



2014/2015

ESPECIFICACIÓN DE REQUISITOS SOFTWARE



Rubén Barrado González

Sergio García Rodríguez

José María López Pulido

Iván María Paredes

Rodrigo de Miguel González

Tomás Muñoz Testón

Universidad Complutense

Facultad de Informática

ÍNDICE

1. Introducción

1.1. Propósito	4
1.2. Alcance	4
1.3. Definiciones, Acrónimos y Abreviaturas	4
1.4. Referencias	5
1.5. Resumen.....	5

2. Descripción General

2.1. Perspectiva del Producto	6
2.2. Funciones del Producto	6
2.3. Características de los Usuarios	7
2.4. Restricciones	7
2.5. Suposiciones y Dependencias	7
2.6. Requisitos Futuros	8

3. Requisitos Específicos

3.1. Interfaces Externas	9
3.2. Funciones	9
3.3. Requisitos de Rendimiento	61
3.4. Requisitos lógicos de la BBDD	61
3.5. Restricciones de Diseño.....	61
3.6. Atributos del Sistema	64

1. INTRODUCCIÓN

1.1. Propósito

El propósito de este documento es hacer la especificación de requisitos (SRS) siguiendo el estándar IEEE Std. 830-1998 para que nuestro sistema permita el correcto funcionamiento de un gestor automático de parkings.

1.2. Alcance.

Nuestro sistema se llama 'AppArcao' y su principal objetivo es organizar un aparcamiento de forma automática, inteligente y cómoda para el cliente, mediante una BBDD que contenga la información de los socios y sus vehículos, la información relevante de las plazas del aparcamiento, el estado del mismo y las reservas de las plazas.

El programa podrá ser utilizado por el administrador que, entre otras, podrá gestionar los socios, las zonas, las reservas, las plazas y los vehículos.

Nada más comenzar la aplicación se activa un panel administrativo común, es decir, no existe un previo 'login'.

1.3. Definiciones, Acrónimos y Abreviaturas.

SW: Software.

HD: Hardware.

BBDD: Base de Datos.

GUI: Graphical User interface (Interfaz Gráfica de Usuario)

VIP: Very Important Person (Persona Muy Importante)

IEEE: Institute of Electrical and Electronics Engineers (Instituto de ingenieros eléctricos y electrónicos).

1.4. Referencias

Nuestra referencia es el Standard de IEEE 830-1998, 20 Oct. 1998 de la IEEE Computer Society.

1.5. Resumen.

Inicialmente se determina el ámbito de la aplicación, especificando cómo se integra en el sistema. En secciones posteriores se identifican funciones principales del sistema de forma muy general así como restricciones, posibles futuros requisitos y los usuarios a los que va destinada.

Finalmente se hace un análisis exhaustivo de los requisitos haciendo una partición del sistema en las funciones identificadas, así como una visión de cómo podrían ser las interfaces externas de la misma.

2. DESCRIPCIÓN GENERAL

El software permitirá llevar a cabo tanto la gestión automática e inteligente de un aparcamiento. Tanto los usuarios ‘socios’ y ‘no socios’ podrán reseras plazas en el parking siempre y cuando haya libres en el mismo día o reservar plazas para otro día.

2.1. Perspectiva del Producto

El sistema que vamos a desarrollar es para un proyecto universitario de la asignatura ‘ingeniería del software’ por lo que es independiente ya que no lo relacionaremos con otros productos.

2.2. Funciones del Producto

2.2.1. Gestión de plazas

El usuario final podrá manejar toda la información de las plazas. Podrá seleccionar que tipo de plaza es, a que zona está asignada y una pequeña descripción de la misma. Se podrá modificar las plazas en cualquier momento , eliminarlas siempre y cuando no haya ninguna resera asignada a dicha plaza o listarlas.

2.2.2. Gestión de Socios

Se encargará de manejar toda la información de los socios (creación, modificación y baja) en el sistema y listarlos. Nunca se podrá dar de baja a un socio con alguna reserva hecha y si se da de baja un socio sus vehículos quedarán, por tanto, eliminados del sistema.

2.2.3. Gestión de vehículos

Este apartado podrá manejar toda la información de los vehículos (creación, modificación y eliminar) en el sistema y listarlos. Nunca se podrá eliminar un vehículo que tiene una reserva asignada.

2.2.4. Gestión de zonas

Se podrán insertar nuevas zonas en el parking, modificarlas y borrarlas siempre y cuando no tengan plazas asignadas. Se pueden listar las zonas en cualquier momento.

2.2.5. Gestión de reservas

El cliente final podrá crear reservas, modificarlas, borrarlas y listarlas en cualquier momento. Además, existe una opción de actualización de reservas que se encarga de asignar las plazas correspondientes a las reservas del día en el que se encuentra el sistema.

2.3. Características de los Usuarios

El producto estará destinado al administrador de un parking que deberá tener unos conocimientos básicos de informática, a nivel de usuario, para el manejo de la herramienta.

La aplicación se verá intuitiva para que no exista ningún problema a la hora de manejarla.

2.4. Restricciones

La implementación del programa se hará en Java utilizando Eclipse como herramienta de desarrollo y NetBeans para la GUI. Para el almacenamiento de datos se usará una base de datos relacional.

Las fechas de las distintas entregas de partes del proyecto son una restricción temporal, por lo que se deben tener en cuenta a la hora de la planificación del mismo.

2.5. Suposiciones y Dependencias

Se supone que el aparcamiento tendrá un único administrador, no se tiene en cuenta una organización con más empleados. También se presupone que el

sistema correrá sobre una plataforma Windows, si no es así afectará a ciertos detalles técnicos y se tendrían que revisar y quizás cambiar los requisitos.

El sistema depende de una BBDD (MySQL) o fichero con varias tablas o secciones para el almacenamiento de los datos de los socios, sus vehículos y de las plazas del aparcamiento así como de las reservas que se generen.

Se supone también que las plazas del aparcamiento tendrán un sistema de balizas que evitará que las plazas libres sean ocupadas por otros vehículos que no sean los asignados a dicha plaza.

2.6. Requisitos Futuros

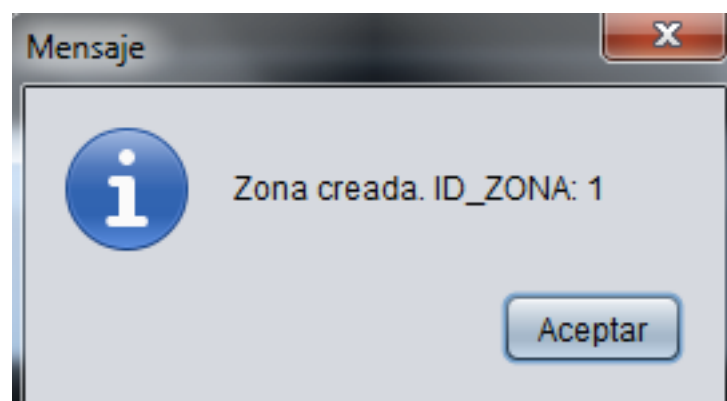
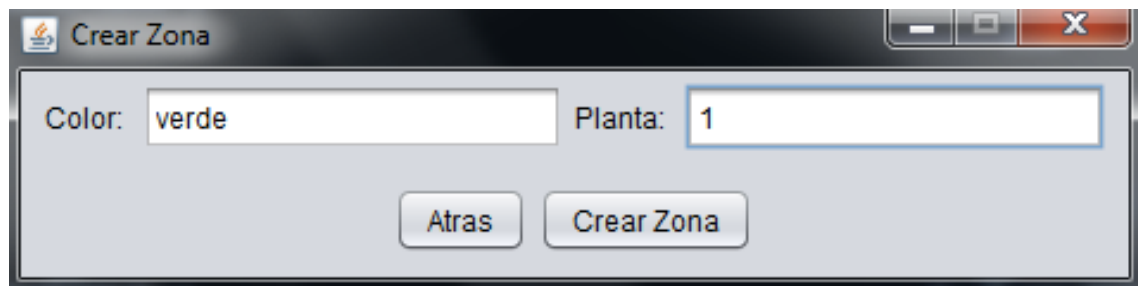
Como futuros requisitos para ampliar la funcionalidad del sistema aparecen diversas ideas:

- Proporcionar a los socios la posibilidad de ampliar sus reservas y la de cancelarlas ellos mismos.
- Un sistema de multas para los usuarios que hayan cometido alguna falta en las normas del aparcamiento.
- La integración de una BlackList para los usuarios a los que se le haya prohibido la entrada al recinto por motivos varios como por el impago de reservas o multas.

