



Práctica Obligatoria

Plataforma web de Gestión de Flotas de Vehículos Eléctricos

Objetivo

El objetivo de esta práctica es diseñar y desarrollar una aplicación web empresarial para la gestión integral de una flota de vehículos eléctricos compartidos. La plataforma debe permitir a una organización controlar la asignación, disponibilidad y uso eficiente de los vehículos, garantizando su accesibilidad a los empleados y ofreciendo a los administradores un sistema de control avanzado.

El sistema deberá integrar:

- Procesos de reserva y devolución, con validaciones de autonomía y disponibilidad.
- Gestión de usuarios y roles diferenciados (empleados, administradores).
- Carga inicial de datos desde JSON y persistencia en una base de datos relacional.
- Configuraciones de accesibilidad que aseguren un uso inclusivo de la plataforma.
- Módulos de análisis y estadísticas (uso de vehículos, kilómetros recorridos, horarios de mayor demanda), útiles para la toma de decisiones empresariales en materia de movilidad sostenible.

Con este proyecto se busca simular una aplicación de uso real en un contexto corporativo actual, donde las empresas incorporan soluciones de movilidad sostenible y optimización de recursos.

Requisitos generales

- Frontend: HTML5, CSS3, Bootstrap (obligatorio).
- Backend: Node.js + Express.js.
- Base de datos relacional MySQL, con tablas bien definidas y normalizadas.
- Carga de datos desde JSON:
 - Si la BD está vacía, al iniciar la aplicación debe solicitarse cargar un fichero JSON con vehículos y concesionarios.
 - El sistema debe permitir cargas posteriores: si el vehículo ya existe (misma matrícula), se pedirá confirmación para actualizarlo.
 - Se mostrarán logs con vehículos añadidos y actualizados.
- Roles de usuario:
 - Empleado: asignado a un concesionario, puede reservar vehículos, consultar histórico y filtrar solo los vehículos de su base.
 - Administrador: puede dar de alta/baja vehículos, crear/editar concesionarios, consultar estadísticas globales.
- Accesibilidad obligatoria: se detallan más adelante.



- Defensa oral obligatoria: el alumno debe explicar el código, la carga desde JSON, la gestión de la BD y demostrar accesibilidad en directo.

Esquema de la Base de Datos

Tabla Usuarios

- id_usuario (PK)
- nombre
- correo (único, validación regex corporativo)
- contraseña (encriptada)
- rol (empleado/admin)
- teléfono (opcional)
- id_concesionario (FK → concesionarios)
- preferencias_accesibilidad (JSON o campos separados)

Tabla Concesionarios

- id_concesionario (PK)
- nombre
- ciudad
- dirección
- teléfono_contacto

Tabla Vehículos

- id_vehiculo (PK)
- matrícula (única)
- marca
- modelo
- año_matriculación
- número_plazas
- autonomía_km
- color
- imagen (ruta/URL)
- estado (disponible/reservado/mantenimiento)
- id_concesionario (FK → concesionarios)



Tabla Reservas

- id_reserva (PK)
- id_usuario (FK)
- id_vehiculo (FK)
- fecha_inicio
- fecha_fin
- estado (activa/finalizada/cancelada)
- kilómetros_recorridos (opcional)
- incidencias_reportadas (opcional)

Opciones de accesibilidad

Todas las funcionalidades de accesibilidad deben ser accesibles desde un menú contextual (modal Bootstrap) en la barra de navegación. Se espera como mínimo lo siguiente:

1. Cambio de contraste y paleta de colores
 - Modo alto contraste (fondo oscuro / texto claro).
 - Paleta adaptada a daltonismo.
 - Implementación: conmutar clases CSS en <body>.
 - Guardar preferencia en sesión y, en nivel avanzado, en BD.
2. Tamaño de fuente ajustable
 - Opciones: Pequeño (80%), Normal (100%), Grande (120%).
 - Se debe aplicar a toda la aplicación.
 - Guardar preferencia en sesión/BD.
3. Navegación completa por teclado
 - Todos los elementos interactivos deben ser accesibles con **Tab**.
 - Orden lógico de tabulación.
 - Nivel avanzado: configuración personalizada del orden de tabulación.
4. Atajos de teclado configurables
 - Ejemplos: **Ctrl+R** → Reservar vehículo, **Ctrl+H** → Mis reservas.
 - Nivel mínimo: atajos predefinidos.
 - Nivel avanzado: configuración por usuario en el modal, guardada en BD.
5. Acciones de ratón replicables con teclado
 - Los menús desplegados deben poder abrirse con Enter o Espacio.
 - Confirmación de reservas también accesible con teclado.
6. Persistencia de preferencias
 - Nivel intermedio: guardar en sesión.



- Nivel avanzado: guardar en BD asociada al usuario.
- 7. Compatibilidad con lectores de pantalla
 - Uso de etiquetas semánticas (**<main>**, **<nav>**, **<header>**).
 - Uso de atributos ARIA (**aria-label**, **aria-expanded**, **role**).
 - Ejemplo: botón de reservar → **aria-label="Reservar vehículo Tesla Model 3 matrícula 1234-ABC"**.

