



# MÓDULO PROYECTO

---

CFGS Desarrollo de Aplicaciones Multiplataforma  
Informática y Comunicaciones

---

## TFG - Control de Flotas

***Tutor individual:*** Silvia Pedrón Hermosa

***Tutor colectivo:*** Cristina Silvan Pardo

***Año:*** 2023-2024

***Fecha de presentación:*** 23/05/2024

**Nombre y Apellidos:** Carlos Samaniego Leo

**Email:** [Carlos.samleo@educa.jcyl.es](mailto:Carlos.samleo@educa.jcyl.es)



## Tabla de contenido

1	Descripción general del proyecto .....	4
1.1	Objetivos.....	4
1.2	Cuestiones metodológicas .....	5
1.3	Entorno de trabajo (tecnologías de desarrollo y herramientas) .....	6
2	Descripción general del producto.....	8
2.1	Visión general del sistema: límites del sistema, funcionalidades básicas, usuarios y/o otros sistemas con los que pueda interactuar.....	8
2.2	Descripción breve de métodos, técnicas o arquitecturas(m/t/a) utilizadas. ....	10
2.3	Despliegue de la aplicación indicando plataforma tecnológica, instalación de la aplicación y puesta en marcha .....	11
3	Planificación y presupuesto .....	13
4	Documentación Técnica: análisis, diseño, implementación y pruebas. ....	15
4.1	Especificación de requisitos.....	15
4.2	Análisis del sistema .....	16
4.3	Diseño del sistema: .....	16
4.3.1	Diseño de la Base de Datos.....	16
4.3.2	Diseño de la Interfaz de usuario. ....	16
4.3.3	Diseño de la Aplicación. ....	19

4.4	Implementación:.....	19
4.4.1	Entorno de desarrollo. ....	19
4.4.2	Estructura del código. ....	20
4.4.3	Cuestiones de diseño e implementación reseñables. ....	21
4.5	Pruebas. ....	23
5	Manuales de usuario .....	25
5.1	Manual de usuario .....	25
5.2	Manual de instalación.....	37
6	Conclusiones y posibles ampliaciones .....	38
7	Bibliografía .....	39
8	Anexos.....	40

## 1 Descripción general del proyecto

### 1.1 *Objetivos*

El principal objetivo de esta aplicación es proporcionar un sistema avanzado de control y gestión de flotas para empresas, permitiendo a los usuarios conocer en tiempo real la ubicación exacta de cada uno de sus vehículos en un mapa. Esta funcionalidad es fundamental para optimizar la logística y el monitoreo de las operaciones de transporte.

Además, la aplicación incluirá una herramienta para la localización de gasolineras en toda España. Los usuarios podrán visualizar todas las estaciones de servicio disponibles en el mapa y, mediante un sistema de notificaciones emergentes (toasts), recibirán información sobre las gasolineras más cercanas a sus vehículos dentro de un radio específico. Este sistema facilitará la planificación de rutas hacia las estaciones de servicio seleccionadas directamente en el mapa, mostrando la ruta desde la ubicación actual del vehículo hasta la gasolinera elegida.

Esta característica no solo mejorará la eficiencia del reabastecimiento de combustible, sino que también permitirá a las empresas anticipar y gestionar mejor posibles imprevistos, como la necesidad urgente de repostar. De este modo, se optimizará la operatividad de la flota y se minimizarán los tiempos de inactividad, contribuyendo a una gestión más efectiva y económica de los recursos de transporte.

En resumen, los objetivos específicos del proyecto son:

**Monitorización en tiempo real de la flota:** Proveer una visualización en tiempo real de la ubicación de todos los vehículos de la empresa en un mapa interactivo.

**Localización de gasolineras:** Ofrecer un sistema integral para la visualización de todas las gasolineras en España, accesible a través de la aplicación.

**Notificaciones de proximidad:** Implementar un sistema de notificaciones que alerte a los usuarios sobre las gasolineras más cercanas a los vehículos dentro de un radio predefinido.