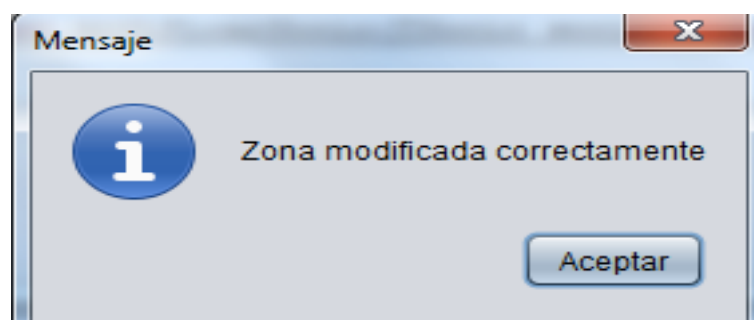
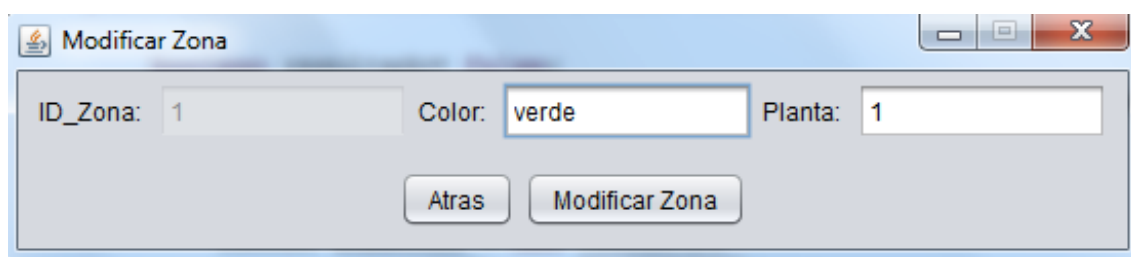
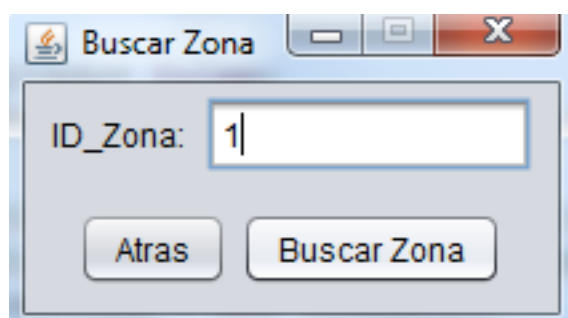
3. REQUISITOS ESPECÍFICOS

3.1. Interfaces externas

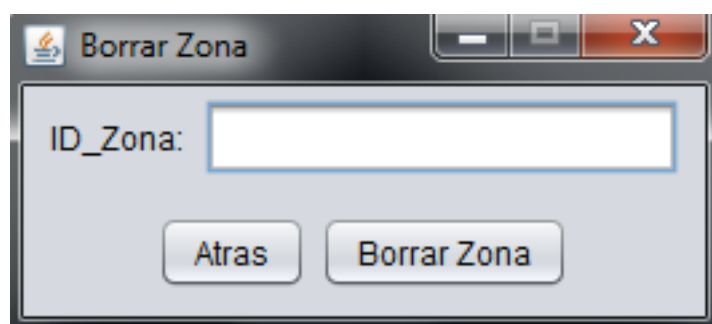
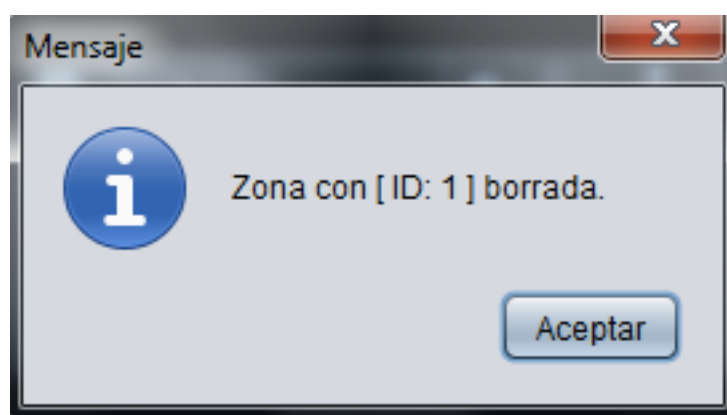
3.1.1.1 Crear zona: Representa al caso de uso “Crear zona”



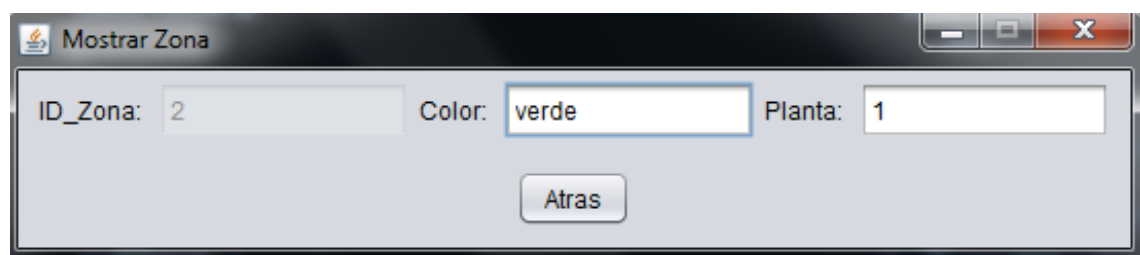
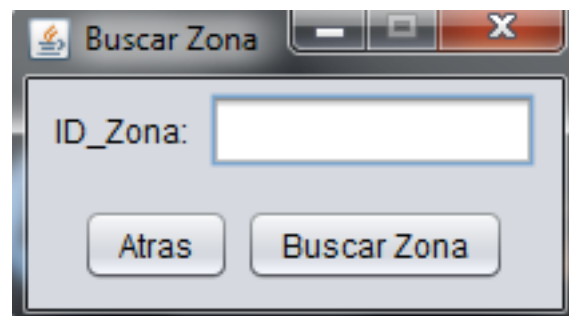
3.1.1.2 Modificar zona: Representa al caso de uso “Modificar zona”



3.1.1.3 Borrar zona: Representa al caso de uso “Borrar zona”



3.1.1.4 Mostrar zona: Representa al caso de uso “Mostrar zona”



3.1.1.5 Listar zonas: Representa al caso de uso “Listar zonas”

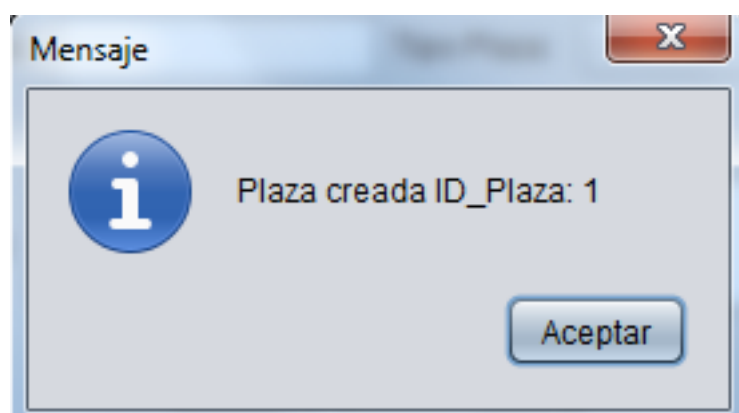
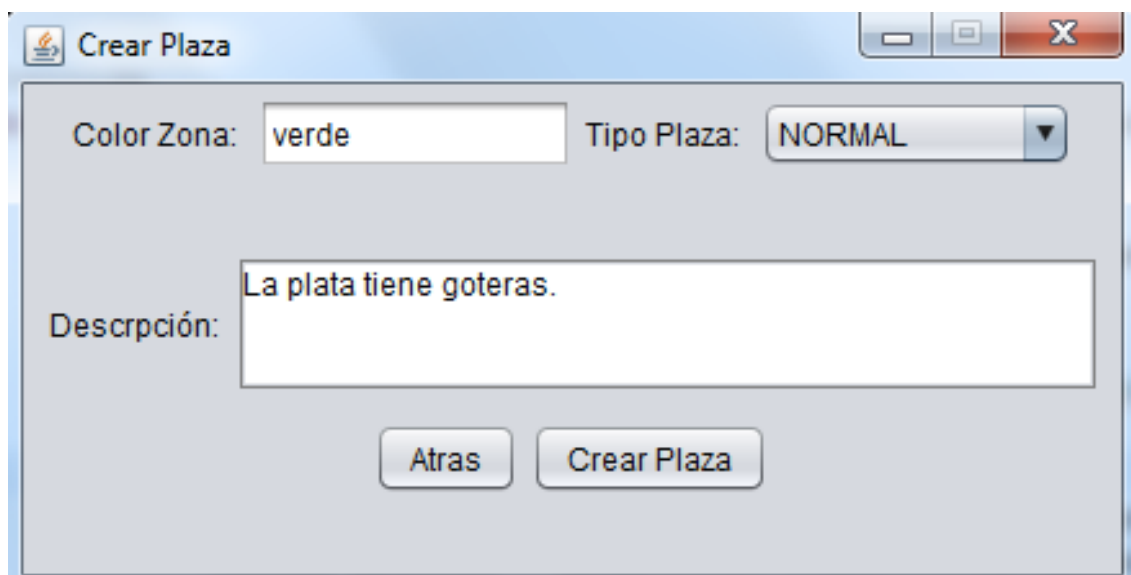


