

INFORME DE LA APLICACIÓN

UPOCarAuto - Concesionario de coches de segunda mano



04/05/2025

INTEGRACIÓN DE TECNOLOGÍAS

Javier Diaz Morales, Teodoro Ruiz Rodríguez, María Vazquez Rodríguez y Marina Benito Torrado

ÍNDICE

1. Descripción del proyecto.....	2
2. Resumen ejecutivo.....	3
2.1 Entidades - Operaciones CRUD.....	3
2.2 Aportaciones extras.....	7
3. Recolección de los requerimientos del problema.....	9
3.1 Diagramas de casos de uso.....	9
3.2 Casos de uso.....	13
-Entidad Reserva:.....	14
-Entidad Cliente:.....	17
-Entidad Concesionario:.....	19
-Entidad Empleado:.....	21
-Entidad Vehículo:.....	23
-Entidad Transacción:.....	26
-Entidad Pago:.....	28
-Entidad Inventario:.....	31
4. Esquema Funcional.....	34
5. Diseño de la base de datos.....	36
5.1 Diseño físico de la base de datos.....	40
6. Casos de prueba.....	41
7. Manual de uso.....	43
8 Alojamiento de la aplicación en un servidor.....	45
8.1 Servicio de alojamiento en la nube elegido.....	45
8.2 Software instalado en la en el servidor:.....	46
8.3 Configuración de un dominio dinámico para el servidor.....	47
8.4 Proceso de instalación y configuración.....	49
8.5 Instalación y configuración del entorno de java jdk-8u281-linux-x64.....	50
8.6 Instalación de xampp-linux-x64-8.0.28-0.....	50
8.7 Instalación de glassfish 4.1.1.....	51
8.8 Configurar ssl para acceder mediante https.....	52
8.9 Despliegue de las aplicación y de los servicios.....	52
8.10 Referencias.....	52

1. Descripción del proyecto

El proyecto UPOCarAuto es una aplicación web diseñada como herramienta interna para la gestión de concesionarios de vehículos de segunda mano. Está orientada exclusivamente al uso por parte de los empleados, quienes utilizarán la plataforma para registrar y gestionar toda la actividad operativa relacionada con clientes, vehículos, ventas...

El sistema permitirá a los empleados:

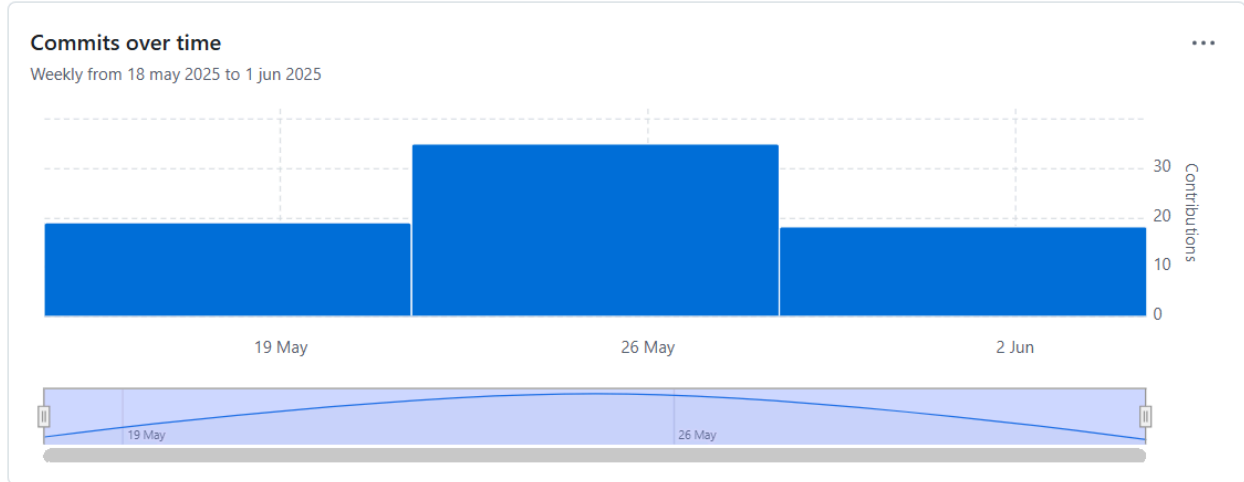
- Registrar y gestionar clientes que visitan el concesionario.
- Consultar y administrar el inventario de vehículos disponibles.
- Realizar reservas de vehículos para clientes interesados.
- Registrar pagos y transacciones, gestionando el ciclo completo de la compraventa.
- Gestionar las sedes del concesionario
- Administrar a los empleados
- Generar documentación como recibos o justificantes de pago.
- Enviar notificaciones por correo electrónico directamente desde el sistema (como confirmaciones de reserva o compra).

La base de datos estará estructurada en torno a las siguientes entidades: Empleado, Concesionario, Inventario, Vehículo, Reserva, Cliente, Pago y Transacción.

El sistema se podrá desplegar tanto en servidores locales como en la nube, garantizando seguridad, escalabilidad y eficiencia en la gestión diaria del negocio, sin requerir acceso por parte de los clientes desde el exterior.

2. Resumen ejecutivo

ENLACE A GITHUB : <https://github.com/mvazrod1/UPOCarAuto.git>



2.1 Entidades - Operaciones CRUD

Entidades \ CRUD	CREAR	LEER	MODIFICAR	ELIMINAR
Empleado	X	X	X	X
Concesionario	X	X	X	X
Inventario	X	X	X	X
Vehículo	X	X	X	X
Reserva	X	X	X	X
Cliente	X	X	X	X
Pago	X	X	X	X
Transacción	X	X	X	X

También se han implementado las funcionalidades de iniciar sesión y cerrar sesión.

Empleado	
Operaciones	Implementación
Crear	El empleado puede realizar todas las funcionalidades disponibles.
Leer	El empleado puede leer los datos de cualquier entidad.
Modificar	El empleado puede modificar todos los datos de las entidades, excepto los atributos autocalculados.
Eliminar	El empleado puede eliminar cualquier objeto de cualquier entidad.

Concesionario	
Operaciones	Implementación
Crear	El empleado puede añadir un concesionario a la base de datos.
Leer	El empleado puede leer los datos de un concesionario.
Modificar	El empleado puede modificar los datos de un concesionario (excepto el id).
Eliminar	El empleado puede eliminar un concesionario.

Inventario	
Operaciones	Implementación
Crear	El empleado puede añadir un inventario que va asociado a únicamente un concesionario y además tiene asociados de uno a muchos vehículos
Leer	El empleado puede leer los datos de un inventario
Modificar	El empleado puede modificar los datos del inventario (excepto el id del concesionario y el total de vehículos que es un campo calculado)
Eliminar	El empleado puede eliminar un inventario y sus vehículos asociados.

Vehículo	
Operaciones	Implementación
Crear	El empleado puede añadir un vehículo, que va asociado a un inventario y de una a muchas reservas
Leer	El empleado puede leer los datos de un vehículo
Modificar	El empleado puede modificar los datos del vehículo
Eliminar	El empleado puede eliminar un vehículo, así como sus reservas asociadas, además se eliminarán los pagos asociados a las reservas.

Reserva	
Operaciones	Implementación
Crear	El empleado puede añadir una reserva, la cual va asociada a un cliente existente y un vehículo disponible
Leer	El empleado puede leer los datos de la reserva
Modificar	El empleado puede modificar los datos de la reserva excepto el id, el dni del cliente y la matrícula
Eliminar	El empleado puede eliminar una reserva siempre y cuando no tenga pagos asociados

Cliente	
Operaciones	Implementación
Crear	El empleado puede añadir clientes nuevos a la base de datos del concesionario
Leer	El empleado puede leer los datos de los clientes registrados
Modificar	El empleado puede modificar los datos de los clientes registrados
Eliminar	El empleado puede eliminar a un cliente de la base de datos

Pago	
Operaciones	Implementación
Crear	El empleado puede añadir pagos nuevos asociados a una reserva.
Leer	El empleado puede leer los datos de los pagos registrados.
Modificar	El empleado puede modificar los datos de cualquier pago (excepto el id de pago y el id de la reserva asociada).
Eliminar	El empleado puede eliminar cualquier pago registrado.

Transacción	
Operaciones	Implementación
Crear	El empleado puede añadir transacciones que están asociadas a un pago.
Leer	El empleado puede ver los datos de cualquier transacción.
Modificar	El empleado puede modificar los datos de cualquier transacción (excepto el id de pago y el id de transacción).
Eliminar	El empleado puede eliminar cualquier transacción asociada a un pago.

2.2 Aportaciones extras

·Envío de correo electrónico con justificante de pago mediante un web service:

Una de las funcionalidades añadidas a UPOCarAuto ha sido el envío automático de un **correo electrónico** al registrar un pago. Esta mejora permite que, una vez se realiza el pago asociado a una reserva, se envíe un mensaje al cliente con los detalles más importantes de la operación. Esto no solo aporta profesionalidad, sino que facilita al cliente tener un comprobante en su bandeja de entrada sin tener que pedirlo manualmente.

El proceso es automático y transparente para el usuario: en cuanto se guarda el pago, se genera la información y se envía por correo. Además, es especialmente útil para llevar un seguimiento de las operaciones sin necesidad de entrar de nuevo en la aplicación.

·Generación de PDF como comprobante mediante un web service:

Junto con el correo, también se genera un **archivo PDF** que contiene el justificante del pago. Este documento se adjunta al correo electrónico para que el cliente pueda guardarlo, imprimirlo o consultarlo cuando lo necesite. De esta forma, cada pago queda documentado con su correspondiente recibo, haciendo que el sistema sea más completo y cómodo tanto para el concesionario como para el cliente.

·Bootstrap:

Es un framework front-end de código abierto que facilita el desarrollo de interfaces web modernas y responsivas. Proporciona una amplia colección de componentes predefinidos y clases CSS que permiten construir páginas web con un diseño limpio, adaptable y coherente en distintos dispositivos y navegadores.

En este proyecto, Bootstrap ha sido utilizado para mejorar la presentación y estructura visual de las vistas, permitiendo una experiencia de usuario más intuitiva y profesional. Se ha usado en algunos archivos como:

- header.jsp y footer.jsp: para estructurar y estilizar los elementos de navegación superior e inferior, asegurando su correcta visualización en pantallas de distintos tamaños.
- index.jsp: se ha incorporado el componente carrusel de Bootstrap, con el fin de mostrar contenido destacado de forma dinámica y visualmente atractiva.