Planificación de rutas hacia gasolineras: Facilitar la planificación de rutas directas desde la ubicación del vehículo hasta la gasolinera seleccionada, mostrando el camino en el mapa.

Optimización de la operatividad: Mejorar la gestión del reabastecimiento de combustible y reducir los tiempos de inactividad de los vehículos, optimizando así la operatividad y eficiencia de la flota.

Con estos objetivos, la aplicación busca ser una herramienta integral que no solo ofrezca un control exhaustivo de la flota, sino que también aporte soluciones prácticas y eficientes para la gestión del reabastecimiento de combustible, mejorando la planificación y respuesta ante posibles imprevistos.

## **1.2 Cuestiones metodológicas**

Para desarrollar la aplicación "Control de Flotas", he decidido seguir algunas prácticas ágiles inspiradas en Scrum. Aunque estoy desarrollando el proyecto solo, estas prácticas pueden ser muy útiles.

La flexibilidad de Scrum es una gran ventaja. Incluso trabajando solo, pueden surgir nuevas ideas o cambios necesarios. Adaptar el trabajo en pequeños sprints me permite ajustar prioridades y asegurarme de que la aplicación cumpla con las necesidades actuales.

Trabajar en incrementos pequeños y manejables me ayuda a mantener el proyecto organizado. Puedo enfocarme en desarrollar y mejorar características específicas, como la ubicación de los vehículos o la integración de las gasolineras, una a la vez. Esto facilita las pruebas y mejoras constantes.

La comunicación constante es importante, incluso si solo estoy comunicando conmigo mismo. Mantener un registro diario de lo que he hecho y lo que necesito hacer a continuación me ayuda a mantenerme enfocado y a identificar problemas rápidamente.

La mejora continua es otra práctica clave. Reflexionar regularmente sobre lo que ha funcionado bien y lo que necesita mejorar me ayuda a optimizar mi trabajo y a mejorar la aplicación de manera constante.

En resumen, aunque estoy desarrollando la aplicación solo, adoptar prácticas ágiles inspiradas en Scrum me permite ser flexible, trabajar de manera organizada y mejorar continuamente. Esto asegura que el proyecto avance de manera eficiente y que la aplicación final cumpla con las expectativas de los usuarios.

### ***1.3 Entorno de trabajo (tecnologías de desarrollo y herramientas)***

El entorno de trabajo para el desarrollo de la aplicación incluirá un conjunto de tecnologías y herramientas cuidadosamente seleccionadas para asegurar una implementación robusta y eficiente del sistema de control de flotas. A continuación, se detallan las principales tecnologías y herramientas utilizadas:

#### Lenguajes de programación

C#: Utilizado principalmente para el desarrollo del backend de la aplicación, aprovechando su robustez y capacidad para manejar operaciones complejas de manera eficiente. C# es adecuado para la construcción de aplicaciones de alto rendimiento y es compatible con numerosas bibliotecas y frameworks que facilitan el desarrollo.

JavaScript: Empleado para el desarrollo de componentes interactivos del frontend. JavaScript permite la creación de interfaces de usuario dinámicas y responsivas, esenciales para una buena experiencia de usuario.

HTML: Utilizado para la estructura del contenido web. HTML es fundamental para definir la presentación y disposición de los elementos de la aplicación en el navegador.

#### Frameworks y bibliotecas

ASP.NET: Framework utilizado para el desarrollo del backend con C#. ASP.NET facilita la creación de aplicaciones web escalables y seguras, proporcionando una sólida infraestructura para el manejo de solicitudes HTTP, autenticación y gestión de sesiones.

OpenLayers: Biblioteca de JavaScript para la visualización de mapas. OpenLayers se utiliza para integrar mapas interactivos en la aplicación, permitiendo a los usuarios visualizar la ubicación de los vehículos y las gasolineras en tiempo real.

Dapper: Dapper es una biblioteca de mapeo de objetos para .NET. Proporciona métodos de extensión simples para realizar consultas SQL y mapear los resultados a objetos .NET de manera eficiente. Dapper elimina gran parte del código repetitivo asociado con la ejecución de consultas SQL y el mapeo de resultados a objetos, al tiempo que ofrece un rendimiento excepcional.