The screenshot shows a window titled 'Listar Zonas' with a table containing the following data:

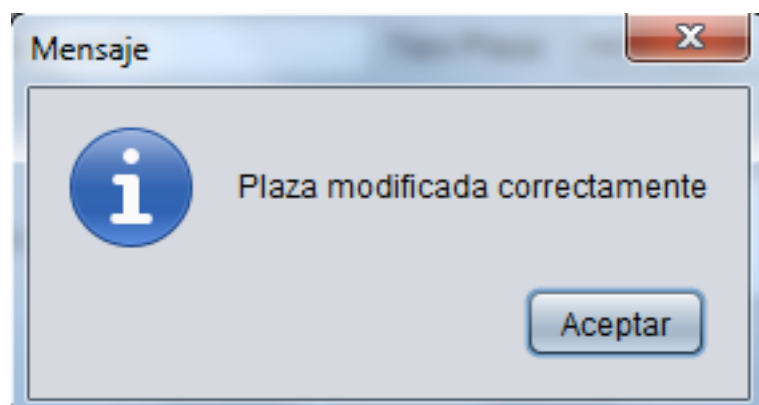
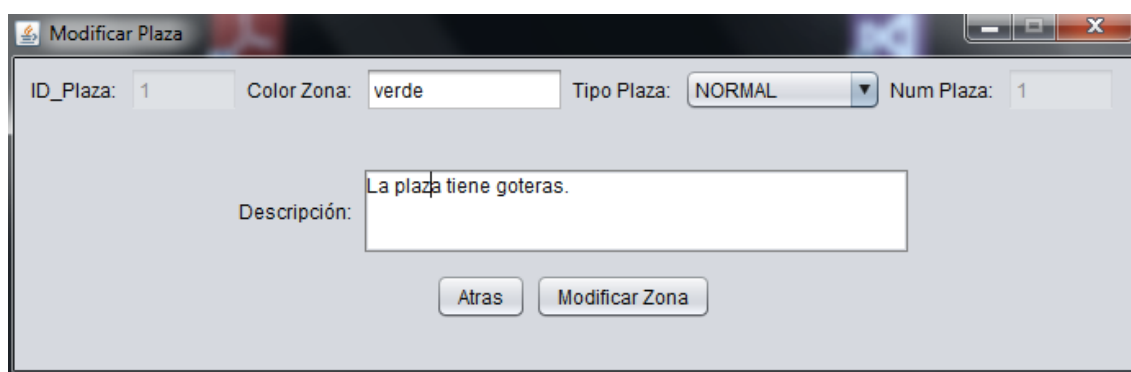
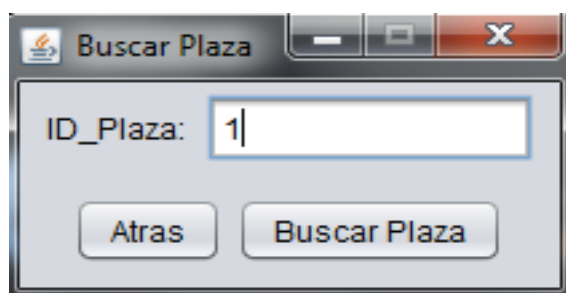
ID	COLOR	PLANTA
2	verde	1

Below the table is a button labeled 'Atras'.

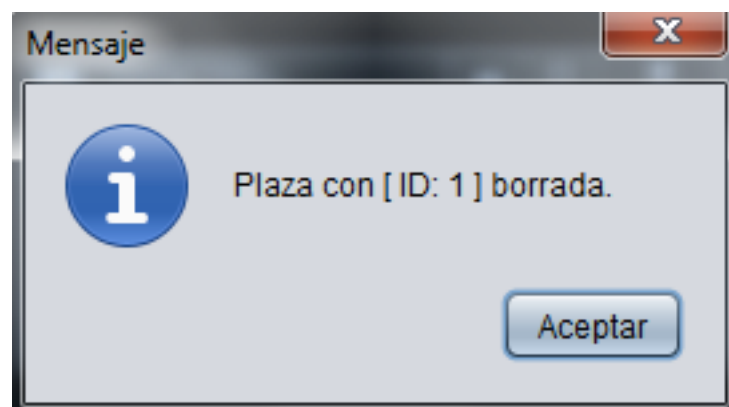
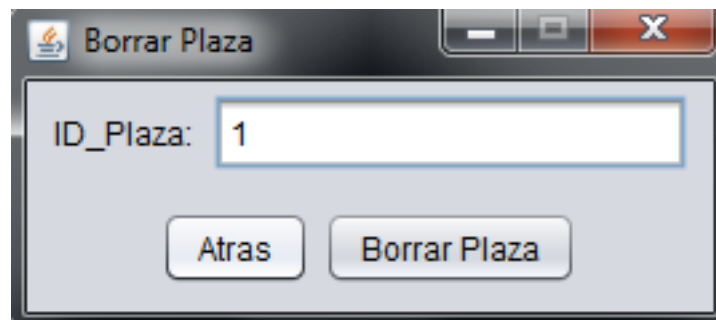
3.1.2.1 : Crear Plaza Representa al caso de uso “Crear plaza”



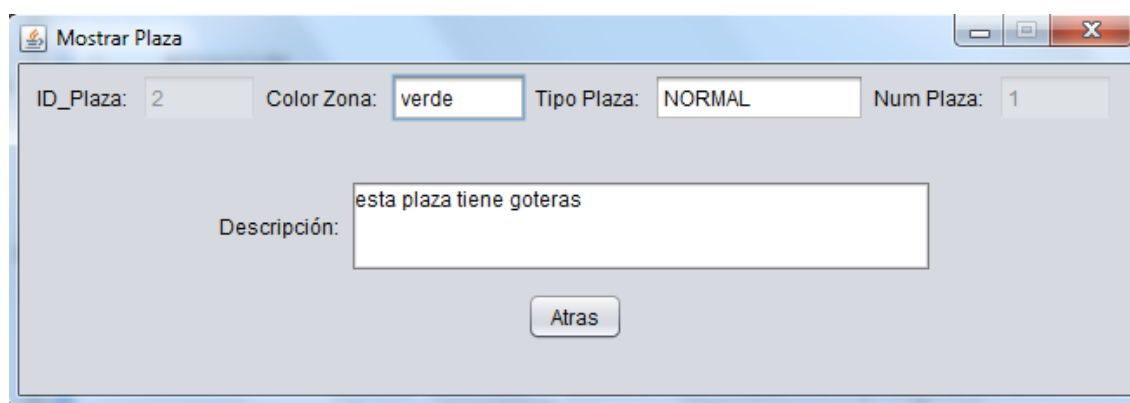
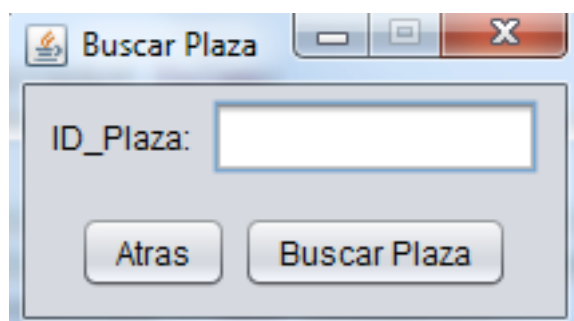
3.1.2.2 : Modificar Plaza Representa al caso de uso “Modificar plaza”



3.1.2.3 : Borrar Plaza Representa al caso de uso “Borrar plaza”



3.1.2.4 : Mostrar Plaza Representa al caso de uso “Mostrar plaza”



3.1.2.5 : Listar Plazas Representa al caso de uso “Listar plazas”



The screenshot shows the 'Listar Plazas' window, which displays a table of parking spaces. The table has the following columns: ID, ID_ZONA, COLOR_ZONA, TIPO_PLAZA, NUM_PLAZA, ESTADO, and DESCRIPCIÓN. The first row of data shows ID 2, ID_ZONA 2, COLOR_ZONA verde, TIPO_PLAZA NORMAL, NUM_PLAZA 1, ESTADO 0, and DESCRIPCIÓN esta plaza tien... Below the table is an 'Atras' button.

ID	ID_ZONA	COLOR_ZONA	TIPO_PLAZA	NUM_PLAZA	ESTADO	DESCRIPCIÓN
2	2	verde	NORMAL	1	0	esta plaza tien...

