

SOFTWARE REQUIREMENTS SPECIFICATION





Pablo Agudo, Alejandro
Cabezas, Iván Fernández, Manuel Monforte,
Gerardo Parra y Alberto Pastor.



GREENGO

Control de Versiones

Nº Versión	Fecha	Autor	Descripción
1.0	04/10/2018	-	Creación SRS
1.1	09/11/2018	-	Renombrado algunos apartados.
1.2	13/12/2018	-	Cambios en algunos apartados.



Índice

1. Introducción	1
1.1 Propósito	1
1.2 Alcance	1
1.3 Definiciones, acrónimos y abreviaturas	2
1.4 Referencias	2
1.5 Resumen	2
2. Descripción general	3
2.1 Perspectiva del producto	3
2.2 Funciones del producto	3
2.3 Características de usuario	12
2.4 Restricciones.....	12
2.5 Supuestos y dependencias	13
2.6 Requisitos futuros	13
3. Requisitos específicos	13
3.1 Interfaces externas	13
3.1.1 Interfaces de usuario	13
3.1.2 Interfaces de software	18
3.1.3 Interfaces de hardware	18
3.2 Funciones.....	19
3.3 Requisitos de rendimiento	97
3.4 Modelo del dominio	97
3.5 Restricciones de diseño	98
3.6 Atributos del Sistema software	98



1. Introducción

En este apartado trataremos determinados puntos del documento SRS del proyecto "GREENGO", en particular:

- Propósito.
- Alcance de la aplicación.
- Referencias tomadas.
- Visión general del documento.

Como nota secundaria, el documento sigue el estándar IEEE 830 - 1998. El desarrollo de la aplicación está destinado a la asignatura Modelado de Software (MS) de la Universidad Complutense de Madrid.

1.1 Propósito

El propósito del documento SRS es definir de forma detallada y precisa los requisitos necesarios para el desarrollo del producto software requerido por el cliente. Este documento estará sujeto a diferentes revisiones que serán validadas por el cliente junto con el equipo y se utilizará como contrato a posteriori por si surgiera algún cambio inesperado.

Los desarrolladores de la aplicación serán los integrantes del equipo compuesto por Manuel Monforte, Iván Fernández, Alberto Pastor, Alejandro Cabezas, Gerardo Parra y Pablo Agudo, alumnos de *Modelado de Software* de tercero E.

El papel de cliente será ejecutado por nuestro profesor Antonio Navarro. El usuario final de la aplicación será el gestor/encargado de la gestión del sistema.

1.2 Alcance

La aplicación por desarrollar recibirá el nombre de "GREENGO", el cual llevará a cabo la gestión de distintos vehículos operativos en ciudades concretas, permitiendo el alquiler de estos por clientes registrados previamente, así como la gestión de diferentes sedes, servicios contratados y empleados.

Dentro de la aplicación, gestionaremos seis módulos principales que son Vehículos, Ciudades, Clientes, Sedes, Servicios y Empleados.

El uso de la aplicación será interno y lo usará el propio encargado de ésta. Dicho encargado dispondrá del manejo completo de la administración, realizando acciones a partir de lo solicitado por el cliente. Este podrá realizar acciones de alta, baja, modificación y visualizar datos.



1.3 Definiciones, acrónimos y abreviaturas

- IEEE (Institute of Electrical and Electronics Engineers): Asociación mundial de técnicos e ingenieros dedicada a la estandarización y el desarrollo en áreas técnicas.
- Vehículo: Entidad referida a coches y/o bicicletas eléctricas pertenecientes a la empresa.
- Ciudad: Espacio en el cual la empresa estaciona los vehículos.
- Cliente: Personas registradas en el sistema con derecho al alquiler de vehículos.
- Sede: Lugar físico perteneciente a la empresa que gestiona la actividad de esta.
- Servicio: Espacio reservado para el mantenimiento de vehículos.
- Empleado: Personas que realizan su actividad en la empresa.
- Interfaz: Señala la conexión que se da de manera física y a nivel de utilidad entre dispositivos o sistemas.
- Base de Datos (BBDD): Conjunto de datos persistentes generados por la actividad de la aplicación.
- Java: Lenguaje de programación utilizado en el desarrollo de la aplicación.
- Herramienta CASE: Programas informáticos destinados a aumentar la productividad en el desarrollo de software.
- UML: Lenguaje Unificado de Modelado.

1.4 Referencias

- Estándar IEEE 830-1998.
- Pressman, R. S. Ingeniería del Software. Un enfoque Práctico. Sexta Edición. McGrawHill, 2005.
- Apuntes de la asignatura Ingeniería del Software.
- Apuntes de la asignatura Modelado de Software.
- Apuntes de otras asignaturas relacionadas.
- Wikipedia (definiciones).

1.5 Resumen

Este documento aporta información sobre el desarrollo, funcionalidad y descripción de la aplicación “GREENGO” y se divide en varias partes:

- En la primera sección se realiza una introducción general e informativa respecto al funcionamiento del software de nuestro programa. Tras esto se expresarán de manera más detallada todas las funciones que tendrá el programa, estas funciones se diferenciarán en módulos para su mejor legibilidad.
- En la segunda sección se expresarán las características del usuario, las restricciones que se imponen sobre los desarrolladores del producto y las dependencias que afectan al desarrollo de la aplicación.
- La tercera sección será donde se realice la especificación detallada de los requisitos de diseño, tales como el desarrollo de interfaces básicas preliminares, modelo del dominio, los casos de uso y demás requisitos y restricciones.

Debido al carácter evolutivo del software y teniendo como base los requisitos aprobados por el cliente, el software se desarrollará de manera que se pueda mantener con el paso del tiempo.



2. Descripción general

El software de gestión que se va a desarrollar se denomina "*GREENGO*" y se trata de un programa informático para gestionar vehículos pertenecientes a la empresa y sus alquileres, así como la información de ciudades y clientes.

La aplicación informática será usada por el gestor para realizar de manera correcta la inyección de vehículos, ciudades, clientes, sedes, servicios y empleados según los datos dados por el cliente de la aplicación. Así, podrá modificar, borrar y listar cualquier información que sea necesaria de esos datos previamente insertados, a partir de especificaciones dadas por los clientes o por interés del propio administrador.

En conclusión, se tratarán todos aquellos datos y entornos de usuario que sean necesarios para la gestión de alquileres de vehículos como principal función de la aplicación.

2.1 Perspectiva del producto

GREENGO está destinado al uso administrativo por la empresa GREENGO. Por tanto, aunque nuestros clientes finales sean las personas que deseen alquilar un vehículo, nuestro usuario principal será el gestor o el encargado.

Se instalará en cualquier ordenador (multiplataforma) ubicado en cualquiera de las sedes pertenecientes a la empresa, tratándose de una aplicación de escritorio para la cual los encargados serán las únicas personas que tengan acceso.

El producto será independiente, es decir, no forma parte de ningún sistema. Es necesario que esté conectada a la base de datos del sistema (local). Los datos de esta deben ser introducidos a mano ya que se trabaja desde una imagen limpia del sistema.

2.2 Funciones del producto

Este producto se trata de una aplicación informática dedicada a la gestión de alquileres de vehículos, por tanto, está destinada, principalmente, al gestor o administrador como usuario final.

La aplicación estará formada por varios módulos, partes diferenciadas del programa.

1. **Vehicles:** Coches y/o bicicletas eléctricas destinados a ser alquilados por los clientes.
2. **Cities:** Espacio que delimita el uso de los vehículos por parte de los clientes.
3. **Clients:** Personas previamente dadas de alta en el programa, las cuáles realizan el alquiler de los vehículos.
4. **Main_offices:** Espacio físico desde el cual se gestionan los servicios contratados y empleados que forman parte de la actividad de la empresa.
5. **Services:** Prestaciones contratadas para llevar a cabo el mantenimiento y puesta a punto de los vehículos pertenecientes a la empresa.
6. **Employees:** Trabajadores de la empresa, se clasifican en empleados con contrato fijo y empleados con contrato temporal.
7. **Rental:** alquileres que realizan los clientes hacia un vehículo, indicando los kilómetros a viajar.
8. **Contract:** contratos que la sede acuerda con los servicios, en los que se indica el nivel de servicios que la sede requiere a ellos.



Estos módulos tendrán las siguientes funciones:

- Vehicle:

Create_Vehicle: Creación de un vehículo en el sistema. Cada vehículo tendrá un ID único auto incremental generado por el sistema gestor de base de datos correspondiente. Al insertar un nuevo vehículo en el sistema, se insertarán todos datos de este por entrada de la interfaz de usuario. También se especificará si el vehículo se trata de una bicicleta o de un coche, completando un número de serie para el primero y una matrícula para el segundo.

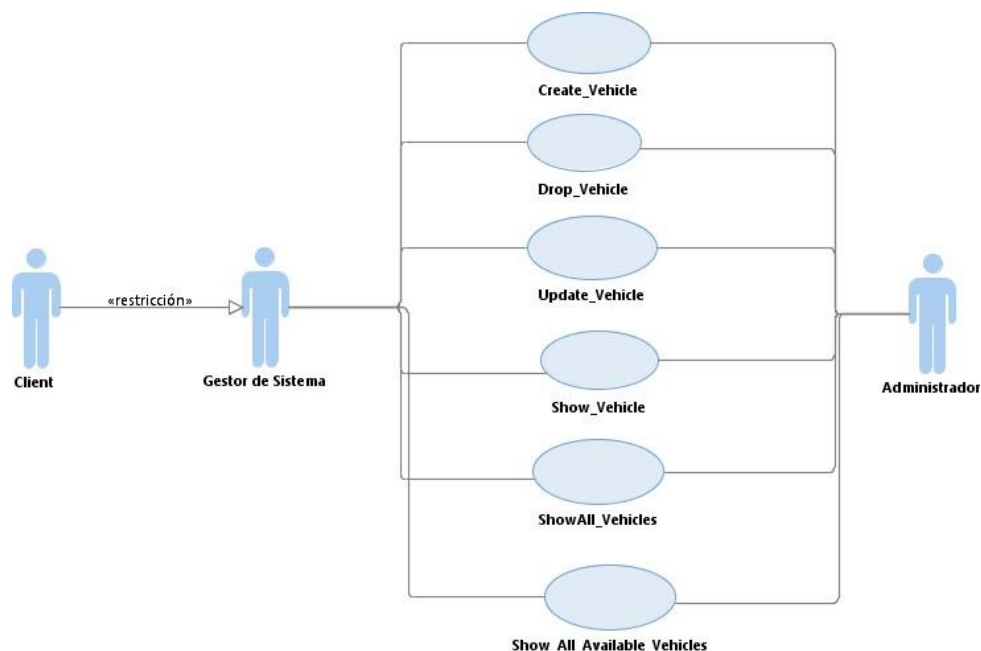
Drop_Vehicle: Borrado lógico de un vehículo activo del sistema a partir de su ID. El borrado de un vehículo implicará el borrado lógico de todos los registros de los clientes que lo han utilizado en el pasado.

ShowAllAvailable_Vehicles: Visualizar listado de vehículos activos disponibles para ser alquilados, es decir, que no están actualmente alquilados por ningún cliente.

ShowAll_Vehicles: Visualizar listado de todos los vehículos activos.

Update_Vehicle: Modificar la información referente a un vehículo a partir de su ID, mostrando previamente el estado original de sus atributos.

Show_Vehicle: Visualizar la información de un vehículo a partir de su ID, mostrando el estado actual de sus atributos, así como sus clientes anteriores.



- Service:

Create_Service: Creación de un servicio en el sistema. Este tendrá un ID único auto-incremental generado por el sistema gestor de base de datos correspondiente, además, en el registro es necesario proveer todos los datos requeridos por los atributos de servicio en el sistema.

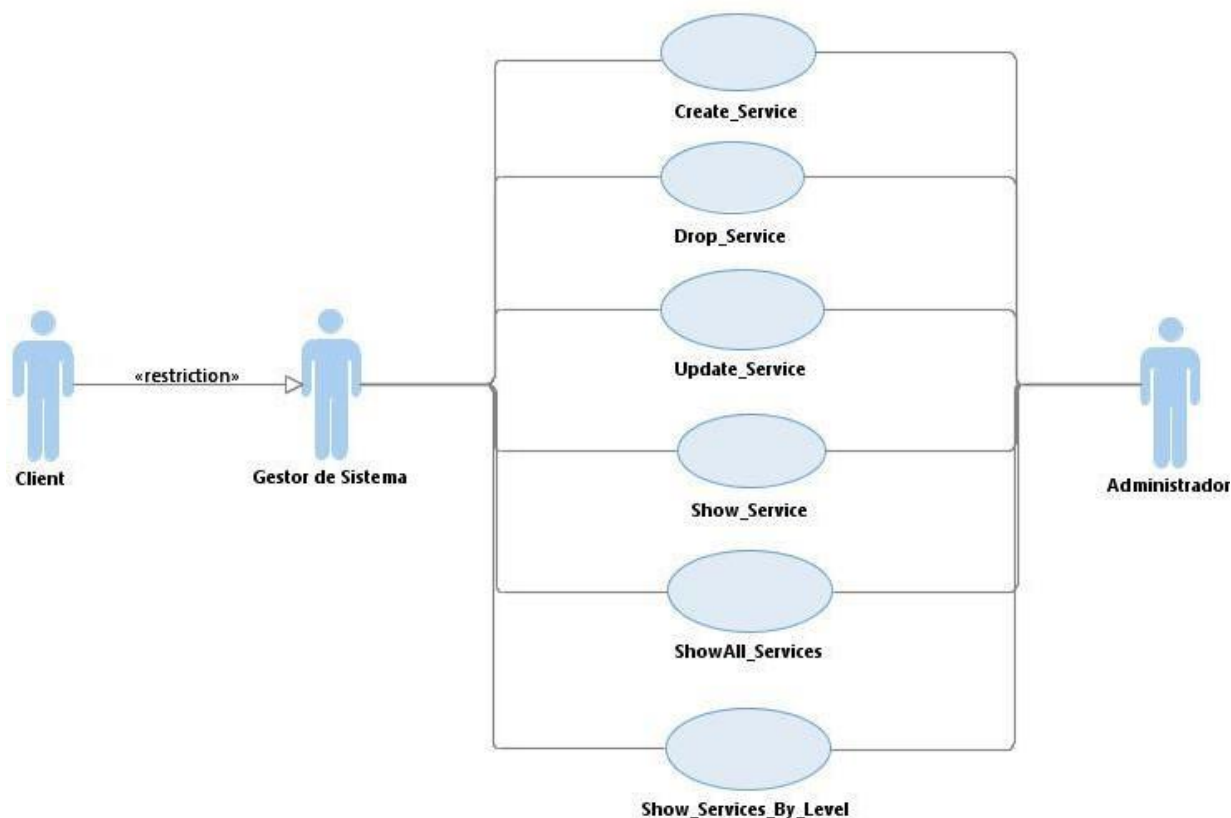
Drop_Service: Borrado lógico de un servicio activo del sistema a partir de su ID.

ShowAll_Services: Visualizar listado de los servicios activos, mostrando sus atributos.

Update_Service: Modificar la información referente a un servicio a partir de su ID, mostrando previamente el estado original de sus atributos.

Show_Service: Visualizar información de un servicio a partir de su ID, mostrando el estado actual de sus atributos.

Show_Services_By_Level: Visualizar listado de los servicios que proporcionan un nivel de servicio concreto.



- Client:

Create_Client: Creación de un cliente en el sistema. Cada cliente tendrá un ID único autoincremental generado por el sistema gestor de base de datos correspondiente. Al insertar un nuevo vehículo en el sistema, se insertarán todos datos de este por entrada del usuario.

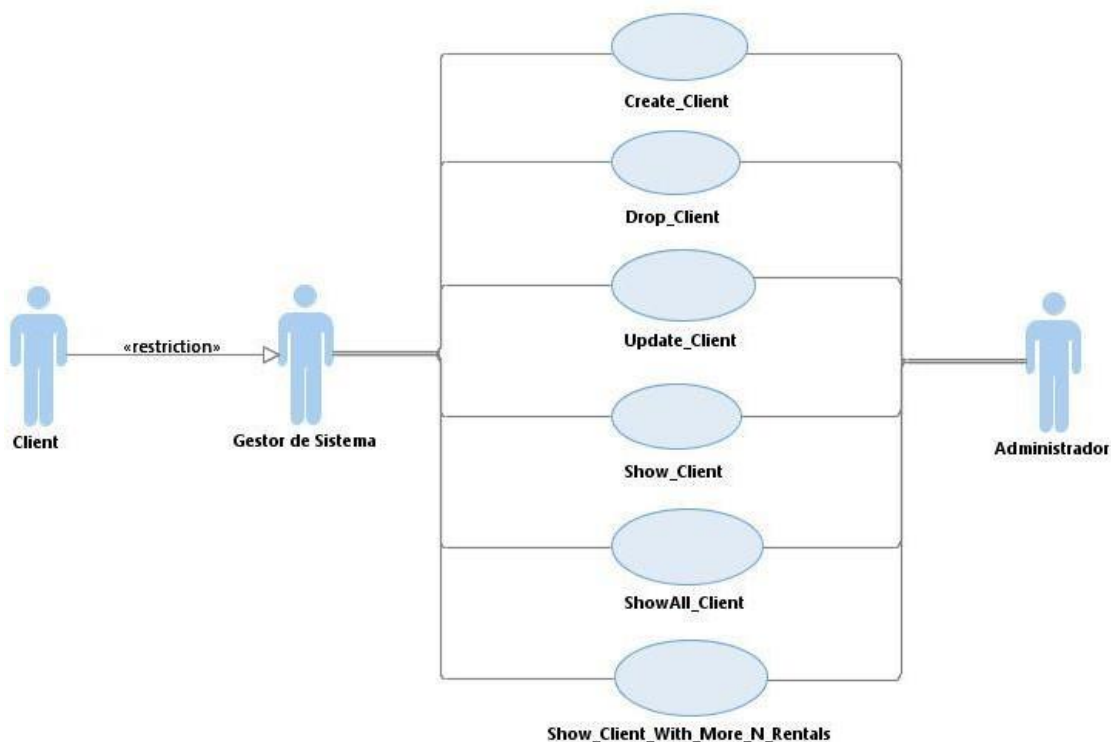
Drop_Client: Borrado lógico de un cliente activo del sistema a partir del ID introducido desde interfaz de usuario.

Update_Client: Modificación del estado de los atributos de un participante a partir de su ID, mostrando previamente el estado original de sus atributos.

Show_Client: Visualización de la información de un cliente a partir de su ID, mostrando el estado actual de sus atributos.

ShowAll_Client: Visualización de la información de todos los clientes activos, mostrando el estado actual de sus atributos.

Show_Clients_With_More_Than_N_rentals: Visualización de la información de todos los clientes que históricamente han realizado más de N alquileres, mostrando el estado actual de sus atributos.



- **Main_Office:**

Create_Main_Office: Creación de una sede en el sistema. Cada sede tendrá un ID único autoincremental generado por el sistema gestor de base de datos correspondiente, además, en el alta de una nueva sede, el usuario debe insertar todos los datos de ésta.

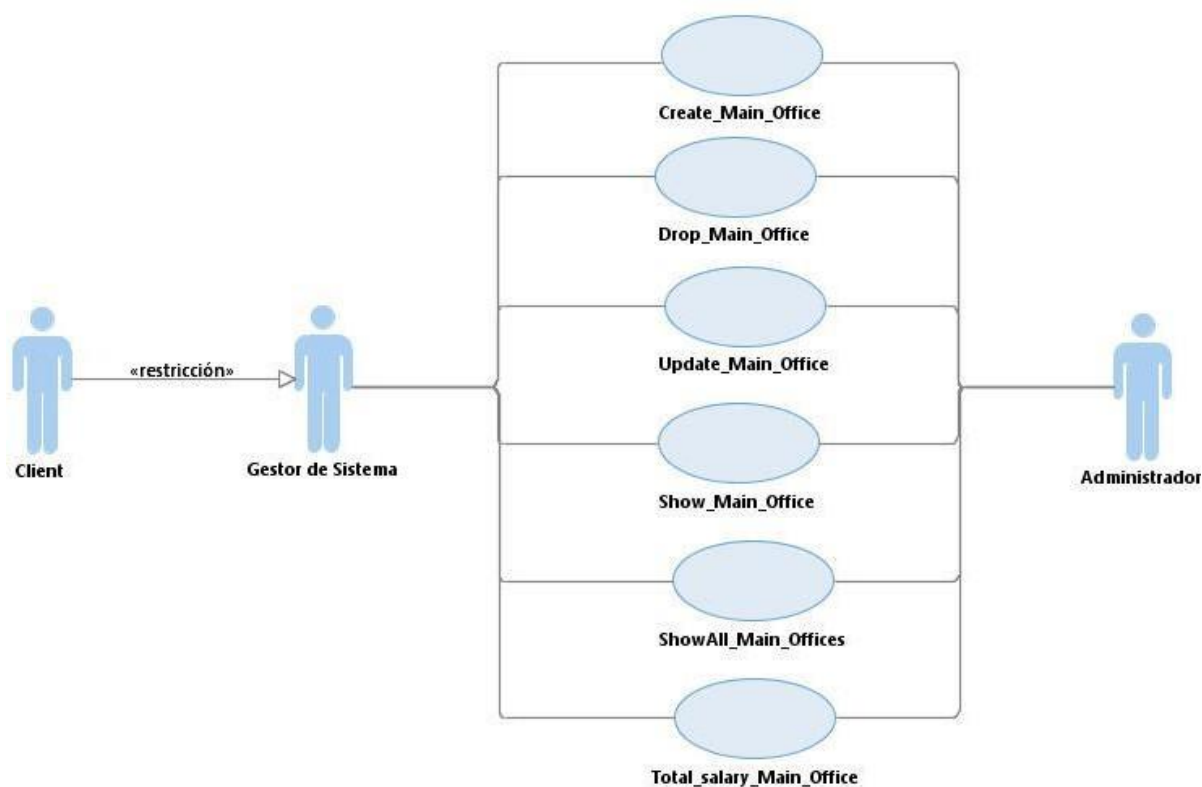
Drop_Main_Office: Borrado lógico de una sede del sistema a partir de su ID. El borrado de una sede implicará el borrado lógico de todos los empleados que desempeñen su trabajo en la misma.

Update_Main_Office: Modificar la información referente a una sede a partir de su ID, mostrando previamente el estado original de sus atributos.

ShowAll_Main_Office: Listado de todas las sedes activas en el sistema, mostrando el estado actual de sus atributos.

Show_Main_Office: Visualizar información de una sede a partir de su ID, mostrando el estado actual de sus atributos.

Total_salary_Main_Office: calcula el salario total de la sede, que está dado como la suma de los salarios de los empleados que trabajan en ella.