### Bases de datos

SQL Server: Sistema de gestión de bases de datos relacional utilizado para el almacenamiento de datos. SQL Server ofrece un rendimiento robusto, seguridad avanzada y la capacidad de manejar grandes volúmenes de datos. Es ideal para gestionar la información crítica relacionada con los vehículos y las gasolineras.

Firebase: Firebase es una plataforma de desarrollo de aplicaciones móviles y web desarrollada por Google. Ofrece una variedad de servicios en la nube, incluidas bases de datos en tiempo real, autenticación de usuarios, almacenamiento de archivos, análisis y más.

### APIs

API del Gobierno para las Gasolineras de España: Proporciona datos actualizados sobre las estaciones de servicio en toda España. Esta API es crucial para obtener información precisa sobre la ubicación, precios y servicios disponibles en las gasolineras, lo cual es esencial para la planificación de rutas de reabastecimiento.

API de OpenLayers: Utilizada para la visualización y manipulación de mapas en la aplicación. Esta API permite mostrar la ubicación de los vehículos y las gasolineras en un mapa interactivo, facilitando la navegación y planificación de rutas.

### Herramientas de desarrollo

Visual Studio Professional 2022: Entorno de desarrollo integrado (IDE) utilizado para el desarrollo de la aplicación. Visual Studio proporciona un conjunto completo de herramientas de desarrollo, incluyendo un depurador avanzado, integración con sistemas de control de versiones y numerosas extensiones que mejoran la productividad del desarrollador.

Visual Studio Code: Editor de código fuente ligero y potente, utilizado especialmente para la edición de archivos HTML, JavaScript y otros recursos web.

GitHub: Plataforma de control de versiones utilizada para gestionar el código fuente del proyecto. GitHub facilita la colaboración entre desarrolladores, el seguimiento de cambios y la integración continua.

### Despliegue

La aplicación se desplegará inicialmente en un entorno local, lo que permite un control completo sobre la configuración y prueba del sistema antes de considerar su despliegue en producción. Este enfoque asegura que todas las funcionalidades sean verificadas y optimizadas en un entorno controlado.

En resumen, el entorno de trabajo está compuesto por tecnologías y herramientas modernas y probadas, seleccionadas para garantizar que la aplicación sea eficiente, segura y fácil de mantener. Cada componente del entorno de trabajo juega un papel crucial en el desarrollo y despliegue exitoso del sistema de control de flotas, asegurando que se cumplan los objetivos del proyecto con alta calidad y confiabilidad.

## **2 Descripción general del producto**

### ***2.1 Visión general del sistema: límites del sistema, funcionalidades básicas, usuarios y/o otros sistemas con los que pueda interactuar.***

La aplicación "Control de Flotas" está diseñada para ser utilizada por empresas que gestionan flotas de vehículos. Los principales usuarios de la aplicación serán los gestores de flotas y los conductores. Esta herramienta tiene como objetivo principal mejorar la eficiencia en la gestión de flotas y facilitar la planificación de rutas de reabastecimiento de combustible.

#### **Límites del Sistema**

El sistema está limitado a la gestión de flotas de vehículos dentro de España. Está diseñado para ofrecer información en tiempo real sobre la ubicación de los vehículos y las gasolineras disponibles, permitiendo a los gestores y conductores tomar decisiones informadas sobre



rutas y reabastecimiento de combustible. La aplicación se desplegará en un entorno local y utilizará bases de datos locales para el almacenamiento de información relevante.