Atras

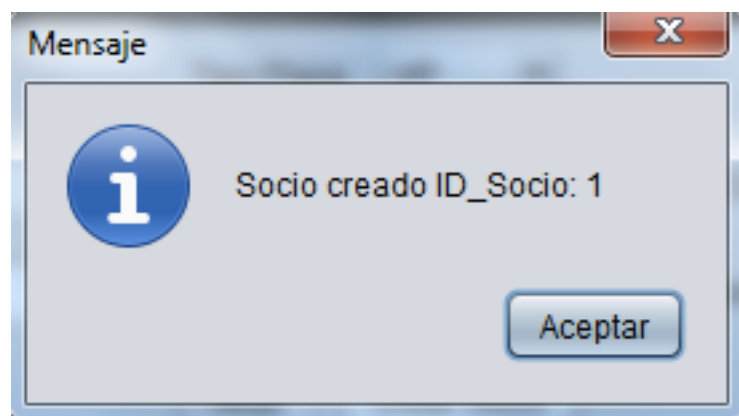
3.1.3.1 : Crear Socio Representa al caso de uso “Crear socio”



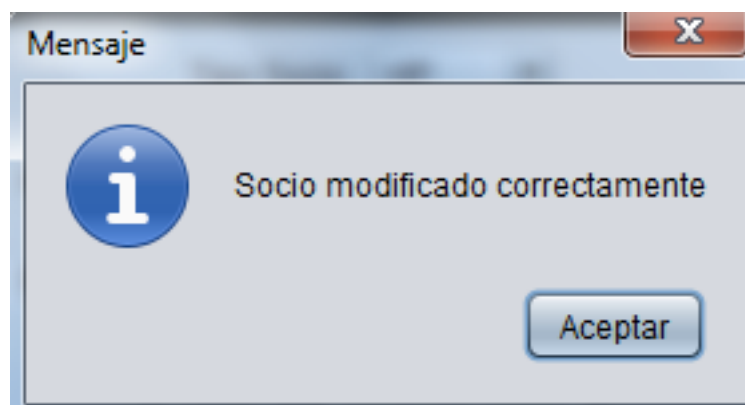
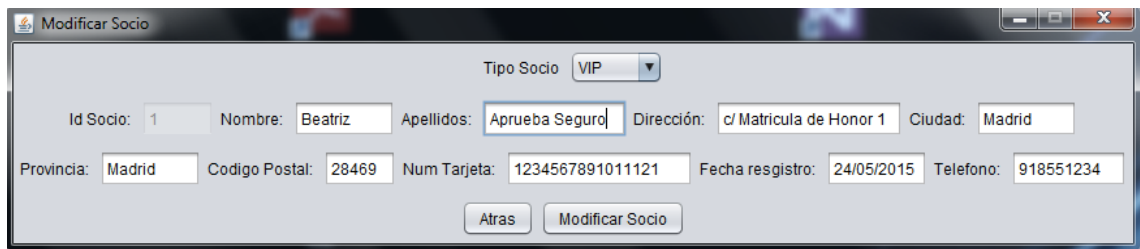
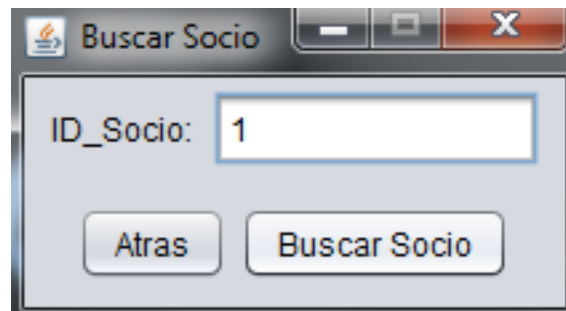
The screenshot shows the 'Crear Socio' dialog box. It contains the following fields and values:

- Tipo Plaza: VIP (dropdown menu)
- Nombre: Beatriz
- Apellidos: Aprueba Segura
- Dirección: c/ Matricula de Honor 1
- Ciudad: Madrid
- Provincia: Madrid
- Codigo Postal: 28469
- Num Tarjeta: 1234567891011121
- Telefono: 918551234

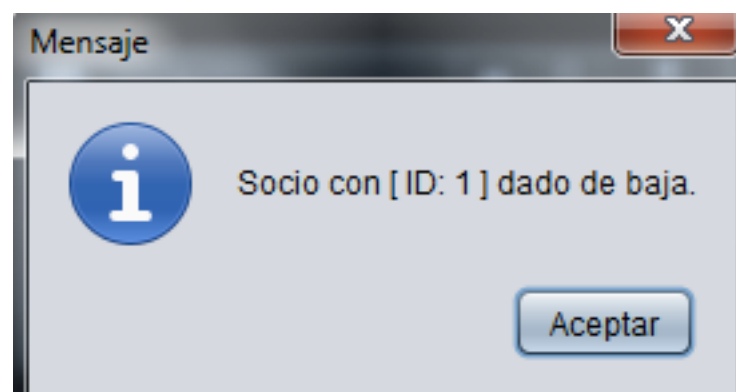
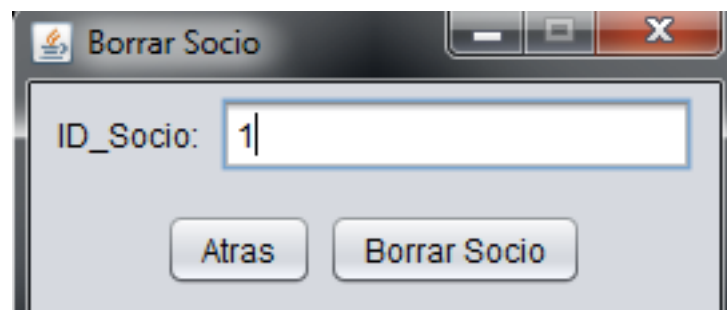
At the bottom, there are two buttons: 'Atras' and 'Crear Socio'.



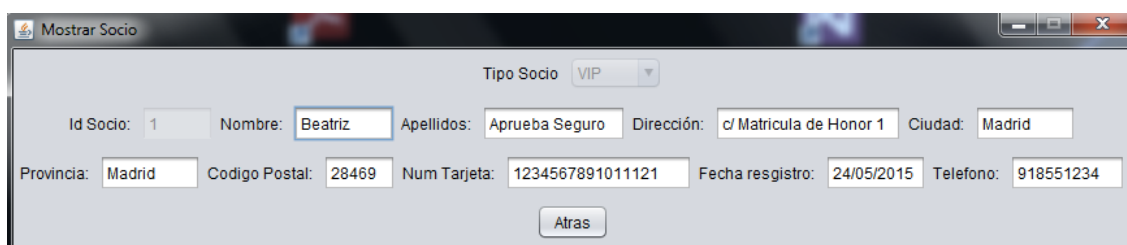
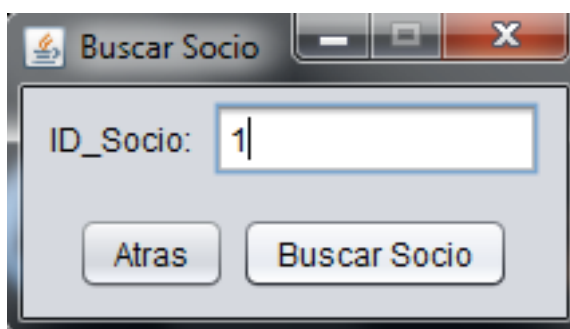
3.1.3.2 : Modificar Socio Representa al caso de uso “Modificar socio”



3.1.3.3 : Borrar Socio Representa al caso de uso “Borrar socio”



3.1.3.4 : Mostrar Socio Representa al caso de uso “Mostrar socio”



3.1.3.5 : Listar Socios Representa al caso de uso “Listar socios”



The screenshot shows the 'Listar Socios' window. It contains a table with the following data:

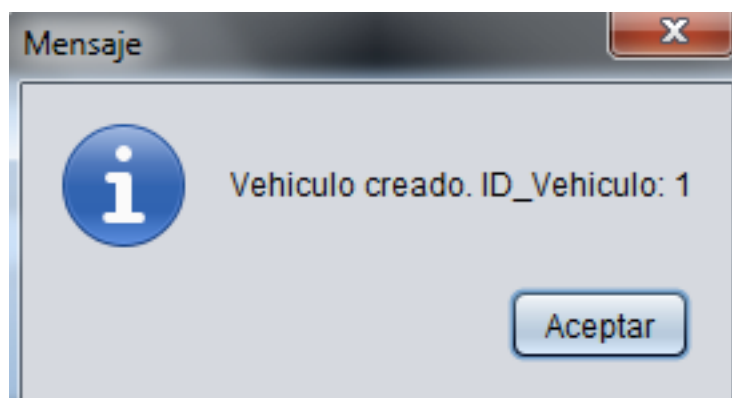
ID_SOCIO	NOMBRE	APELLIDOS	DIRECCI...	CIUDAD	PROVINCIA	COD_PO...	NUM_TAR...	FECHA_R...	ESTADO	TIPO_SO...	TELEFONO	EMAIL
1	Beatriz	Aprueba S...	c/ Matricul...	Madrid	Madrid	28469	12345678...	24/05/2015	1	VIP	918551234	

Below the table is a button labeled 'Atras'.