- Employee:

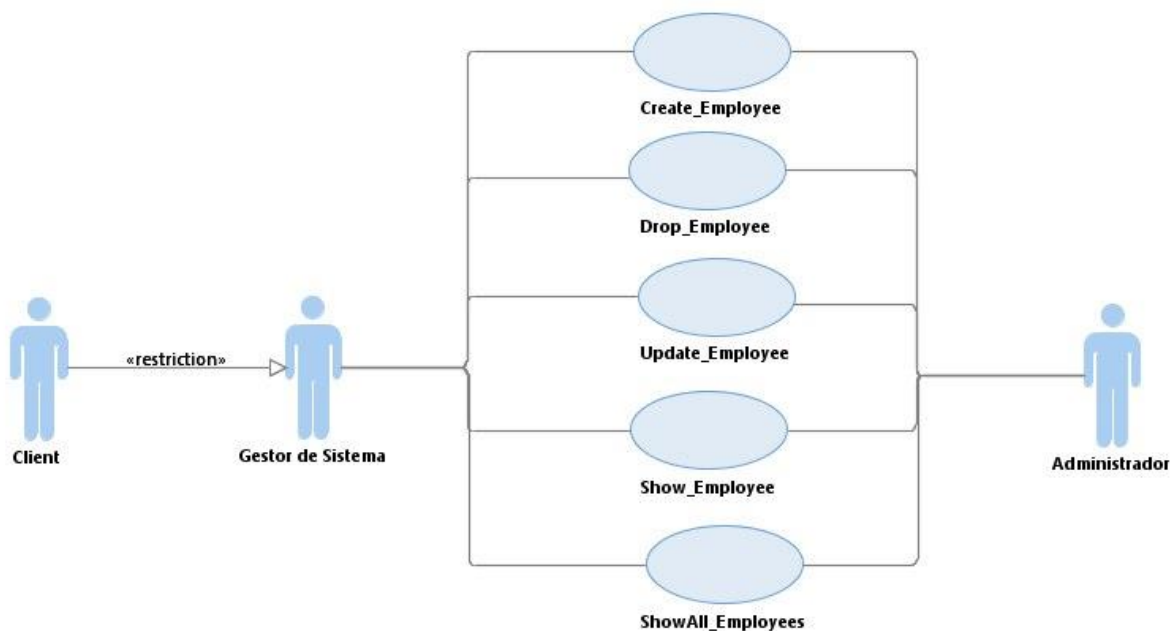
Create_Employee: Creación de un empleado en el sistema. Cada empleado tendrá un ID único autoincremental generado por el sistema gestor de base de datos correspondiente, además, se deberá especificar en el momento de alta si se trata de un empleado a tiempo completo o a tiempo parcial, así como el resto de sus atributos en el sistema.

Drop_Employee: Borrado lógico de un empleado a partir de su ID.

ShowAll_Employee: Listado de todos los empleados activos en el sistema independientemente del contrato que esté vigente (sean empleados a tiempo completo o a tiempo parcial), mostrando el estado actual de sus atributos.

Update_Employee: Modificar la información referente al empleado a partir de su ID, mostrando previamente el estado original de sus atributos.

Show_Employee: Visualizar información de un empleado a partir de su ID, mostrando el estado actual de sus atributos.



- City:

Create_City: Creación de una ciudad en el sistema. Cada ciudad tendrá un ID único autoincremental generado por el sistema gestor de base de datos correspondiente, además, en el alta de una nueva ciudad, el usuario debe insertar todos los datos de ésta.

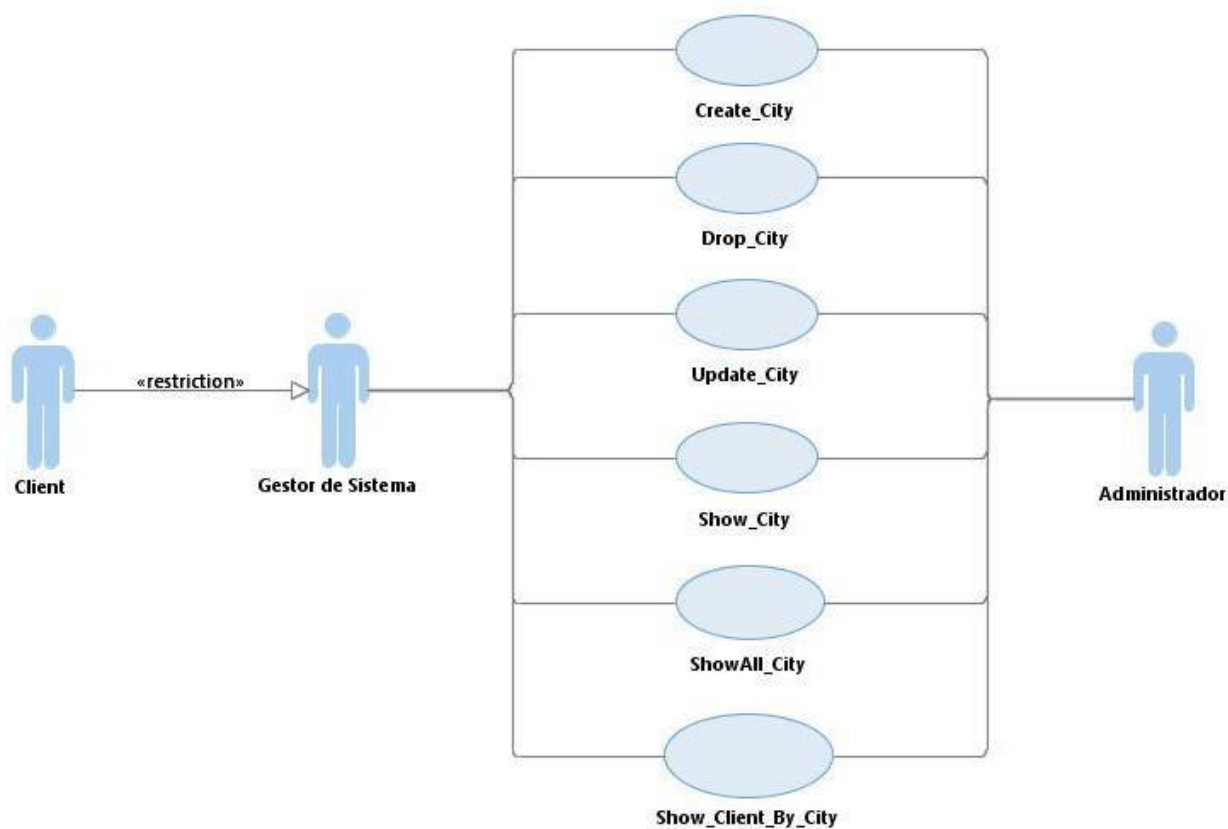
Drop_City: Borrado lógico de una ciudad activa en el sistema a partir de su ID.

Update_City: Modificación de la información referente a una ciudad en el sistema a partir de su ID, mostrando previamente el estado original de sus atributos.

Show_City: Visualizar información de una ciudad a partir de su ID, mostrando el estado actual de sus atributos.

ShowAll_City: Listado de todas las ciudades activas en el sistema, mostrando el estado actual de sus atributos.

Show_Clients_By_City: Listado de todos los clientes que han alquilado algún vehículo en la ciudad introducida en la interfaz de usuario.



- Rental:

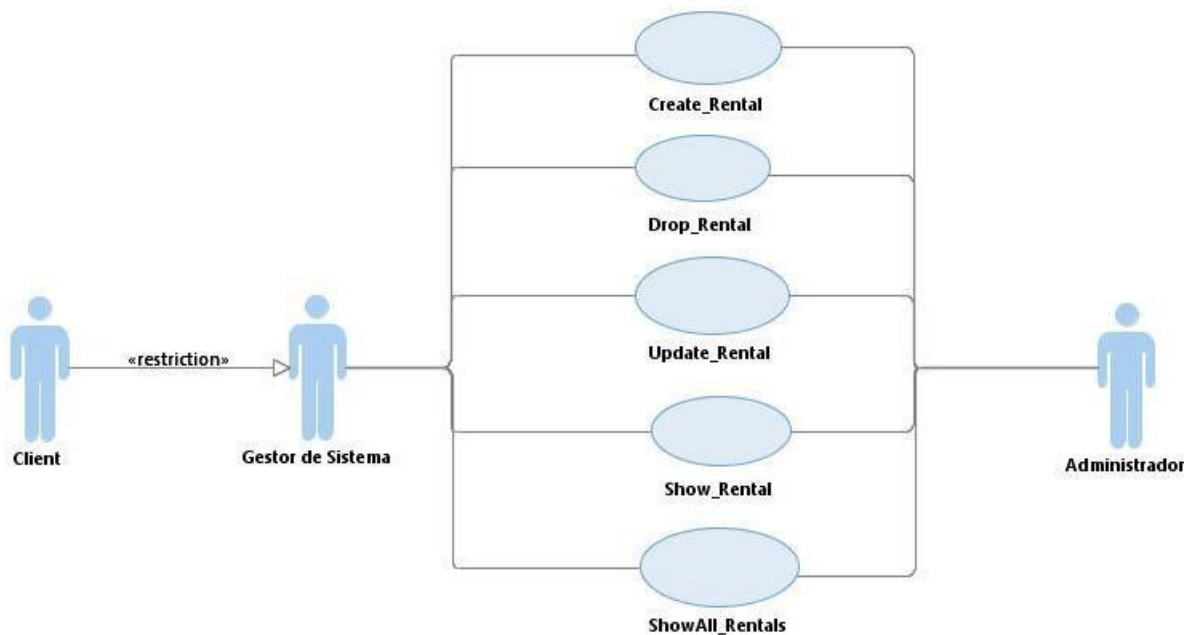
Create_Rental: Creación de un alquiler dado un cliente, un vehículo y el kilometraje que el cliente indica. Cada alquiler tendrá un ID único autogenerado por el sistema gestor de base de datos correspondiente.

Drop_Rental: Borrado lógico de un alquiler activo en el sistema a partir de su ID.

Update_Rental: Modificación de la información referente al kilometraje del alquiler a partir del ID de este, mostrando previamente el estado original de sus atributos.

Show_Rental: Visualizar información de un alquiler a partir de su ID, mostrando el estado actual de sus atributos.

ShowAll_Rental: Listado de todos los alquileres activos en el sistema, mostrando el estado actual de sus atributos.



- Contract:

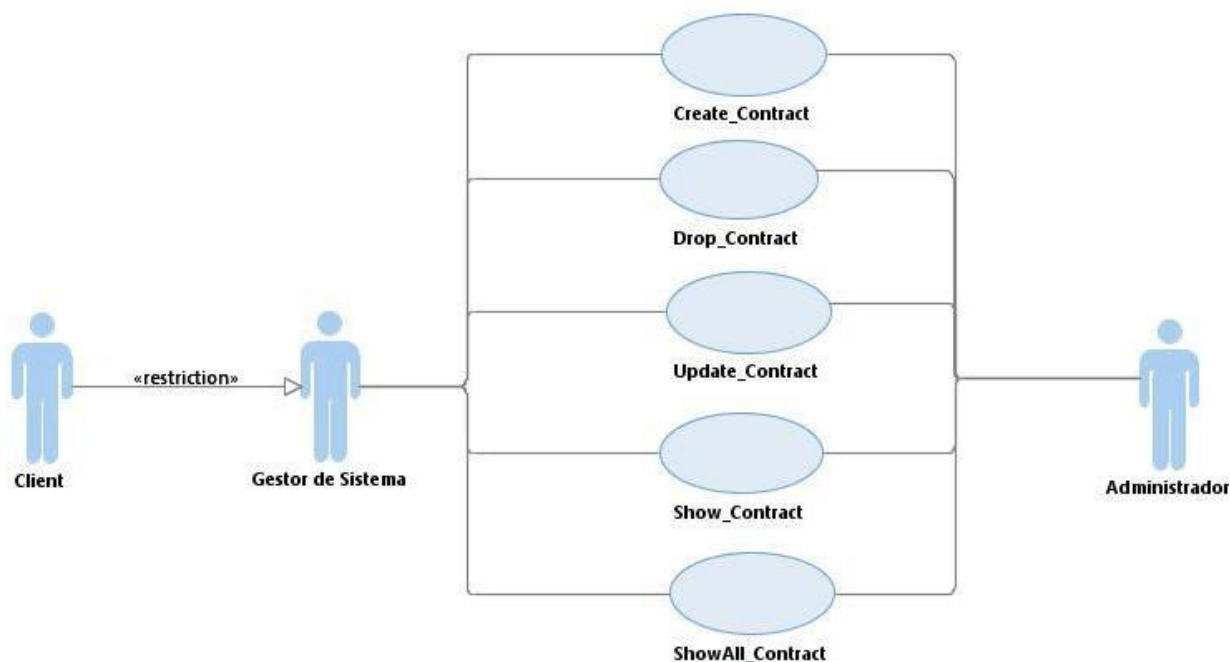
Create_Contract: Creación de un contrato dado una sede, un servicio y un nivel que el servicio proporciona. Cada contrato tendrá un ID único autogenerado por el sistema gestor de base de datos correspondiente.

Drop_Contract: Borrado lógico de un contrato dado su identificador.

Update_Contract: Modificación de la información.

Show_Contract: Visualizar información de un contrato a partir de su identificador, mostrando el estado actual de sus atributos.

ShowAll_Contract: Listado de todos los contratos activos en el sistema, mostrando el estado actual de sus atributos.



2.3 Características de usuario

La aplicación software “GREENGO” está destinada al usuario que sea gerente o encargado, es decir, no está destinado al usuario final que desea alquilar un vehículo.

Aunque está destinada a que cualquier persona con conocimientos básicos en informática sepa usarla, la complejidad viene dada por el campo profesional a la que está destinada. Dicha aplicación tendrá un único usuario principal:

- **Trabajador/encargado:** Este se centrará en realizar la tarea solicitada por el cliente, para ello debe saber cómo realizar las funciones del sistema a partir de lo solicitado por el cliente.

De manera independiente podemos distinguir otro agente en la aplicación, se trata del administrador del sistema, el cual debe arreglar cualquier problema que pueda surgir, como la inserción duplicada de datos, errores al visualizar información, etc.

2.4 Restricciones

En este apartado se describen las restricciones en las que se basa el desarrollo de la aplicación *GREENGO*:

- **Políticas de empresa:**
 - La empresa sólo poseerá dos tipos de vehículos: coches y bicicletas, ambos eléctricos.
 - Los clientes sólo podrán realizar un alquiler simultáneo.
 - Los clientes tienen acceso al alquiler de cualquier vehículo, tanto coche como bicicleta.
- **Políticas de regulación:**
 - Se seguirá el estándar “IEEE Std. 830 – 1998”.
- **Limitaciones de hardware:**
 - Al tratarse de una aplicación de escritorio quedaría limitada en dispositivos portátiles como móviles o *tablets*. Aunque sea multiplataforma no está destinada a otros dispositivos diferentes a un ordenador.
 - Es necesario una conexión a la red para realizar conexión a la base de datos.
- **Operaciones en paralelo:**
 - Se permite la introducción de varias conexiones concurrentemente, utilizando la base de datos como un método de sincronización en línea.
- **Limitaciones del lenguaje de alto nivel:**
 - El lenguaje utilizado será Java.
- **Requisitos de fiabilidad:**
 - Los datos se guardarán de forma segura, restringiendo acceso a terceros y almacenándolos en la base de datos. Se comprobará que todas las operaciones de almacenamiento o carga de datos se hayan realizado correctamente.



- **Tiempos de aplicación:**

- Las consultas a la base de datos no serán superiores a 2 segundos y las operaciones que se hagan en la aplicación que repercuten conexiones a la base de datos no superarán los 2 minutos y 30 segundos. Las operaciones de listado no tienen tiempo máximo de ejecución por la posibilidad de manejar tamaños elevados.

2.5 Supuestos y dependencias

GREENGO se diseña para ordenadores de cualquier sistema operativo debido a que su tecnología es multiplataforma, además cabe destacar que no sería necesario un dispositivo con grandes requisitos debido a su naturaleza básica. Las entradas del programa serán proporcionadas a través de teclado y ratón.

La aplicación está basada para usarse con una conexión a una base de datos, siendo esta de manera local, por tanto, si se extrapola a conexiones externas, hay que adaptar la conexión para que todos los dispositivos puedan acceder a la base de datos desde donde estén situados.

2.6 Requisitos futuros

En un futuro se podría expandir la aplicación para que los clientes pudieran gestionar sus alquileres. De este modo, disminuiríamos la carga de trabajo del encargado o gerente.

También se daría la posibilidad de una conexión a partir de una web y que se pueda gestionar a través de esta. Se podría adaptar el programa para más sistemas operativos llevándolo a dispositivos móviles.

Todas propuestas quedarían a expensas de que el cliente las acepte.

3. Requisitos específicos

3.1 Interfaces externas

3.1.1 Interfaces de usuario

La interfaz de *GREENGO* será lo más clara y sencilla posible, siendo utilizable de manera eficiente desde el primer uso. Estará compuesta de ventanas que incluyen botones para añadir y quitar elementos, listas de los elementos y áreas de texto para insertar datos de los elementos.

Las entradas se harán mediante el uso de ratón, o "TouchPad", y teclado, teniendo la salida de los datos a través de la pantalla del ordenador con la interfaz del sistema.

Con respecto a la interfaz de usuario, en los elementos despegables figurarán múltiples elementos correspondientes a la función en la que se encuentre el usuario.

Especificaciones de los atributos de los elementos de la interfaz:



- **Vehicle:**
 - **ID:** Código numérico autoincrementado que identifica a los distintos vehículos que la aplicación maneje.
 - **Brand:** Cadena de caracteres que identifica la casa de vehículos de donde este procede. Por ejemplo: *Skoda*.
 - **Occupied:** Booleano que indica si el vehículo está disponible o no para los clientes.
 - **Estimated_duration:** Cantidad de kilómetros estimada que puede realizar un vehículo antes de dejar de funcionar.
 - **Active:** Booleano que nos indica si dicho vehículo está operativo y en servicio.
 - **Km_travelled:** Entero que nos indica la cantidad de kilómetros que dicho vehículo ha recorrido hasta la fecha.



Especialización de vehículos:

En este apartado veremos los tipos de vehículos que el sistema acepta:

- **Car:**

Plate: Cadena de caracteres que identifica de manera inequívoca al vehículo en cuestión.

- **Bicycle:**

Serial_number: Entero que identifica de manera única a la bicicleta.

- **Client:**
 - **ID:** Código numérico autoincrementado que identifica unívocamente a los clientes.
 - **Id_card_number:** Cadena de caracteres que acredita al cliente en cuestión.
 - **Active:** Booleano que nos indica si dicho cliente tiene alquilado actualmente algún vehículo.
 - **Rentals_number:** Entero que denota cuántos alquileres ha realizado el cliente hasta la fecha.
- **City:**
 - **ID:** Código numérico autoincrementado que identifica unívocamente la ciudad.
 - **Name:** Cadena de caracteres que denota la denominación de la ciudad
 - **Active:** Booleano que indica si la ciudad está disponible para la circulación de vehículos.
- **Service:**
 - **ID:** Código numérico autoincrementado que identifica unívocamente el servicio.
 - **Type:** Cadena de caracteres que indica la especificación del servicio.
 - **Address:** Cadena de caracteres que da información acerca de la localización del servicio.
 - **Capacity:** Entero que indica el número de vehículos máximo que el servicio puede gestionar.
 - **Active:** Booleano que indica si el servicio está disponible para la recepción de vehículos averiados.
 - **Vehicles_Attended:** Entero que refleja el número de vehículos atendidos por el servicio hasta la fecha.



- **Main_Office:**
 - **ID:** Código numérico autoincrementado que identifica unívocamente las sedes.
 - **City:** Cadena de caracteres que indica la ciudad en la que se ubica la sede.
 - **Active:** Booleano que indica si la sede se encuentra disponible o no.
 - **Adress:** Cadena de caracteres cuyo objetivo es el de identificar la ubicación exacta de la sede dentro de la ciudad concretada.
- **Employee:**
 - **ID:** Código numérico autoincrementado que identifica de manera única al empleado.
 - **Id_card_number:** Cadena de caracteres que refleja el Documento Nacional de Identidad del empleado.
 - **Salary:** Número real que indica la cantidad de dinero que el empleado recibe mensualmente.
 - **Active:** booleano que nos indica si el empleado está actualmente dado de alta o, por el contrario, está dado de baja.
 - **Id_main_office:** Identificador correspondiente a la sede a la que el empleado pertenece.
- ❖ **Especialización de empleados:**
 En este apartado concretaremos entre los distintos tipos de empleados que la aplicación soporta actualmente.
 - **Temporary:**
 - **Worked_hours:** Entero que indica la cantidad de horas que debe trabajar el empleado al día.
 - **Permanent:**
 - **Apportionment:** Float que expresa el prorratio que le corresponde al empleado.
- **Rental:**
 - **ID:** Código numérico autoincrementado que identifica de manera única el alquiler.
 - **Km_rented:** Entero que indica el kilometraje contratado por el cliente.
 - **ID_Client:** Identificador correspondiente al cliente que alquila el vehículo.
 - **ID_Vehicle:** Identificador correspondiente al vehículo que dicho cliente contrata.
 - **Active:** Booleano que indica la disponibilidad del alquiler.
 - **Date:** tipo date que indica la fecha en la que realiza el alquiler. Esta se divide a su vez en “**Date_from**” y “**Date_to**” que indican la fecha de inicio y vencimiento del alquiler respectivamente.
- **Contract:**
 - **ID:** Código numérico autoincrementado que identifica de manera única el contrato.
 - **Service_Level:** Entero que indica el nivel de servicio que este ofrece a la sede.
 - **ID_Main_Office:** Identificador correspondiente a la sede que contrata los servicios.
 - **ID_Service:** Identificador que corresponde al servicio contratado con la sede.
 - **Active:** Booleano que indica la disponibilidad del contrato.



La siguiente pantalla sigue la configuración estándar, pero se especifica como seria en la parte del programa de las entidades de sede, empleados y servicios.

Title

Main Office

Employees

Services

Contract

Create Main Office

Drop Main Office

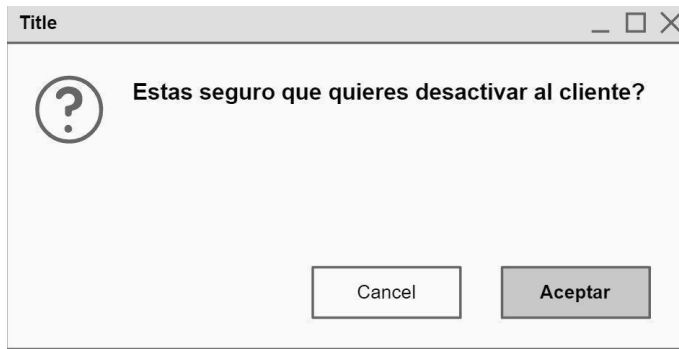
View Main Office

List Main Office

Update Main Office

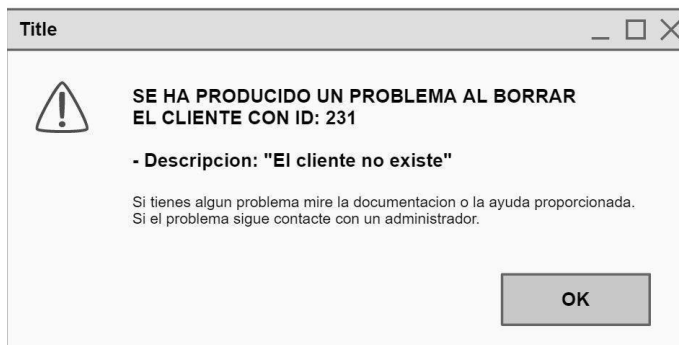
ID	ADRESS	CITY	ACTIVE
1	Friedrichstraße 96, 10117	Berlin	True
2	Av. de los Canteros, 4, 28430	Alpedrete	True
3	99 Rue de Rivoli, 75001	Paris	True
4	Calle de la Montera, 37, 28013	Madrid	True
5	Antonio Machado, 20, 28863	Cobena	True
6	Calle Virgen de la Soterraña, 20, 05005	Avila	False





Como se ha mostrado en las imágenes se ve un ejemplo de la función listar.