### **Funcionalidades Básicas**

1. Visualización en tiempo real de la ubicación de los vehículos: La aplicación permitirá a los gestores de flotas y conductores ver la ubicación exacta de cada vehículo en un mapa interactivo. Esta funcionalidad es crucial para la gestión eficiente de la flota, ya que proporciona visibilidad completa sobre la posición y el movimiento de los vehículos.
2. Localización y visualización de gasolineras en toda España: Utilizando la API de OpenLayers y la API del gobierno español para las gasolineras, la aplicación mostrará todas las gasolineras disponibles en el mapa. Los usuarios podrán ver detalles como el precio del combustible y los servicios disponibles en cada estación.
3. Sistema de notificaciones sobre gasolineras cercanas: La aplicación enviará notificaciones a los conductores sobre las gasolineras más cercanas en función de su ubicación actual. Esto es especialmente útil para la planificación de rutas y para asegurar que los vehículos siempre tengan acceso a reabastecimiento de combustible cuando sea necesario.
4. Planificación de rutas desde el vehículo hasta la gasolinera seleccionada: Una de las funcionalidades clave es la capacidad de planificar rutas óptimas desde la ubicación actual del vehículo hasta la gasolinera seleccionada. Esto ayudará a los conductores a encontrar las rutas más eficientes y rápidas, ahorrando tiempo y combustible.

### **Usuarios**

Todos los usuarios, incluidos los gestores de flotas y los conductores, tendrán acceso a las mismas funcionalidades dentro de la aplicación. Esto incluye la capacidad de visualizar en tiempo real la ubicación de los vehículos, recibir notificaciones sobre gasolineras cercanas y planificar rutas de reabastecimiento. La decisión de ofrecer las mismas funcionalidades a todos los usuarios se basa en promover la igualdad de acceso a la información y simplificar la experiencia de usuario, lo que garantiza que todos los usuarios puedan utilizar la aplicación de manera efectiva para sus necesidades individuales. Interacción con Otros Sistemas

**API de OpenLayers:** La aplicación utilizará la API de OpenLayers para la visualización de mapas interactivos. Esta API permitirá mostrar la ubicación de los vehículos y las gasolineras en un formato claro y fácil de usar, mejorando la experiencia del usuario.

**API del Gobierno Español para las Gasolineras:** La aplicación se integrará con la API del gobierno español para obtener datos precisos y actualizados sobre las gasolineras en toda España. Esta información incluirá la ubicación, precios del combustible y servicios disponibles, que serán mostrados en el mapa interactivo.

**Bases de Datos Locales:** La información sobre los vehículos se almacenará en bases de datos locales. SQL Server será utilizado para gestionar estos datos, asegurando un rendimiento robusto y una alta disponibilidad de la información.

## ***2.2 Descripción breve de métodos, técnicas o arquitecturas(m/t/a) utilizadas.***

En el desarrollo de la aplicación "Control de Flotas", se emplean diversas metodologías, técnicas y arquitecturas para garantizar su eficiencia, escalabilidad y facilidad de mantenimiento.

**Metodologías Ágiles Inspiradas en Scrum:** Para la gestión del proyecto, se adoptan prácticas ágiles inspiradas en Scrum. Aunque el desarrollo se realiza de forma individual, estas metodologías ofrecen flexibilidad y permiten adaptarse a cambios y nuevas ideas. Trabajar en sprints cortos facilita la priorización de tareas y asegura que la aplicación cumpla con las necesidades actuales. La comunicación constante, incluso con uno mismo, mediante registros diarios, garantiza la claridad en el progreso del proyecto. Además, la reflexión continua sobre el trabajo realizado permite identificar áreas de mejora y optimizar el desarrollo de manera constante.

**Desarrollo Frontend y Backend:** El desarrollo del frontend se realiza principalmente utilizando JavaScript, que permite crear interfaces de usuario dinámicas y responsivas. HTML se utiliza para la estructura del contenido web, garantizando una presentación clara y ordenada de los elementos en el navegador. En el backend, se emplea C# junto con el framework ASP.NET, aprovechando su robustez y capacidad para manejar operaciones

complejas de manera eficiente. ASP.NET proporciona una sólida infraestructura para el manejo de solicitudes HTTP, autenticación y gestión de sesiones, asegurando un desarrollo escalable y seguro.