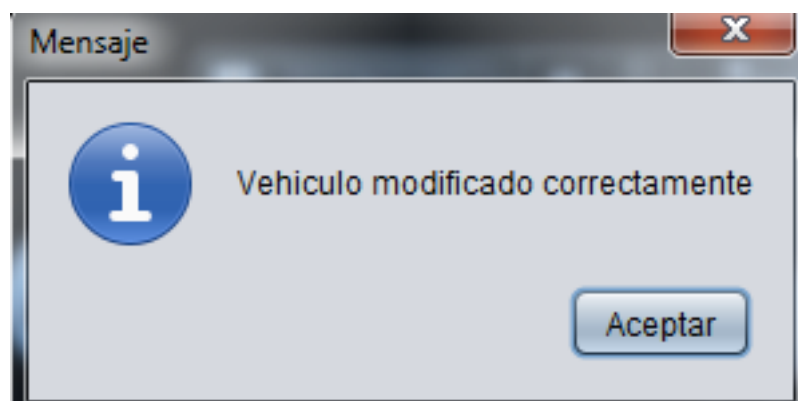
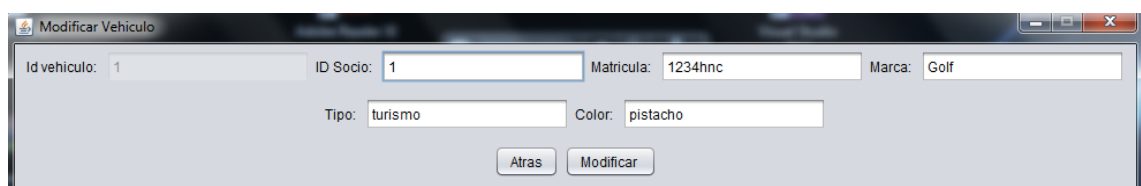
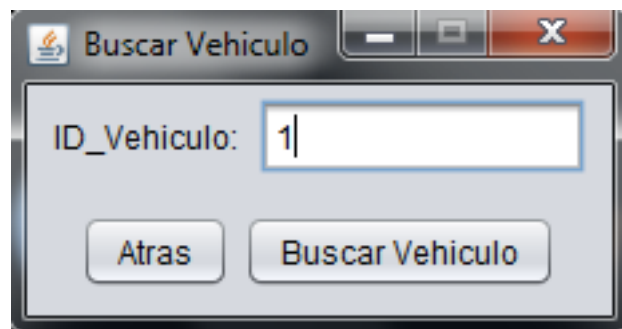
3.1.4.1 : Crear vehículo Representa al caso de uso “Crear vehículo”



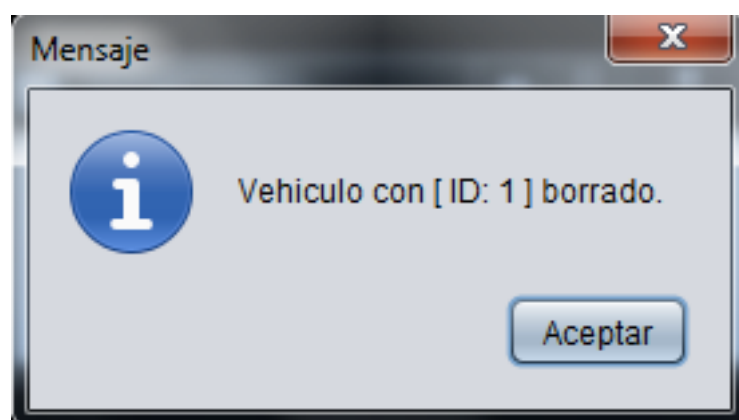
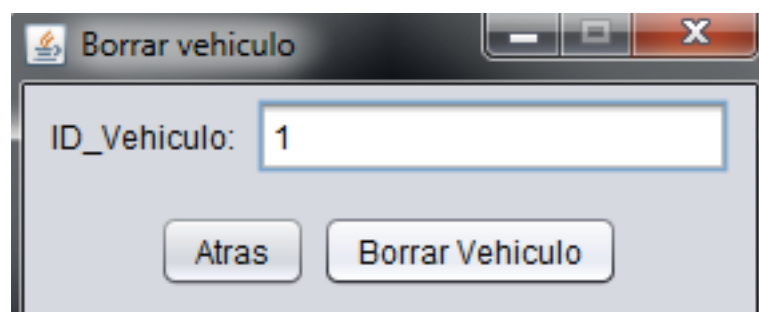
The screenshot shows the 'Crear vehiculo' dialog box. It contains several input fields: 'ID Socio' with the value '1', 'Matricula' with the value '1234hnc', 'Marca' with the value 'Golf', 'Tipo' with the value 'turismo', and 'Color' with the value 'pistacho'. At the bottom, there are two buttons: 'Atras' and 'Crear Vehiculo'.



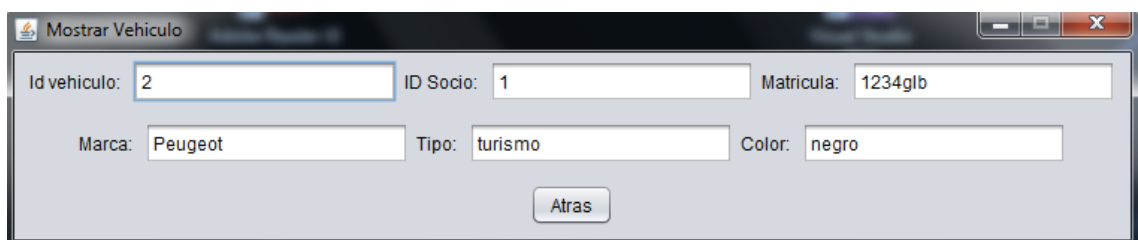
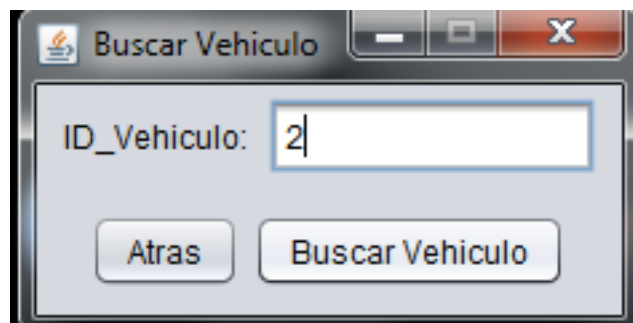
3.1.4.2 : Modificar vehículo Representa al caso de uso “Modificar vehículo”



3.1.4.3 : Borrar vehículo Representa al caso de uso “Borrar vehículo”



3.1.4.4 : Mostrar vehículo Representa al caso de uso “Mostrar vehículo”



3.1.4.5 : Listar vehículos Representa al caso de uso “Listar vehículos”

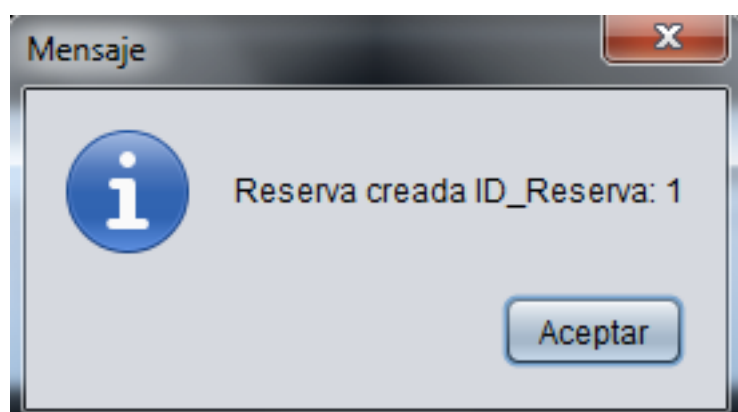
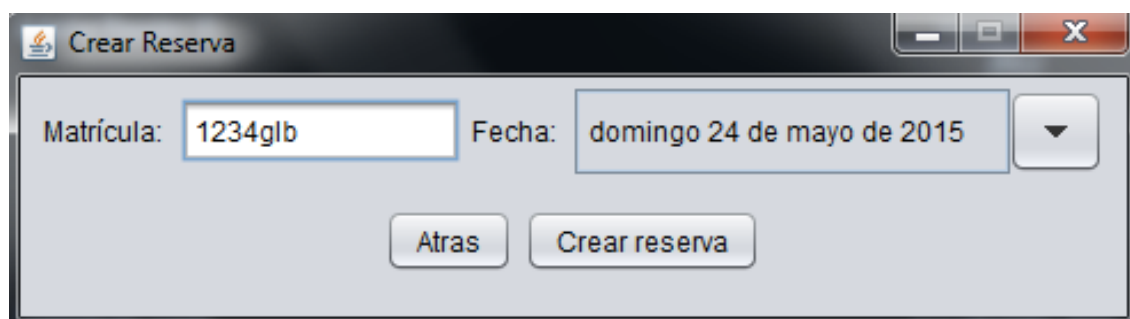