El uso de este framework ha contribuido a acelerar el desarrollo del diseño front-end y a mantener la coherencia visual en toda la aplicación.

-Despliegue de la aplicación en un servidor en la nube:

El alojamiento elegido para la aplicación se realiza en un servidor físico dedicado, propiedad de uno de los miembros del grupo, ubicado en su red local privada. Esta decisión se tomó para aprovechar los recursos disponibles y tener control total sobre el entorno de despliegue. El proceso completo de configuración del servidor y el despliegue de las aplicaciones se explica en el apartado 8 de este documento.

La url de la aplicación es: <https://upocarauto.sytes.net/> . Hay que tener en cuenta que sólo funcionará mientras el servidor esté encendido y en el primer acceso hay que aceptar el certificado en la opción avanzado.

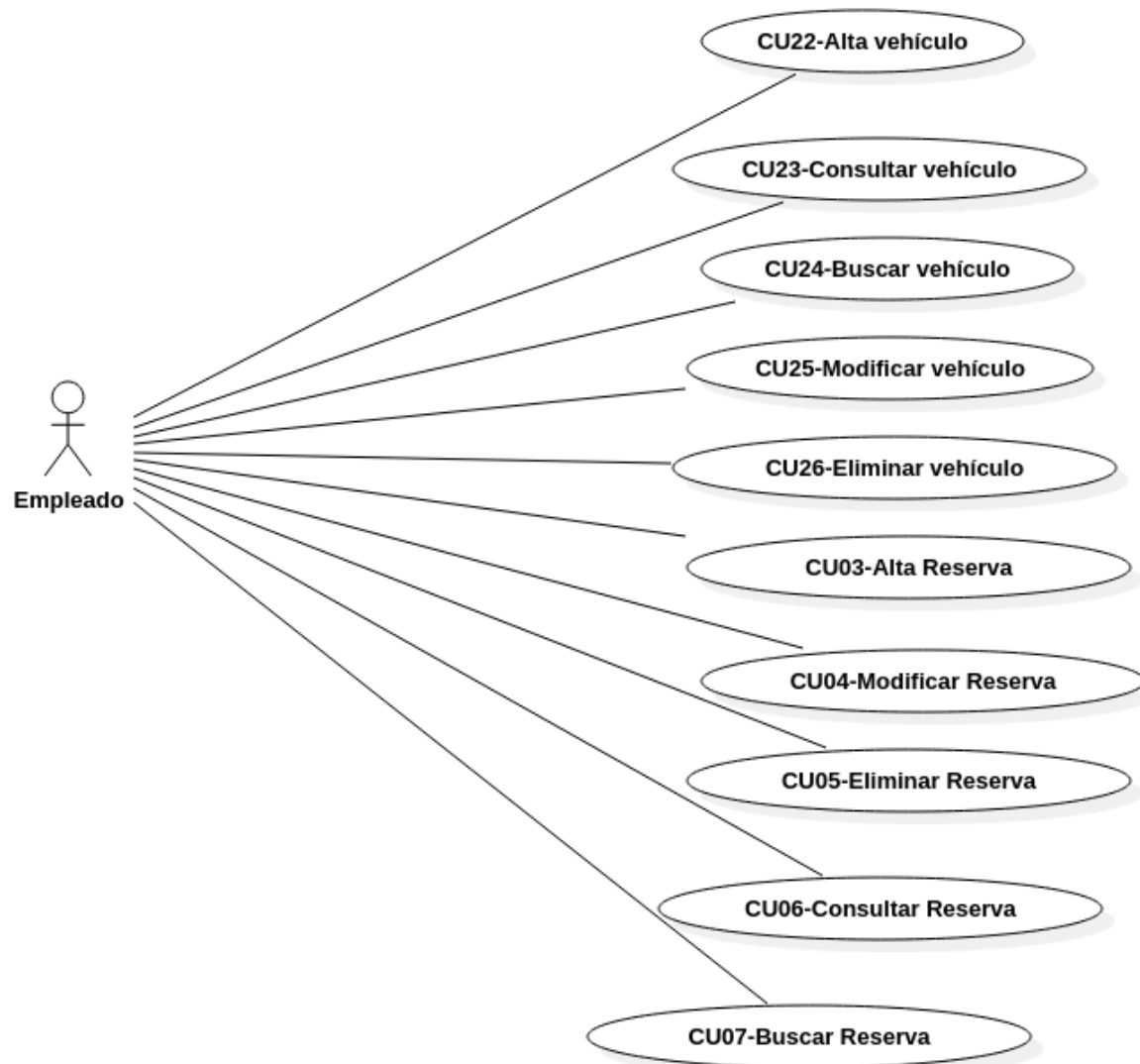
3. Recolección de los requerimientos del problema

3.1 Diagramas de casos de uso

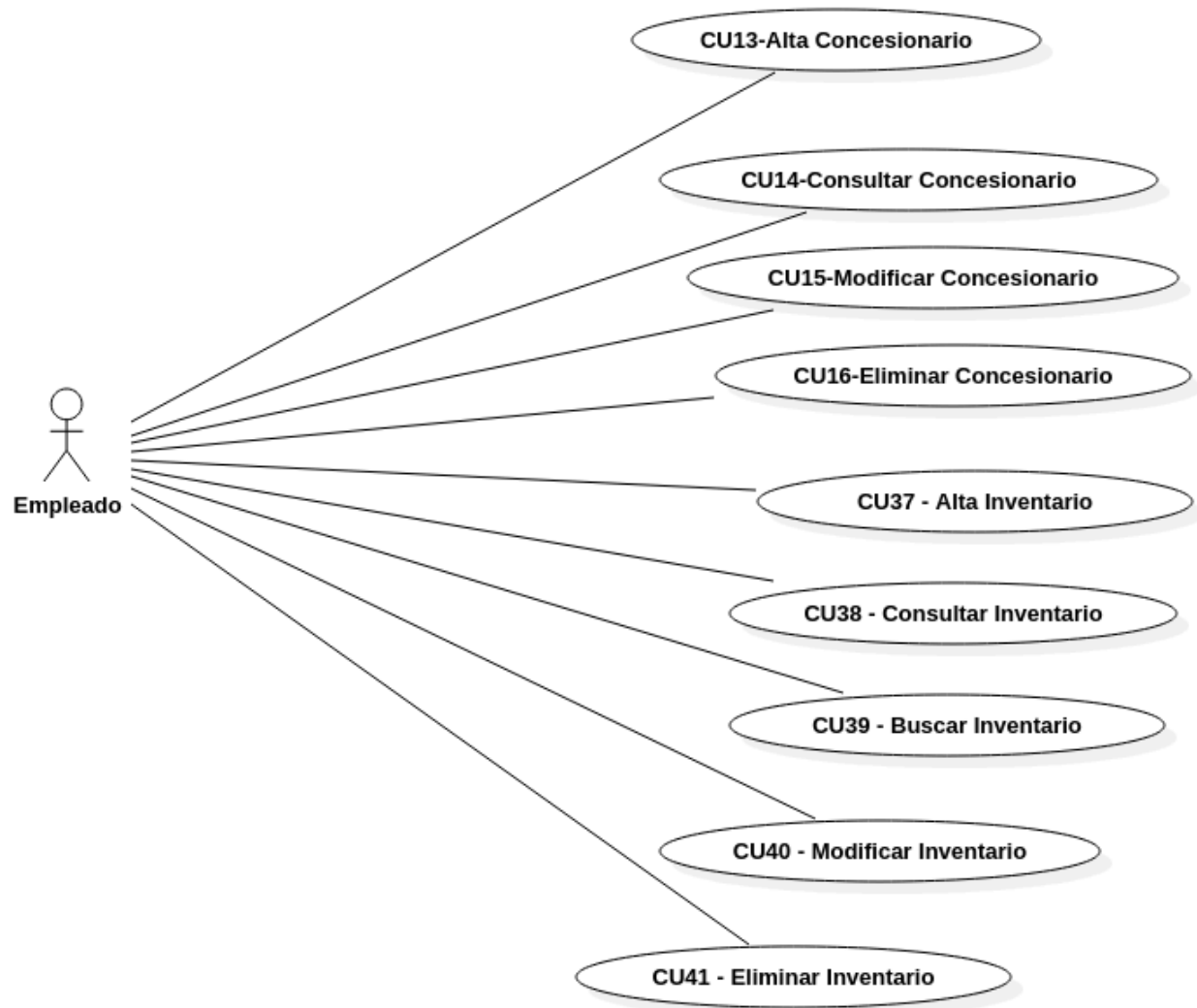
- Diagrama de casos de uso para la gestión general de usuarios:



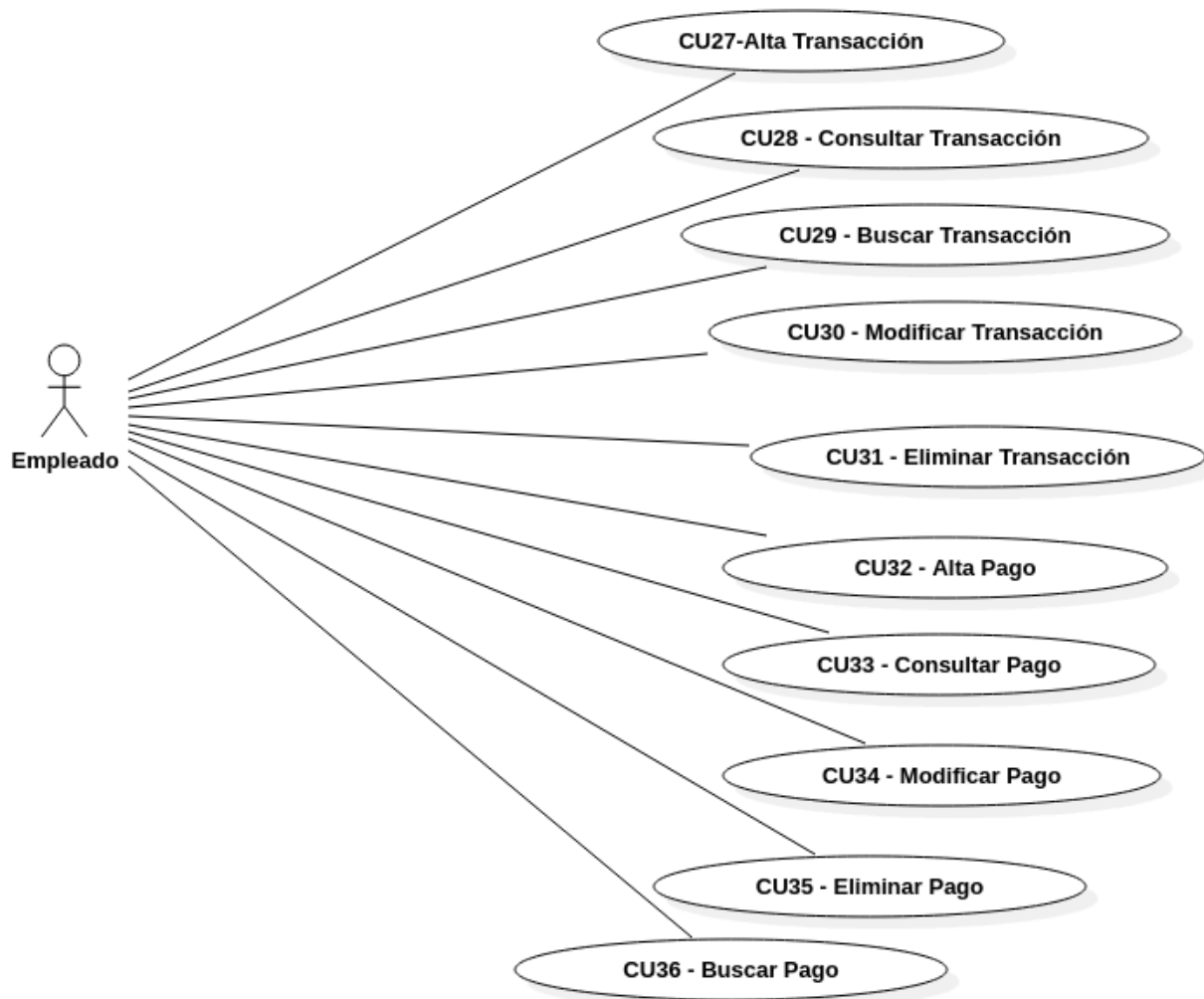
- Diagrama de casos de uso para la gestión de vehículos y de reservas:



- Diagrama de casos de uso de la gestión de inventarios y de concesionarios:



- Diagrama de casos de uso de la gestión de pagos y de transacciones:



3.2 Casos de uso

CU01-Iniciar Sesión	
Actor	Empleado
Descripción	Permite al empleado autenticarse en el sistema para acceder a sus funcionalidades.
Precondición	El empleado debe estar registrado en el sistema.
Flujo Principal	<ol style="list-style-type: none">1. El empleado accede a la pantalla de inicio de sesión.2. El empleado introduce su DNI y contraseña.3. El sistema valida las credenciales.4. Si las credenciales son correctas, el sistema inicia la sesión del empleado.5. El sistema redirige al empleado al menú principal.
Postcondición	El empleado accede al sistema con una sesión activa.

CU02-Cerrar Sesión	
Actor	Empleado
Descripción	Permite al empleado cerrar su sesión actual en el sistema.
Precondición	El empleado debe haber iniciado sesión previamente.
Flujo Principal	<ol style="list-style-type: none">1. El empleado selecciona la opción “Cerrar sesión”.2. El sistema finaliza la sesión del empleado.3. El sistema redirige al usuario a la pantalla de inicio de sesión.
Postcondición	La sesión del empleado se cierra y se libera el acceso a las funcionalidades del sistema.

-Entidad Reserva:

CU03-Alta Reserva	
Actor	Empleado
Descripción	Permite al empleado registrar una nueva reserva para un vehículo por parte de un cliente.
Precondición	El empleado debe estar autenticado y deben existir clientes y vehículos disponibles.
Flujo Principal	<ol style="list-style-type: none">1. El empleado accede a la página con el formulario de nueva reserva.2. El sistema muestra un formulario donde se ingresan los datos de la reserva (cliente, vehículo, estado, fecha de recogida).3. El empleado introduce los datos y envía el formulario.4. El sistema valida los datos introducidos (incluyendo el formato de la fecha y que no sea pasada).5. El sistema registra la reserva en la base de datos y marca el vehículo como no disponible.6. El sistema confirma la creación de la nueva reserva.
Postcondición	La reserva se registra en el sistema y el vehículo queda asignado como reservado.

CU04-Modificar Reserva	
Actor	Empleado
Descripción	Permite al empleado modificar los datos de una reserva ya existente.
Precondición	El empleado debe estar autenticado y la reserva debe existir.
Flujo Principal	<ol style="list-style-type: none"> 1. El empleado accede a la opción para modificar una reserva. 2. El sistema carga los datos actuales de la reserva. 3. El empleado modifica los datos (estado, fecha de recogida) y envía el formulario. 4. El sistema valida los datos ingresados. 5. El sistema actualiza la reserva en la base de datos. 6. Si la reserva es rechazada, el sistema actualiza la disponibilidad del vehículo. 7. El sistema confirma la modificación.
Postcondición	Los datos de la reserva se actualizan en el sistema y la disponibilidad del vehículo se ajusta según corresponda.

CU05-Eliminar Reserva	
Actor	Empleado
Descripción	Permite eliminar una reserva existente y liberar el vehículo asociado.
Precondición	El empleado debe estar autenticado y la reserva debe existir.
Flujo Principal	<ol style="list-style-type: none"> 1. El empleado selecciona una reserva para eliminarla. 2. El sistema verifica que la reserva exista. 3. El sistema elimina la reserva de la base de datos. 4. El sistema marca el vehículo como disponible nuevamente. 5. El sistema confirma la eliminación.
Postcondición	La reserva es eliminada del sistema y el vehículo queda disponible.

CU06-Consultar Reserva	
Actor	Empleado
Descripción	Permite al empleado consultar los detalles de una reserva específica.
Precondición	El empleado debe estar autenticado.
Flujo Principal	<ol style="list-style-type: none"> 1. El empleado accede al módulo de consulta de reservas. 2. El sistema solicita el ID de la reserva a consultar. 3. El sistema muestra los datos de la reserva si esta existe.
Postcondición	El sistema muestra la información de la reserva solicitada.

CU07-Buscar Reserva	
Actor	Empleado
Descripción	Permite buscar y visualizar una reserva específica a partir de su ID.
Precondición	El empleado debe estar autenticado.
Flujo Principal	<ol style="list-style-type: none"> 1. El empleado introduce un ID de reserva en el campo de búsqueda. 2. El sistema consulta la base de datos con el ID ingresado. 3. Si la reserva existe, se muestra en la lista de resultados. 4. Si no existe, se muestra un mensaje de advertencia.
Postcondición	Se muestra la información de la reserva, o un mensaje indicando que no existe.

-Entidad Cliente:

CU08-Alta Cliente	
Actor	Empleado
Descripción	Permite al empleado registrar un nuevo cliente en el sistema.
Precondición	El empleado debe estar autenticado.
Flujo Principal	<ol style="list-style-type: none">1. El empleado accede al formulario de alta de cliente.2. El sistema muestra un formulario para ingresar los datos del cliente (DNI, nombre, apellidos, email, teléfono, dirección).3. El empleado completa el formulario y lo envía.4. El sistema valida los datos ingresados.5. El sistema registra al nuevo cliente en la base de datos.6. El sistema confirma el registro del cliente.
Postcondición	El cliente queda registrado en el sistema con sus datos completos.

CU09-Modificar Cliente	
Actor	Empleado
Descripción	Permite al empleado modificar los datos de un cliente existente.
Precondición	El empleado debe estar autenticado y el cliente debe existir en el sistema.
Flujo Principal	<ol style="list-style-type: none">1. El empleado accede a la opción de modificar cliente.2. El sistema recupera y muestra los datos actuales del cliente.3. El empleado modifica los campos deseados y envía el formulario.4. El sistema valida los nuevos datos ingresados.5. El sistema actualiza la información del cliente en la base de datos.6. El sistema confirma la modificación.
Postcondición	La información del cliente queda actualizada en el sistema.

CU10-Eliminar Cliente	
Actor	Empleado
Descripción	Permite al empleado eliminar a un cliente del sistema.
Precondición	El empleado debe estar autenticado y el cliente debe existir.
Flujo Principal	<ol style="list-style-type: none"> 1. El empleado selecciona el cliente a eliminar. 2. El sistema elimina al cliente de la base de datos. 3. El sistema confirma la eliminación.
Postcondición	El cliente ya no existe en el sistema.

CU11-Consultar Cliente	
Actor	Empleado
Descripción	Permite visualizar los datos de un cliente específico.
Precondición	El empleado debe estar autenticado.
Flujo Principal	<ol style="list-style-type: none"> 1. El empleado accede a la opción de consulta de clientes. 2. El sistema solicita el DNI del cliente. 3. El sistema recupera y muestra los datos del cliente si existe.
Postcondición	Se muestran los datos del cliente.

CU12-Buscar Cliente	
Actor	Empleado
Descripción	Permite buscar un cliente específico a través de su DNI.
Precondición	El empleado debe estar autenticado.
Flujo Principal	<ol style="list-style-type: none"> 1. El empleado introduce el DNI del cliente en el campo de búsqueda. 2. El sistema consulta la base de datos con ese DNI. 3. Si el cliente existe, se muestra en la lista de resultados. 4. Si no existe, el sistema muestra un mensaje de advertencia.
Postcondición	Se muestra la información del cliente o un mensaje indicando que no existe.

-Entidad Concesionario:

CU13-Alta Concesionario	
Actor	Empleado
Descripción	Permite registrar un nuevo concesionario al sistema.
Precondición	El empleado debe estar autenticado
Flujo Principal	<ol style="list-style-type: none">1. El empleado accede al formulario de alta de concesionario2. El sistema muestra los campos requeridos: nombre, dirección, teléfono y correo.3.El empleado completa los campos y envía el formulario.4.El sistema valida los datos ingresados.5.El sistema registra el nuevo concesionario en la base de datos.6.El sistema confirma el alta y muestra un mensaje de éxito
Postcondición	El concesionario queda registrado y disponible en el sistema.