Destaca que la inserción de datos será a partir de formulario en paneles diferentes, sin modificar el panel principal. También se mostrará paneles independientes en caso de error o cuando haya que mostrar un mensaje informativo.



En la imagen a continuación se ven dos ejemplos de paneles independientes, uno de ellos en el que se solicita asegurar de una desactivación de un cliente y en el segundo, un mensaje de error por la no existencia del cliente a borrar.

3.1.2 Interfaces de software

GREENGO dispone de una base de datos local que auto gestiona con un único acceso para garantizar la transcendencia de las operaciones realizadas por el usuario de la aplicación.

Como se expresa en otros apartados se puede acceder a esta desde diferentes puntos de acceso, lo cual permite realizar acciones sobre la base de datos de manera paralela.

En las pruebas y el propósito del proyecto es general un software completo, pero debido a limitaciones se realizará el sistema en local con un dispositivo. Esto proporcionara robustez en el uso al realizar las pruebas y en el comienzo de su uso.

3.1.3 Interfaces de hardware

El hardware necesario para ejecutar nuestra aplicación tendrá como requisitos mínimos:

- **Procesador:** 1GHz o superior.
- **Memoria mínima:** 1 GB RAM.
- **Espacio libre:** 500 MB.
- **Sistema Operativo:** Windows 7.0+, Mac OS OS X 10.9.0+ y Linux.
- **Pantalla:** 7.0" o superior.



3.2 Funciones

A continuación, se mostrarán todas las funciones de manera detallada y con el diagrama de actividades específico de cada uno.

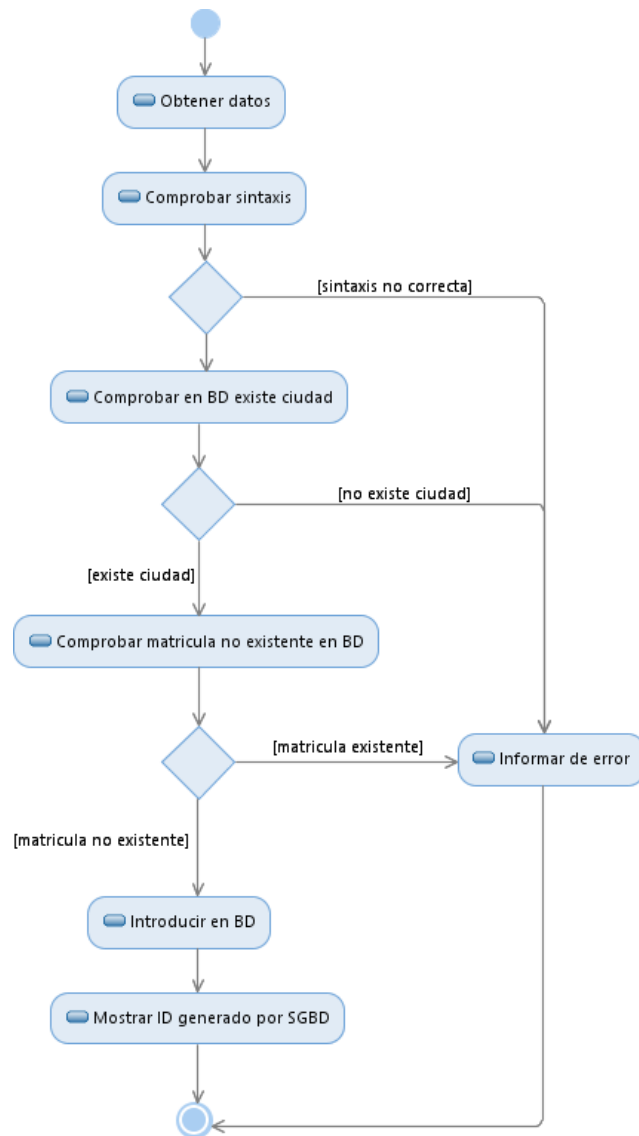
Aclaraciones:

- Cuando se trata la “Comprobación de datos insertados” además de realizar el paseo necesario para ver si se han insertado en formato correcto, se comprueba si estos datos existen en la base de datos (Teniendo la misma iteración recursiva en caso de fallo).
- Para mantener constancia en los datos en determinadas ocasiones será necesario comprobar que entidades dependen de la borrada y borrar esas dependencias si es necesario. “Comprobar dependencias de...”

- **Vehicle**

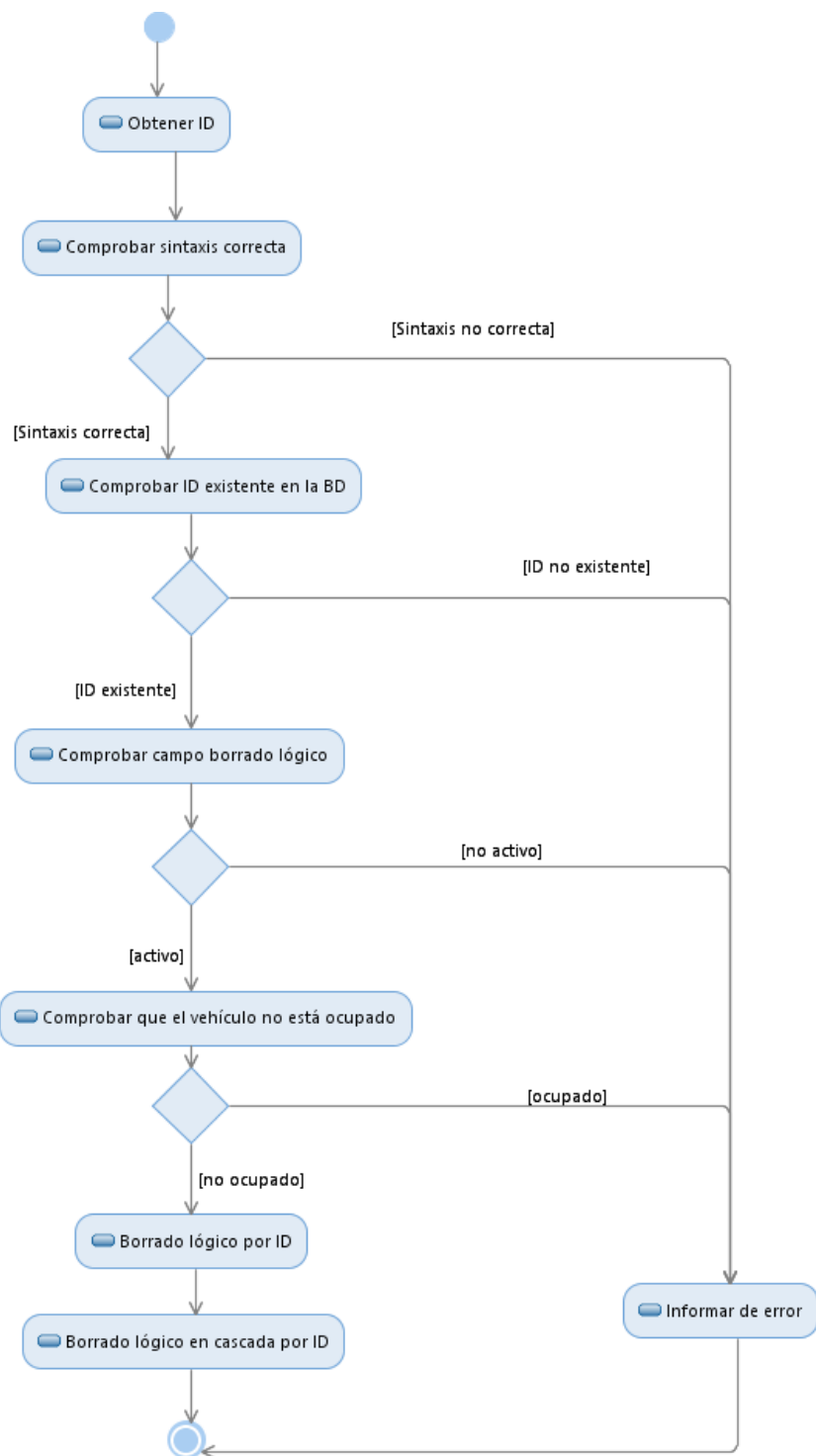
3.2.1.1 CREATE_VEHICLE	
Descrita por	(nombre gestor)
Prioridad	Alta
Estabilidad	Alta
Descripción	Añadir un vehículo con un estado concreto al sistema de GREENGO
Actor	Encargado/usuario de software
Entrada	Insertar valores que satisfagan los atributos de la entidad vehículo
Salida	Mensaje con ID o de error
Origen	Interfaz usuario - Gestor del sistema
Destino	Sistema/BD de GREENGO
Necesita	BD de GREENGO
Acciones	Descritas en el diagrama de actividades
Precondición	No existe vehículo en el sistema que posea el ID introducido y no existe vehículo con la misma matrícula o número de serie en el sistema.
Postcondición	Añadir vehículo con valores en sus atributos iguales a los datos introducidos por entrada a la BD
Efectos laterales	-





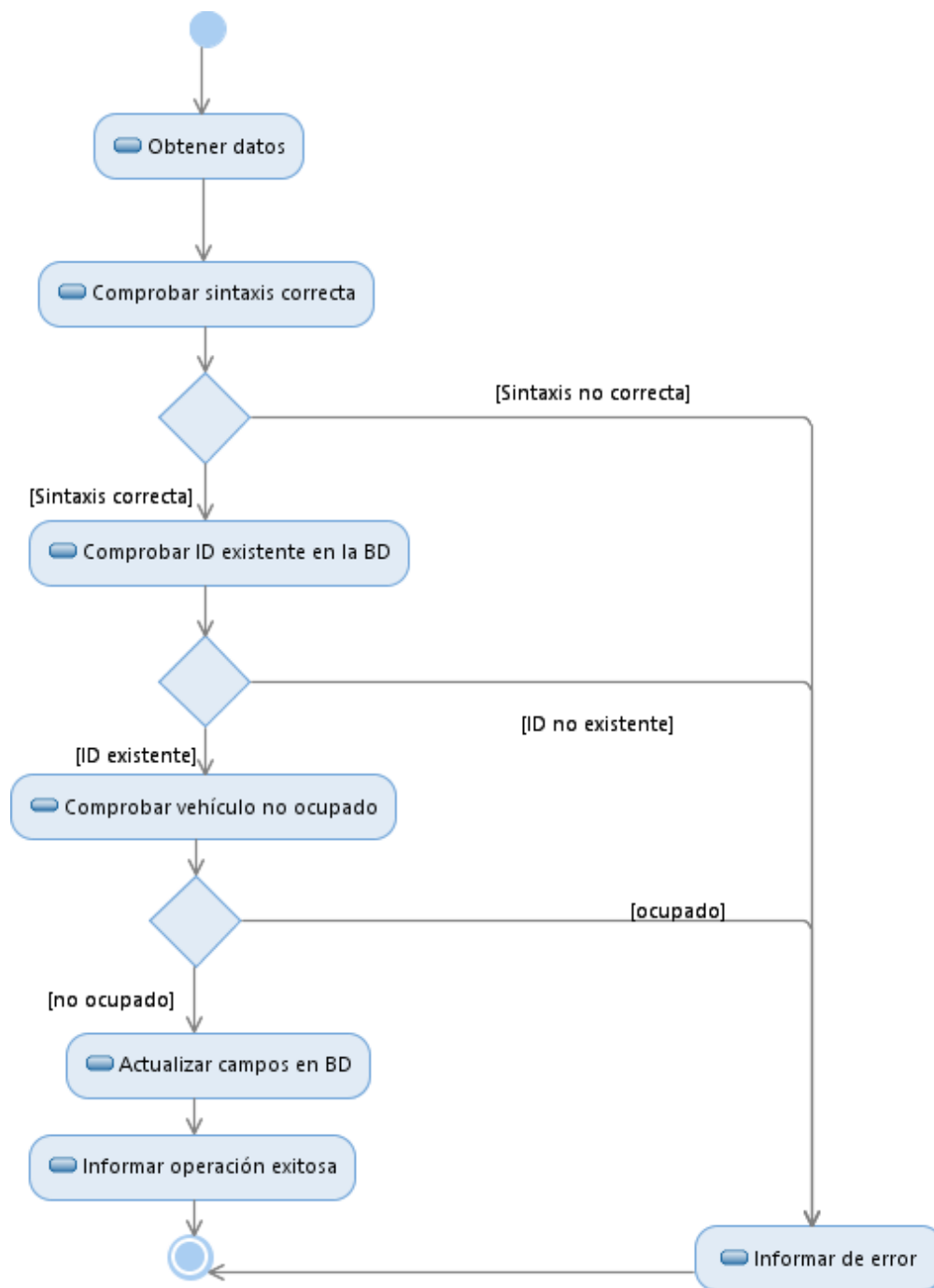
3.2.1.2 DROP_VEHICLE	
Descrita por	(nombre gestor)
Prioridad	Alta
Estabilidad	Alta
Descripción	Inserta el valor 'FALSE' en el atributo <i>activo</i> de un vehículo de la BD
Actor	Encargado/usuario de software
Entrada	ID del vehículo que se quiera dar de baja.
Salida	Mensaje de confirmación o error
Origen	Interfaz usuario - Gestor del sistema
Destino	Sistema/BD de GREENGO
Necesita	BD de GREENGO
Acciones	Descritas en el diagrama de actividades
Precondición	Existe un vehículo con el ID insertado activo y desocupado.
Postcondición	Atributo <i>activo</i> con valor 'FALSE'
Efectos laterales	-





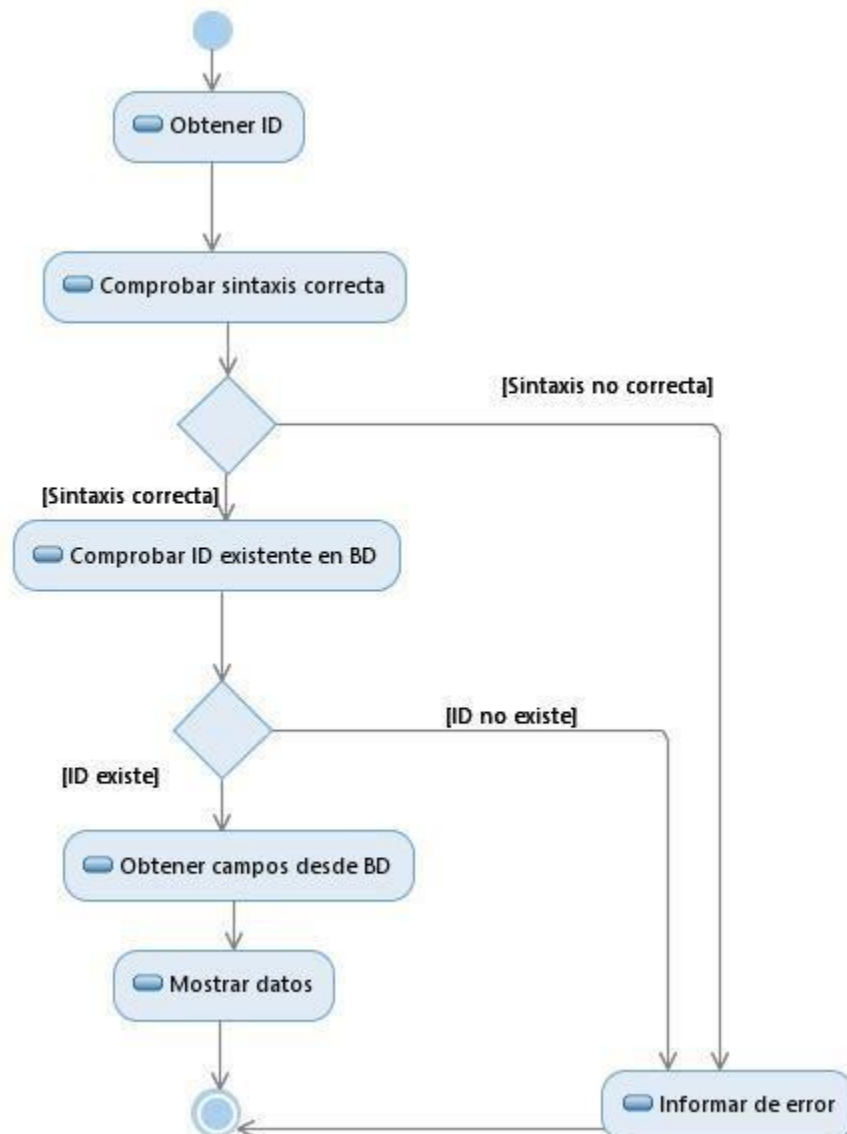
3.2.1.3 UPDATE_VEHICLE	
Descrita por	(nombre gestor)
Prioridad	Media
Estabilidad	Alta
Descripción	Modificar el estado de los atributos de un vehículo concreto
Actor	Gestor Base de datos
Entrada	ID, nuevo estado atributos a modificar
Salida	Mensaje de confirmación o error
Origen	Interfaz usuario - Gestor del sistema
Destino	Sistema/BD de GREENGO
Necesita	BD de GREENGO
Acciones	Descritas en el diagrama de actividades
Precondición	Vehículo existente en la BD y desocupado.
Postcondición	Modificar datos del vehículo
Efectos laterales	-



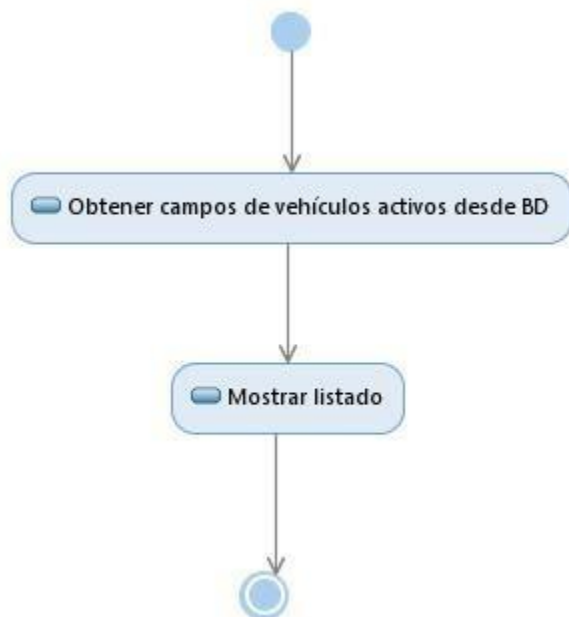


3.2.1.4 SHOW_VEHICLE	
Descrita por	(nombre gestor)
Prioridad	Media
Estabilidad	Alta
Descripción	Visualización del estado actual de los atributos de un vehículo.
Actor	Gestor Base de datos
Entrada	ID
Salida	Estado actual atributos o error.
Origen	Interfaz usuario - Gestor del sistema
Destino	Interfaz usuario
Necesita	BD de GREENGO
Acciones	Descritas en el diagrama de actividades
Precondición	Vehículo existente en la BD.
Postcondición	-
Efectos laterales	-

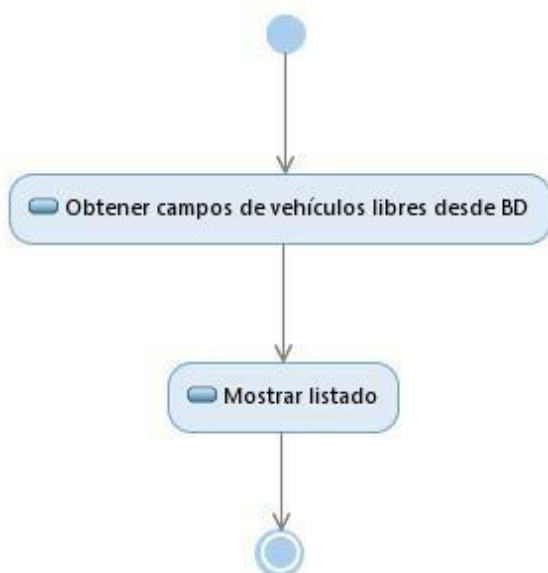




3.2.1.5 SHOWALL_VEHICLES	
Descrita por	(nombre gestor)
Prioridad	Media
Estabilidad	Alta
Descripción	Visualización del estado actual de los atributos de los vehículos pertenecientes al listado de vehículos con el atributo <i>activo</i> con estado actual 'TRUE'.
Actor	Gestor Base de datos
Entrada	-
Salida	Listado de vehículos
Origen	Interfaz usuario - Gestor del sistema
Destino	Interfaz usuario
Necesita	BD de GREENGO
Acciones	Descritas en el diagrama de actividades
Precondición	-
Postcondición	-
Efectos laterales	-

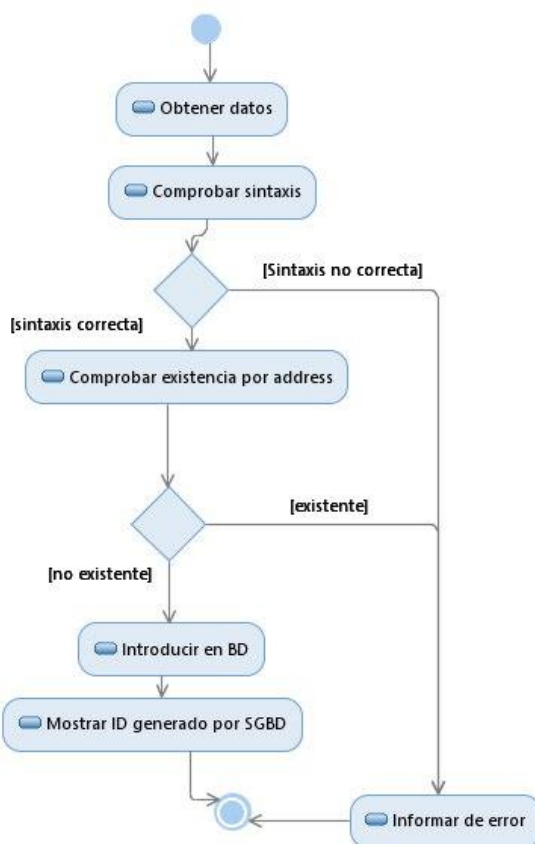


3.2.1.6 SHOWALLAVAILABLE_ VEHICLES	
Descrita por	(nombre gestor)
Prioridad	Media
Estabilidad	Alta
Descripción	Visualización del estado actual de los atributos de los vehículos pertenecientes al listado de vehículos con el atributo <i>ocupado</i> con estado actual 'FALSE'.
Actor	Gestor Base de datos
Entrada	-
Salida	Listado de vehículos
Origen	Interfaz usuario - Gestor del sistema
Destino	Interfaz usuario
Necesita	BD de GREENGO
Acciones	Descritas en el diagrama de actividades
Precondición	-
Postcondición	-
Efectos laterales	-