Visualización de Mapas y Geolocalización: Para la visualización de mapas interactivos, se integra la biblioteca OpenLayers en el frontend de la aplicación. OpenLayers permite mostrar la ubicación de los vehículos y las gasolineras en tiempo real, facilitando la navegación y planificación de rutas. Además, se utiliza la API del Gobierno Español para las Gasolineras para obtener datos actualizados sobre las estaciones de servicio en toda España, lo que permite mostrar información precisa sobre precios del combustible y servicios disponibles.

Gestión de Datos: La información sobre los vehículos se gestiona utilizando bases de datos relacionales, específicamente SQL Server. SQL Server ofrece un rendimiento robusto, seguridad avanzada y capacidad para manejar grandes volúmenes de datos, garantizando la disponibilidad y confiabilidad de la información crítica para la aplicación.

Herramientas de Desarrollo y Control de Versiones: Para el desarrollo de la aplicación, se utilizan herramientas como Visual Studio Professional 2022 y Visual Studio Code, que proporcionan un entorno de desarrollo integrado y un editor de código fuente potente. Además, se emplea GitHub como plataforma de control de versiones, facilitando la colaboración entre desarrolladores, el seguimiento de cambios y la integración continua del código.

En resumen, la aplicación "Control de Flotas" se desarrolla utilizando metodologías ágiles, tecnologías modernas y herramientas de desarrollo robustas, lo que garantiza un producto final eficiente, seguro y fácil de mantener. La combinación de estas metodologías, técnicas y arquitecturas permite cumplir con los objetivos del proyecto y ofrecer una solución integral para la gestión de flotas y el reabastecimiento de combustible.

### ***2.3 Despliegue de la aplicación indicando plataforma tecnológica, instalación de la aplicación y puesta en marcha***

El despliegue de la aplicación "Control de Flotas" se lleva a cabo de manera cuidadosa y planificada, asegurando una implementación exitosa en el entorno de producción. A

continuación, se detalla el proceso de despliegue, incluyendo la plataforma tecnológica utilizada, la instalación de la aplicación y su puesta en marcha:

Plataforma Tecnológica: La aplicación se despliega inicialmente en un entorno local para garantizar un control completo sobre la configuración y prueba del sistema antes de considerar su despliegue en producción. Esto se logra utilizando un servidor local o una máquina virtual que cumpla con los requisitos de hardware y software necesarios para ejecutar la aplicación de manera óptima.

Instalación de la Aplicación:

Configuración del Entorno: Se configura el servidor local o la máquina virtual con el sistema operativo adecuado, las dependencias de software necesarias y las herramientas de desarrollo correspondientes, como Visual Studio y SQL Server.

Despliegue del Backend: Se compila y despliega el backend de la aplicación, desarrollado utilizando C# y ASP.NET, en el servidor local. Esto implica la configuración de la aplicación en el servidor web, asegurando que esté disponible para recibir solicitudes HTTP.

Despliegue del Frontend: Se copian los archivos estáticos del frontend, desarrollado con JavaScript y HTML, en el servidor web. Estos archivos incluyen las páginas HTML, archivos JavaScript y recursos como imágenes y estilos CSS.

Configuración de la Base de Datos: Se configura la base de datos utilizando SQL Server, asegurando que esté correctamente instalada y configurada en el servidor local. Se crean las tablas necesarias y se insertan los datos iniciales requeridos para el funcionamiento de la aplicación.

Integración con APIs Externas: Se realiza la integración con las APIs externas necesarias, como la API del Gobierno Español para las Gasolineras, asegurando que la aplicación pueda acceder a los datos actualizados sobre las estaciones de servicio en España.

Puesta en Marcha:

Pruebas de Funcionalidad: Se realizan pruebas para verificar que todas las funcionalidades de la aplicación funcionen según lo esperado. Esto incluye pruebas de integración para

asegurar que los componentes frontend y backend se comuniquen correctamente, así como pruebas de usuario para validar la experiencia del usuario final.