CU14-Consultar Concesionario	
Actor	Empleado
Descripción	Permite consultar los datos de un concesionario por su ID.
Precondición	El empleado debe estar autenticado
Flujo Principal	<ol style="list-style-type: none">1. El empleado introduce un ID de concesionario.2. El sistema busca el concesionario correspondiente.3. Si el concesionario existe, se muestra su información.4. Si no existe, se muestra un mensaje de error.
Postcondición	Se muestran los datos del concesionario o un mensaje informando que no existe.

CU15-Modificar Concesionario	
Actor	Empleado
Descripción	Permite modificar los datos de un concesionario existente.
Precondición	El empleado debe estar autenticado y el concesionario debe existir.
Flujo Principal	<ol style="list-style-type: none"> 1. El empleado accede al formulario de modificación del concesionario. 2. El sistema carga los datos actuales del concesionario. 3. El empleado modifica los campos deseados. 4. El sistema valida los datos ingresados. 5. El sistema actualiza los datos en la base de datos. 6. El sistema muestra un mensaje de confirmación.
Postcondición	El concesionario queda actualizado con los nuevos datos.

CU16-Eliminar Concesionario	
Actor	Empleado
Descripción	Permite eliminar un concesionario del sistema.
Precondición	El empleado debe estar autenticado y el concesionario debe existir.
Flujo Principal	<ol style="list-style-type: none"> 1. El empleado selecciona un concesionario para eliminar. 2. El sistema elimina el concesionario de la base de datos. 3. El sistema muestra un mensaje de confirmación.
Postcondición	El concesionario ya no existe en el sistema.

-Entidad Empleado:

CU17-Alta Empleado	
Actor	Empleado
Descripción	Permite registrar un nuevo empleado en el sistema y asignarlo a un concesionario.
Precondición	El administrador debe estar autenticado y deben existir concesionarios registrados.
Flujo Principal	<ol style="list-style-type: none">1. El administrador accede al formulario de alta de empleado.2. El sistema muestra un formulario con los datos requeridos (DNI, nombre, apellidos, email, teléfono, dirección, puesto, contraseña y concesionario).3. El administrador completa el formulario y lo envía.4. El sistema valida los datos ingresados.5. El sistema registra al nuevo empleado y lo asocia al concesionario indicado.6. Se muestra un mensaje confirmando el registro.
Postcondición	El empleado queda registrado y disponible en el sistema.

CU18-Consultar Empleado	
Actor	Empleado
Descripción	Permite consultar los datos de un empleado mediante su DNI.
Precondición	El empleado debe estar autenticado.
Flujo Principal	<ol style="list-style-type: none">1. El empleado introduce un DNI en el campo de búsqueda.2. El sistema busca al empleado en la base de datos.3. Si lo encuentra, muestra su información.4. Si no, muestra un mensaje de error.
Postcondición	Se muestran los datos del empleado o un mensaje de que no existe.

CU19-Modificar Empleado	
Actor	Empleado
Descripción	Permite modificar los datos personales y profesionales de un empleado.
Precondición	El empleado debe estar autenticado y el empleado buscado debe existir.
Flujo Principal	<ol style="list-style-type: none"> 1. El administrador accede a la opción de editar un empleado existente. 2. El sistema carga los datos actuales del empleado. 3. El administrador modifica los datos necesarios. 4. El sistema valida los nuevos datos. 5. El sistema guarda los cambios. 6. Se muestra un mensaje de confirmación.
Postcondición	Los datos del empleado quedan actualizados en el sistema.

CU20-Eliminar Empleado	
Actor	Empleado
Descripción	Permite eliminar a un empleado del sistema.
Precondición	El empleado debe estar autenticado y el DNI debe existir.
Flujo Principal	<ol style="list-style-type: none"> 1. El administrador selecciona un empleado a eliminar. 2. El sistema elimina al empleado de la base de datos. 3. Se muestra un mensaje de confirmación.
Postcondición	El empleado ya no figura en el sistema.

CU21-Buscar Empleado	
Actor	Empleado
Descripción	Permite buscar un empleado específico a través de su DNI.
Precondición	El empleado debe estar autenticado.
Flujo Principal	<ol style="list-style-type: none"> 1. El empleado introduce el DNI del empleado en el campo de búsqueda. 2. El sistema consulta la base de datos con ese DNI. 3. Si el empleado existe, se muestra en la lista de resultados. 4. Si no existe, el sistema muestra un mensaje de advertencia.
Postcondición	Se muestra la información del empleado o un mensaje indicando que no existe.

-Entidad Vehículo:

CU22-Alta Vehículo	
Actor	Empleado
Descripción	Permite registrar un nuevo vehículo en el sistema asignándole a un inventario.
Precondición	El empleado debe estar autenticado y debe existir al menos un inventario.
Flujo Principal	<ol style="list-style-type: none"> 1. El empleado accede al formulario de alta de vehículo. 2. El sistema solicita los siguientes datos: matrícula, inventario, marca, modelo, año, precio, estado y disponibilidad. 3. El empleado completa el formulario y lo envía. 4. El sistema valida los datos ingresados. 5. El sistema guarda el nuevo vehículo en la base de datos. 6. Se confirma el alta con un mensaje de éxito.
Postcondición	El vehículo queda registrado en el sistema y disponible según el estado indicado.

CU23-Consultar Vehículo	
Actor	Empleado
Descripción	Permite consultar los datos de un vehículo a partir de su matrícula.
Precondición	El empleado debe estar autenticado y debe existir el vehículo.
Flujo Principal	<ol style="list-style-type: none"> 1. El empleado introduce una matrícula. 2. El sistema consulta la base de datos con la matrícula ingresada. 3. Si el vehículo existe, se muestran sus datos. 4. Si no, se muestra un mensaje de error.
Postcondición	Se muestran los datos del vehículo consultado o un mensaje indicando que no existe.

CU24-Buscar Vehículo	
Actor	Empleado
Descripción	Permite buscar un vehículo específico mediante su matrícula y mostrarlo en una lista.
Precondición	El empleado debe estar autenticado.
Flujo Principal	<ol style="list-style-type: none"> 1. El empleado introduce una matrícula. 2. El sistema valida el formato de la matrícula. 3. El sistema consulta el vehículo correspondiente. 4. Se muestra el vehículo encontrado en una lista.
Postcondición	El vehículo buscado aparece listado si existe.

CU25-Modificar Vehículo	
Actor	Empleado
Descripción	Permite buscar un vehículo específico mediante su matrícula y mostrarlo en una lista.
Precondición	El vehículo debe existir en el sistema.
Flujo Principal	<ol style="list-style-type: none"> 1. El empleado accede al formulario de edición del vehículo. 2. El sistema carga los datos actuales del vehículo. 3. El empleado realiza los cambios y los guarda. 4. El sistema valida los datos modificados. 5. El sistema actualiza la información en la base de datos. 6. Se muestra un mensaje de confirmación.
Postcondición	Los nuevos datos del vehículo quedan registrados en el sistema.

CU26-Eliminar Vehículo	
Actor	Empleado
Descripción	Permite eliminar un vehículo registrado en el sistema.
Precondición	Debe seleccionarse un vehículo existente.
Flujo Principal	<ol style="list-style-type: none"> 1. El empleado selecciona un vehículo e inicia la acción de eliminar. 2. El sistema valida la matrícula. 3. El sistema elimina el vehículo de la base de datos. 4. Se muestra un mensaje confirmando la eliminación.
Postcondición	El vehículo ya no está disponible en el sistema.

-Entidad Transacción:

CU27-Alta Transacción	
Actor	Empleado
Descripción	Permite registrar una nueva transacción económica dentro del sistema.
Precondición	El empleado debe estar autenticado.
Flujo Principal	<ol style="list-style-type: none">1. El empleado accede al formulario de alta de transacción.2. El sistema solicita datos como fecha de transacción, precio, método de pago y estado.3. El empleado completa y envía el formulario.4. El sistema valida los datos ingresados.5. El sistema registra la transacción en la base de datos.6. Se muestra una confirmación del alta.
Postcondición	La transacción queda registrada en el sistema.

CU28 - Consultar Transacción	
Actor	Empleado
Descripción	Permite visualizar los detalles de una transacción específica mediante su ID.
Precondición	El empleado debe estar autenticado.
Flujo Principal	<ol style="list-style-type: none">1. El empleado accede a la opción de consulta de transacción.2. El sistema solicita el ID de la transacción.3. El sistema recupera y muestra los datos si la transacción existe.
Postcondición	Se muestran los datos de la transacción o un mensaje indicando que no existe.

CU29 - Buscar Transacción	
Actor	Empleado
Descripción	Permite buscar una transacción por su ID y mostrarla en una lista.
Precondición	El empleado debe estar autenticado.
Flujo Principal	<ol style="list-style-type: none"> 1. El empleado introduce el ID de la transacción en el campo de búsqueda. 2. El sistema busca la transacción correspondiente. 3. Si la transacción existe, se muestra en la lista. 4. Si no existe, se muestra un mensaje de advertencia.
Postcondición	Se muestra la transacción buscada o un mensaje indicando que no existe.

CU30 - Modificar Transacción	
Actor	Empleado
Descripción	Permite modificar los datos de una transacción existente.
Precondición	La transacción debe existir en el sistema.
Flujo Principal	<ol style="list-style-type: none"> 1. El empleado accede a la opción de modificar transacción. 2. El sistema carga los datos actuales. 3. El empleado realiza los cambios necesarios. 4. El sistema valida los nuevos datos. 5. El sistema actualiza los datos en la base de datos. 6. Se muestra un mensaje de confirmación.
Postcondición	Los datos de la transacción quedan actualizados.

CU31 - Eliminar Transacción	
Actor	Empleado
Descripción	Permite eliminar una transacción registrada en el sistema.
Precondición	El empleado debe estar autenticado y la transacción debe existir.
Flujo Principal	<ol style="list-style-type: none"> 1. El empleado selecciona una transacción para eliminarla. 2. El sistema verifica su existencia. 3. El sistema elimina la transacción. 4. Se muestra un mensaje confirmando la eliminación.
Postcondición	La transacción ya no figura en el sistema.