The screenshot shows a window titled 'Listar Vehiculos' containing a table with the following data:

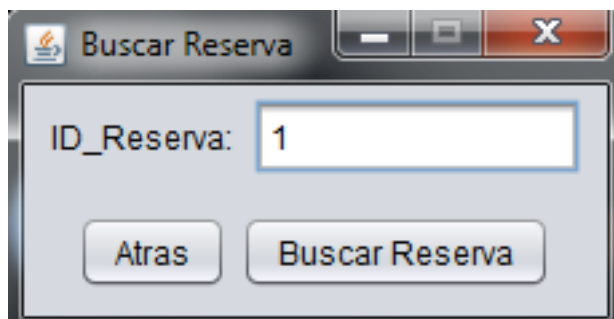
ID_VEHICULO	ID_SOCIO	MATRICULA	MARCA	TIPO	COLOR
2	1	1234gib	Peugeot	turismo	negro

Below the table is a button labeled 'Atras'.

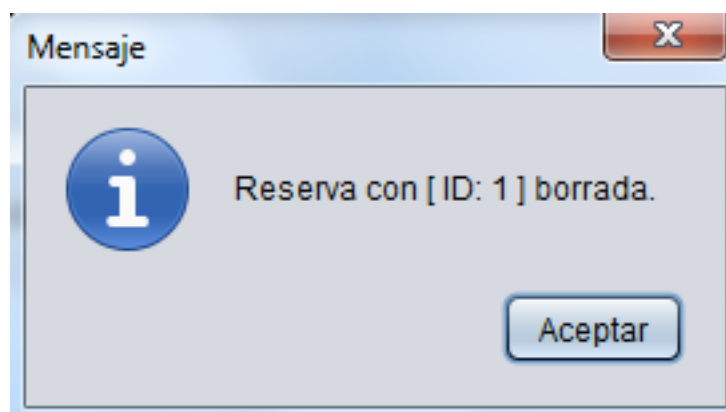
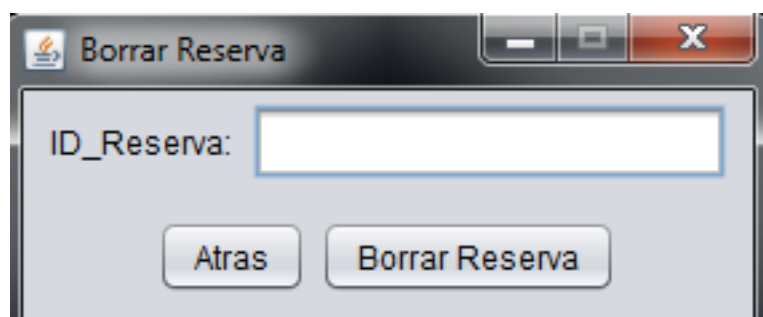
3.1.5.1 : Crear reserva Representa al caso de uso “Crear reserva”



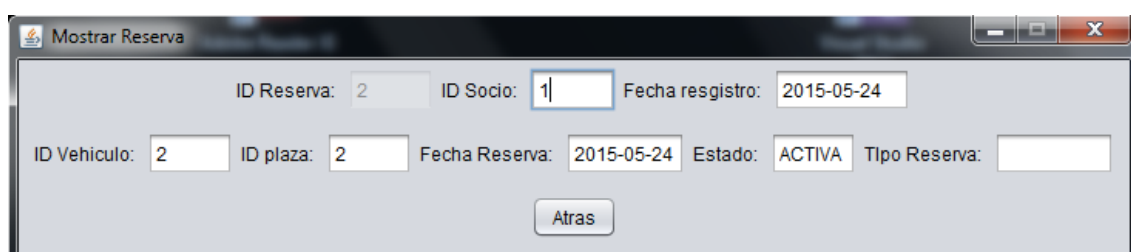
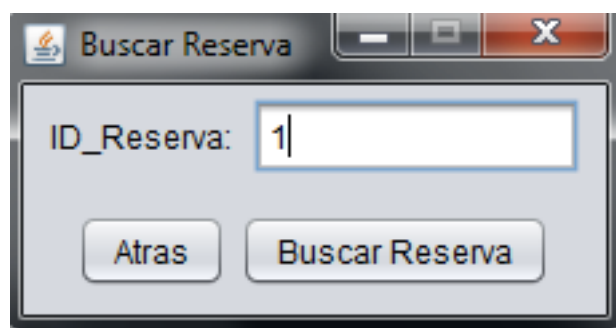
3.1.5.2 : Modificar reserva Representa al caso de uso “Modificar reserva”



3.1.5.3 : Borrar reserva Representa al caso de uso “Borrar reserva”



3.1.5.4 : Mostrar reserva Representa al caso de uso “Mostrar reserva”



3.1.5.5 : Listar reservas Representa al caso de uso “Listar reservas”

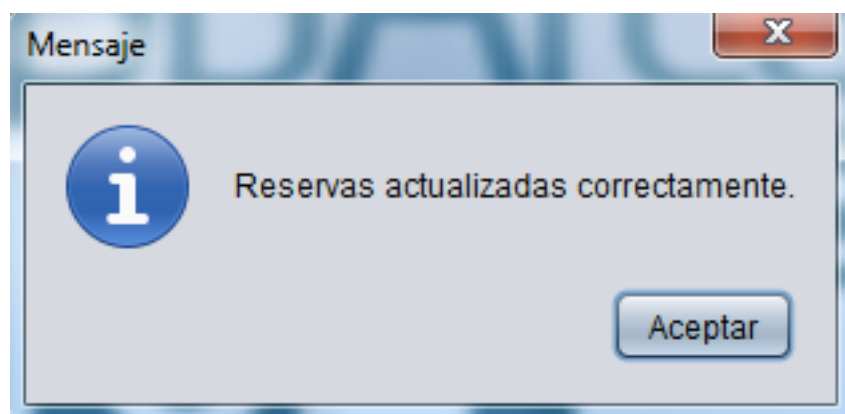


The screenshot shows a window titled 'Listar Reservas' containing a table with reservation data. The table has seven columns: ID_RESERVA, ID_SOCIO, MATRICULA, FECHA DE RE..., FECHA RESE..., ID_PLAZA, and ID_VEHICULO. The first row of data shows reservation ID 2 for a member with ID 1 and license plate 1234glb, made on 2015-05-24. Below the table is an 'Atras' button.

ID_RESERVA	ID_SOCIO	MATRICULA	FECHA DE RE...	FECHA RESE...	ID_PLAZA	ID_VEHICULO
2	1	1234glb	2015-05-24 0...	2015-05-24 0...	2	2

Atras

3.1.5.6 : Actualizar reservas.



3.2. Funciones

- Módulo PLAZAS

Función	Mostrar plaza (PLA01)
Prioridad	Media
Estabilidad	Alta
Descripción	Muestra los datos de una plaza.
Entrada	Id_plaza, id_zona, tipo_plaza, num_plaza, descripción, estado.
Salida	Muestra los datos de la plaza.
Origen	Sistema
Destino	GUI
Necesita	Id_plaza
Acción	Muestra información de la plaza
Precondición	Que la plaza exista en el sistema.
Postcondición	Datos mostrados por pantalla.
Efectos lat.	Error en la BBDD.

Función	Listar plazas (PLA02)
Prioridad	Baja
Estabilidad	Media
Descripción	Muestra los datos de todas las plazas.
Entrada	-
Salida	Listado de las plazas.
Origen	Sistema
Destino	GUI
Necesita	-
Acción	Lista por pantalla las plazas.
Precondición	Existen plazas registradas en el sistema.
Postcondición	Listado de plazas por pantalla.
Efectos lat.	No muestra nada al estar la lista vacía o error en la BBDD.

Función	Crear plaza (PLA03)
Prioridad	Alta
Estabilidad	Media
Descripción	Crea una plaza en el sistema.
Entrada	id_zona, tipo_plaza, num_plaza, descripción, estado.
Salida	Mensaje de éxito.
Origen	Sistema
Destino	GUI
Necesita	id_zona, tipo_plaza, num_plaza, descripción, estado.
Acción	Registra una plaza en el sistema.
Precondición	Todos los datos rellenos.
Postcondición	Plaza registrada en el sistema.
Efectos lat.	Campos no están completos o error en la BBDD.