Pruebas de Rendimiento y Seguridad: Se realizan pruebas de rendimiento para evaluar la capacidad de la aplicación para manejar cargas de trabajo esperadas y pruebas de seguridad para identificar posibles vulnerabilidades y asegurar la protección de los datos del usuario.

### 3 Planificación y presupuesto

#### 1. Recursos Humanos:

- Desarrollador Backend (C# y ASP.NET): 2 meses a tiempo completo.
- Desarrollador Frontend (JavaScript, HTML, CSS, OpenLayers): 2 meses a tiempo completo.
- Diseñador UI/UX: 1 mes a tiempo parcial.
- Ingeniero de DevOps: 1 mes a tiempo parcial para configuración de servidores y despliegue.

#### Costos por roles (salarios mensuales estimados en España):

- Desarrollador Backend: 2500 €/mes
- Desarrollador Frontend: 2500 €/mes
- Diseñador UI/UX: 2000 €/mes
- Ingeniero de DevOps: 3000 €/mes (tiempo parcial, estimado 50%)

#### 2. Tecnologías y Herramientas:

- Visual Studio Professional 2022: Licencia anual 1200 €
- SQL Server: Puede variar, pero considerando una versión básica, 1500 €
- Firebase: Costos según uso, estimado inicial 500 €
- API del Gobierno Español para Gasolineras: Gratuita

#### 3. Infraestructura:

- Servidor para despliegue:
  - Servidor de desarrollo: 50 €/mes

- Servidor de producción: 100 €/mes
- Total anual:  $(50 + 100) * 12 = 1800$  €

### **Presupuesto Estimado:**

#### **1. Recursos Humanos:**

- Desarrollador Backend: 2 meses x 2500 € = 5000 €
- Desarrollador Frontend: 2 meses x 2500 € = 5000 €
- Diseñador UI/UX: 1 mes x 2000 € = 2000 €
- Ingeniero de DevOps: 1 mes (tiempo parcial) x 1500 € = 1500 €
- Total Recursos Humanos:  $5000 + 5000 + 2000 + 1500 = 13500$  €

#### **2. Tecnologías y Herramientas:**

- Visual Studio Professional 2022: 1200 €
- SQL Server: 1500 €
- Firebase: 500 €
- Total Tecnologías y Herramientas: 3200 €

#### **3. Infraestructura:**

- Servidores (anual): 1800 €

### **Total del Presupuesto Estimado:**

- Recursos Humanos: 13500 €
- Tecnologías y Herramientas: 3200 €
- Infraestructura: 1800 €
- **Total Estimado: 18500 €**

*Consideraciones Adicionales:* Este presupuesto es una estimación inicial y puede variar en función de factores como la duración real del desarrollo, cambios en el alcance del proyecto y costos adicionales no contemplados. Se recomienda incluir un margen adicional del 10-15% para cubrir imprevistos.

## **4 Documentación Técnica: análisis, diseño, implementación y pruebas.**

### ***4.1 Especificación de Requisitos Funcionales***

#### Integración con APIs externas para la visualización de gasolineras

Descripción: La aplicación debe integrar información sobre gasolineras, obtenida de fuentes externas mediante APIs. Esta integración permite a los usuarios ver las ubicaciones de las gasolineras en el mismo mapa donde se monitorean los vehículos.

Detalles Técnicos: Utilización de la API del Gobierno Español para obtener datos de las gasolineras, incluyendo ubicación, precios de combustible y otros detalles relevantes.

Procesamiento y almacenamiento de los datos recibidos para una visualización eficiente y rápida.

Actualización periódica de la información de las gasolineras para mantener la precisión y relevancia de los datos mostrados.

Implementación de un sistema de capas en el mapa con OpenLayers para alternar la visualización entre vehículos y gasolineras.

#### Notificaciones emergentes para alertar sobre gasolineras cercanas

Descripción: La aplicación debe generar notificaciones emergentes para alertar a los conductores sobre la proximidad de gasolineras, especialmente cuando el nivel de combustible es bajo o se detecta una necesidad de reabastecimiento.