-Entidad Pago:

CU32 - Alta Pago	
Actor	Empleado
Descripción	Permite registrar un nuevo pago asociado a una reserva existente.
Precondición	El empleado debe estar autenticado y debe existir una reserva válida.
Flujo Principal	<ol style="list-style-type: none"> 1. El empleado accede al formulario de alta de pago. 2. El sistema solicita los datos requeridos: ID de reserva, fecha de pago, precio total, método de pago y estado. 3. El empleado completa el formulario y lo envía. 4. El sistema valida los datos ingresados (formato de fecha, método de pago, etc.). 5. El sistema guarda el pago en la base de datos. 6. El sistema genera automáticamente un informe PDF del pago. 7. El sistema muestra un mensaje de éxito.
Postcondición	El pago queda registrado correctamente en el sistema y asociado a una reserva.

CU33 - Consultar Pago	
Actor	Empleado
Descripción	Permite consultar los detalles de un pago específico mediante su ID.
Precondición	El empleado debe estar autenticado y el ID del pago debe existir.
Flujo Principal	<ol style="list-style-type: none"> 1. El empleado accede al módulo de consulta de pagos. 2. El sistema solicita el ID del pago. 3. El sistema recupera y muestra los datos del pago si existe.
Postcondición	El sistema muestra la información detallada del pago consultado.

CU34 - Modificar Pago	
Actor	Empleado
Descripción	Permite modificar los datos de un pago ya existente.
Precondición	El empleado debe estar autenticado y el pago debe estar previamente registrado.
Flujo Principal	<ol style="list-style-type: none"> 1. El empleado accede a la opción para modificar pagos. 2. El sistema carga los datos actuales del pago. 3. El empleado modifica los campos permitidos (fecha, precio, método, estado). 4. El sistema valida los datos actualizados. 5. El sistema guarda los cambios en la base de datos. 6. El sistema muestra un mensaje de confirmación.
Postcondición	Los datos del pago se actualizan correctamente en el sistema.

CU35 - Eliminar Pago	
Actor	Empleado
Descripción	Permite eliminar un pago registrado en el sistema.
Precondición	El empleado debe estar autenticado y debe indicar un ID de pago válido.
Flujo Principal	<ol style="list-style-type: none"> 1. El empleado selecciona un pago para eliminarlo. 2. El sistema verifica la existencia del ID de pago. 3. El sistema elimina el pago de la base de datos. 4. El sistema muestra un mensaje de confirmación.
Postcondición	El pago queda eliminado y ya no figura en el sistema.

CU36 - Buscar Pago	
Actor	Empleado
Descripción	Permite buscar un pago por su ID y mostrarlo en una lista.
Precondición	El empleado debe estar autenticado.
Flujo Principal	<ol style="list-style-type: none"> 1. El empleado introduce el ID del pago en el campo de búsqueda. 2. El sistema busca el pago correspondiente. 3. Si el pago existe, se muestra en la lista. 4. Si no existe, se muestra un mensaje de advertencia.
Postcondición	Se muestra el pago buscada o un mensaje indicando que no existe.

-Entidad Inventario:

CU37 - Alta Inventario	
Actor	Empleado
Descripción	Permite registrar un nuevo inventario en el sistema y asignarlo a un concesionario.
Precondición	El empleado debe estar autenticado. El concesionario no debe tener ya un inventario asociado. El empleado responsable y la fecha de actualización deben ser válidos.
Flujo Principal	<ol style="list-style-type: none">1. El empleado accede al formulario de alta de inventario.2. El sistema solicita los siguientes datos: ID del concesionario, DNI del empleado responsable y fecha de última actualización.3. El empleado completa el formulario y lo envía.4. El sistema valida que el concesionario exista, que no tenga ya un inventario y que el empleado también exista.5. El sistema valida el formato de la fecha.6. El sistema registra el inventario en la base de datos.7. El sistema confirma el registro con un mensaje de éxito.
Postcondición	El inventario queda registrado y vinculado al concesionario correspondiente.

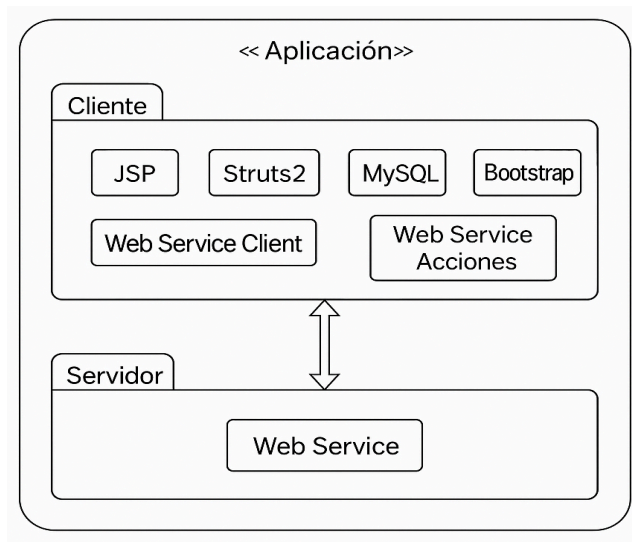
CU38 - Consultar Inventario	
Actor	Empleado
Descripción	Permite consultar los datos de un inventario específico mediante su ID.
Precondición	El empleado debe estar autenticado.
Flujo Principal	<ol style="list-style-type: none"> 1. El empleado accede al módulo de consulta de inventario. 2. El sistema solicita el ID del inventario. 3. El sistema busca en la base de datos y muestra los datos si existe.
Postcondición	Se muestran los datos del inventario o un mensaje indicando que no existe.

CU39 - Buscar Inventario	
Actor	Empleado
Descripción	Permite buscar un inventario por su ID para mostrarlo en una lista.
Precondición	El empleado debe estar autenticado.
Flujo Principal	<ol style="list-style-type: none"> 1. El empleado introduce un ID de inventario en el campo de búsqueda. 2. El sistema busca el inventario correspondiente. 3. Si existe, lo muestra en una lista. 4. Si no existe, se muestra un mensaje de advertencia.
Postcondición	El sistema muestra el inventario si existe o un mensaje indicando su ausencia.

CU40 - Modificar Inventario	
Actor	Empleado
Descripción	Permite modificar los datos de un inventario existente.
Precondición	El inventario debe existir. El concesionario y el empleado deben ser válidos.
Flujo Principal	<ol style="list-style-type: none"> 1. El empleado accede al formulario de modificación. 2. El sistema carga los datos actuales del inventario. 3. El empleado realiza los cambios necesarios: concesionario, empleado y fecha. 4. El sistema valida los nuevos datos y relaciones. 5. El sistema actualiza el inventario en la base de datos. 6. El sistema muestra un mensaje de confirmación.
Postcondición	El inventario queda actualizado con los nuevos datos.

CU41 - Eliminar Inventario	
Actor	Empleado
Descripción	Permite eliminar un inventario del sistema.
Precondición	El empleado debe estar autenticado y el inventario debe existir.
Flujo Principal	<ol style="list-style-type: none"> 1. El empleado selecciona un inventario para eliminarlo. 2. El sistema verifica que el ID es válido y que el inventario existe. 3. El sistema elimina el inventario de la base de datos. 4. El sistema muestra un mensaje confirmando la eliminación.
Postcondición	El inventario ya no figura en el sistema.

4. Esquema Funcional



Para hacer funcionar nuestra aplicación **UPOCarAuto**, nos hemos basado en el patrón MVC, que básicamente separa cada parte de la app por su función: una parte se encarga de mostrar cosas al usuario, otra de controlar lo que pasa y otra de hablar con la base de datos.

·Cliente

Esta es la parte de la app que usa el empleado. Aquí es donde se hacen cosas como registrar un cliente, añadir un vehículo, consultar una reserva o eliminar un pago. Está formada por varios componentes:

- **JSP:** Son las páginas que el empleado ve. Ahí mezclamos HTML con Java para mostrar cosas dinámicas como formularios, tablas o mensajes de error.
- **Struts2:** Es el “cerebro” que se encarga de recibir las acciones del usuario (como hacer clic en un botón), procesarlas y decidir qué hacer. Gracias a este framework, organizamos bien el código y todo está más limpio.
- **Bootstrap:** Usamos esta herramienta para que la aplicación tenga un diseño moderno, con botones, tablas y formularios bonitos sin tener que hacer todo el estilo a mano.
- **Acciones:** Son las clases que se activan cuando haces algo, como guardar un cliente o eliminar una reserva. Estas clases se encargan de recoger los datos del formulario, validar que estén bien, y hablar con la base de datos.
- **MySQL:** Es el sistema donde guardamos todos los datos: clientes, empleados, coches, reservas, pagos, etc. Usamos SQL para acceder a esos datos.
- **Web Service Client:** Es una parte que se conecta con un servicio web externo. ¿Para qué? Pues, por ejemplo, cuando se hace un pago, se genera automáticamente un informe en PDF gracias a este servicio.

·Servidor

Esta parte es como el motor que está detrás del sistema. Recibe lo que pide el cliente, lo procesa y da una respuesta.

- **Web Service:** Es un servicio especial que usamos para cosas concretas, como generar documentos PDF de los pagos. Funciona con un estándar llamado SOAP, que permite intercambiar información con otros sistemas aunque estén hechos en diferentes lenguajes.

En resumen, hemos montado una aplicación dividida por partes para que todo esté organizado. El empleado solo ve lo que necesita y puede hacer todo desde un menú muy intuitivo: gestionar clientes, coches, concesionarios, reservas, pagos y más. Detrás, hay una base de datos bien estructurada y un sistema que se encarga de validar todo, generar informes, y mantenerlo todo funcionando como debe.