Función	Modificar plaza (PLA04)
Prioridad	Alta
Estabilidad	Baja
Descripción	Se muestran los datos por pantalla, el usuario los modifica y el sistema registra los cambios.
Entrada	id_zona, tipo_plaza, num_plaza, descripción, estado.
Salida	Mensaje de éxito.
Origen	Sistema
Destino	GUI
Necesita	id_zona.
Acción	Modifica los datos de una plaza ya registrada.
Precondición	Todos los datos están completos.
Postcondición	Datos modificados correctamente.
Efectos lat.	-

Función	Borrar plaza (PLA05)
Prioridad	Media
Estabilidad	Media
Descripción	Elimina del sistema una plaza registrada.
Entrada	ID_plaza
Salida	Mensaje de éxito.
Origen	Sistema
Destino	GUI
Necesita	ID_plaza
Acción	Elimina del sistema una plaza.
Precondición	ID_plaza válida. No haya reservas asignadas con dicha plaza.
Postcondición	-
Efectos lat.	-

- **Módulo ZONAS**

Función	Mostrar zona (ZO01)
Prioridad	Baja
Estabilidad	Alta
Descripción	Muestra los datos de una zona.
Entrada	ID_zona, color, planta
Salida	Muestra los datos de la zona.
Origen	Sistema
Destino	GUI
Necesita	ID_zona
Acción	Muestra información de la zona
Precondición	Que la zona exista en el sistema.
Postcondición	Datos mostrados por pantalla.
Efectos lat.	Error en la BBDD.

Función	Listar zonas (ZO02)
Prioridad	Baja
Estabilidad	Media
Descripción	Muestra los datos de todas las zonas.
Entrada	-
Salida	Listado de las zonas.
Origen	Sistema
Destino	GUI
Necesita	-
Acción	Lista por pantalla las zonas.
Precondición	Existen zonas registradas en el sistema.
Postcondición	Listado de zonas por pantalla.
Efectos lat.	No muestra nada al estar la lista vacía o error en la BBDD.

Función	Crear zona (ZO03)
Prioridad	Alta
Estabilidad	Media
Descripción	Crea una zona en el sistema.
Entrada	ID_zona, color, planta
Salida	Mensaje de éxito.
Origen	Sistema
Destino	GUI
Necesita	color, planta
Acción	Registra una zona en el sistema.
Precondición	Todos los datos rellenos.
Postcondición	Zona registrada en el sistema.
Efectos lat.	Campos no están completos o error en la BBDD.

Función	Modificar zona (ZO04)
Prioridad	Alta
Estabilidad	Baja
Descripción	Se muestran los datos por pantalla, el usuario los modifica y el sistema registra los cambios.
Entrada	ID_zona, color, planta
Salida	Mensaje de éxito.
Origen	Sistema
Destino	GUI
Necesita	ID_zona
Acción	Modifica los datos de una zona ya registrada.
Precondición	Todos los datos están completos.
Postcondición	Datos modificados correctamente.
Efectos lat.	-

Función	Borrar zona (ZO05)
Prioridad	Media
Estabilidad	Media
Descripción	Elimina del sistema una zona registrada.
Entrada	ID_zona
Salida	Mensaje de éxito.
Origen	Sistema
Destino	GUI
Necesita	ID_zona
Acción	Elimina del sistema una plaza.
Precondición	ID_zona válida y no hay plazas registradas en dicha zona.
Postcondición	-
Efectos lat.	-

- **Módulo RESERVA.**

Función	Registrar reserva (RES01)
Prioridad	Alta
Estabilidad	Alta
Descripción	Realiza una petición de reserva al sistema.
Entrada	-
Salida	Mensaje de éxito.
Origen	Sistema
Destino	GUI
Necesita	Matrícula, fecha_reserva
Acción	Registra una reserva de un vehículo.
Precondición	El vehículo existe en el sistema.
Postcondición	La reserva queda asignada a un vehículo
Efectos lat.	El parking se encuentre lleno y no se pueda realizar la reserva.

Función	Modificar reserva (RES02)
Prioridad	Alta
Estabilidad	Alta
Descripción	Realiza una modificación de una reserva.
Entrada	Id_reserva, fecha_reserva
Salida	Confirmación de la operación.
Origen	Sistema
Destino	GUI
Necesita	Id_reserva
Acción	Modifica los datos de una reserva de un vehículo.
Precondición	Id_reserva no existe en el sistema.
Postcondición	La modificación queda realizada.
Efectos lat.	El parking se encuentra lleno y no se puede modificar la reserva.

Función	Borrar reserva (RES03)
Prioridad	Media
Estabilidad	Media
Descripción	Borrar una reserva.
Entrada	ID_reserva
Salida	Confirmación de la operación.
Origen	Sistema
Destino	GUI
Necesita	ID_reserva
Acción	Borrar una reserva asignada a un vehículo.
Precondición	Id_reserva existe en el sistema, el vehículo no se encuentra en el parking.
Postcondición	Se elimina la reserva de la base de datos, la plaza queda liberada en el caso de que estuviera ocupada.
Efectos lat.	-

Función	Listar reservas (RES04)
Prioridad	Baja
Estabilidad	Media
Descripción	Petición de listado por pantalla de todas las reservas.
Entrada	-
Salida	Listado de las reservas.
Origen	Sistema
Destino	GUI
Necesita	-
Acción	Lista todas las reservas por pantalla.
Precondición	Existe al menos una reserva.
Postcondición	Muestra por pantalla todo el listado de reservas.
Efectos lat.	-

Función	Mostrar reserva (RES05)
Prioridad	Alta
Estabilidad	Media
Descripción	Petición de búsqueda de una reserva
Entrada	ID_reserva
Salida	Mensaje de éxito.
Origen	Sistema
Destino	GUI
Necesita	ID_Reserva
Acción	Nos devuelve los datos de una reserva.
Precondición	ID_reserva registrado en el sistema.
Postcondición	Muestra por pantalla datos de una reserva.
Efectos lat.	-

Función	Listar reservas (RES04)
Prioridad	Baja
Estabilidad	Media
Descripción	Petición de listado por pantalla de todas las reservas.
Entrada	-
Salida	Listado de las reservas.
Origen	Sistema
Destino	GUI
Necesita	-
Acción	Lista todas las reservas por pantalla.
Precondición	Existe al menos una reserva.
Postcondición	Muestra por pantalla todo el listado de reservas.
Efectos lat.	-

- **Módulo SOCIOS.**

Función	Registrar socio (SO01)
Prioridad	Alta
Estabilidad	Media
Descripción	Registra a un socio.
Entrada	Nombre,apellidos,dirección,ciudad,provincia,código_postal, num_tarjeta, fecha_registro, estado, tipo_socio.
Salida	Mensaje de éxito.
Origen	Sistema
Destino	GUI
Necesita	Nombre,apellidos,dirección,ciudad,provincia,código_postal, num_tarjeta, fecha_registro, estado, tipo_socio.
Acción	Registra a un socio en el sistema.
Precondición	Todos los datos rellenos.
Postcondición	-
Efectos lat.	-

Función	Borrar socio (SO02)
Prioridad	Media
Estabilidad	Media
Descripción	Da de baja a un socio registrado.
Entrada	ID_socio
Salida	Mensaje de éxito.
Origen	Sistema
Destino	GUI
Necesita	ID_socio
Acción	Da de baja a un socio del sistema.
Precondición	Tiene que existir el socio en el sistema.
Postcondición	El socio se da de baja del sistema.
Efectos lat.	-

Función	Modificar socio (SO03)
Prioridad	Alta
Estabilidad	Alta
Descripción	Modifica a un socio ya registrado.
Entrada	Nombre,apellidos,dirección,ciudad,provincia,código_postal, num_tarjeta, fecha_registro, estado, tipo_socio.
Salida	Mensaje de éxito.
Origen	Sistema
Destino	GUI
Necesita	ID_socio.
Acción	Modifica a un socio registrado en el sistema.
Precondición	Tiene que existir el ID_socio en el sistema.
Postcondición	-
Efectos lat.	-

Función	Listar socios (SO04)
Prioridad	Baja
Estabilidad	Alta
Descripción	Lista los datos de los socios por pantalla.
Entrada	-
Salida	Muestra los datos por pantalla.
Origen	Sistema
Destino	GUI
Necesita	-
Acción	Muestra los datos de los socios por pantalla.
Precondición	Tiene que existir al menos un socio.
Postcondición	-
Efectos lat.	-