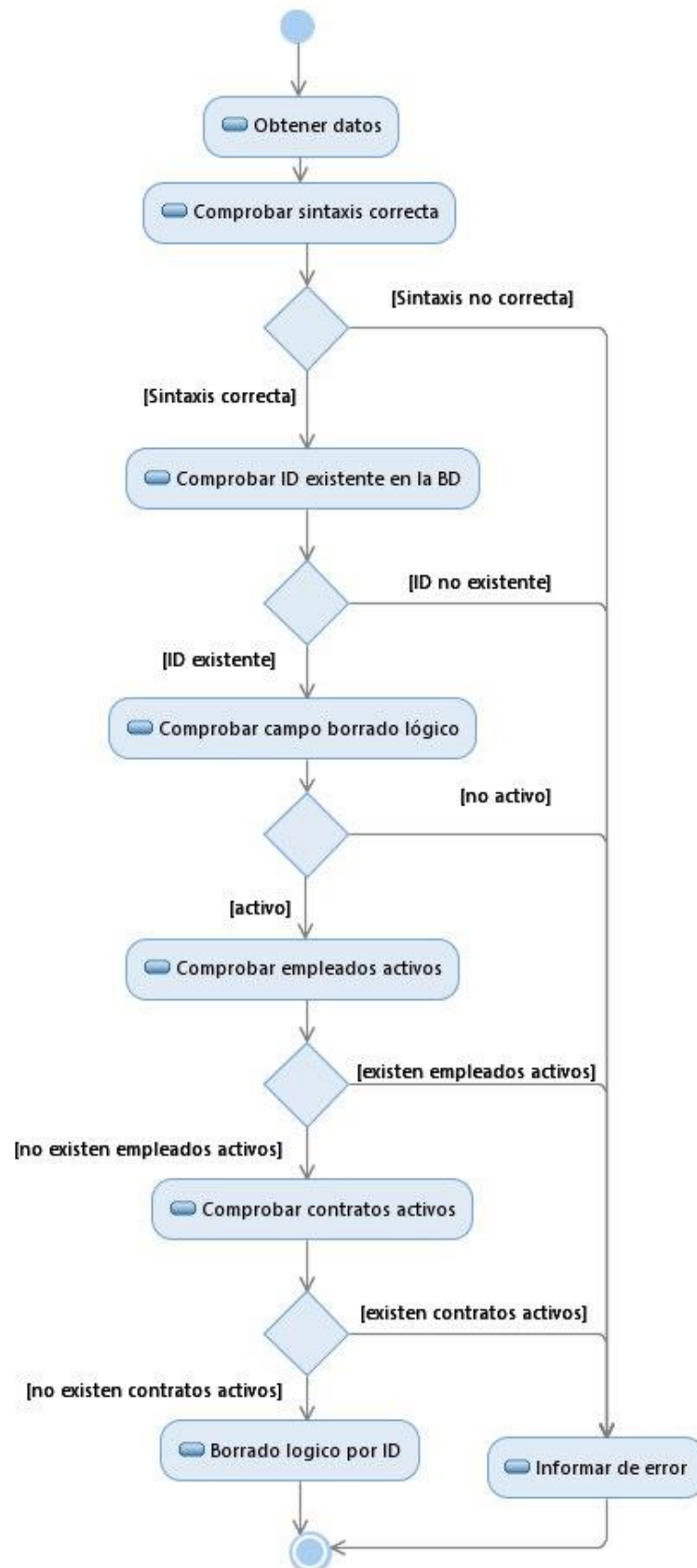
• Main Office

3.2.2.1 CREATE_MAIN_OFFICE	
Descrita por	(nombre gestor)
Prioridad	Alta
Estabilidad	Alta
Descripción	Añadir una sede con un estado concreto al sistema de GREENGO
Actor	Encargado/usuario de software
Entrada	Insertar valores que satisfagan los atributos de la entidad sede
Salida	Mensaje con ID o de error
Origen	Interfaz usuario - Gestor del sistema
Destino	Sistema/BD de GREENGO
Necesita	BD de GREENGO
Acciones	Descritas en el diagrama de actividades
Precondición	No existe sede en el sistema que posea el ID introducido
Postcondición	Añadir sede con valores en sus atributos iguales a los datos introducidos por entrada a la BD
Efectos laterales	-



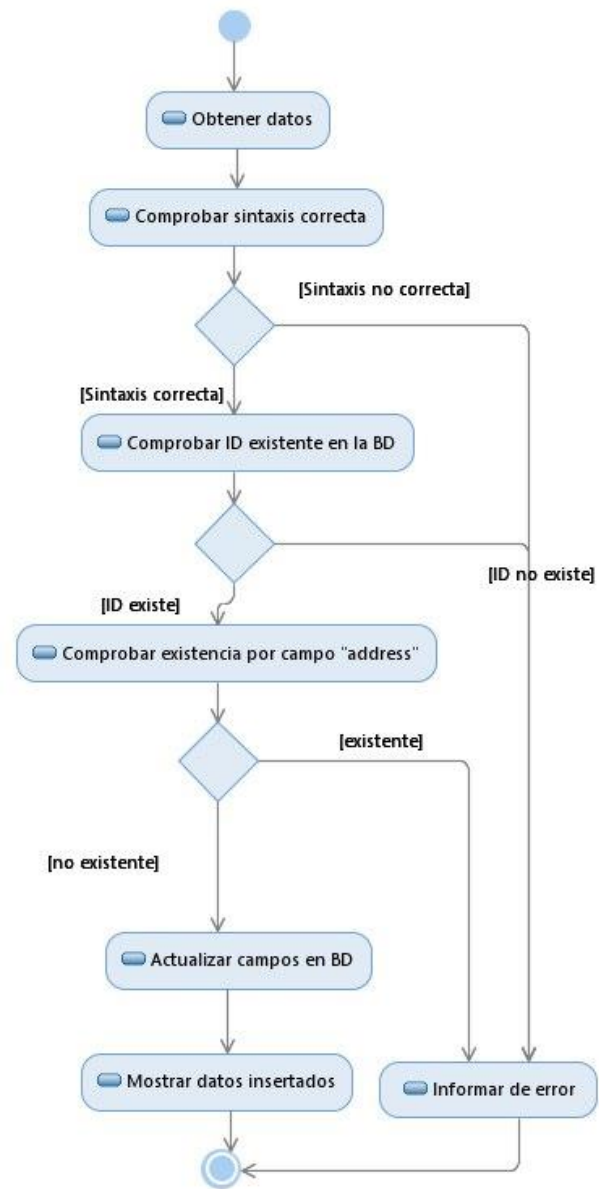
3.2.2.2 DROP_MAIN_OFFICE	
Descrita por	(nombre gestor)
Prioridad	Alta
Estabilidad	Alta
Descripción	Inserta el valor 'FALSE' en el atributo <i>activo</i> de una sede de la BD
Actor	Encargado/usuario de software
Entrada	D de la sede que se quiera dar de baja.
Salida	Mensaje de confirmación o error
Origen	Interfaz usuario - Gestor del sistema
Destino	Sistema/BD de GREENGO
Necesita	BD de GREENGO
Acciones	Descritas en el diagrama de actividades
Precondición	Existe una sede con el ID insertado
Postcondición	Atributo <i>activo</i> con valor 'FALSE'
Efectos laterales	Eliminación lógica de los empleados asociados a dicha sede





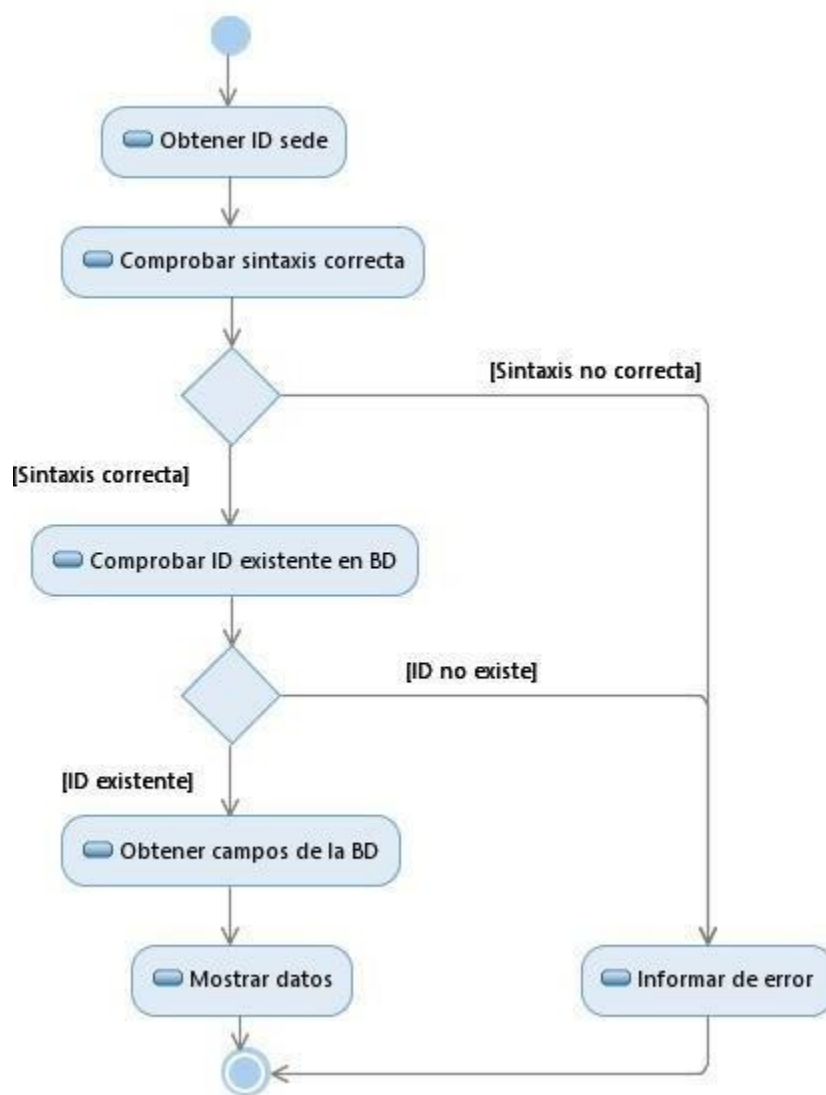
3.2.2.3 UPDATE_MAIN_OFFICE	
Descrita por	(nombre gestor)
Prioridad	Media
Estabilidad	Alta
Descripción	Modificar el estado de los atributos de una sede concreta
Actor	Gestor Base de datos
Entrada	ID, nuevo estado atributos a modificar
Salida	Mensaje de confirmación o error
Origen	Interfaz usuario - Gestor del sistema
Destino	Sistema/BD de GREENGO
Necesita	BD
Acciones	Descritas en el diagrama de actividades
Precondición	Sede existente en la BD
Postcondición	Modificar datos de la sede
Efectos laterales	-





3.2.2.4 SHOW_MAIN_OFFICE	
Descrita por	(nombre gestor)
Prioridad	Media
Estabilidad	Alta
Descripción	Visualización del estado actual de los atributos de una sede
Actor	Gestor Base de datos
Entrada	ID
Salida	Estado actual atributos.
Origen	Interfaz usuario - Gestor del sistema
Destino	Interfaz usuario
Necesita	BD de GREENGO
Acciones	Descritas en el diagrama de actividades
Precondición	Participante existente en la BD y con el campo activo a "TRUE"
Postcondición	-
Efectos laterales	-



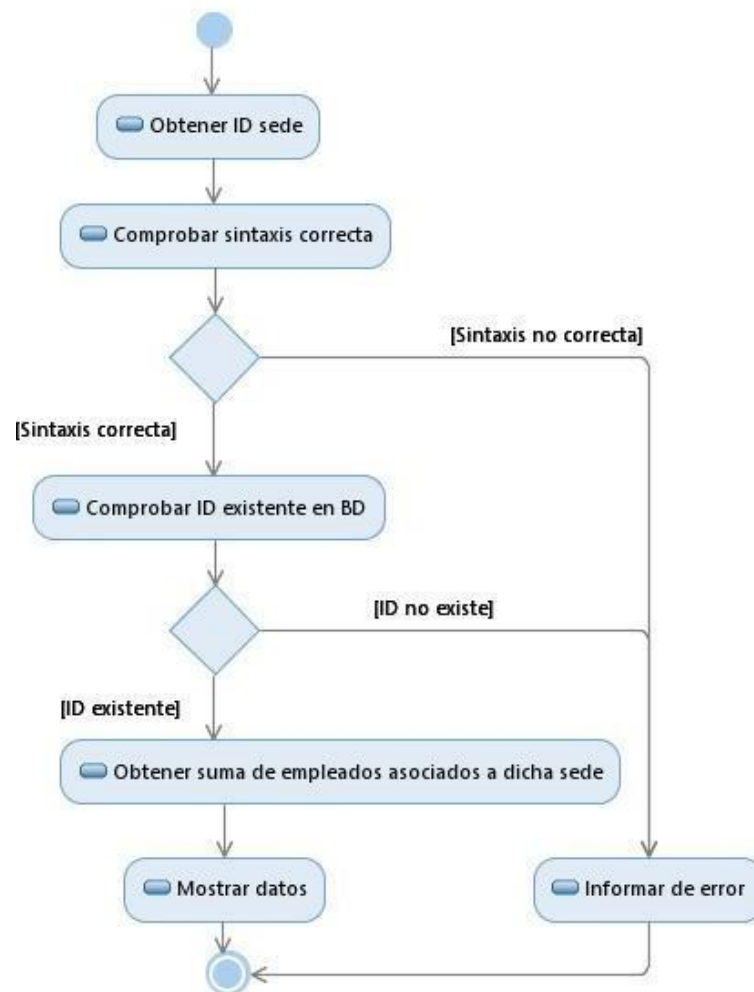


3.2.2.5 SHOWALL_MAIN _OFFICE	
Descrita por	(nombre gestor)
Prioridad	Media
Estabilidad	Alta
Descripción	Visualización del estado actual de los atributos de las sedes pertenecientes al listado de sedes con el atributo <i>activo</i> con estado actual 'TRUE'.
Actor	Gestor Base de datos
Entrada	-
Salida	Listado de sedes
Origen	Interfaz usuario - Gestor del sistema
Destino	Interfaz usuario
Necesita	BD de GREENGO
Acciones	Descritas en el diagrama de actividades
Precondición	-
Postcondición	-
Efectos laterales	-



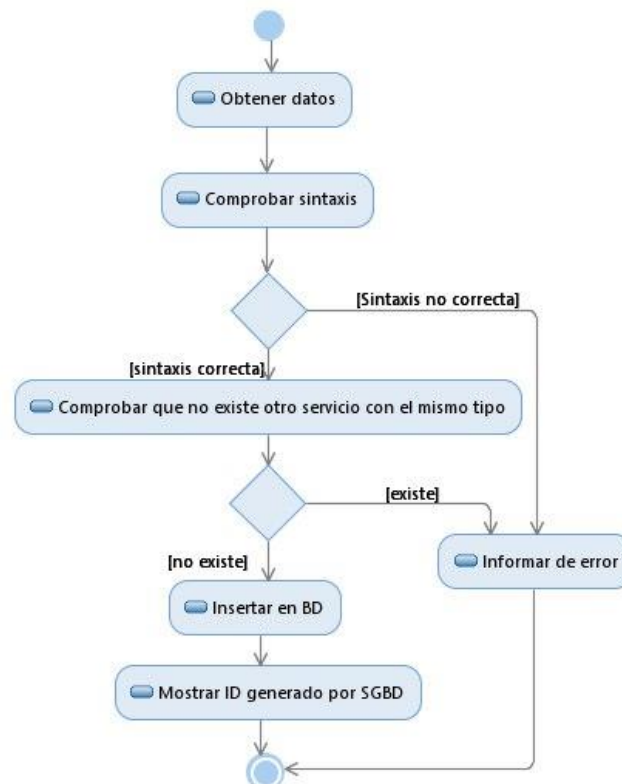
3.2.2.3 TOTAL_SALARY_ MAIN_OFFICE	
Descrita por	(nombre gestor)
Prioridad	Media.
Estabilidad	Alta
Descripción	Muestra el salario de la sede correspondiente, que se corresponde con la suma de los salarios de los empleados que trabajan en dicha sede.
Actor	Gestor Base de datos
Entrada	ID de la sede.
Salida	Salario total o error.
Origen	Interfaz usuario - Gestor del sistema
Destino	Sistema/BD de GREENGO
Necesita	BD
Acciones	Descritas en el diagrama de actividades
Precondición	Sede existente en la BD
Postcondición	Mostrar salario de la sede.
Efectos laterales	-





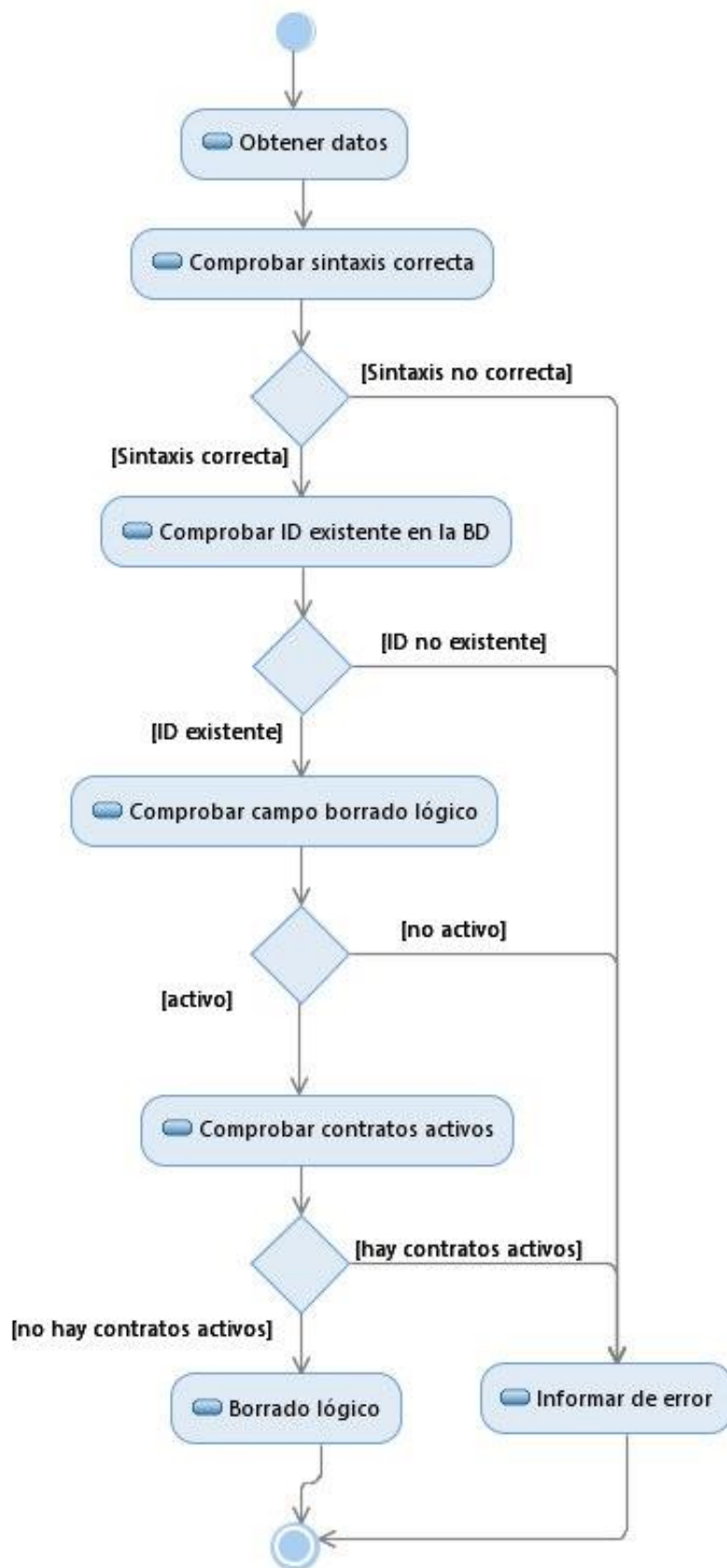
- Service

3.2.3.1 CREATE_SERVICE	
Descrita por	(nombre gestor)
Prioridad	Alta
Estabilidad	Alta
Descripción	Añadir un servicio con un estado concreto al sistema de GREENGO
Actor	Gestor Base de datos
Entrada	Insertar valores que satisfagan los atributos de la entidad servicio
Salida	Mensaje con ID o error
Origen	Interfaz usuario - Gestor del sistema
Destino	Sistema/BD de GREENGO
Necesita	BD de GREENGO
Acciones	Descritas en el diagrama de actividades
Precondición	No existe el tipo de servicio en la BD
Postcondición	Añadir servicio a la BD
Efectos laterales	-



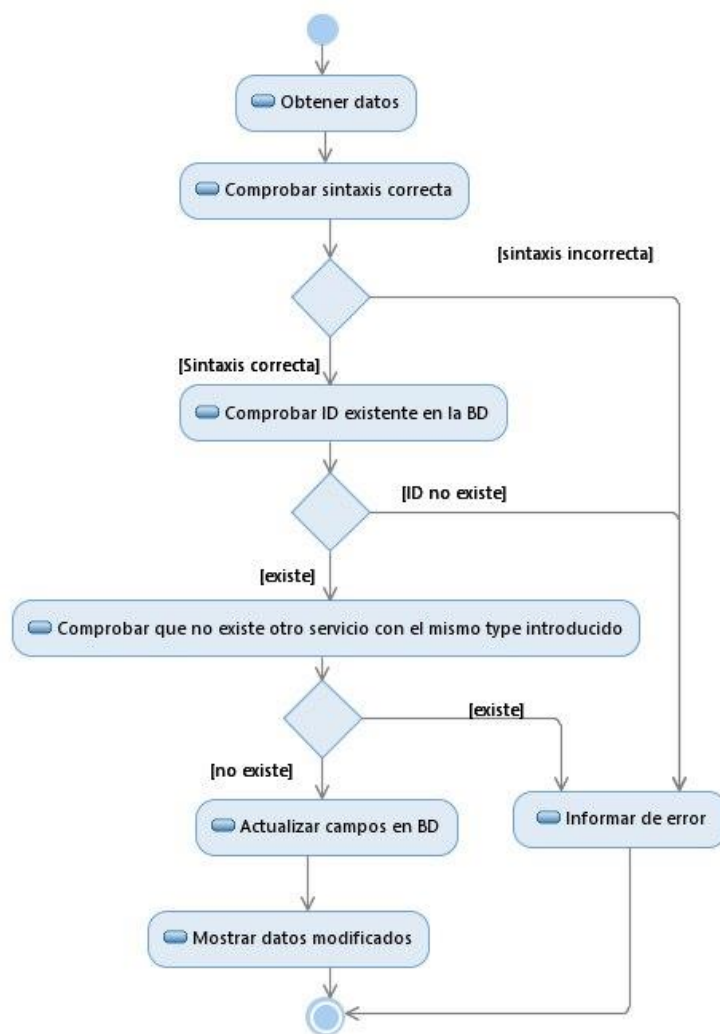
3.2.3.2 DROP_SERVICE	
Descrita por	(nombre gestor)
Prioridad	Alta
Estabilidad	Alta
Descripción	Inserta el valor 'FALSE' en el atributo activo de un servicio de la BD
Actor	Encargado/usuario de software
Entrada	ID del servicio que se quiere dar de baja
Salida	Mensaje de confirmación o error
Origen	Interfaz usuario - Gestor del sistema
Destino	Sistema/BD de GREENGO
Necesita	BD de GREENGO
Acciones	Descritas en el diagrama de actividades
Precondición	Existe un servicio con el ID insertado
Postcondición	Atributo <i>active</i> del servicio con valor "FALSE".
Efectos laterales	-