Menús de la aplicación

Menú de navegación (barra superior)

- Inicio: información general.
- Vehículos disponibles: listado filtrable.
- Reservar vehículo: formulario accesible.
- Perfil de usuario: visible tras login.
- Accesibilidad: abre el modal con opciones anteriores.
- Administración (solo admin): panel de gestión de vehículos, usuarios y concesionarios.
- Cerrar sesión: disponible tras login.

Panel de administrador

- Gestión de vehículos: alta, baja, edición, carga desde JSON.
- Gestión de concesionarios: crear, editar, eliminar.
- Gestión de usuarios: asignar empleados a una base, modificar roles.
- Estadísticas: uso de flota, kms, incidencias.

Panel de empleado

- Mis reservas: histórico personal.
- Vehículos de mi base: listado filtrado por concesionario asignado.

Formularios de la aplicación

Registro de usuario

- Nombre completo (obligatorio).
- Correo corporativo (validado mediante expresión regular para restringir el dominio, por ejemplo: **^[A-Za-z0-9._%+-]+@ucm.es\$**).
- Contraseña (mín. 8 caracteres, mayúsculas, minúsculas y número).
- Teléfono (opcional).
- Concesionario asignado (selector).



Login

- Correo electrónico.
- Contraseña.

Reserva de vehículo

- Selección de vehículo.
- Fecha/hora inicio y fin (usando componentes de fecha y hora de Bootstrap).
- Confirmación en modal.

Alta/Edición de vehículo (admin)

- Matrícula (única).
- Marca.
- Modelo.
- Año matriculación.
- Plazas.
- Autonomía.
- Color.
- Imagen (validación).
- Concesionario (selector).

Alta/Edición de concesionario (admin)

- Nombre.
- Ciudad.
- Dirección.
- Teléfono.

Desarrollo y Evaluación de la práctica

1. **Funcionalidad:**
La aplicación debe funcionar correctamente, cumpliendo con las especificaciones exigidas.
2. **Accesibilidad:**
Se evaluará la implementación de configuraciones de accesibilidad y que el diseño permita un uso inclusivo mediante el teclado y lectores de pantalla.
3. **Calidad del Código:**
El código debe estar organizado y documentado. Se espera un uso adecuado de comentarios y buenas prácticas de programación.



4. Usabilidad y Diseño:

Se evaluará la usabilidad de la aplicación en dispositivos móviles y de escritorio, con un enfoque en Bootstrap y diseño responsive.

5. Documentación y Defensa:

Incluir una breve guía de usuario sobre el funcionamiento de las configuraciones de accesibilidad y la gestión de eventos. La práctica se debe defender en los laboratorio utilizando sus equipos y **obligatoriamente** tendrán que participar en la defensa los autores de la práctica.

La calificación máxima de la práctica varía en función de las funcionalidades implementadas. En la siguiente tabla se describe el sistema de calificación.

Funcionalidades implementadas	Calificación (máx. 10)
<ul style="list-style-type: none">• Registro/Login/Logout: permitir crear cuentas con correo corporativo validado, iniciar y cerrar sesión de forma segura.• Carga inicial desde JSON: al iniciar con la BD vacía, la aplicación solicitará cargar un fichero JSON con concesionarios y vehículos.• Listado básico de vehículos: mostrar tabla o tarjetas con matrícula, marca, modelo, autonomía, plazas y concesionario.• Reserva de vehículos (empleado): seleccionar vehículo disponible y reservarlo con fecha/hora inicio y fin (usando Bootstrap date/time pickers).• Panel de administrador básico: acceso a un área donde el administrador puede:• Panel de administrador básico con acceso restringido a usuarios con rol admin, que permita:<ul style="list-style-type: none">○ Gestión básica de concesionarios: alta manual de concesionarios y listado.○ Gestión básica de vehículos: alta, edición y baja.○ Visualización general de usuarios registrados (opcional, listado simple sin edición).• Diseño responsive con Bootstrap: obligatorio en todas las pantallas• Accesibilidad básica: cambio de contraste, tamaño de fuente y navegación por teclado, disponibles en un modal accesible.	5
<ul style="list-style-type: none">• Todas las anteriores• Gestión dinámica de disponibilidad: los vehículos reservados no deben mostrarse como disponibles en ese rango.• Filtros avanzados de búsqueda: por autonomía mínima, número de plazas y color.• AJAX: actualización dinámica de listados y reservas sin recargar la página.• Accesibilidad intermedia: preferencias guardadas en sesión.• Estadísticas básicas para administrador: número de reservas por concesionario, vehículo más usado.	8



<ul style="list-style-type: none">• Todas las anteriores• Integración con API de mapas: mostrar concesionarios en mapa y calcular distancia desde la ubicación del usuario.• Alertas automáticas: aviso de autonomía baja y recordatorio de devolución próxima.• Accesibilidad avanzada: atajos de teclado configurables y control del orden de tabulación.• Estadísticas avanzadas: kilómetros totales recorridos, reservas por franjas horarias e incidencias reportadas.• Sistema de feedback de usuarios: puntuación y comentarios sobre los vehículos tras el uso.• Accesibilidad persistente en BD: cargar preferencias de accesibilidad automáticamente al iniciar sesión.	10
--	-----------

Se recuerda que la entrega de una práctica total o parcialmente copiada conllevará la calificación de **NO APTO**. Los casos de copia serán tratados de acuerdo a la **Normativa Oficial de la Facultad de Informática de la UCM**. Para ello se utilizará el software actualizado desarrollado en esta Facultad.