5. Diseño de la base de datos

Concesionario		
Descripción:	Representa una sucursal que vende vehículos.	
Relación	Atributo	Descripción
	id_concesionario	Identificador único del concesionario
	nombre	Nombre del Concesionario
	direccion	Dirección física del concesionario
	telefono	Número de referencia del concesionario(único)
	correo	Correo electrónico del concesionario(único)

Empleado		
Descripción:	Representa a una persona que trabaja en un concesionario.	
Relación	Atributo	Descripción
	dni	Identifiador único del empleado
	nombre	Nombre del empleado
	apellidos	Apellidos del empleado
	email	Correo electrónico del empleado(único)
	telefono	Número de teléfono del empleado(único)
	direccion	Dirección de residencia del empleado
	puesto	Cargo o función del empleado dentro del concesionario
	contrasenya	Contraseña del empleado para el acceso
	id_concesionario	Identificador del concesionario donde trabaja el empleado

Vehículo		
Descripción:	Representa un vehículo disponible en el inventario de un concesionario.	
Relación	Atributo	Descripción
	matricula	Matrícula del vehículo. Es única y PK
	id_inventario	Identificador del inventario al que pertenece el vehículo
	marca	Marca del vehículo
	modelo	Modelo específico del vehículo
	anio	Año de fabricación del vehículo
	precio	Precio actual del vehículo
	estado	Estado del vehículo(Nuevo, Usado o Reservado)
	disponibilidad	Indica si el vehículo está disponible(1) o no (0) para ser reservado

Inventario		
Descripción:	Representa el stock de vehículos disponible en un concesionario específico.	
Relación	Atributo	Descripción
	id_inventario	Identificador único de inventario
	id_concesionario	Concesionario al que pertenece este inventario
	dni_empleado	Empleado responsable de la gestión del inventario
	total_vehiculos	Campo calculado que representa al número total de vehículos registrados en el inventario
	ultima_actualizacion	Fecha de la última actualización del inventario

Reserva		
Descripción:	Representa la acción de reservar un vehículo por parte de un cliente.	
Relación	Atributo	Descripción
	id_reserva	Identificador único de la reserva
	dni_cliente	Cliente que realiza la reserva
	matricula	Matrícula del vehículo reservado(único)
	estado	Estado de la reserva(Pendiente, Confirmada o Rechazada)
	fecha_creacion	Fecha en la que se creó la reserva
	fecha_recogida	Fecha planificada para recoger el vehículo, puede ser null

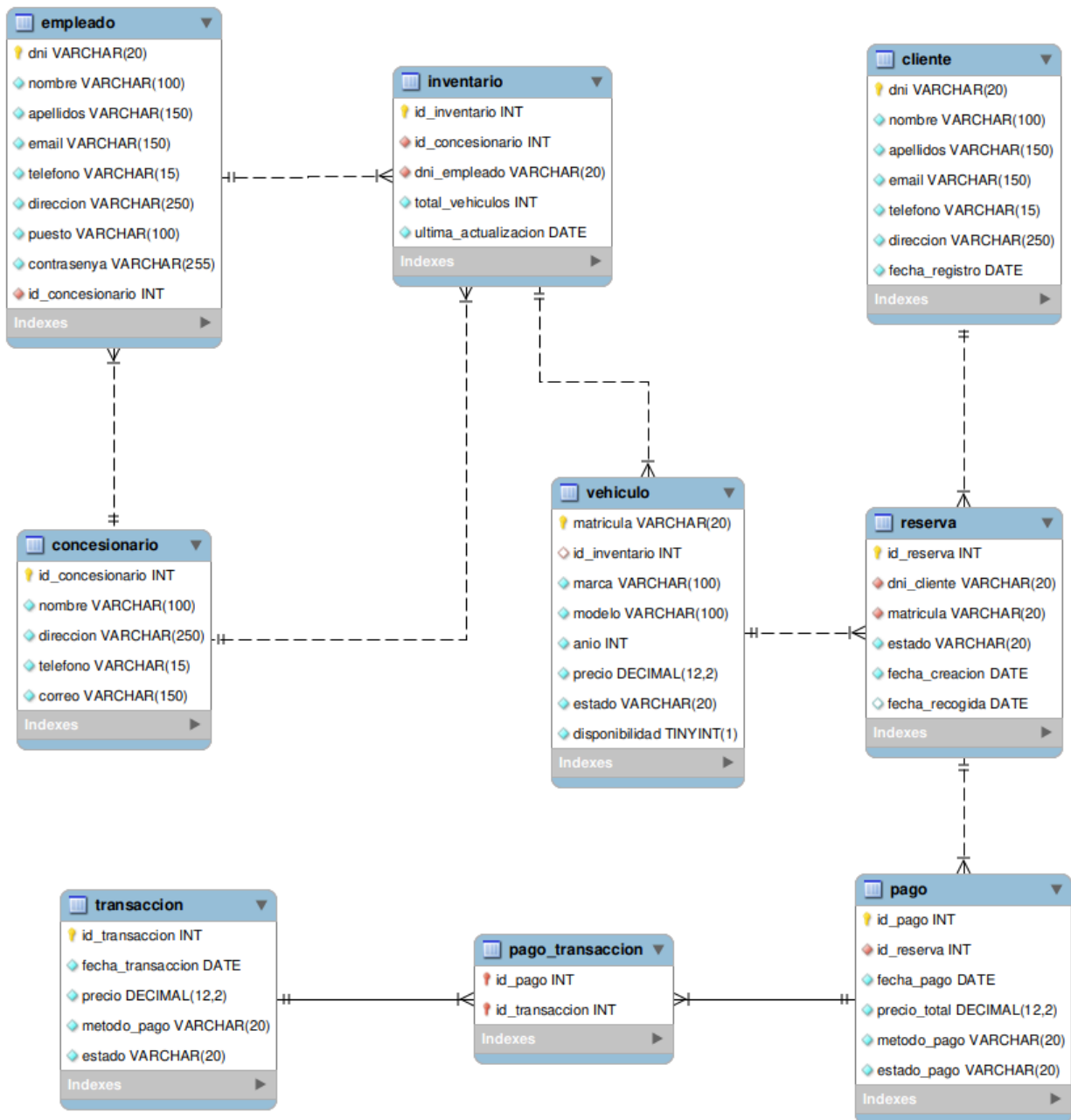
Pago		
Descripción:	Representa un pago asociado a una reserva de vehículo.	
Relación	Atributo	Descripción
	id_pago	Identificador único del pago
	id_reserva	Reserva a la que está asociado el pago
	fecha_pago	Fecha en la que se realizó el pago
	precio_total	Importe total del pago
	metodo_pago	Método de pago usado(Tarjeta, Bizum o Transferencia)
	estado_pago	Estado del pago(Pendiente, Completado, Fallido o Cancelado)

Transacción		
Descripción:	Representa una transacción financiera individual dentro del sistema.	
Relación	Atributos	Descripción
	id_transaccion	Identificador único de la transacción
	fecha_transaccion	Fecha en la que se ejecutó la transacción
	precio	Monto económico de la transacción
	metodo_pago	Medio por el cual se efectuó la transacción
	estado	Estado actual de la transacción(Completada, Pendiente o Cancelada)

Pago_Transaccion		
Descripción:	Representa la relación muchos a muchos entre pagos y transacciones.	
Relación	Atributo	Descripción
	id_pago	Identificador único del pago asociado
	id_transaccion	Identificador de la transacción asociada

Cliente		
Descripción:	Representa a una persona que utiliza el sistema para realizar reservas de vehículos.	
Relación	Atributo	Descripción
	dni	Identificador único del cliente
	nombre	Nombre del cliente
	apellidos	Apellidos del cliente
	email	Correo electrónico del cliente
	telefono	Número de teléfono del cliente
	direccion	Dirección física del cliente
	fecha_registro	Fecha en la que el cliente se registró en el sistema

5.1 Diseño físico de la base de datos



6. Casos de prueba

Durante el desarrollo de la aplicación, se han implementado casos de prueba con el fin de verificar que cada funcionalidad se comporte según lo esperado. Estos casos simulan distintos escenarios de uso, tanto correctos como erróneos, y permiten detectar posibles fallos o comportamientos no deseados.

Los casos de prueba abarcan todas las entidades del sistema (Empleado, Cliente, Pago, Transacción, Concesionario, Inventario, Reserva y Vehículo), contemplando sus respectivas operaciones CRUD (Crear, Leer, Actualizar y Eliminar).

A continuación, se detallan algunos de los aspectos cubiertos durante las pruebas:

- **Validaciones de formularios:** Se ha verificado que los formularios no permitan el envío si existen campos obligatorios vacíos. En dichos casos, el sistema lanza un error tipo `FieldError`, señalando al usuario los campos que deben completarse.
- **Inicio de sesión:** Se han desarrollado pruebas para comprobar que el sistema no permite el acceso si las credenciales (DNI y contraseña) no son válidas. En ese caso, se muestra un mensaje de error informando que los datos son incorrectos.
- **Eliminación de datos:** Antes de eliminar cualquier entidad, el sistema solicita una confirmación al usuario mediante un mensaje emergente, evitando eliminaciones accidentales. Se han realizado pruebas tanto confirmando como cancelando la operación.
- **Validación de formatos:** El sistema comprueba que ciertos campos específicos cumplan con un formato válido, como:
 - **Email:** debe tener un formato estándar `usuario@dominio.com`.
 - **Número de teléfono:** debe contener únicamente números y cumplir con la longitud esperada.
 - **DNI:** debe validarse tanto en longitud como en estructura (letra final incluida).
- **Reservas:** Al realizar una reserva, el formulario incluye selectores que permiten elegir únicamente vehículos disponibles. Se ha verificado que los vehículos ya reservados o no disponibles no se muestren en el listado.

Todos estos casos de prueba han sido documentados junto con los resultados obtenidos, facilitando así el seguimiento de errores, su corrección y la validación de mejoras posteriores.