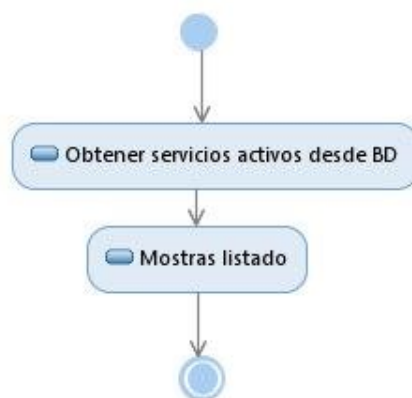


3.2.3.3 UPDATE_SERVICE	
Descrita por	(nombre gestor)
Prioridad	Media
Estabilidad	Alta
Descripción	Modificar el estado de los atributos de un servicio concreto
Actor	Gestor Base de datos
Entrada	ID, nuevo estado atributos a modificar
Salida	Mensaje de confirmación o error
Origen	Interfaz usuario - Gestor del sistema
Destino	Sistema/BD de GREENGO
Necesita	BD de GREENGO
Acciones	Descritas en el diagrama de actividades
Precondición	Servicio existente en la BD
Postcondición	Modificar datos del servicio
Efectos laterales	-



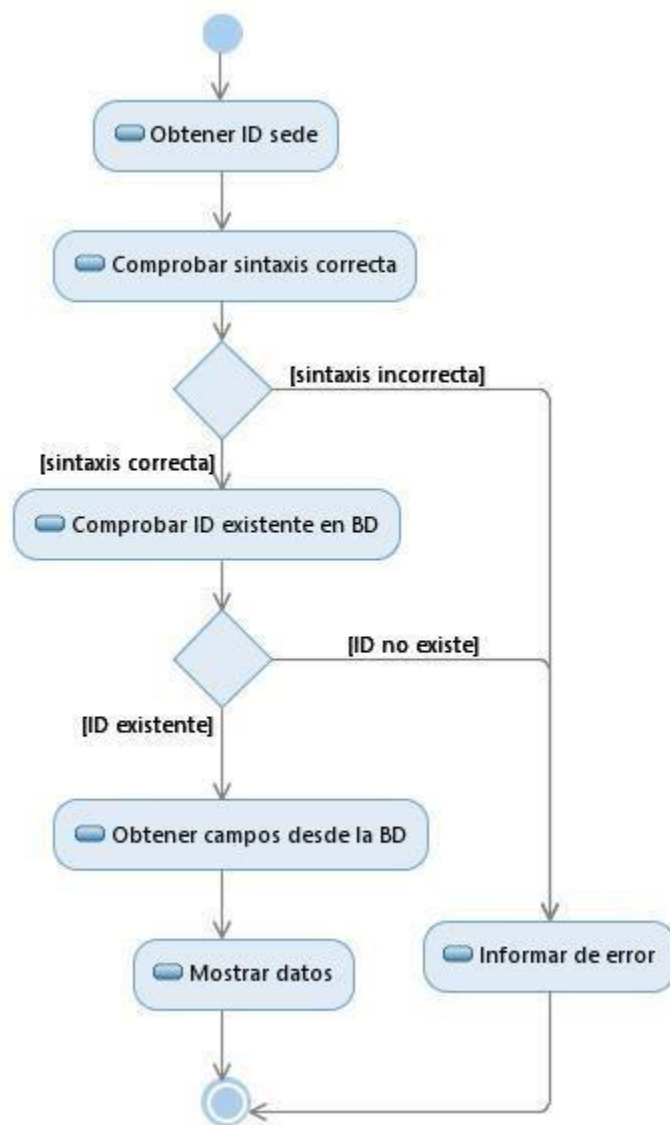


3.2.3.4 SHOWALL_SERVICE	
Descrita por	(nombre gestor)
Prioridad	Media
Estabilidad	Alta
Descripción	Visualización del estado actual de los atributos de los servicios pertenecientes al listado de servicios con el atributo activo con estado actual 'TRUE'.
Actor	Gestor Base de datos
Entrada	-
Salida	Listado de los servicios
Origen	Interfaz usuario - Gestor del sistema
Destino	Interfaz usuario
Necesita	BD de GREENGO
Acciones	Descritas en el diagrama de actividades
Precondición	-
Postcondición	-
Efectos laterales	-



3.2.3.5 SHOW_SERVICE	
Descrita por	(nombre gestor)
Prioridad	Media
Estabilidad	Alta
Descripción	Visualización del estado actual de los atributos de un servicio.
Actor	Gestor Base de datos
Entrada	ID del servicio
Salida	Datos del servicio
Origen	Interfaz usuario - Gestor del sistema
Destino	Interfaz usuario
Necesita	BD de GREENGO
Acciones	Descritas en el diagrama de actividades
Precondición	Servicio existente en la BD y con el campo activo a "TRUE"
Postcondición	-
Efectos laterales	-





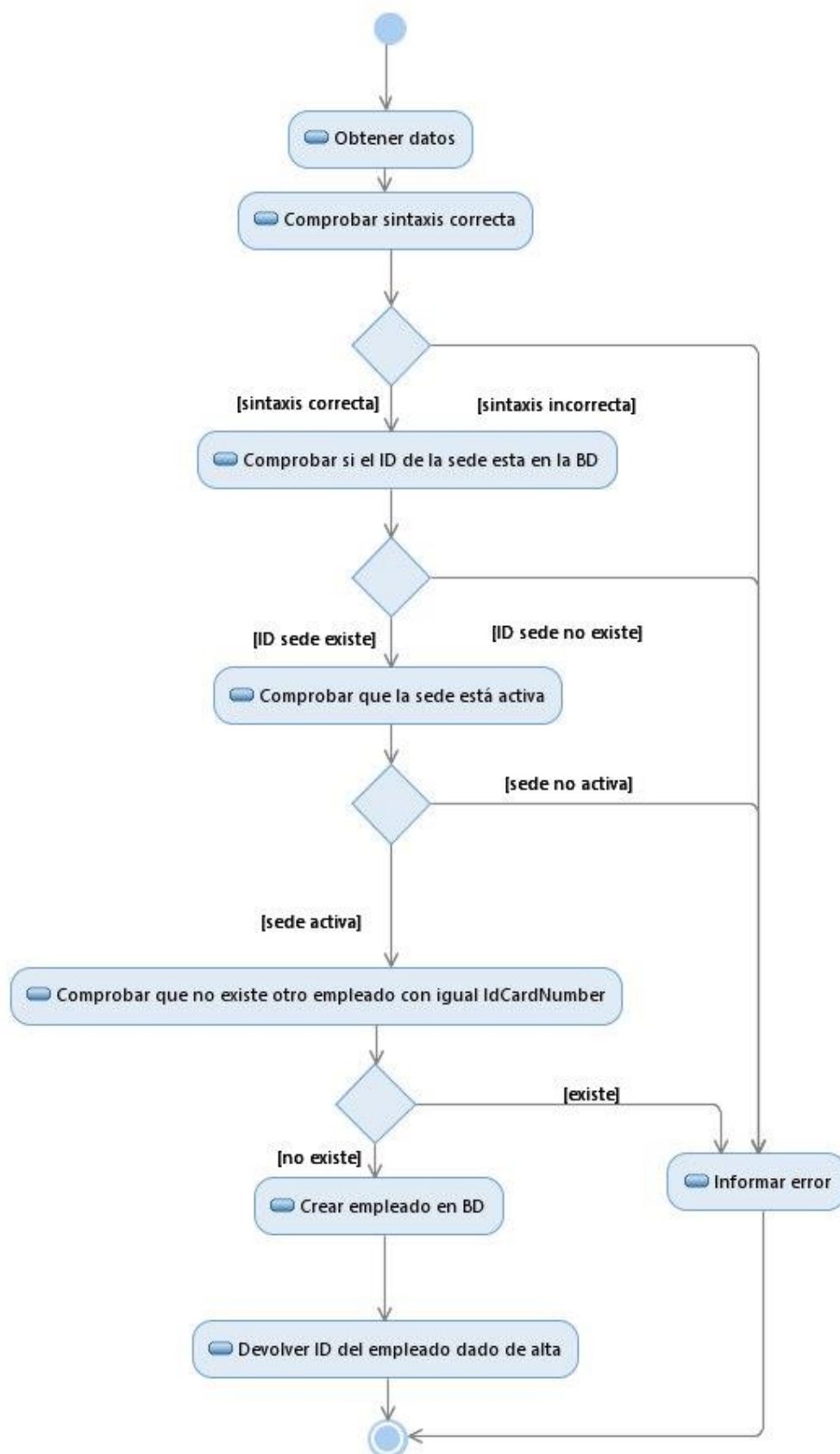
3.2.3.6 SHOW_SERVICES_ BY_LEVEL	
Descrita por	(nombre gestor)
Prioridad	Media
Estabilidad	Alta
Descripción	Visualización de los servicios activos que ofrecen un nivel determinado.
Actor	Gestor Base de datos
Entrada	Nivel del servicio.
Salida	Servicios por nivel.
Origen	Interfaz usuario - Gestor del sistema
Destino	Interfaz usuario
Necesita	BD de GREENGO
Acciones	Descritas en el diagrama de actividades
Precondición	Servicio existente en la BD con el campo activo a "TRUE" y atributo <i>Service_Level</i> estrictamente mayor que cero.
Postcondición	-
Efectos laterales	-



- **Employee**

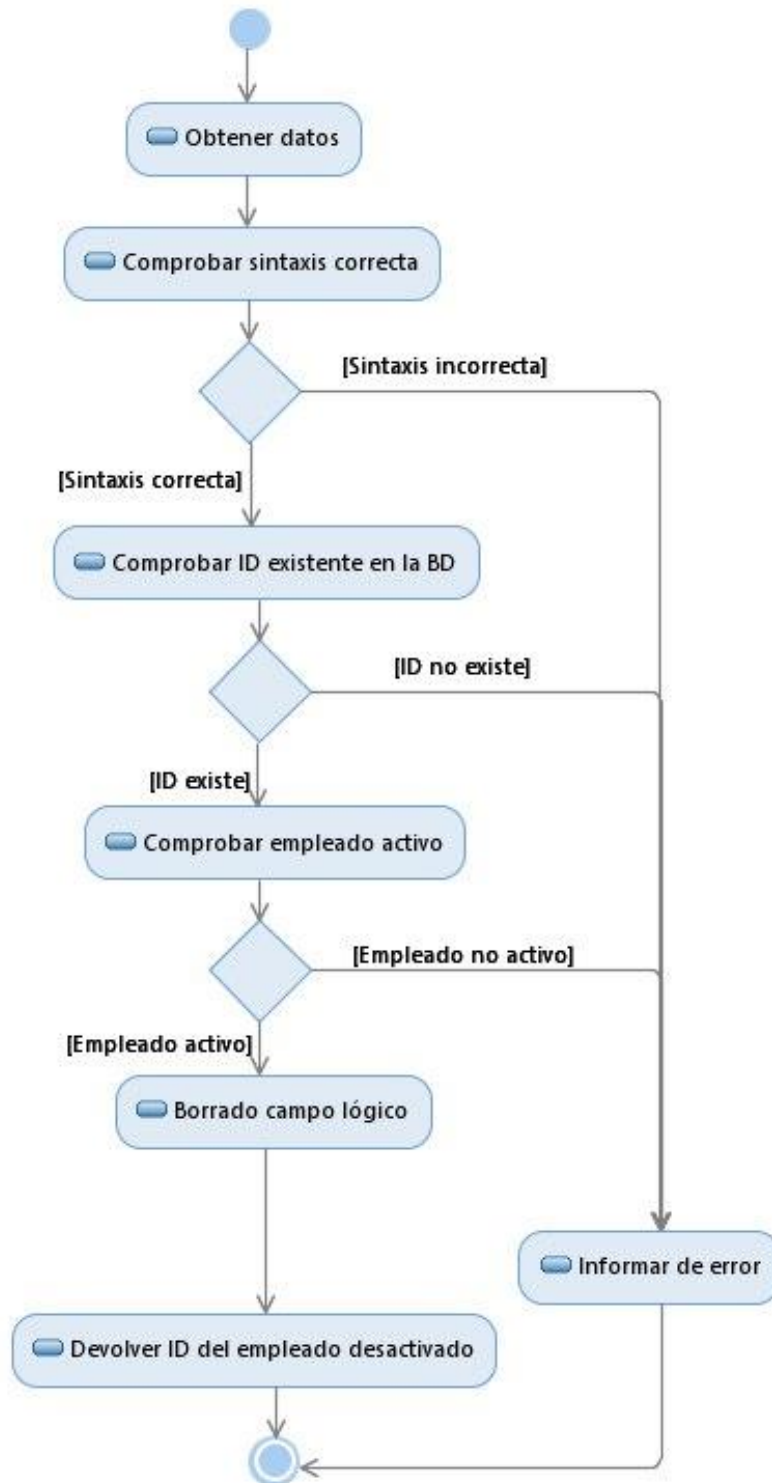
3.2.4.1 CREATE_EMPLOYEE	
Descrita por	(nombre gestor)
Prioridad	Alta
Estabilidad	Alta
Descripción	Añade un empleado con unas determinadas características asociado a una sede al sistema de GREENGO.
Actor	Encargado/usuario de software
Entrada	Insertar valores que satisfagan los atributos de la entidad empleado y el identificador de la sede a la que está asociado.
Salida	Mensaje de confirmación (ID Empleado) o error
Origen	Interfaz usuario - Gestor del sistema
Destino	Sistema/BD de GREENGO
Necesita	BD de GREENGO
Acciones	Descritas en el diagrama de actividades
Precondición	No hay mismo ID (ya existente) en la BD, no existe ningún empleado con el mismo DNI que el del empleado a dar de alta.
Postcondición	Añadir empleado a la BD
Efectos laterales	-





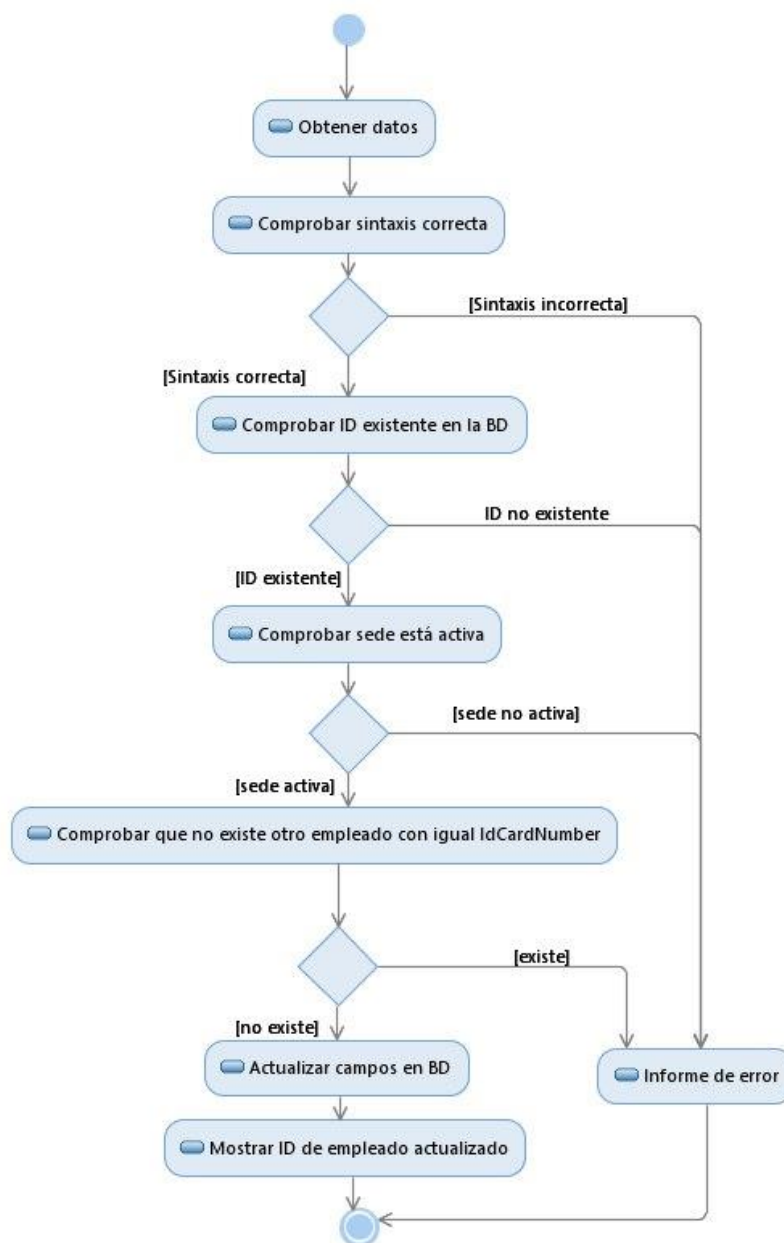
3.2.4.2 DROP_EMPLOYEE	
Descrita por	(nombre gestor)
Prioridad	Alta
Estabilidad	Alta
Descripción	Inserta el valor 'FALSE' en el atributo <i>activo</i> de un empleado de la BD
Actor	Encargado/usuario de software
Entrada	ID del empleado a eliminar.
Salida	Mensaje de confirmación (Mostrar ID borrado) o error
Origen	Interfaz usuario - Gestor del sistema
Destino	Sistema/BD de GREENGO
Necesita	BD de GREENGO
Acciones	Descritas en el diagrama de actividades
Precondición	Existe el empleado a eliminar en la BD.
Postcondición	Atributo <i>activo</i> con valor 'FALSE'
Efectos laterales	-





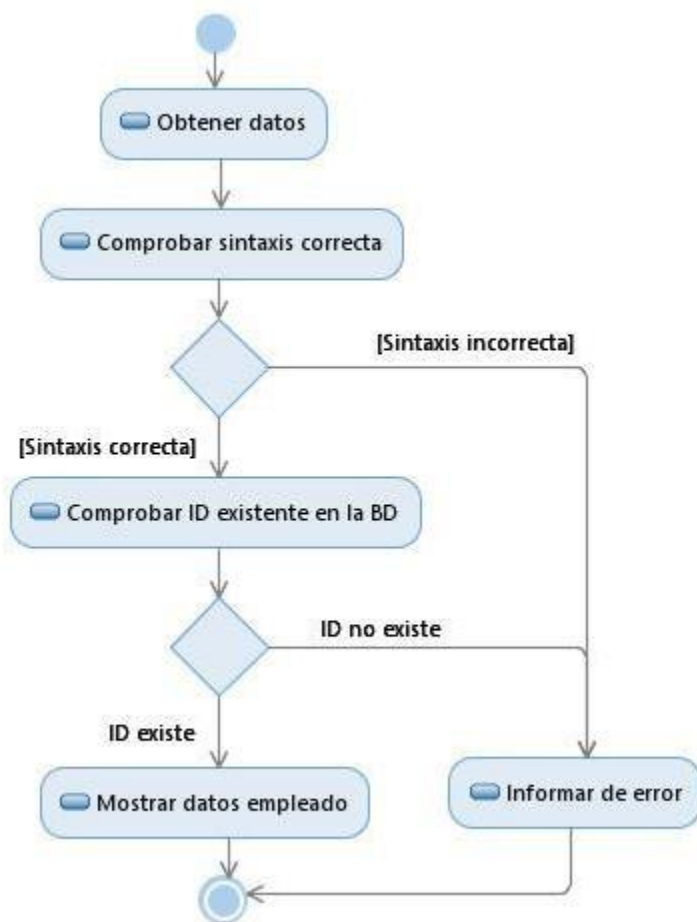
3.2.4.3 UPDATE_EMPLOYEE	
Descrita por	(nombre gestor)
Prioridad	Media
Estabilidad	Alta
Descripción	Modificar el estado de los atributos de un empleado concreto.
Actor	Gestor Base de datos
Entrada	ID, estado de los atributos a modificar.
Salida	Mensaje de confirmación o error
Origen	Interfaz usuario - Gestor del sistema
Destino	Sistema/BD de GREENGO
Necesita	BD
Acciones	Descritas en el diagrama de actividades
Precondición	Empleado existente en la BD
Postcondición	Modificar datos del empleado.
Efectos laterales	-





3.2.4.4 SHOW_EMPLOYEE	
Descrita por	(nombre gestor)
Prioridad	Alta
Estabilidad	Alta
Descripción	Visualización del estado actual de los atributos de un empleado.
Actor	Gestor Base de datos
Entrada	Insertar ID
Salida	Datos del empleado.
Origen	Interfaz usuario - Gestor del sistema
Destino	Interfaz usuario
Necesita	BD
Acciones	Descritas en el diagrama de actividades
Precondición	ID existente en la BD.
Postcondición	-
Efectos laterales	-





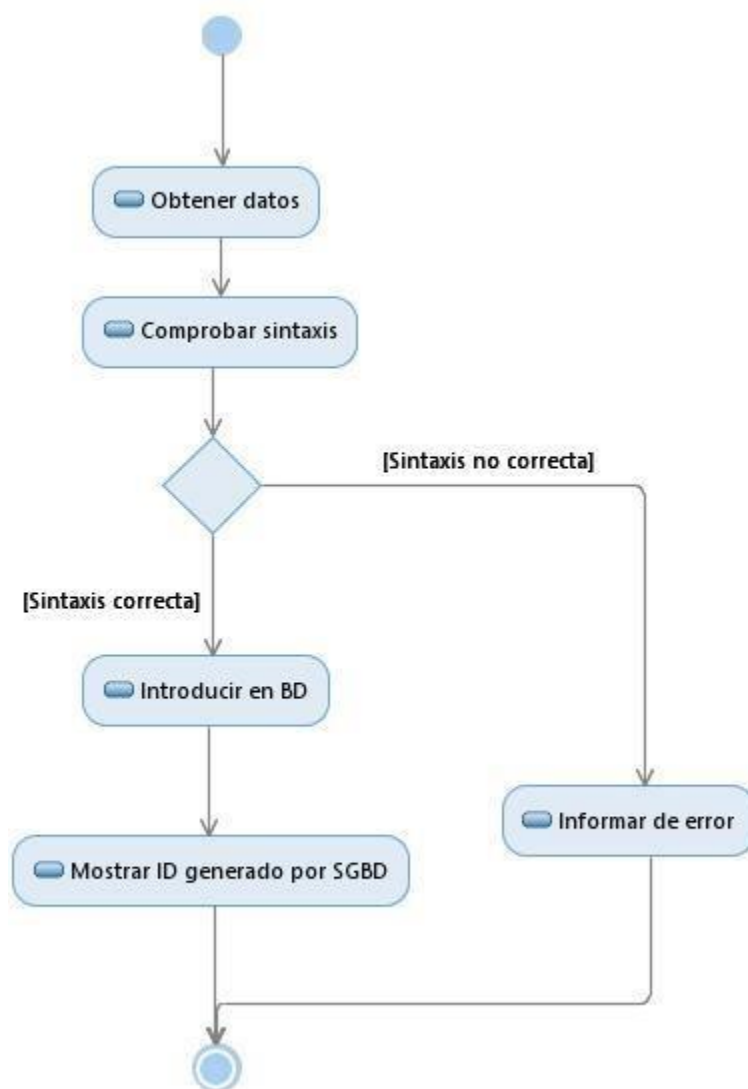
3.2.4.5 SHOWALL_EMPLOYEE	
Descrita por	(nombre gestor)
Prioridad	Media
Estabilidad	Alta
Descripción	Visualización del estado actual de los atributos de los empleados pertenecientes al listado de empleados con el atributo <i>activo</i> con estado actual 'TRUE'.
Actor	Gestor Base de datos
Entrada	-
Salida	Listado de empleados
Origen	Interfaz usuario - Gestor del sistema
Destino	Interfaz usuario
Necesita	BD
Acciones	Descritas en el diagrama de actividades
Precondición	Empleados existentes en la BD
Postcondición	-
Efectos laterales	-