En la corrección se valorará:

- El correcto funcionamiento de la aplicación web (las funcionalidades implementadas).
- El correcto uso de Express.js para estructurar adecuadamente la cadena de peticiones al servidor.
- El diseño de la página web. Cada grupo deberá realizar su propio diseño, que será valorado en cuanto a su usabilidad y al correcto uso de las propiedades CSS.

ADVERTENCIAS

- Si con estos datos de conexión vistos anteriormente no se puede acceder a la base de datos la práctica estará suspensa.
- Si la práctica no sigue TODAS las directrices explicadas tanto de organización de carpetas y ficheros como de nomenclatura de carpetas, tablas, ficheros, etc, explicados en clase, la práctica estará suspensa.
- Si en algunas de las pantallas los enlaces a imágenes están rotos, es decir, no se ven las imágenes, esa funcionalidad no contará como válida, aunque todo lo demás funcione correctamente. El tema de las rutas se considera un fallo bastante significativo
- Se valorará negativamente la no utilización de ventanas modales para el desarrollo global de la práctica, así como la falta de utilización del framework Bootstrap
- Se valorará negativamente la falta de usabilidad y claridad de la aplicación. Para evitarlo se recomienda que los alumnos consulten en los laboratorios o bien en las jornadas prácticas semanales las dudas que puedan surgir antes de seguir caminos que llevan a soluciones poco defendibles.
- Se valorará positivamente que las páginas validen por el 3WC indicándolo con los iconos correspondientes de la 3WC



Entrega de la práctica

- Respetar la fecha límite para la entrega de la práctica (El enlace la para la entrega se cierra a las 23:59)
- En el campus se entregará un fichero **gxx_po.zip** (xx es el número de grupo de laboratorio) con todos los elementos necesarios para la visualización de la página (HTML, CSS, imágenes, etc.) respetando la estructura de carpetas explicadas en las clases teóricas.

Las prácticas que no cumplan este requisito serán consideradas como NO ENTREGADAS.



Validación de funcionalidades

Funcionalidades básicas (hasta 5 puntos)

Funcionalidad	Cumple (✓/X)
Registro/Login/Logout implementado con validaciones estrictas	
Correo corporativo validado con regex	
Contraseña segura (mínimo 8 caracteres, 1 mayúscula, 1 número)	
Empleado asignado a un concesionario	
Carga inicial desde JSON si BD está vacía (vehículos + concesionarios)	
Panel de administrador básico (alta/baja/edición de vehículos, alta de concesionarios)	
Listado de vehículos disponibles para empleados	
Filtros básicos (por concesionario y ciudad)	
Reserva de vehículos con fecha/hora (Bootstrap date/time pickers)	
Histórico de reservas de cada usuario (activas y finalizadas)	
Diseño responsive con Bootstrap (rejilla y modales)	
Accesibilidad básica (contraste, tamaño de letra, navegación teclado)	

Funcionalidades intermedias (hasta 8 puntos)

Funcionalidad	Cumple (✓/X)
Roles diferenciados correctamente implementados (empleado/admin)	
Gestión dinámica de disponibilidad de vehículos	
Filtros avanzados (autonomía mínima, plazas, color)	
AJAX en reservas y listados	
Accesibilidad intermedia (preferencias guardadas en sesión)	



Estadísticas básicas: nº de reservas por concesionario, vehículo más usado, total reservas	
--	--

Funcionalidades avanzadas (hasta 10 puntos)

Funcionalidad	Cumple (✓/X)
Integración con API de mapas (mostrar concesionarios y calcular distancias)	
Gestión avanzada de imágenes (subida validada + miniatura)	
Alertas automáticas: autonomía baja y recordatorio de devolución	
Accesibilidad avanzada (atajos configurables + orden de tabulación)	
Estadísticas avanzadas: kms recorridos, reservas por franjas horarias, incidencias	
Sistema de feedback de usuarios (puntuación + comentarios)	
Accesibilidad persistente en BD (cargada al iniciar sesión)	

En la defensa (obligatoria en todos los casos)

Requisito	Cumple (✓/X)
Demuestra funcionamiento en vivo (reservas, filtros, panel admin)	
Explica carga inicial y actualización desde JSON	
Muestra gestión de concesionarios y asignación de empleados	
Ejecuta accesibilidad en directo (contraste, fuentes, atajos teclado)	
Explica el código (rutas, middleware, consultas SQL)	