ID	Módulo	Descripción	Datos de Entrada	Resultado Esperado	Resultado Obtenido	Estado
CP01	Inicio Sesión	Verificar login con credenciales correctas	Usuario: 12345678A Contraseña: Pwd_Luis!2025	Acceso concedido y redirección al panel principal	Acceso concedido y redirección correcta	Éxito
CP02	Inicio Sesión	Verificar login con credenciales incorrectas	Usuario: 23456789B Contraseña: Pwd_Marta!20	Mensaje de error: "DNI o contraseña incorrectos"	Mensaje de error mostrado correctamente	Éxito
CP03	Cliente	Crear cliente con campos vacíos	Nombre vacío	Mensaje de error: "Introduce el nombre"	Error lanzado correctamente	Éxito
CP04	Cliente	Crear cliente con email mal formado	Email: juangmail.com	Mensaje de error: "Introduce un email válido"	Mensaje mostrado correctamente	Éxito
CP05	Empleado	Eliminar empleado con confirmación cancelada	Confirmación: No	No se elimina el empleado, vuelve a la lista de empleados	El empleado no fue eliminado	Éxito
CP06	Empleado	Eliminar empleado con confirmación aceptada	Confirmación: Sí	Empleado eliminado y vuelve a la lista de empleados	Eliminación completada	Éxito
CP07	Cliente	Crear cliente con número de teléfono incorrecto	Teléfono: 12345abc	Mensaje de error: "El teléfono debe tener exactamente 9 dígitos"	Error mostrado correctamente	Éxito
CP08	Reserva	Modificar reserva y cambiar estado a "Rechazada"	Estado: Rechazada	Reserva actualizada y vehículo liberado para disponibilidad	Reserva y disponibilidad actualizadas correctamente	Éxito

7. Manual de uso

La aplicación **UPOCarAuto** está pensada para que cualquier empleado del concesionario pueda gestionar fácilmente todo lo relacionado con vehículos de segunda mano. No está hecha para el cliente, sino para quienes trabajan dentro del sistema: vendedores, administrativos o responsables de concesionarios.

Nada más acceder, ya como empleado autorizado, se presenta un menú general desde el que puedes gestionar clientes, vehículos, reservas, pagos, concesionarios, empleados, inventarios y transacciones.

Por ejemplo, si entra un nuevo cliente, simplemente vas al apartado de clientes, pulsas “Registrar cliente”, rellenas los datos básicos (DNI, nombre, dirección, etc.) y listo. Desde esa misma sección puedes buscar clientes por DNI, verlos en una lista, consultar sus datos, actualizarlos si cambian de teléfono o dirección, o incluso eliminarlos si ya no están activos.

Lo mismo ocurre con los empleados: si entra uno nuevo, lo das de alta con su concesionario asignado y todos sus datos. Puedes consultarlos, editarlos o darlos de baja desde su propio módulo.

En cuanto a los concesionarios, el funcionamiento es igual de simple. Puedes añadir uno nuevo con sus datos de contacto, ver la lista completa, editar cualquier dato si ha cambiado o eliminar uno si cierra.

El inventario de vehículos está asociado a cada concesionario. Solo puede haber uno por sede, y dentro de él se registran todos los vehículos disponibles. Puedes crear inventarios nuevos, verlos, editarlos (como cambiar la persona responsable o la fecha de actualización) o eliminarlos si ya no hacen falta.

Una vez tengas vehículos en el inventario, gestionarlos es muy intuitivo. Desde la sección de vehículos puedes dar de alta uno nuevo indicando su matrícula, marca, año, precio, etc., y asociarlo al inventario que corresponda. También puedes modificarlo, eliminarlo (lo cual borra también sus reservas y pagos), o buscarlo directamente por matrícula.

Con los vehículos listos, puedes empezar a registrar reservas para los clientes interesados. Solo tienes que indicar qué cliente y qué coche están implicados, establecer la fecha de recogida y guardar. A partir de ahí, podrás modificar la reserva (por ejemplo si cambia la fecha), rechazarla si no se va a hacer, o consultarla cuando quieras.

Cuando una reserva se confirma, viene la parte del pago. En el módulo de pagos puedes crear uno nuevo indicando cuánto se ha pagado, cómo (tarjeta, efectivo, etc.), y su estado (completado, pendiente...). Una vez guardado, automáticamente se genera un informe en PDF con los datos del pago, gracias a un servicio web que se conecta al sistema. Además, puedes consultar pagos por ID, modificarlos o eliminarlos si ha habido algún error.

También puedes gestionar transacciones financieras por separado. Funcionan de forma parecida a los pagos: se registran con fecha, importe, método y estado, y luego puedes consultarlas, actualizarlas o eliminarlas si es necesario.

En toda la app se ha intentado mantener la experiencia lo más fluida posible. Hay buscadores por ID, DNI o matrícula en casi todas las secciones, mensajes claros cuando algo se hace bien o hay errores, y los formularios vienen ya rellenos cuando entras a modificar algo.

En resumen, UPOCarAuto busca que cualquier empleado del concesionario pueda gestionar toda la operativa diaria de manera rápida y clara, sin complicaciones.

8 Alojamiento de la aplicación en un servidor

8.1 Servicio de alojamiento en la nube elegido.

El alojamiento elegido para la aplicación se realiza en un servidor físico dedicado, propiedad de uno de los miembros del grupo, ubicado en su red local privada. Esta decisión se tomó para aprovechar los recursos disponibles y tener control total sobre el entorno de despliegue.

El servidor utilizado es un ordenador personal con un procesador AMD Ryzen 5 5200X, equipado con una considerable cantidad de memoria RAM (48 GB) y una rápida unidad de almacenamiento SSD de 1 TB. Estas especificaciones superan ampliamente las de una instancia de nube básica, proporcionando un entorno con recursos de sobra para ejecutar la aplicación, la base de datos y cualquier otro servicio auxiliar necesario.

La conexión a internet del servidor se realiza a través de una conexión de fibra óptica simétrica de 1 Gigabit, lo que garantiza un ancho de banda elevado tanto de subida como de bajada, crucial para el acceso rápido y fluido de los usuarios a la aplicación desplegada.

El sistema operativo instalado es Linux, concretamente la versión **Ubuntu 24.04 LTS (Long Term Support)**. Esta elección se basa en su estabilidad, seguridad, gratuidad y amplio soporte comunitario, además de ser una plataforma estándar en el despliegue de aplicaciones web a nivel profesional.

8.2 Software instalado en la en el servidor:

El software instalado en el servidor y necesario para que la aplicación funcione es el siguiente:

- Entorno de Java (JDK 8u281)
- Servicios XAMPP (solo se usa MySQL Server)
- Servidor de aplicaciones GlassFish 4.1.1

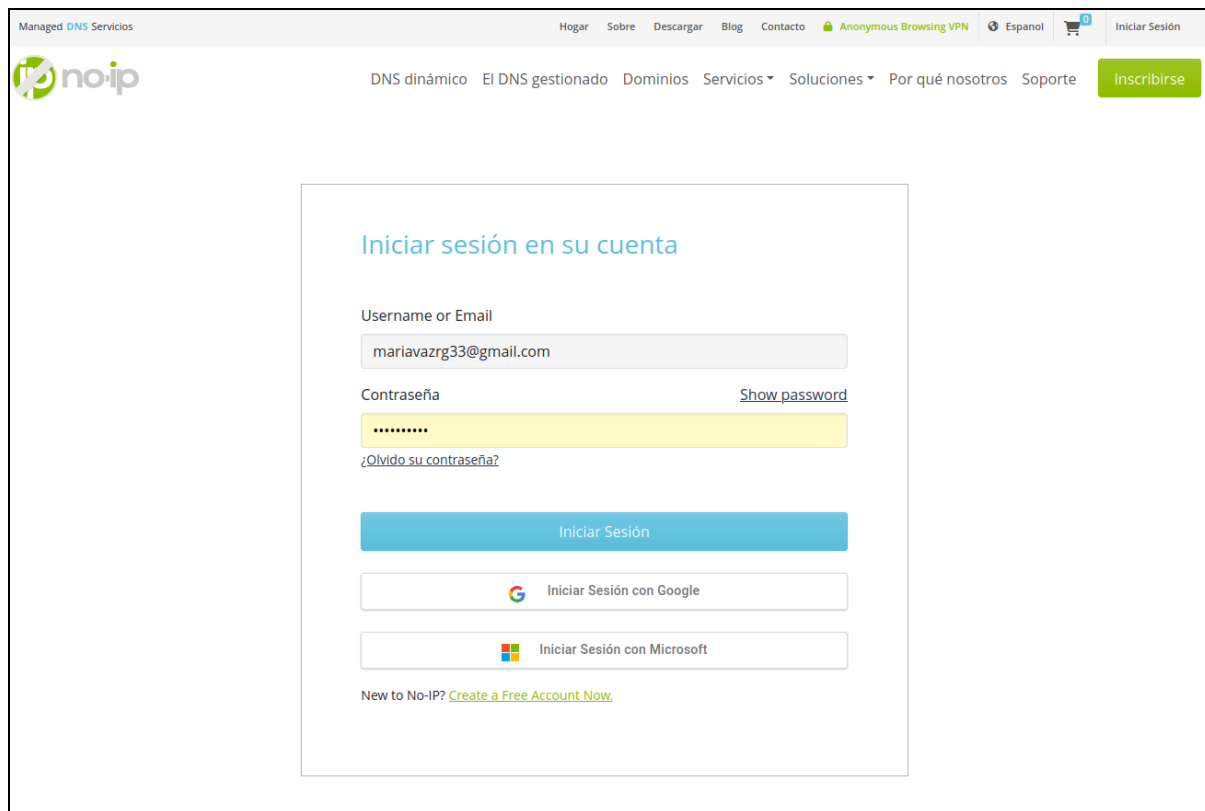
La instalación de este software (JDK, MySQL Server de XAMPP, GlassFish) y su configuración se ha llevado a cabo enteramente mediante **línea de comandos**, usando una terminal. Para automatizar y replicar eficientemente este proceso, se ha ejecutado una **serie de scripts programados en bash**. Estos scripts no solo gestionan la instalación desatendida de los componentes listados, sino que también realizan las configuraciones post-instalación, incluyendo los pasos necesarios para configurar el servidor de aplicaciones **GlassFish** para habilitar el soporte **SSL** y aceptar conexiones seguras a través del protocolo HTTPS.

Las versiones de Java (jdk-8u281) y GlassFish (4.1.1) elegidas son las utilizadas por Netbeans 8.2, asegurando así la compatibilidad del entorno de desarrollo. En el caso de XAMPP (v8.2.12-0), se seleccionó la última versión estable disponible en el momento de la instalación, aunque únicamente se utiliza el componente MySQL Server para la base de datos.

8.3 Configuración de un dominio dinámico para el servidor.

Con el objeto de no tener que indicar la ip en la url para acceder a la aplicación se ha configurado un dominio dinámico en no-ip.com que asocia la ip dinámica del servidor con el dominio gratuito upocarauto.sytes.net.

Lo primero que hay que hacer es crear una cuenta en no-ip.com.