- City

3.2.5.1 CREATE_CITY	
Descrita por	(nombre gestor)
Prioridad	Alta
Estabilidad	Alta
Descripción	Añadir una ciudad con un estado concreto al sistema de GREENGO
Actor	Encargado/usuario de software
Entrada	Insertar valores que satisfagan los atributos de la entidad ciudad
Salida	Mensaje con ID o de error
Origen	Interfaz usuario - Gestor del sistema
Destino	Sistema/BD de GREENGO
Necesita	BD de GREENGO
Acciones	Descritas en el diagrama de actividades
Precondición	No existe la ciudad en el sistema que posea el mismo ID.
Postcondición	Añadir ciudad con valores en sus atributos iguales a los datos introducidos por entrada a la BD
Efectos laterales	-

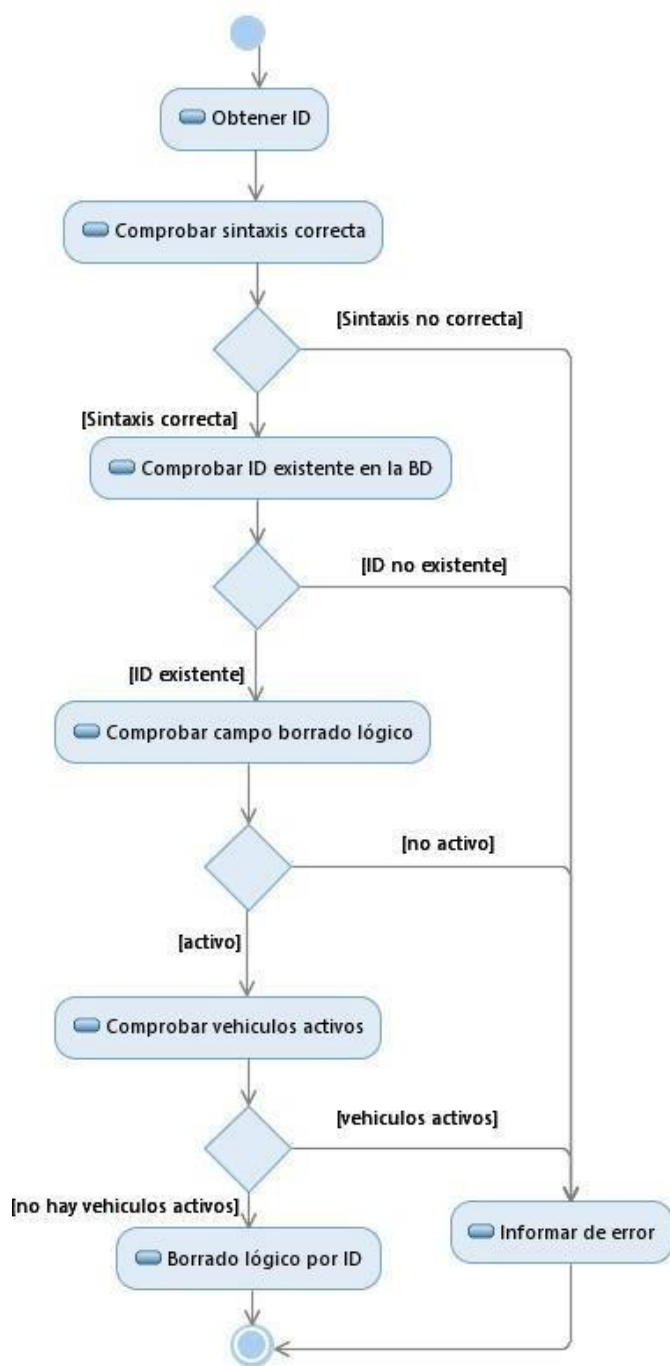




3.2.5.2 DROP_CITY

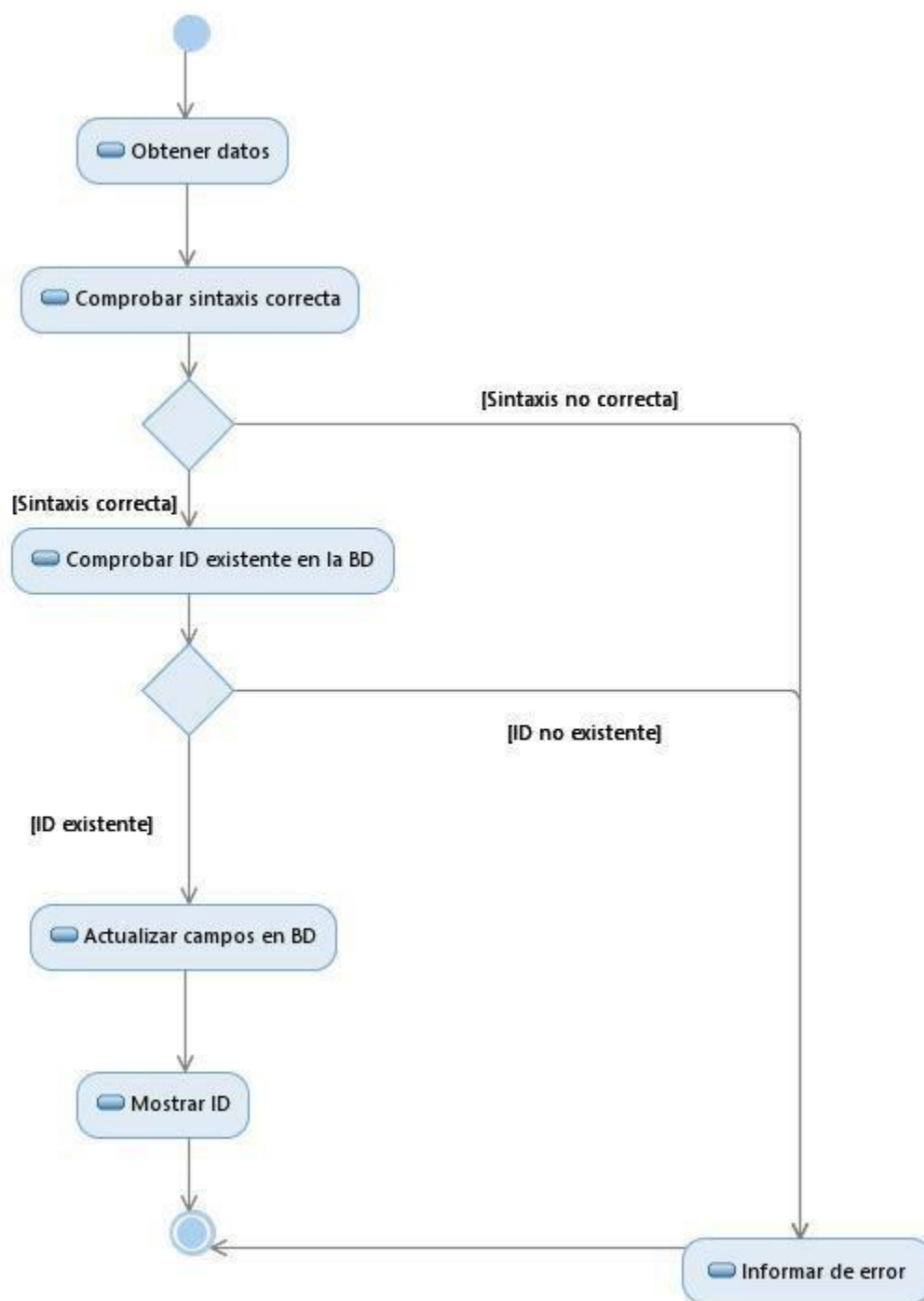
Descrita por	(nombre gestor)
Prioridad	Alta
Estabilidad	Alta
Descripción	Inserta el valor 'FALSE' en el atributo <i>activo</i> de una ciudad de la BD
Actor	Encargado/usuario de software
Entrada	ID de la ciudad que se quiera dar de baja.
Salida	Mostrar id o error
Origen	Interfaz usuario - Gestor del sistema
Destino	Sistema/BD de GREENGO
Necesita	BD de GREENGO
Acciones	Descritas en el diagrama de actividades
Precondición	Existe una ciudad con el ID insertado con atributo <i>activo</i> con valor 'TRUE' y sin vehículos activos relacionados con ella.
Postcondición	Atributo <i>activo</i> con valor 'FALSE'
Efectos laterales	-





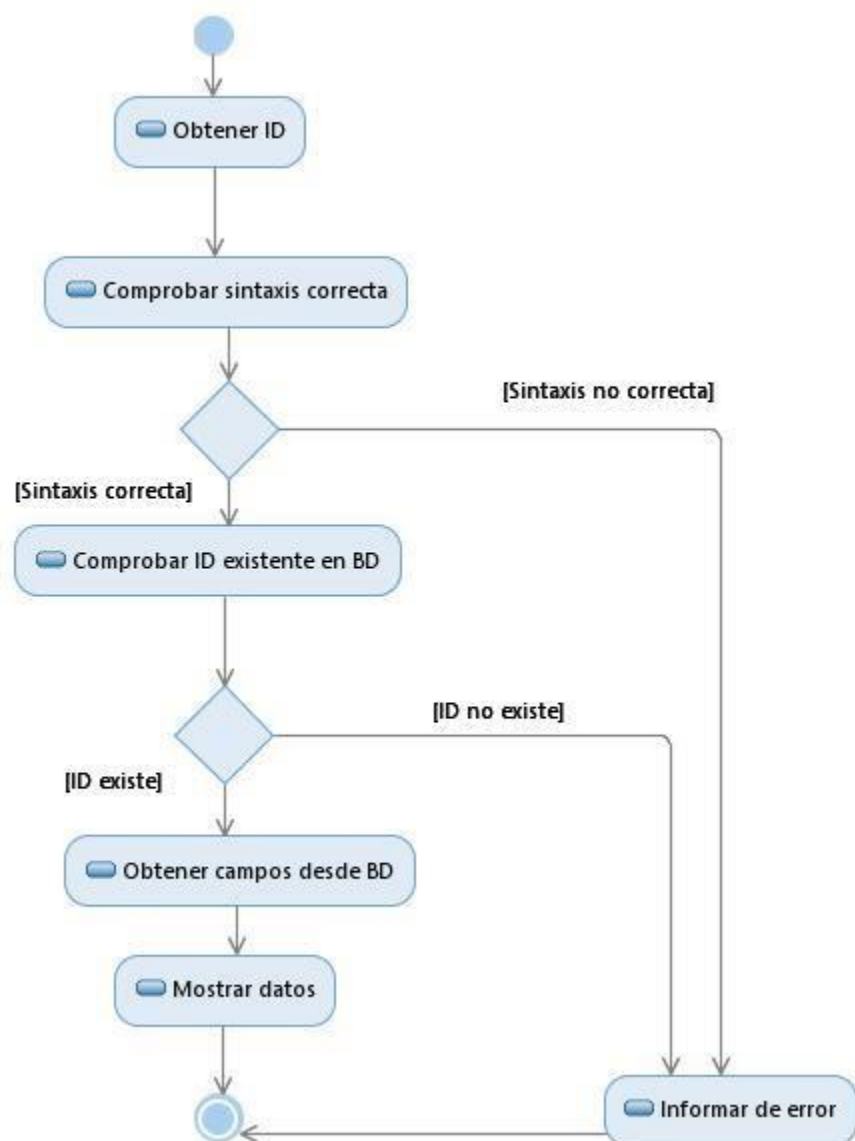
3.2.5.3 UPDATE_ CITY	
Descrita por	(nombre gestor)
Prioridad	Media
Estabilidad	Alta
Descripción	Modificar el estado de los atributos de una ciudad concreta
Actor	Gestor Base de datos
Entrada	ID, nuevo estado atributos a modificar
Salida	Mensaje de confirmación o error
Origen	Interfaz usuario - Gestor del sistema
Destino	Sistema/BD de GREENGO
Necesita	BD de GREENGO
Acciones	Descritas en el diagrama de actividades
Precondición	Vehículo existente en la BD
Postcondición	Modificar datos del vehículo
Efectos laterales	-



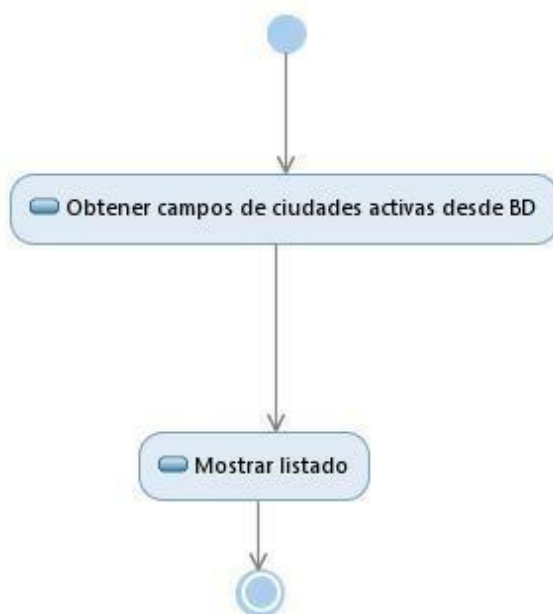


3.2.5.4 SHOW_CITY	
Descrita por	(nombre gestor)
Prioridad	Media
Estabilidad	Alta
Descripción	Visualización del estado actual de los atributos de una determinada ciudad.
Actor	Gestor Base de datos
Entrada	ID
Salida	Estado actual atributos o error.
Origen	Interfaz usuario - Gestor del sistema
Destino	Interfaz gráfica de usuario
Necesita	BD de GREENGO
Acciones	Descritas en el diagrama de actividades
Precondición	Ciudad existente en la BD.
Postcondición	-
Efectos laterales	-





3.2.5.5 SHOWALL_CITY	
Descrita por	(nombre gestor)
Prioridad	Media
Estabilidad	Alta
Descripción	Visualización del estado actual de los atributos de las ciudades pertenecientes al listado de ciudades con el atributo <i>activo</i> con estado actual 'TRUE'.
Actor	Gestor Base de datos
Entrada	-
Salida	Listado de ciudades por la interfaz de usuario
Origen	Interfaz usuario - Gestor del sistema
Destino	Interfaz gráfica de usuario
Necesita	BD de GREENGO
Acciones	Descritas en el diagrama de actividades
Precondición	-
Postcondición	-
Efectos laterales	-



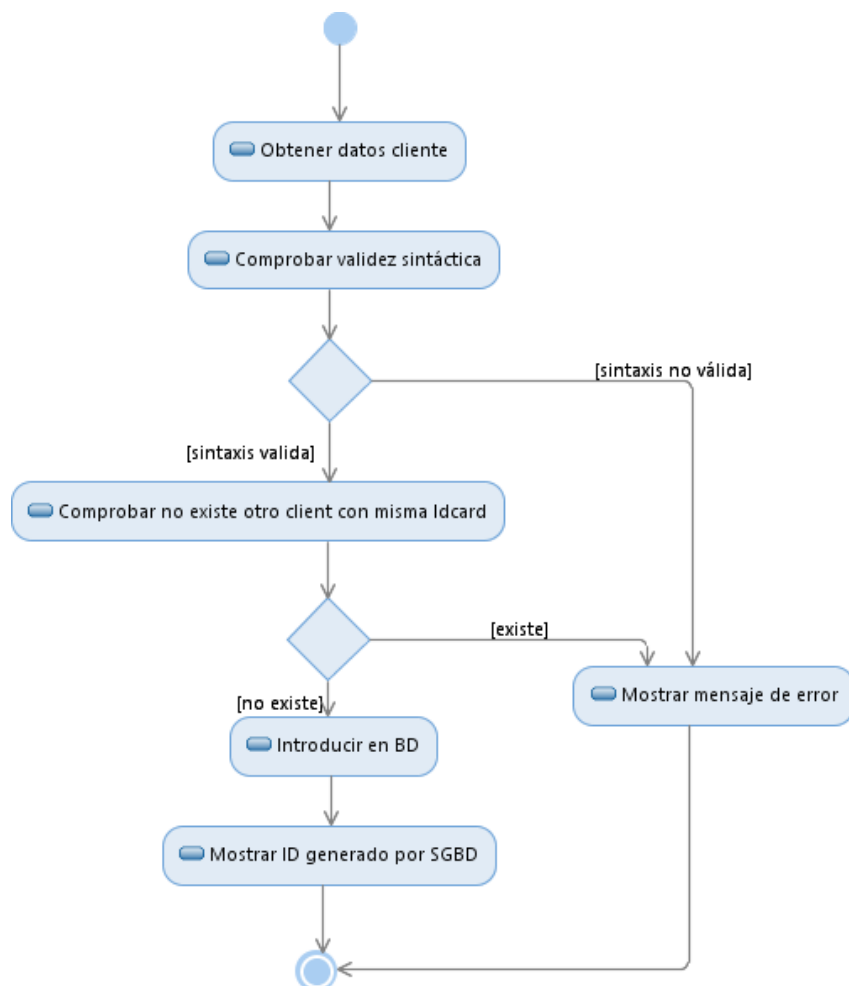
3.2.5.6 SHOW_CLIENTS_BY_CITY	
Descrita por	(nombre gestor)
Prioridad	Media
Estabilidad	Alta
Descripción	Visualización del estado actual de los atributos de los clientes pertenecientes al listado de clientes con el atributo <i>activo</i> con estado actual 'TRUE' y que hayan alquilado un vehículo determinado en una ciudad dada.
Actor	Gestor Base de datos
Entrada	ID ciudad
Salida	Listado de clientes por la interfaz de usuario
Origen	Interfaz usuario - Gestor del sistema
Destino	Interfaz gráfica de usuario
Necesita	BD de GREENGO
Acciones	Descritas en el diagrama de actividades
Precondición	Ciudad existente en la BD y con el campo activo a "TRUE"
Postcondición	-
Efectos laterales	-



- Client

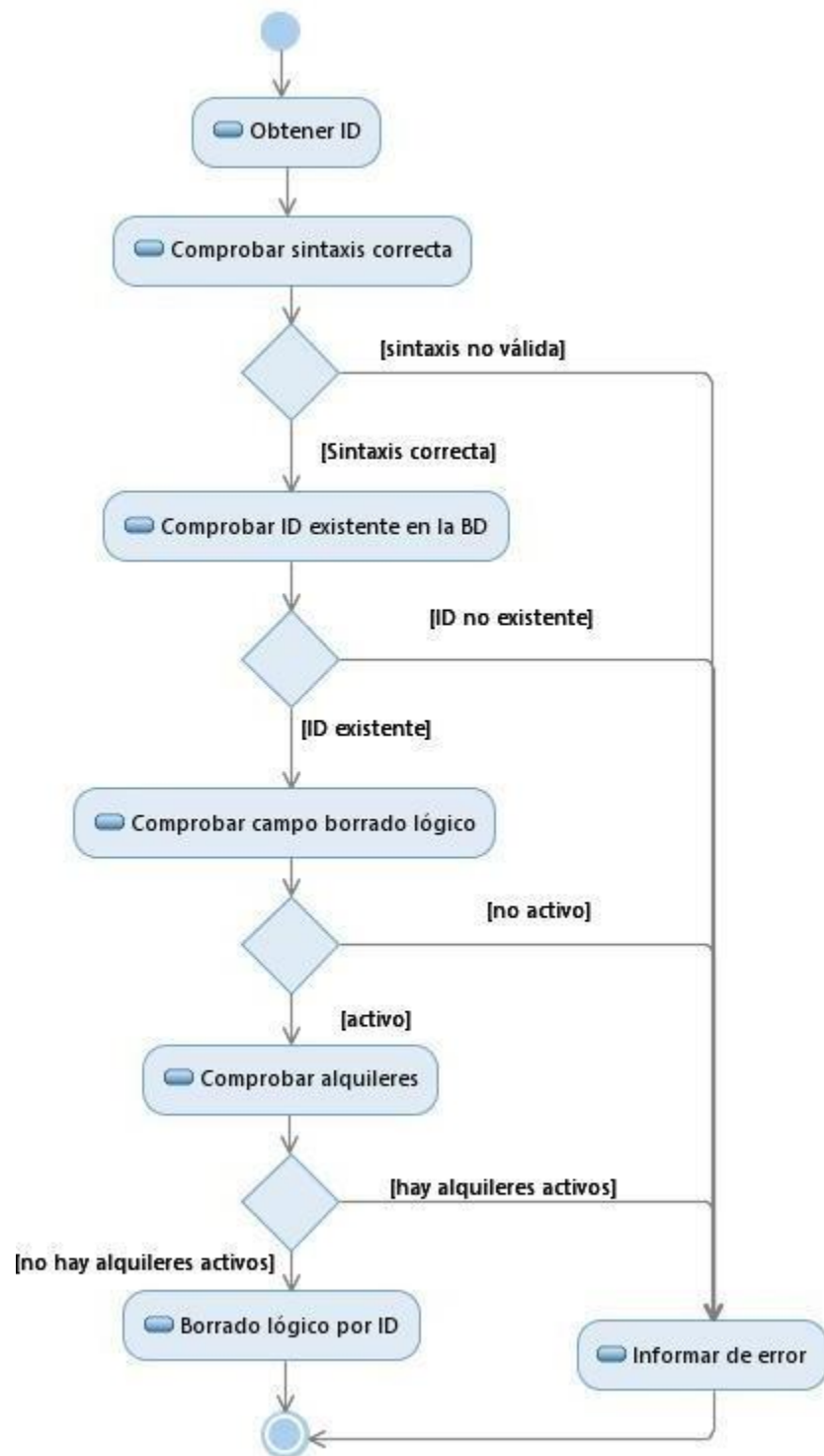
3.2.6.1 CREATE_CLIENT	
Descrita por	(nombre gestor)
Prioridad	Alta
Estabilidad	Alta
Descripción	Añadir un cliente con un estado concreto al sistema de GREENGO
Actor	Gestor Base de Datos
Entrada	Insertar valores que satisfagan los atributos de la entidad servicio
Salida	Mensaje con ID o error
Origen	Interfaz usuario - Gestor del sistema
Destino	Sistema/BD de GREENGO
Necesita	BD de GREENGO
Acciones	Descritas en el diagrama de actividades
Precondición	No existe el cliente en la BD con el mismo id_card_number
Postcondición	Añadir cliente a la BD
Efectos laterales	-





