de la reserva.
- Precondición: Reserva existente.
- Post condición: -
- Efectos laterales: -
- Usuarios: Usuario y administrador.

e) Funciones vehículo:

17. Función: Alta vehículo

- Prioridad: Alta.
- Estabilidad: Alta.
- Descripción: Permite dar de alta vehículos en la aplicación.
- Entrada: Vehículo (detallado en la sección 3.1.).
- Salida: Ventana emergente de confirmación de alta de vehículo.
- Precondición: Datos correctos (matrícula no existente en la BBDD).
- Post condición: Aplicación con el nuevo vehículo registrado.
- Efectos laterales: -
- Usuarios: Administrador.

18. Función: Baja vehículo

- Prioridad: Media.
- Estabilidad: Alta.
- Descripción: Permite dar de baja un vehículo.
- Entrada: Datos del vehículo que se va a dar de baja.
- Salida: -
- Precondición: Vehículo existente y disponible.
- Post condición: Aplicación sin el vehículo eliminado.
- Efectos laterales: -
- Usuarios: Administrador

19. Función: Modificar vehículo

- Prioridad: Media.
- Estabilidad: Alta.
- Descripción: Permite modificar los datos de un vehículo.
- Entrada: Vehículo a modificar.
- Salida: Ventana emergente de confirmación de la modificación del vehículo.
- Precondición: Vehículo existente.
- Post condición: Vehículo actualizado en la BBDD.
- Efectos laterales: -
- Usuarios: Administrador.

20. Función: Consultar vehículo

- Prioridad: Media.
- Estabilidad: Alta.
- Descripción: Permite realizar una búsqueda de vehículos mediante los filtros indicados.
- Entrada: Filtros de búsqueda (marca, tipo, carburante y número de plazas).
- Salida: Lista de vehículos cuyos datos coinciden con los filtros establecidos.
- Precondición: -
- Post condición: -
- Efectos laterales: -
- Usuarios: Administrador

21. Función: Mostrar vehículo

- Prioridad: Media.
- Estabilidad: Alta.
- Descripción: Muestra la información de un vehículo.
- Entrada: Vehículo.
- Salida: La aplicación muestra los datos del vehículo.
- Precondición: Vehículo existente.
- Post condición: -
- Efectos laterales: -
- Usuarios: Administrador y usuario.

3.3 Requisitos de rendimiento

En el siguiente apartado vamos a detallar los requisitos de la aplicación. El tiempo para iniciar sesión será de unos cuatro segundos, en los que se cargarán los datos y se verificará que sean correctos y que existan. Las búsquedas serán rápidas, pues la base de datos es pequeña y será cuestión de pocos segundos encontrar los vehículos, ordenarlos y filtrarlos.

3.4 Requisitos lógicos de la BDs

En la tabla siguiente vamos a presentar las relaciones entre las distintas entidades:

	Vehículos	Usuario	Alquiler	Reservas
Vehículos	X	-Puede ser alquilado. -Puede ser comprado. -Puede ser vendido.	-Forma parte del alquiler.	-Forma parte de las reservas
Usuario	-Puede alquilar un vehículo. -Puede comprar un vehículo. -Puede vender un vehículo.	X	-Forma parte del alquiler.	-Forma parte de las reservas.
Alquiler	-El alquiler está formado por vehículos.	-El alquiler está formado por usuarios.	X	-Un alquiler se puede confirmar gracias a una reserva.
Reservas	-Una reserva está formada por vehículos	-Una reserva está formada por usuarios.	-Una reserva puede pasar a ser un alquiler.	X

3.5 Restricciones de diseño

El programa de alquiler de vehículos se ejecutará sobre un solo ordenador, el cual estará localizado en la tienda donde se recogerán los vehículos. La limitación a nivel de hardware estará sujeta al ordenador donde se usará el programa. Existen varias restricciones por el uso de la programación orientada a objetos, así como al uso de patrones (MVC, fachada, DAO, SA y transfer), y a las limitaciones que nos dará el profesor en el segundo cuatrimestre.

3.6 Atributos del sistema software

En esta sección vamos a tratar los atributos de calidad (las "ilities") del sistema, que corresponden a fiabilidad, disponibilidad, seguridad, mantenibilidad y portabilidad. Estos atributos son imprescindibles para hacer un buen software. A continuación vamos a describir cada uno de ellos:

-**Fiabilidad**: el principal objetivo del proyecto es hacer un software 100% fiable, para garantizarlo vamos a realizar diversos casos prueba basados en la SRS para comprobar la funcionalidad del programa. Cada módulo del programa va a ser revisado y comprobado por una persona diferente a la que lo ha realizado para ver posibles fallos en la implementación.

-**Disponibilidad**: Debido a que solo se puede acceder a él a través de un único ordenador, únicamente va a estar disponible en el periodo en el que la tienda se encuentre abierta (9 h-21h) y no haya otro usuario utilizándolo.

-**Seguridad**: este es uno de los aspectos más importantes, cada usuario tiene su propio Nick y una contraseña (figura1) que haya decidido el mismo. Al entrar en la aplicación puede alquilar o reservar un vehículo y si lo desea cambiar la información de su perfil (figura 5). Para acceder como administrador se tiene que tener una contraseña específica, esto le permitirá hacer otras funciones a la que no están autorizados otros usuarios como ver la lista de clientes (puede eliminarlos) o vender y comprar nuevos vehículos.

-**Mantenibilidad**: debido a que es un proyecto universitario, la aplicación no tiene soporte para internet por lo que el cliente no puede comunicarse inmediatamente con los desarrolladores. Como se accede desde un único lugar, se lo puede comunicar al encargado en ese momento. Sin embargo se va a comprobar continuamente el correcto funcionamiento del programa para corregir posibles errores. La mantenibilidad se garantizará gracias a la aplicación de las técnicas de la Ingeniería del Software.

-**Portabilidad**: el programa va a estar disponible para escritorio, es válido para cualquier sistema operativo que soporte JAVA (es multiplataforma) y sólo se puede utilizar en ordenadores. Es un programa de poco tamaño por lo que cualquier equipo lo puede utilizar.