The screenshot shows the login page of no-ip.com. The header includes navigation links: 'Hogar', 'Sobre', 'Descargar', 'Blog', 'Contacto', 'Anonymous Browsing VPN', 'Espanol', and a shopping cart icon. The main navigation bar contains links for 'DNS dinámico', 'El DNS gestionado', 'Dominios', 'Servicios', 'Soluciones', 'Por qué nosotros', and 'Soporte', along with an 'Inscribirse' button. The central content area is titled 'Iniciar sesión en su cuenta' and contains a login form. The form has two input fields: 'Username or Email' (containing 'mariavazrg33@gmail.com') and 'Contraseña' (masked with dots). A 'Show password' link is next to the password field. Below the fields is a blue 'Iniciar Sesión' button. Further down are two buttons for social login: 'Iniciar Sesión con Google' and 'Iniciar Sesión con Microsoft'. At the bottom, there is a link for new users: 'New to No-IP? Create a Free Account Now.'

A continuación se crea el dominio dinámico **upocarauto.sytes.net**

New customers get 25% off Enhanced or Pro Dynamic DNS, but hurry - this offer expires in 7 days! Use code **NEW25** at checkout!

Espanol

Tablero de control

DDNS y acceso remoto

Nombres de host sin IP

Nombres de host personales

Claves DDNS

Cliente de actualización dinámica

Actualizar clientes

Eliminar asistente de configuración

DNS administrado

Dominios

Correo electrónico

Monitoreo de servidor

Nombres de host

My No-IP > Nombres de host

La configuración dinámica de DNS requiere la creación de DDNS Key.

Crear nombre de host

Buscar...

Nombre de host	Last Update	IP / Objetivo	Type	DDNS Key
upocarauto.sytes.net	Jun 1, 2025 09:21 PDT	79.116.25.8	A	Create DDNS Key

Ayuda con nombres de host

Configurar su nombre de host de No-IP

Managed DNS API

Para asociar la ip del router al dominio dinámico creado en no-ip hay que configurarlo para que comunique a No-ip cualquier cambio en la ip usando las credenciales de acceso a no-ip.

DIGI

Hora actual: 2025-06-02T08:38

user Cerrar sesión Español

Inicio Topología Internet Red local Gestión y diagnóstico

Estado

Seguridad

Control parental

DDNS

SNTP

Información de la página

Esta página proporciona la función de configuración de los parámetros DDNS.

DDNS

Proveedor: No-IP

DDNS: ☒ Activado ☐ Desactivado

URL del proveedor: http://www.no-ip.com

Nombre de usuario: mariavazrg33

Contraseña:

Nombre de host: upocarauto.sytes.net

Aplicar Cancelar

Seguidamente se configura el router para redirigir el puerto 443 (https) al 8181 y permitir conexiones entrantes https dirigidas al servidor glashfish ubicado en la máquina con ip local 192.168.1.23

The screenshot shows the Digi router's web interface. The top navigation bar includes 'Inicio', 'Topología', 'Internet', 'Red local', and 'Gestión y diagnóstico'. The 'Internet' tab is selected, and the 'Port Forwarding' sub-tab is active. On the left, a sidebar lists 'Estado', 'Seguridad', 'Control parental', 'DDNS', and 'SNTP'. The main content area is titled 'Información de la página' and 'Port Forwarding'. It includes a table of existing rules and a detailed configuration form for the 'https_upo' rule.

Nombre	Estado	Acción
ssh_2299	Activado	[Icono de borrar]
cam_tcp	Activado	[Icono de borrar]
cam_udp	Activado	[Icono de borrar]
rdp_server	Activado	[Icono de borrar]
https_upo	Activado	[Icono de borrar]

Configuración de https_upo:

- Nombre: https_upo
- Protocolo: TCP
- Dirección IP del host WAN: 0.0.0.0 ~ 0.0.0.0
- LAN Host: 192.168.1.23
- Puerto WAN: 443 ~ 443
- Puerto de host LAN: 8181 ~ 8181

Botones: [Aplicar] [Cancelar]

[+] Crear elemento nuevo

8.4 Proceso de instalación y configuración

A continuación, se describe el proceso de instalación del servidor GlassFish. Este se abordó desarrollando una serie de scripts para automatizar cada fase. Estos scripts no solo se han usado para la instalación inicial, sino que han sido fundamentales para realizar pruebas exhaustivas, instalando y desinstalando componentes y servicios repetidamente para afinar el procedimiento. Gracias a esta automatización, poner en marcha y configurar el servidor ha resultado mucho más fácil. El proceso cubre desde la instalación de sus diversos elementos y servicios hasta el despliegue de las aplicaciones web (.war), que también queda totalmente integrado. Los siguientes apartados muestran el proceso completo.

8.5 Instalación y configuración del entorno de java jdk-8u281-linux-x64.

En primer lugar se realiza la instalación de jdk-8u281-linux-x64, automatizada mediante el script `install_jdk-8u281-linux.sh`. Dicho script se encarga de extraer el JDK en `/usr/lib/jvm`, configurar las variables de entorno `JAVA_HOME` y `PATH` en el `.bashrc` del usuario, y establecer esta versión como la predeterminada del sistema utilizando `update-alternatives`. El script también verifica la existencia previa del JDK y los permisos necesarios para su ejecución.

```
sudo ./install_jdk-8u281-linux.sh
```

8.6 Instalación de xampp-linux-x64-8.0.28-0

Seguidamente, se abordó la instalación de XAMPP (versión 8.2.12). El script utilizado es **`install_xampp.sh`** y automatiza la descarga del instalador, su ejecución y la configuración necesaria para que XAMPP funcione como un servicio del sistema. Es importante precisar que, de todos los componentes incluidos en XAMPP, únicamente se desplegará y utilizará el servidor MySQL. Esta versión específica de XAMPP (y por ende, de MySQL) fue seleccionada por ser la utilizada y requerida en el contexto de la asignatura.

```
sudo ./install_xampp.sh
```

El fichero `xampp.service` sirve para configurar el arranque del servidor de base de datos `mysql` y que pueda ser iniciado en el arranque del ordenador.

```
[Unit]  
Description=XAMPP  
After = syslog.target network.target  
  
[Service]  
ExecStart=/opt/lampp/bin/mysql.server start  
ExecStop=/opt/lampp/bin/mysql.server stop  
Type=forking  
  
[Install]  
WantedBy=multi-user.target
```

El comando que ejecuta el script para habilitar el servicio en el arranque que es:

```
sudo systemctl enable xampp.service
```

Una vez instalado mysql hay que importar la base de datos desde un fichero .sql de la siguiente forma:

```
/opt/lampp/bin/mysql -u root < upocarauto.sql
```

El comando anterior creará la base de datos upocarauto, las tablas correspondientes e insertará datos para las pruebas.

8.7 Instalación de glassfish 4.1.1

Tras instalar el servidor de base de datos mysql de xampp, se inicia la instalación del servidor **GlassFish 4.1.1**. Este proceso también ha sido automatizado mediante el script **install_glassfish4.sh** que se encarga de varios pasos clave. Inicialmente, el script descarga el archivo de instalación (o usando uno local) y lo extrae en el directorio /opt. Posteriormente, crea el dominio por defecto 'domain1', configura las variables de entorno GLASSFISH_HOME y añade su ruta al PATH en el .bashrc del usuario, y finalmente, establece GlassFish para ejecutarse como un servicio del sistema a través de systemd.

```
sudo ./install_glassfish4.sh
```

El fichero de servicio glassfish4.service es el siguiente:

```
[Unit]  
Description = GlassFish Server v4.1.1  
After = syslog.target network.target  
  
[Service]  
ExecStart=/opt/glassfish4/bin/asadmin start-domain  
ExecReload=/opt/glassfish4/bin/asadmin restart-domain  
ExecStop=/opt/glassfish4/bin/asadmin stop-domain  
Type = forking  
  
[Install]  
WantedBy = multi-user.target
```

El comando que ejecuta el script para habilitar el servicio en el arranque que es:

```
sudo systemctl enable glassfish4.service
```

8.8 Configurar ssl para acceder mediante https

Seguidamente, se configura el acceso seguro mediante HTTPS. El script `setup_ssl_server.sh` automatiza la creación de un certificado SSL autofirmado y su instalación en el keystore de GlassFish. Configura el listener HTTPS para utilizar este certificado y deshabilita el listener HTTP por defecto, forzando así que las conexiones se realicen a través del protocolo seguro https.

```
sudo ./setup_ssl_server.sh
```

8.9 Despliegue de las aplicación y de los servicios

Finalmente, se procede al despliegue de la aplicación principal (`UPOCarAuto.war`) y de dos servicios web (`UPOCarAuto_InformeWS.war` y `UPOCarAuto_NotificacionWS.war`) mediante el script `deploy.sh`. Éste se encarga de verificar si el servidor GlassFish está activo y, si no lo está, lo inicia. Seguidamente, usa el comando **asadmin deploy --force** para desplegar o actualizar cada uno de los tres archivos `.war` proporcionados, asegurando que las aplicaciones y servicios web queden disponibles en el servidor.

```
sudo ./deploy.sh UPOCarAuto_InformeWS.war UPOCarAuto_NotificacionWS.war UPOCarAuto.war
```

8.10 Referencias

Glassfish 4.1.1. <https://download.oracle.com/glassfish/4.1.1/release/index.html>

Cómo instalar glassfish en Ubuntu 20.04

<https://orcacore.com/set-up-glassfish-ubuntu-20-04/>

Despliegue de aplicaciones glassfish usando comandos

<https://docs.oracle.com/cd/E19798-01/821-1757/ghgjj/index.html>

Línea de comando de XAMPP mysql

<https://www.delftstack.com/howto/mysql/xampp-mysql-command-line/>

Preguntas frecuentes de XAMPP https://www.apachefriends.org/faq_linux.html

Configurar servicio para noip2

<https://askubuntu.com/questions/1089704/cant-get-service-noip2-to-start-on-boot>

<https://www.noip.com/support/knowledgebase/installing-the-linux-dynamic-update-client-on-ubuntu/>

Configurar servicio XAMPP

<https://www.computernetworkingnotes.com/linux-tutorials/auto-start-xampp-at-startup-in-ubuntu-linux.html>

Url directa a un servidor glassfish

<https://stackoverflow.com/questions/16094858/direct-a-url-directly-to-a-glassfish-application-in-a-virtual-server>