Función	Muestra socio (SO05)
Prioridad	Media
Estabilidad	Alta
Descripción	Muestra los datos de un socio por pantalla.
Entrada	ID_socio
Salida	Muestra los datos por pantalla.
Origen	Sistema
Destino	GUI
Necesita	ID_socio
Acción	Muestra los datos del socio por pantalla.
Precondición	Tiene que existir el socio en el sistema.
Postcondición	-
Efectos lat.	-

- **Módulo VEHÍCULOS.**

Función	Registrar vehículo (VE01)
Prioridad	Alta
Estabilidad	Baja
Descripción	Registra un vehículo en el sistema.
Entrada	Tipo, Marca, matricula, color, ID_socio
Salida	Mensaje de éxito.
Origen	Sistema
Destino	GUI
Necesita	Tipo, Marca, matricula, color, ID_socio
Acción	Registra a un vehículo en el sistema.
Precondición	Se han rellenado todos los datos del vehículo
Postcondición	-
Efectos lat.	-

Función	Borrar vehículo (VE02)
Prioridad	Media
Estabilidad	Alta
Descripción	Elimina a un vehículo registrado en el sistema.
Entrada	ID_vehiculo
Salida	Mensaje de éxito.
Origen	Sistema
Destino	GUI
Necesita	ID_vehiculo
Acción	Elimina a un vehículo del sistema.
Precondición	Tiene que existir el vehículo en el sistema.
Postcondición	El vehículo se elimina del sistema.
Efectos lat.	-

Función	Modificar vehículo (VE03)
Prioridad	Alta
Estabilidad	Baja
Descripción	Modifica a un vehículo ya registrado en el sistema.
Entrada	Tipo, Marca, matricula, color, ID_socio
Salida	Mensaje de éxito.
Origen	Sistema
Destino	GUI
Necesita	Tipo, Marca, matricula, color, ID_socio
Acción	Modifica a un vehículo registrado en el sistema.
Precondición	Tiene que existir el vehículo en el sistema.
Postcondición	-
Efectos lat.	-

Función	Listar vehículos (VE04)
Prioridad	Baja
Estabilidad	Alta
Descripción	Lista los datos de los vehículos por pantalla.
Entrada	-
Salida	Muestra los datos por pantalla.
Origen	Sistema
Destino	GUI
Necesita	Lista de los vehículos.
Acción	Muestra los datos de los vehículos por pantalla.
Precondición	Tiene que existir el vehículo en el sistema.
Postcondición	-
Efectos lat.	-

Función	Muestra vehículo (VE05)
Prioridad	Baja
Estabilidad	Alta
Descripción	Lista los datos de un vehículo por pantalla.
Entrada	ID_vehiculo
Salida	Muestra los datos por pantalla.
Origen	Sistema
Destino	GUI
Necesita	ID_vehiculo
Acción	Muestra los datos del vehículo por pantalla.
Precondición	Tiene que existir el vehículo en el sistema.
Postcondición	-
Efectos lat.	-

3.3. Requisitos de rendimiento

Solo se va a utilizar un terminal, por tanto no habrá operaciones simultáneas.

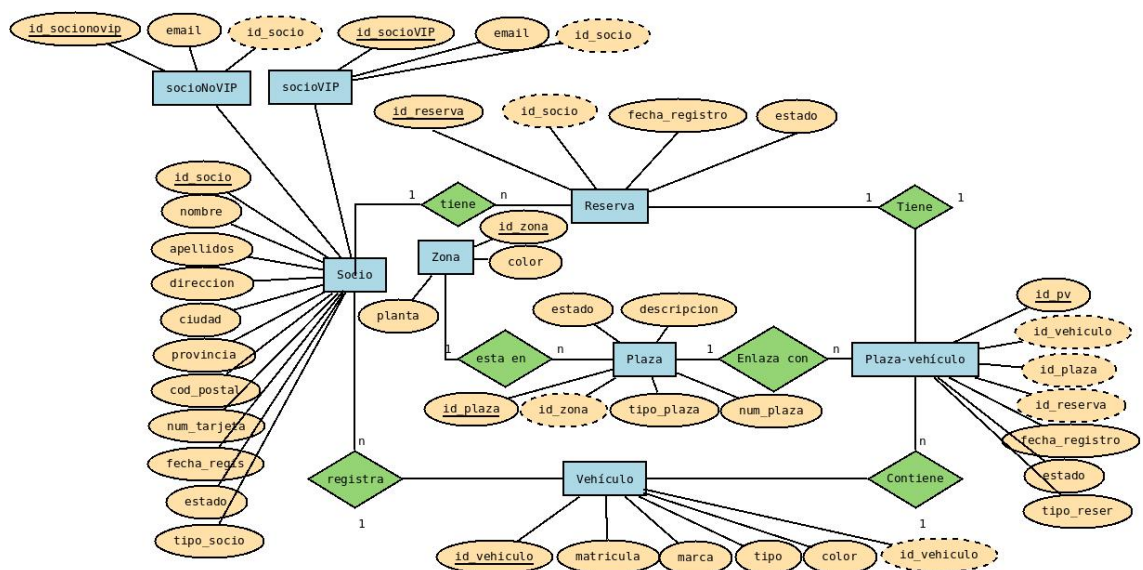
La BBDD se modificará cada vez que se produzca una variación en la información almacenada en dicha BBDD.

Los datos se almacenarán hasta el máximo que nos permita nuestra BBDD.

3.4. Requisitos lógicos de la BBDD.

- **DIAGRAMA ENTIDAD RELACIÓN**

Este diagrama representa el funcionamiento de la base de dato relacional del sistema pudiéndose diferenciar los distintos módulos de nuestra aplicación.



3.5. Restricciones de diseño.

Construye una aplicación de gestión de información (por ejemplo, una biblioteca).

Restricciones I

- Es obligatorio aplicar técnicas de IS (tanto a nivel gestión, como a nivel técnico).
- La aplicación no podrá ser ni una biblioteca, ni un videoclub.

Restricciones II

- La primera entrega del proyecto será el día 26/01/2015. Consistirá del plan del proyecto del software y la especificación de requisitos software de la aplicación.

- La segunda entrega del proyecto será el día 25/05/2015. Consistirá en el producto final.

Restricciones III

- El lenguaje para caracterizar el diseño de la aplicación debe ser UML 2.x.
- Respecto al producto final:
 - La aplicación debe ser de escritorio o web.
 - El lenguaje de implementación debe ser Java o C++.
 - La persistencia de los datos debe hacerse en formato texto, formato XML o en formato relacional. En el caso relacional, el sistema de gestión de bases se recomienda que sea MySQL.
 - En el caso web el lenguaje de programación debe ser Java y la persistencia relacional.
 - Debe haber polimorfismo.
- Se utilizará el sistema de control de versiones (SCV) de la Facultad de Informática para gestionar la documentación, modelo y código.

Restricciones IV

- El proyecto debe realizarse en equipo.
- El número de miembros de cada equipo debe ser exactamente seis.
- En casos excepcionales, y previa consulta con el profesor de esta asignatura antes del día 24/11/2014, se podría permitir la existencia de algún equipo de otro tamaño.
- En caso de existir equipos de alumnos de tamaño distinto de seis, el profesor se reserva el derecho de modificar la composición de estos equipos.
- Lo antes posible, los equipos deberán enviar un e-mail al profesor (anavarro@fdi.ucm.es) con el nombre del proyecto, los nombres de los integrantes del equipo, y la dirección de correo electrónico del responsable del equipo para los repositorios del SCV de la facultad.
- Cada entrega es prerequisite de la siguiente.
- No se admiten entregas después de acabar la sesión de la asignatura correspondiente al día de cada entrega.

Restricciones V

- El día de la entrega del producto final se procederá a una ejecución del proyecto para comprobar la implementación de los requisitos.
- Todas las entregas deben hacerse en un CD-ROM etiquetado con el nombre del proyecto.
- En todas las entregas debe incluirse el archivo sobre la planificación temporal (o su revisión) en formato MS Project.
- En la entrega del producto final deben incluirse una memoria.
- En la entrega del producto final deben incluirse los archivos correspondientes al diseño en formato IBM Rational Software Architect.
- En la entrega final deben incluirse los archivos de cada aplicación, tanto los archivos fuente como los compilados (que en particular deben ser ejecutables).
- La existencia de cualquier tipo de virus en cualquier soporte informático entregado al profesor invalidará la entrega.
- Los alumnos son los únicos responsables del contenido almacenado en el SCV. La presencia de contenido inadecuado en el SCV conllevará su borrado y la invalidación del proyecto.
- Los repositorios de la facultad serán BORRADOS el día 01/10/2015.

Nota: El profesor de la asignatura se reserva la posibilidad de modificar este proyecto con el fin de poder corregir cualquier error que pudiera existir en su enunciado.

3.6. Atributos del sistema software.

Se diferencia entre Socio no vip y Socio vip. Esta diferenciación se lleva a cabo en función de si en la base de datos queda registrado con teléfono o con email.