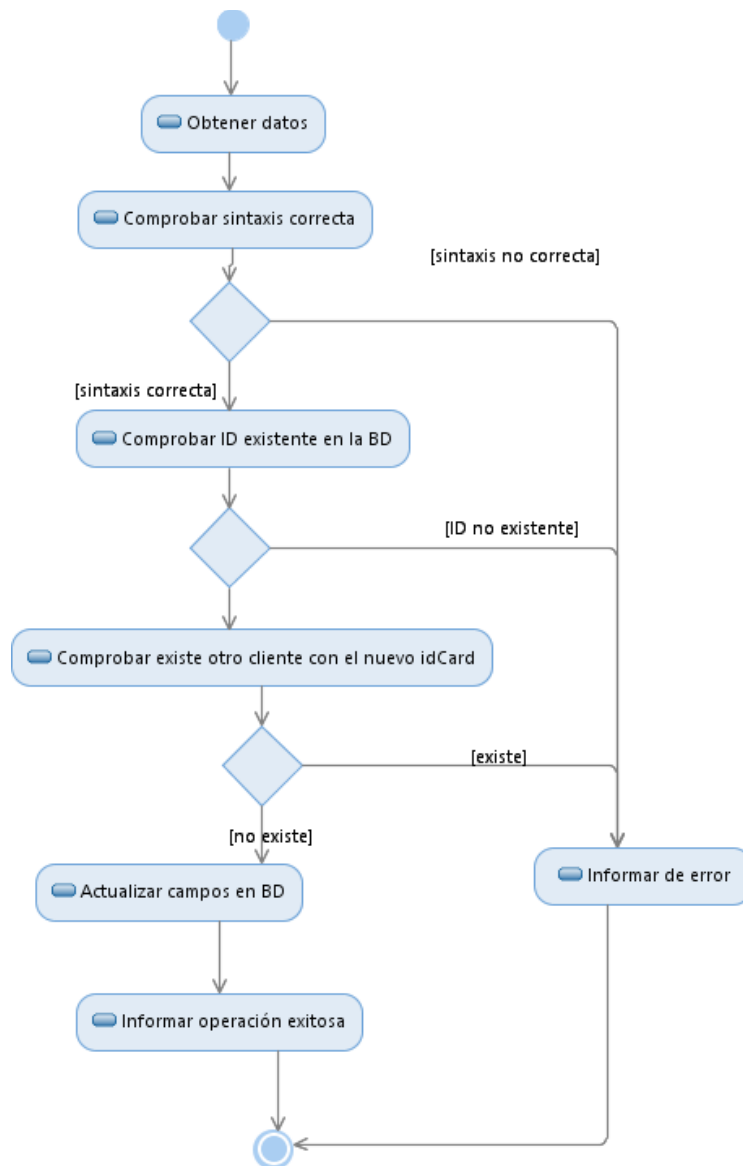
3.2.6.2 DROP_CLIENT	
Descrita por	(nombre gestor)
Prioridad	Alta
Estabilidad	Alta
Descripción	Inserta el valor 'FALSE' en el atributo <i>activo</i> de un cliente de la BD
Actor	Encargado/usuario de software
Entrada	ID del cliente que se da de baja.
Salida	Mostrar id o error
Origen	Interfaz usuario - Gestor del sistema
Destino	Sistema/BD de GREENGO
Necesita	BD de GREENGO
Acciones	Descritas en el diagrama de actividades
Precondición	Existe un cliente con el ID insertado y atributo <i>activo</i> con valor 'TRUE'
Postcondición	Atributo <i>activo</i> con valor 'FALSE'
Efectos laterales	-





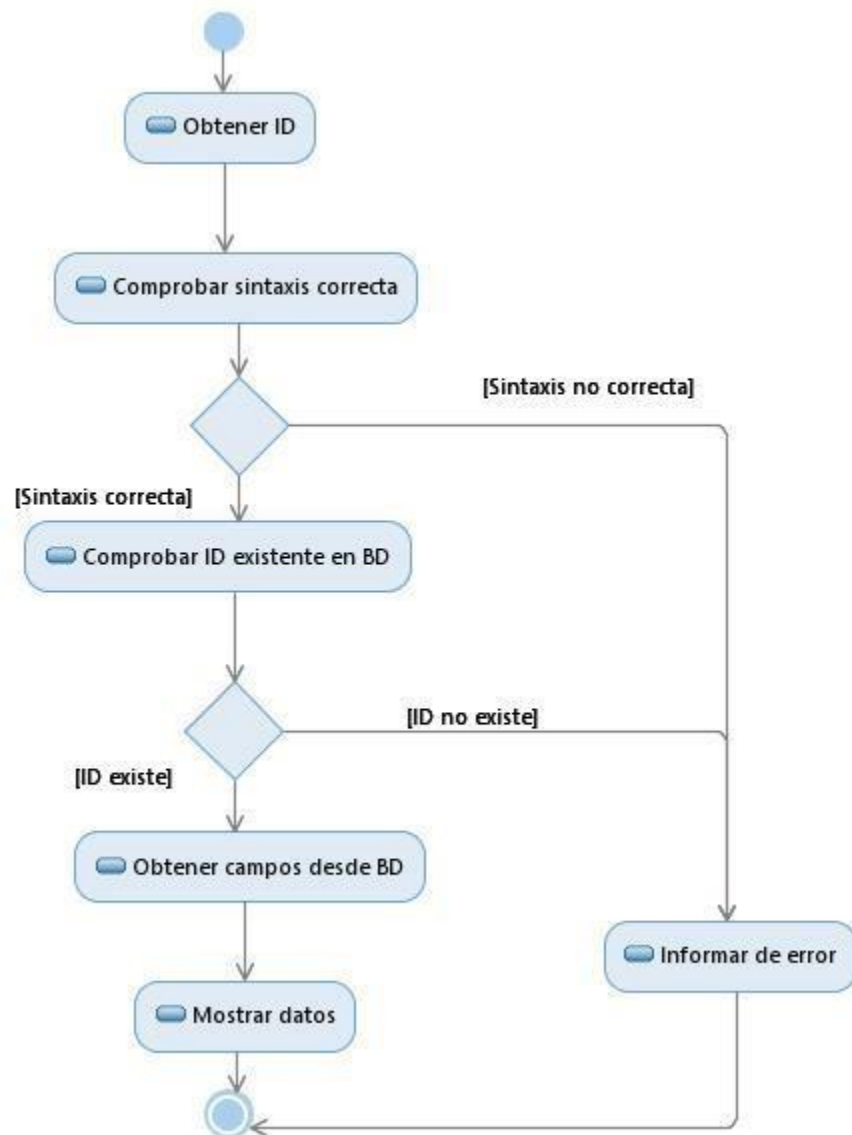
3.2.6.3 UPDATE_CLIENT	
Descrita por	(nombre gestor)
Prioridad	Media
Estabilidad	Alta
Descripción	Modificar el estado de los atributos de un cliente
Actor	Gestor Base de datos
Entrada	ID, nuevo estado de los atributos a modificar
Salida	Mostrar id o error
Origen	Interfaz usuario - Gestor del sistema
Destino	Sistema/BD de GREENGO
Necesita	BD de GREENGO
Acciones	Descritas en el diagrama de actividades
Precondición	Cliente existente en la BD
Postcondición	Datos de cliente modificados
Efectos laterales	-





3.2.6.4 SHOW_CLIENT	
Descrita por	(nombre gestor)
Prioridad	Media
Estabilidad	Alta
Descripción	Visualización del estado actual de los atributos de una determinada cliente.
Actor	Gestor Base de datos
Entrada	ID
Salida	Estado actual atributos o error.
Origen	Interfaz usuario - Gestor del sistema
Destino	Interfaz gráfica de usuario
Necesita	BD de GREENGO
Acciones	Descritas en el diagrama de actividades
Precondición	Cliente existente en la BD.
Postcondición	-
Efectos laterales	-





3.2.6.5 SHOWALL_CLIENT	
Descrita por	(nombre gestor)
Prioridad	Media
Estabilidad	Alta
Descripción	Visualización del estado actual de los atributos de los clientes pertenecientes al listado de clientes con el atributo <i>activo</i> con estado actual 'TRUE'.
Actor	Gestor Base de datos
Entrada	-
Salida	Listado de clientes por la interfaz de usuario
Origen	Interfaz usuario - Gestor del sistema
Destino	Interfaz gráfica de usuario
Necesita	BD de GREENGO
Acciones	Descritas en el diagrama de actividades
Precondición	-
Postcondición	-
Efectos laterales	-



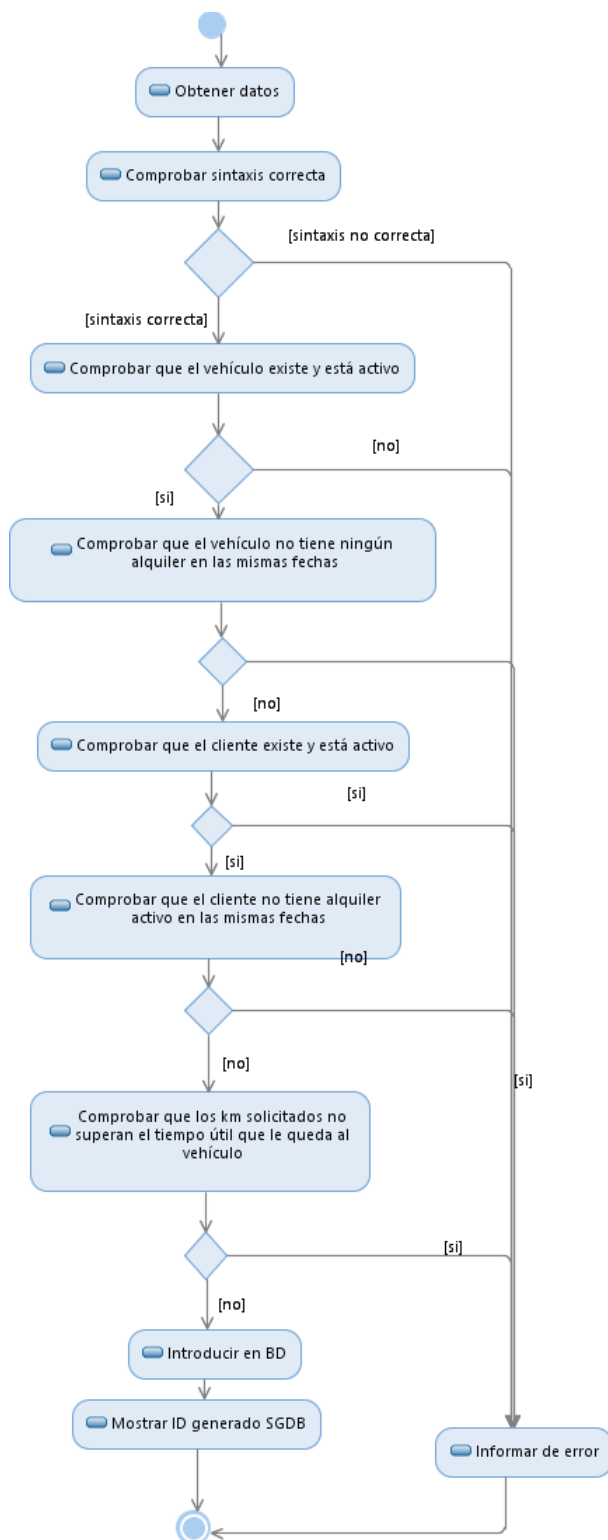
3.2.6.6 SHOW_CLIENTS_WITH_ MORE_THAN_N_RENTALS	
Descrita por	(nombre gestor)
Prioridad	Media
Estabilidad	Alta
Descripción	Visualización de los clientes que han realizado más de N alquileres.
Actor	Gestor Base de datos.
Entrada	N alquileres a consultar.
Salida	Listado de clientes por la interfaz de usuario.
Origen	Interfaz usuario - Gestor del sistema.
Destino	Interfaz gráfica de usuario.
Necesita	BD de GREENGO.
Acciones	Descritas en el diagrama de actividades.
Precondición	El número de alquileres es mayor o igual que 0.
Postcondición	-
Efectos laterales	-



- Rental

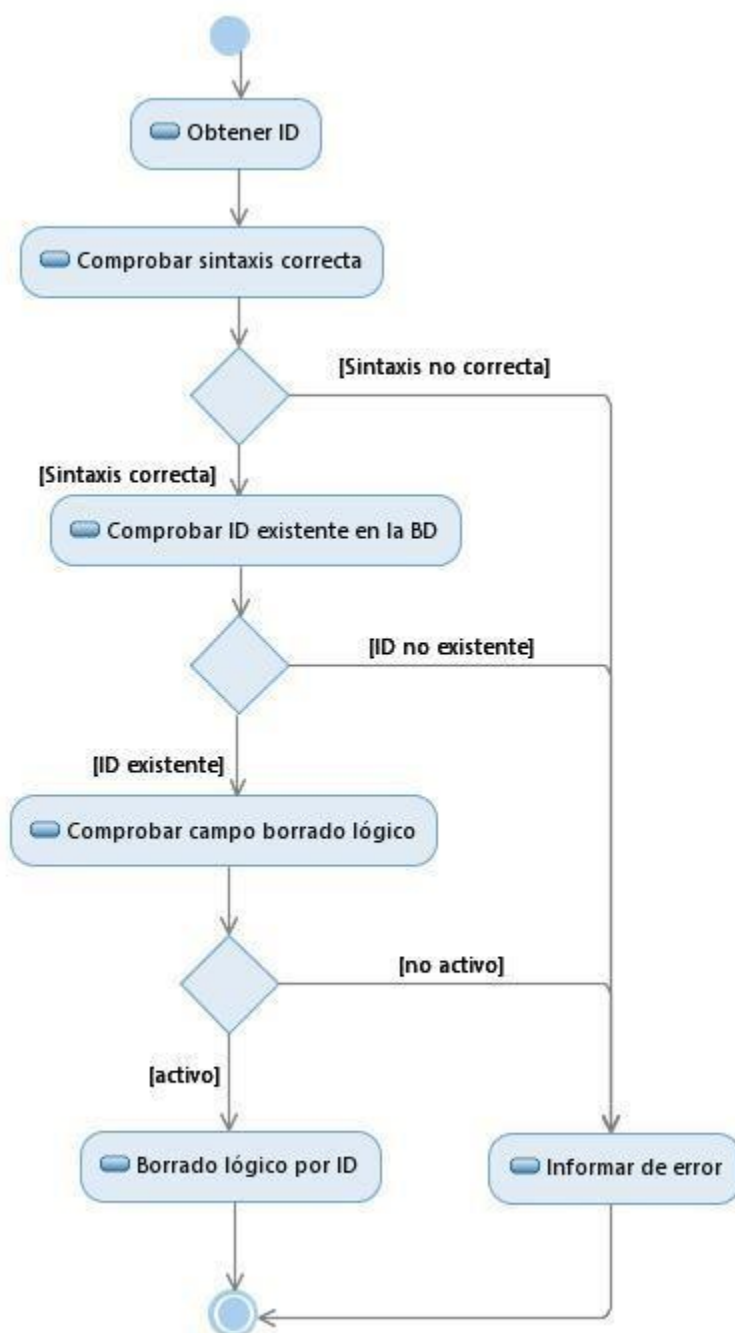
3.2.7.1 CREATE_RENTAL	
Descrita por	(nombre gestor)
Prioridad	Alta
Estabilidad	Alta
Descripción	Añade un alquiler con un estado concreto al sistema de GREENGO
Actor	Gestor Base de datos
Entrada	ID del vehículo, ID del cliente, kilometraje a recorrer, fecha de inicio y fecha de vencimiento.
Salida	Mensaje con ID o error.
Origen	Interfaz usuario - Gestor del sistema
Destino	Interfaz gráfica de usuario
Necesita	BD de GREENGO
Acciones	Descritas en el diagrama de actividades
Precondición	Cliente existente en la BD con el campo activo a "TRUE", vehículo existente en la BD con el campo activo a "TRUE", kilometraje estrictamente mayor que cero y fecha de inicio estrictamente menor que fecha de vencimiento
Postcondición	Añadir alquiler con valores en sus atributos a los datos introducidos por entrada a la BD.
Efectos laterales	-





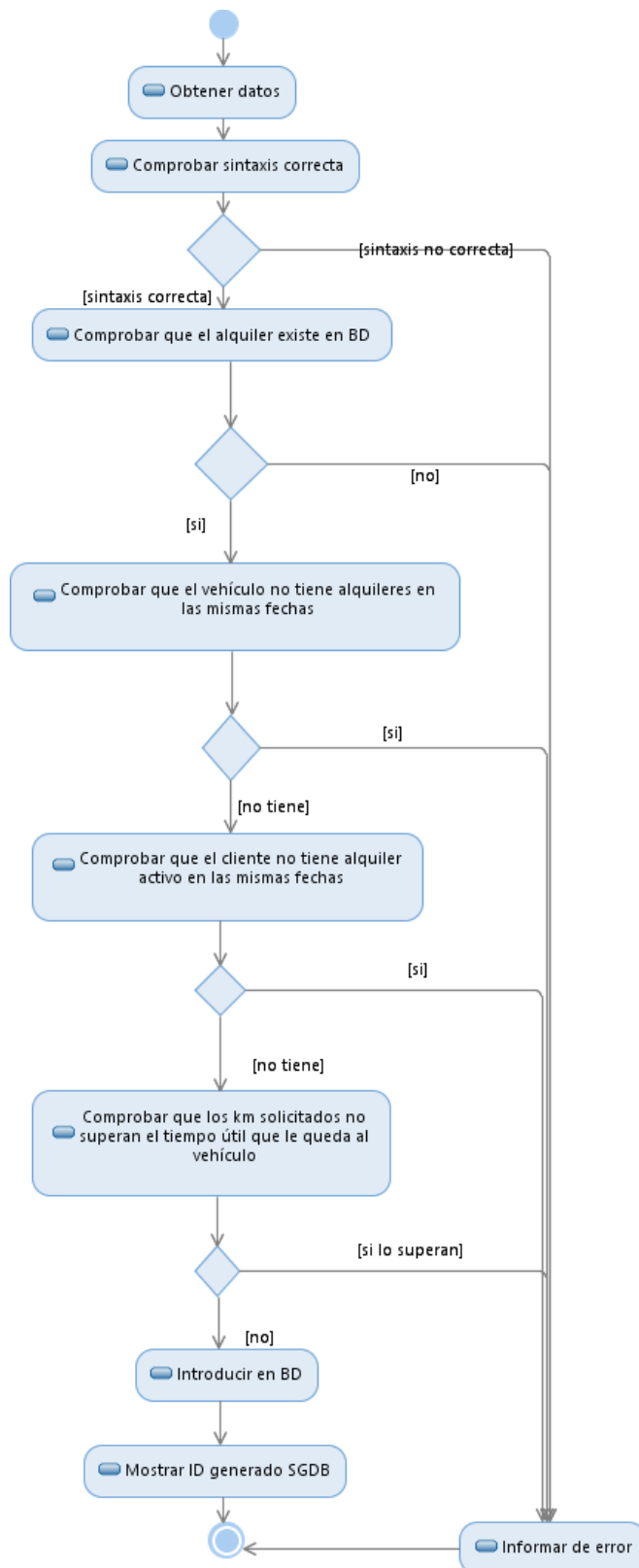
3.2.7.2 DROP_RENTAL	
Descrita por	(nombre gestor)
Prioridad	Alta
Estabilidad	Alta
Descripción	Inserta el valor "FALSE" en el atributo activo de un alquiler de la BD
Actor	Gestor Base de datos
Entrada	ID del alquiler.
Salida	Mostrar id o error.
Origen	Interfaz usuario - Gestor del sistema
Destino	Interfaz gráfica de usuario
Necesita	BD de GREENGO
Acciones	Descritas en el diagrama de actividades
Precondición	Existe el alquiler en la BD y este está activo.
Postcondición	Campo activo a "FALSE".
Efectos laterales	-





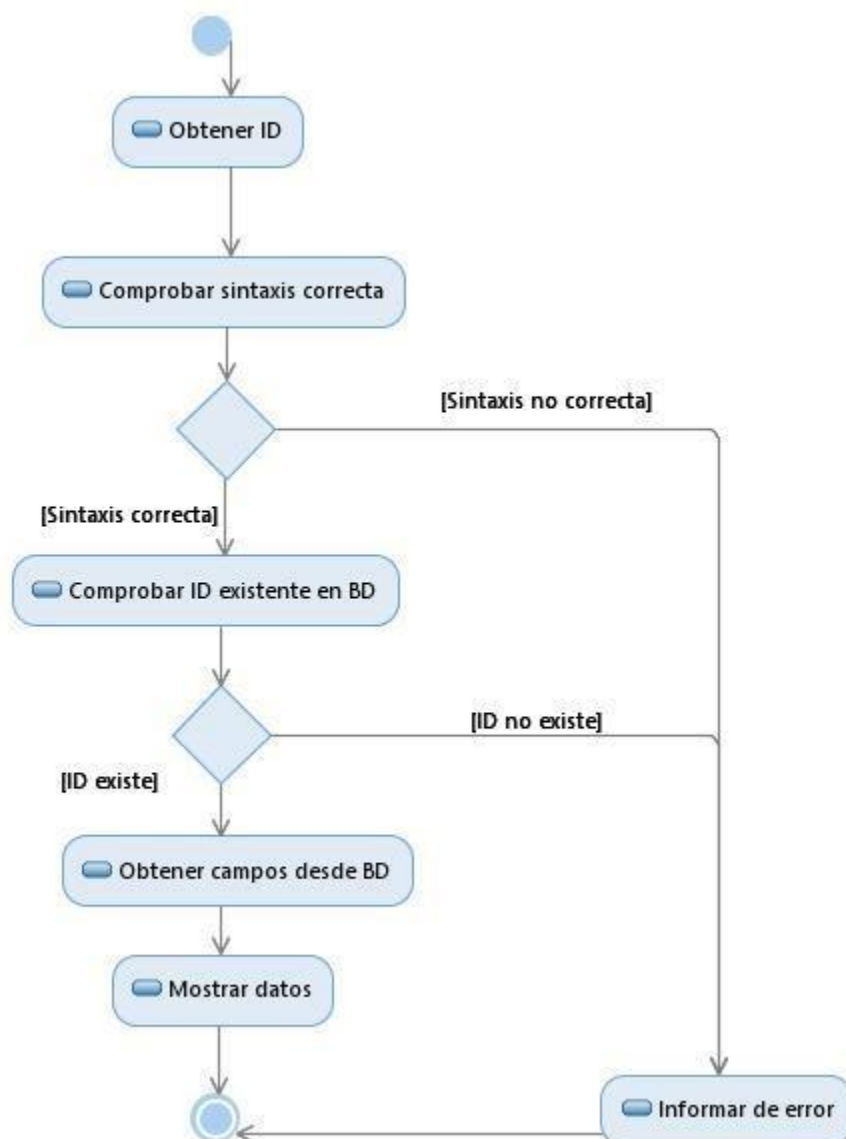
3.2.7.3 UPDATE_RENTAL	
Descrita por	(nombre gestor)
Prioridad	Media
Estabilidad	Alta
Descripción	Modificar el estado de los atributos <i>km_rented</i> y <i>active</i> de un alquiler concreto.
Actor	Gestor Base de datos
Entrada	ID del alquiler, valor nuevo de los atributos.
Salida	Mensaje de confirmación o error.
Origen	Interfaz usuario - Gestor del sistema
Destino	Interfaz gráfica de usuario
Necesita	BD de GREENGO
Acciones	Descritas en el diagrama de actividades
Precondición	Alquiler existente en la BD, valor nuevo de <i>km_rented</i> estrictamente mayor que cero y nueva fecha inicio menor que nueva fecha vencimiento.
Postcondición	Datos del alquiler modificados.
Efectos laterales	-





3.2.7.4 SHOW_RENTAL	
Descrita por	(nombre gestor)
Prioridad	Media
Estabilidad	Alta
Descripción	Visualización del estado actual de los atributos de un determinado alquiler.
Actor	Gestor Base de datos
Entrada	ID del alquiler.
Salida	Estado actual atributos o error.
Origen	Interfaz usuario - Gestor del sistema
Destino	Interfaz gráfica de usuario
Necesita	BD de GREENGO
Acciones	Descritas en el diagrama de actividades
Precondición	Alquiler existente en la BD.
Postcondición	-
Efectos laterales	-





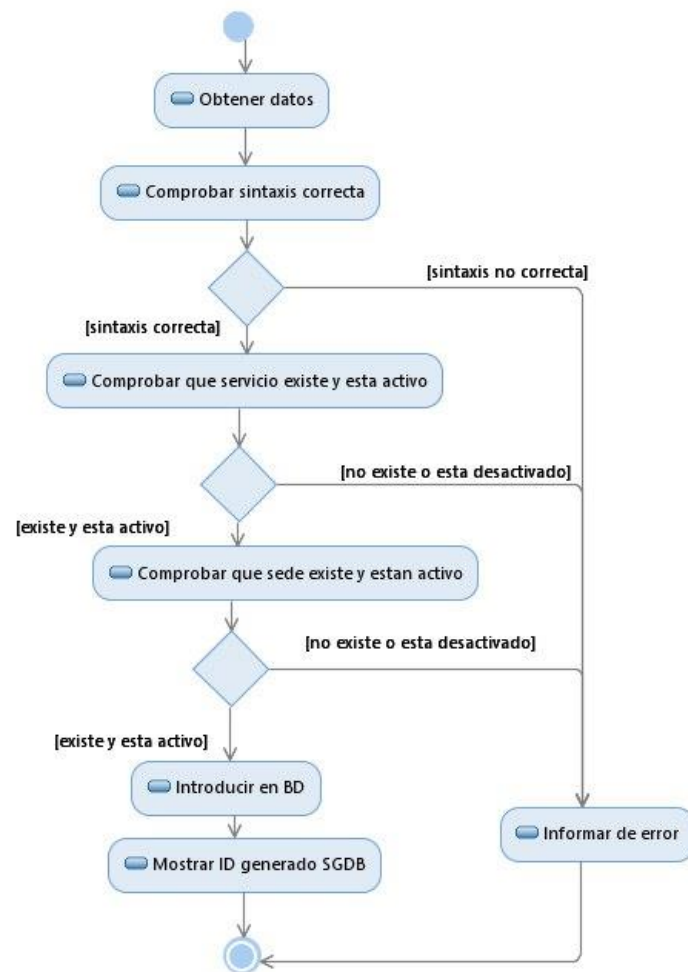
3.2.7.5	
SHOWALL_RENTAL	
Descrita por	(nombre gestor)
Prioridad	Media
Estabilidad	Alta
Descripción	Visualización del estado actual de los atributos de los alquileres pertenecientes al listado de alquileres con el atributo activo con estado actual "TRUE".
Actor	Gestor Base de datos
Entrada	-
Salida	Listado de alquileres.
Origen	Interfaz usuario - Gestor del sistema
Destino	Interfaz gráfica de usuario
Necesita	BD de GREENGO
Acciones	Descritas en el diagrama de actividades
Precondición	-
Postcondición	-
Efectos laterales	-



- Contract

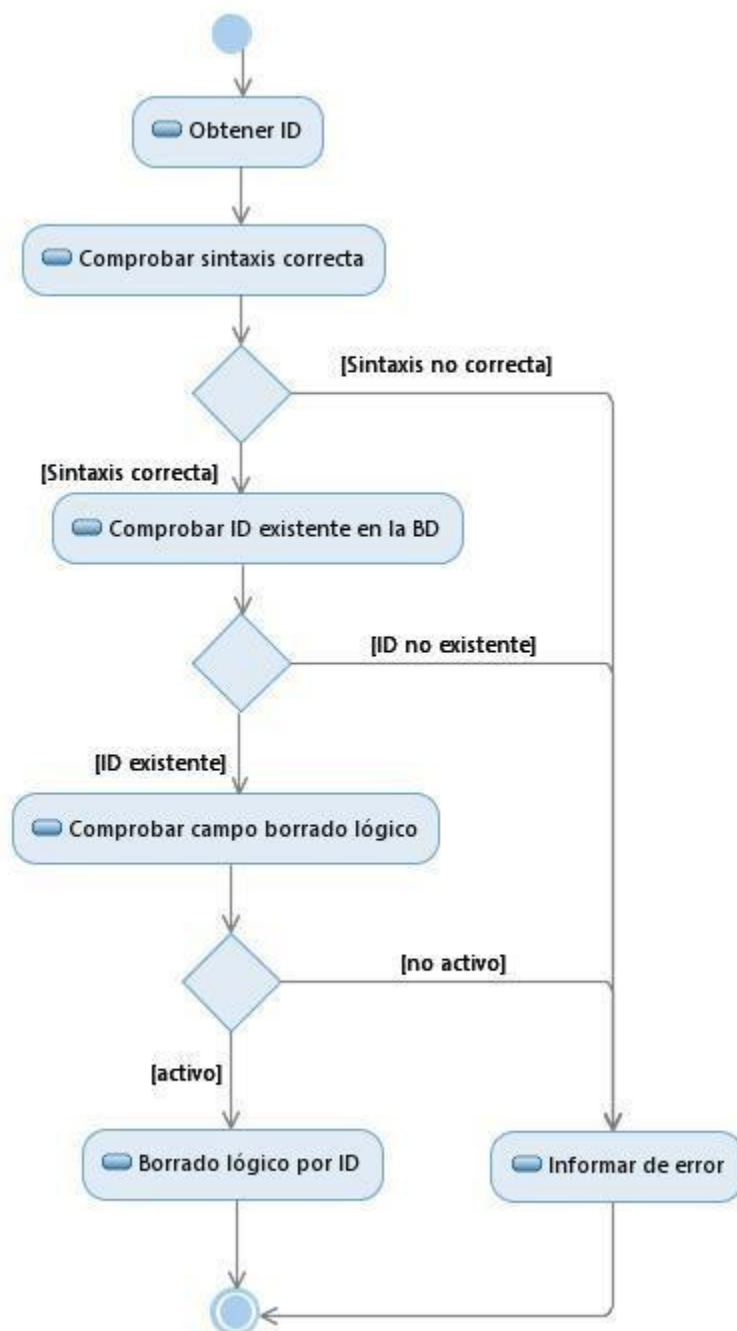
3.2.8.1 CREATE_CONTRACT	
Descrita por	(nombre gestor)
Prioridad	Alta
Estabilidad	Alta
Descripción	Añadir un contrato con un estado concreto al sistema de GREENGO
Actor	Gestor Base de datos
Entrada	ID de la Sede, ID del servicio, nivel del servicio.
Salida	Mensaje con ID o de error.
Origen	Interfaz usuario - Gestor del sistema
Destino	Interfaz gráfica de usuario
Necesita	BD de GREENGO
Acciones	Descritas en el diagrama de actividades
Precondición	No existe contrato en el sistema que posea el ID introducido.
Postcondición	Añadir contrato con valores en sus atributos iguales a los datos introducidos por entrada a la BD
Efectos laterales	-





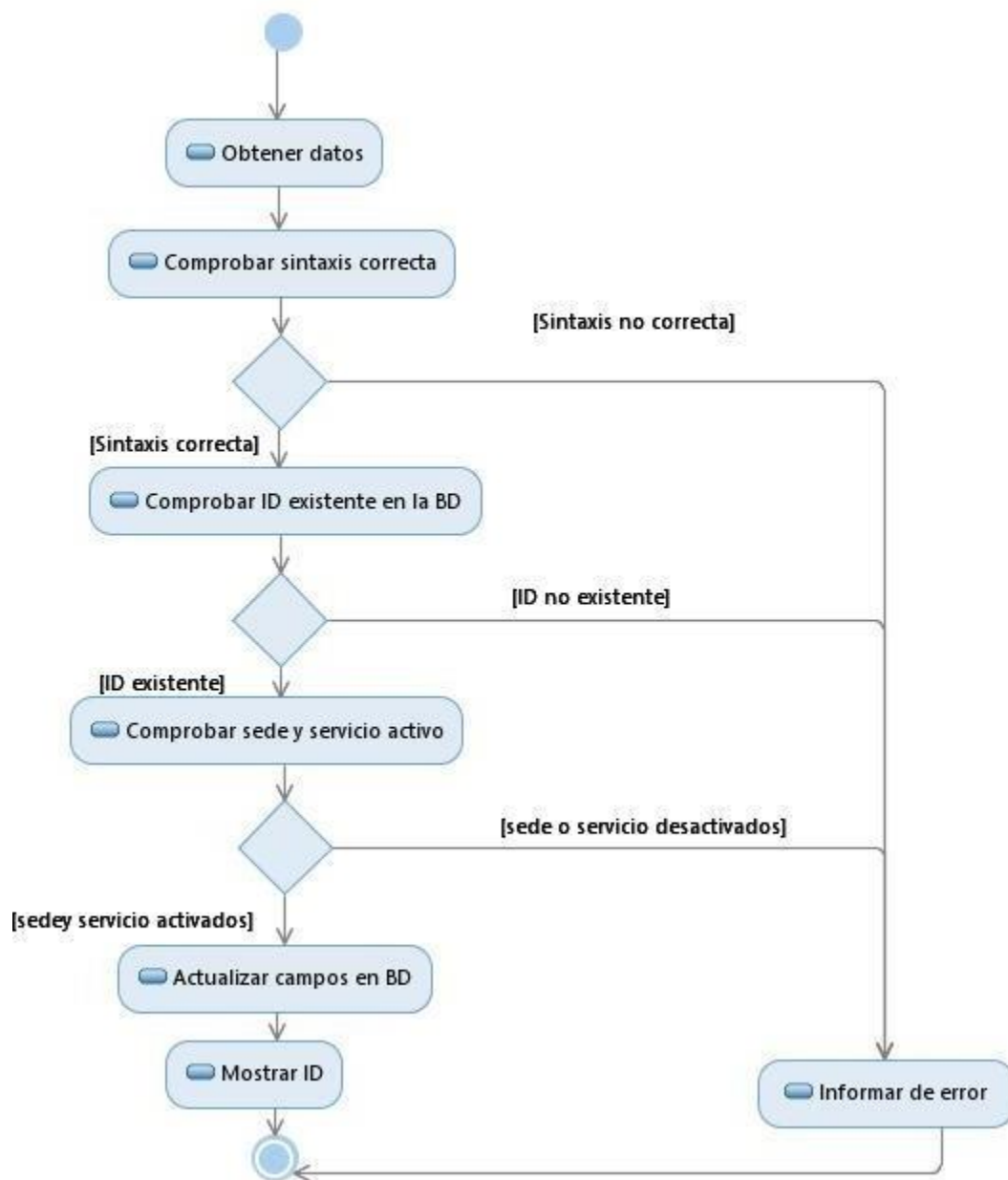
3.2.8.2 DROP_ CONTRACT	
Descrita por	(nombre gestor)
Prioridad	Alta
Estabilidad	Alta
Descripción	Inserta el valor "FALSE" en el atributo <i>activo</i> de un contrato de la BD.
Actor	Gestor Base de datos
Entrada	ID del contrato.
Salida	Mensaje con ID o error.
Origen	Interfaz usuario - Gestor del sistema
Destino	Interfaz gráfica de usuario
Necesita	BD de GREENGO
Acciones	Descritas en el diagrama de actividades
Precondición	Contrato existente en la BD y campo activo a "TRUE"
Postcondición	Atributo <i>activo</i> con valor "FALSE".
Efectos laterales	-





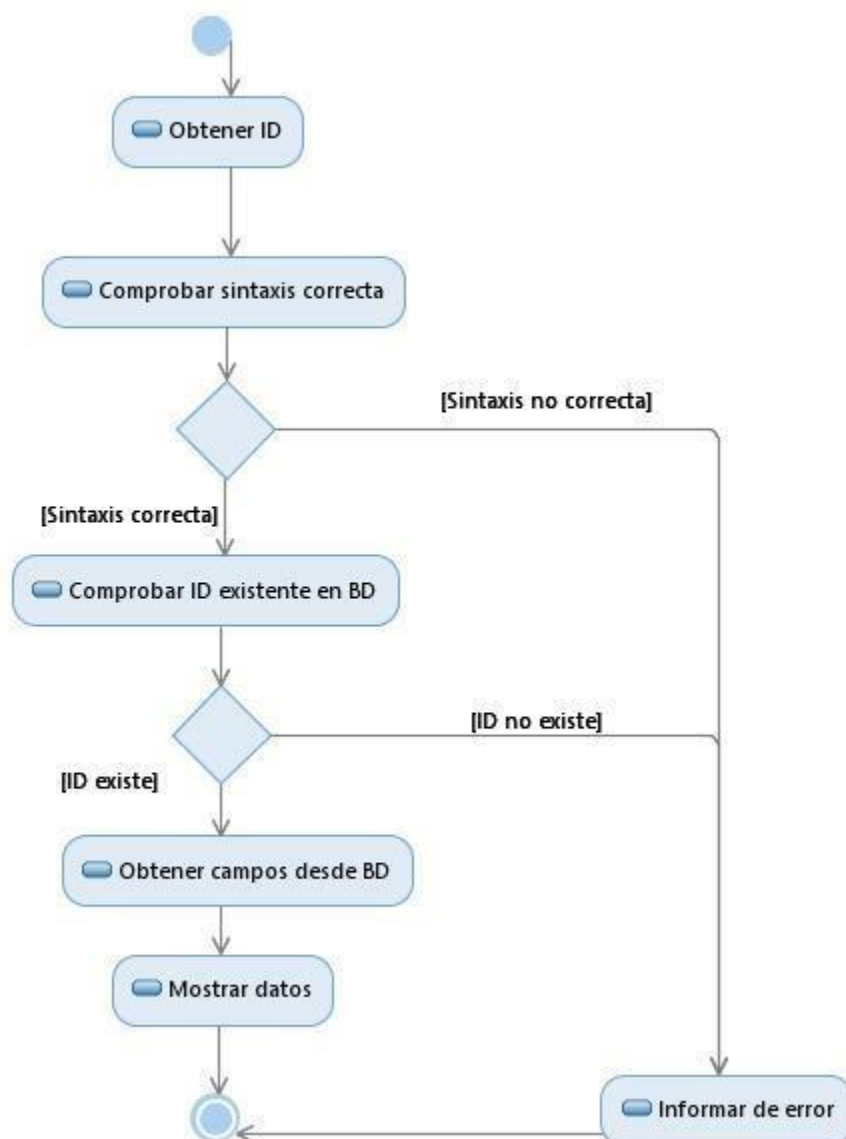
3.2.8.3 UPDATE_ CONTRACT	
Descrita por	(nombre gestor)
Prioridad	Media
Estabilidad	Alta
Descripción	Modificar los atributos <i>active</i> y <i>Service_Level</i> de un contrato específico
Actor	Gestor Base de datos
Entrada	ID del contrato, nuevo estado de los atributos a modificar.
Salida	Mensaje de ID o error.
Origen	Interfaz usuario - Gestor del sistema
Destino	Interfaz gráfica de usuario
Necesita	BD de GREENGO
Acciones	Descritas en el diagrama de actividades
Precondición	Contrato existente en la BD.
Postcondición	Modificar datos del contrato.
Efectos laterales	-





3.2.8.4 SHOW_CONTRACT	
Descrita por	(nombre gestor)
Prioridad	Media
Estabilidad	Alta
Descripción	Visualización del estado actual de los atributos de un determinado contrato.
Actor	Gestor Base de datos
Entrada	ID del contrato.
Salida	Estado actual atributos.
Origen	Interfaz usuario - Gestor del sistema
Destino	Interfaz gráfica de usuario
Necesita	BD de GREENGO
Acciones	Descritas en el diagrama de actividades
Precondición	Contrato existente en la BD y atributo <i>active</i> con valor "TRUE".
Postcondición	-
Efectos laterales	-





3.2.8.5 SHOWALL_CONTRACT	
Descrita por	(nombre gestor)
Prioridad	Media
Estabilidad	Alta
Descripción	Visualización del estado actual de los atributos de los contratos pertenecientes al listado de contratos con el atributo activo con estado actual "TRUE".
Actor	Gestor Base de datos
Entrada	-
Salida	Estado actual atributos.
Origen	Interfaz usuario - Gestor del sistema
Destino	Interfaz gráfica de usuario
Necesita	BD de GREENGO
Acciones	Descritas en el diagrama de actividades
Precondición	-
Postcondición	-
Efectos laterales	-



3.3 Requisitos de rendimiento

La aplicación "**GREENGO**" es desarrollada para trabajar en ordenadores, fijos o portátiles, con algunas características hardware mínimas, véase Punto 3.1.2. Además, el desarrollo de la aplicación debe satisfacer las necesidades funcionales del gestor de la misma.

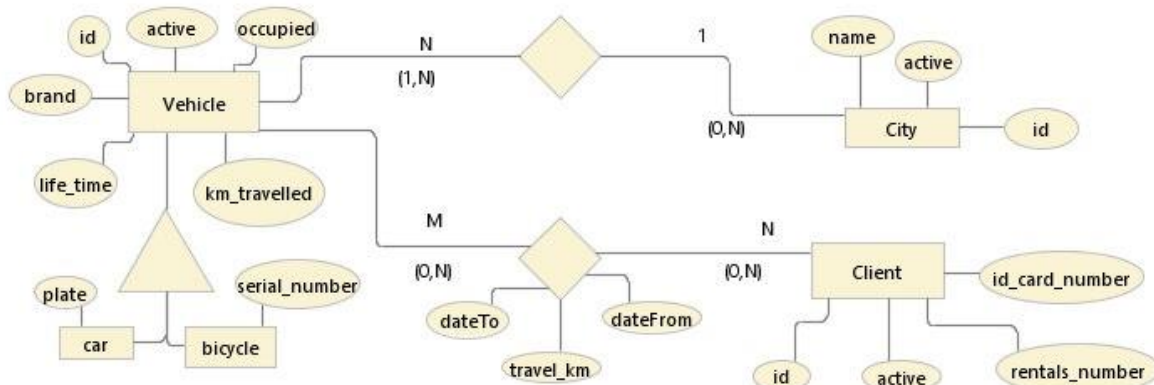
Los requisitos de rendimiento tendrán que garantizar que las consultas u otros procesos del sistema no afecten al desempeño de las operaciones, por ello se garantiza que las operaciones se realizarán en tiempos bajos cercanos a dos segundos y medio como máximo, véase el punto 2.4.

La aplicación funciona sin una conexión a internet ya que todos los datos que utiliza para su funcionamiento óptimo se almacenan en una base de datos local que la propia aplicación gestiona.

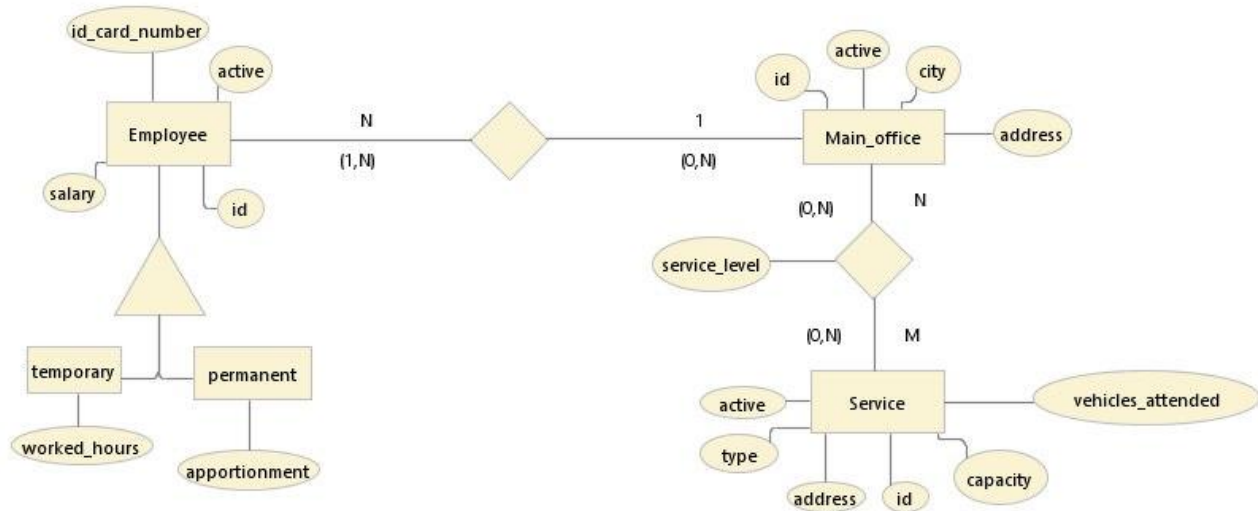
3.4 Modelo del dominio

El modelo del dominio siguiente representa la abstracción realizada por el equipo para basar el diseño de la base de datos y de la arquitectura de la aplicación.

Para la primera versión, realizada con la implementación de los DAOS, se utilizará el siguiente modelo de dominio:



Para la segunda versión, realizada con la implementación JPA, se utilizará el siguiente modelo de dominio:



3.5 Restricciones de diseño

El desarrollo de la aplicación GREENGO tiene una serie de restricciones sobre el diseño que se listan a continuación:

- El diseño de la aplicación se realiza bajo los principios del paradigma Orientado a Objetos.
- El diseño de la aplicación se hace bajo los principios impartidos en la asignatura Modelado de Software.
- La primera versión de la aplicación (correspondiente a los módulos Ciudad, Vehículos y Empleados) persistirá los datos directamente en una base relacional, utilizando objetos transferencia en la capa de negocio.
La segunda versión persistirá los datos utilizando JPA, utilizando objetos del negocio en la capa de negocio.
- El lenguaje de programación, en coherencia con el paradigma, es java.
- La herramienta CASE para el análisis y diseño es:
IBM Rational Software Architect Designer.

3.6 Atributos del Sistema software

- **Eficiencia:** Será capaz de utilizar de manera más rápida posible todas funciones con las que cuenta la aplicación y con el mínimo número de recursos posible.



- **Robustez:** Será capaz de reaccionar ante cualquier condición, es decir, controlara todos los posibles caminos que generen ralentizaciones y fallos en la aplicación.
- **Extensibilidad:** Sera escalable y ampliable a futuras utilidades de la manera más sencilla posible.
- **Corrección:** Cumplirá los objetivos que se han marcado en la especificación.
- **Verificabilidad:** Se creará métodos sencillos y fáciles de verificar, mostrando así claramente los resultados de los ensayos y pruebas.
- **Facilidad de uso:** Uso del sistema será fácil y simple para cualquier usuario con conocimientos básicos del ecosistema que rodea a la aplicación.
- **Legibilidad:** Código será sencillo y legible, así como con un buen estilo de programación. Con comentarios explicativos completos y una documentación detallada.
- **Fácil mantenimiento:** Se podrá realizar modificaciones en el software sin dificultad ante la aparición de posibles fallos o futuras ampliaciones.
- **Integridad:** Se controlará el acceso a información del sistema, de modo que no se pueda producir un deterioro de los datos por el acceso no permitido de los usuarios a otros componentes.

Como conclusión, se puede afirmar que el objetivo principal en el desarrollo de esta aplicación software es producir un sistema de gestión de datos de alta calidad y de coste mínimo: barato de diseñar, implementar, operar, mantener y modificar.