Detalles Técnicos: Configuración de umbrales de alerta basados en niveles de combustible, distancia a la gasolinera y otras condiciones definidas por el usuario.

Desarrollo de una interfaz de notificaciones emergentes, tanto en la aplicación móvil del conductor como en el panel de control del gestor de flotas.

#### Planificación de rutas hacia las gasolineras seleccionadas

**Descripción:** La aplicación debe proporcionar herramientas para la planificación de rutas optimizadas hacia las gasolineras seleccionadas por el usuario. Esta funcionalidad es esencial para asegurar que los vehículos sigan las rutas más eficientes y seguras.

**Detalles Técnicos:** Integración con OpenLayers para el cálculo de rutas.

Interfaz de usuario intuitiva que permita a los gestores de flotas y conductores seleccionar una gasolinera y obtener una ruta sugerida. Análisis del sistema

## 4.2 *Diseño del sistema:*

### 4.2.1 Diseño de la Base de Datos

La base de datos solo contendrá dos tablas, una para poder cargar todas las marcas y modelos disponibles:

TiposCoches	
id	
marca	
modelo	

Y otra tabla que contendrá toda la información de los vehículos:

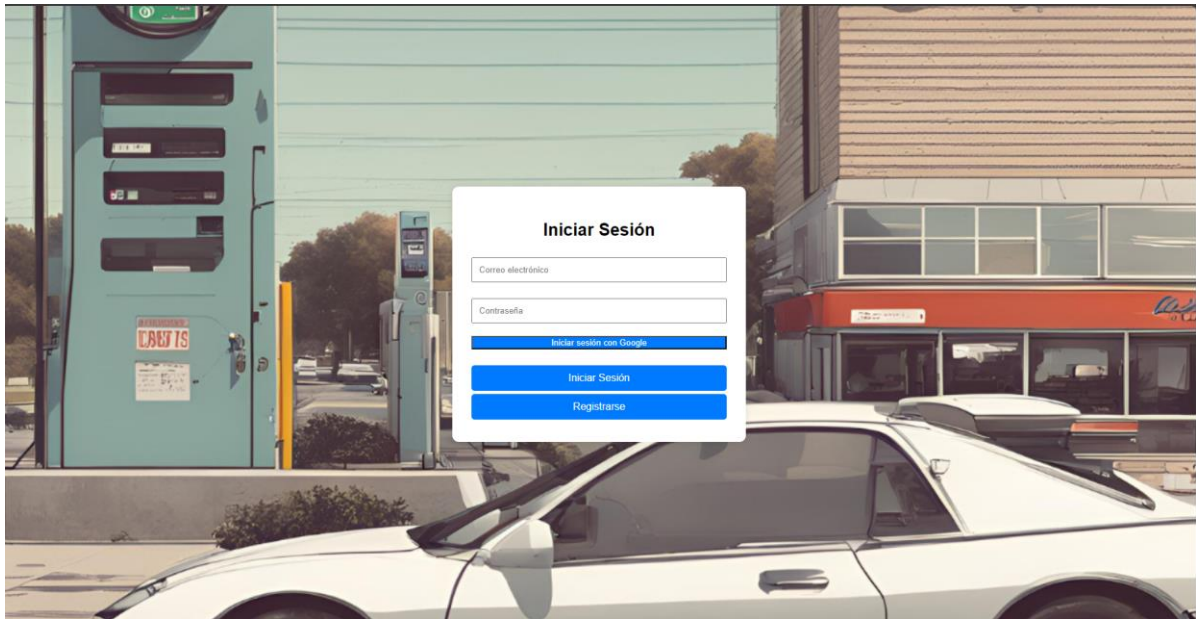
Coches	
idCoche	
Latitud	
Longitud	
Marca	
Modelo	
Año	
AutonomiaTotal	
AutonomiaRestante	
idUsuario	
matricula	

### 4.2.2 Diseño de la Interfaz de usuario.

La interfaz de usuario es muy sencilla, entendiendo directamente la funcionalidad de cada uno de los botones.

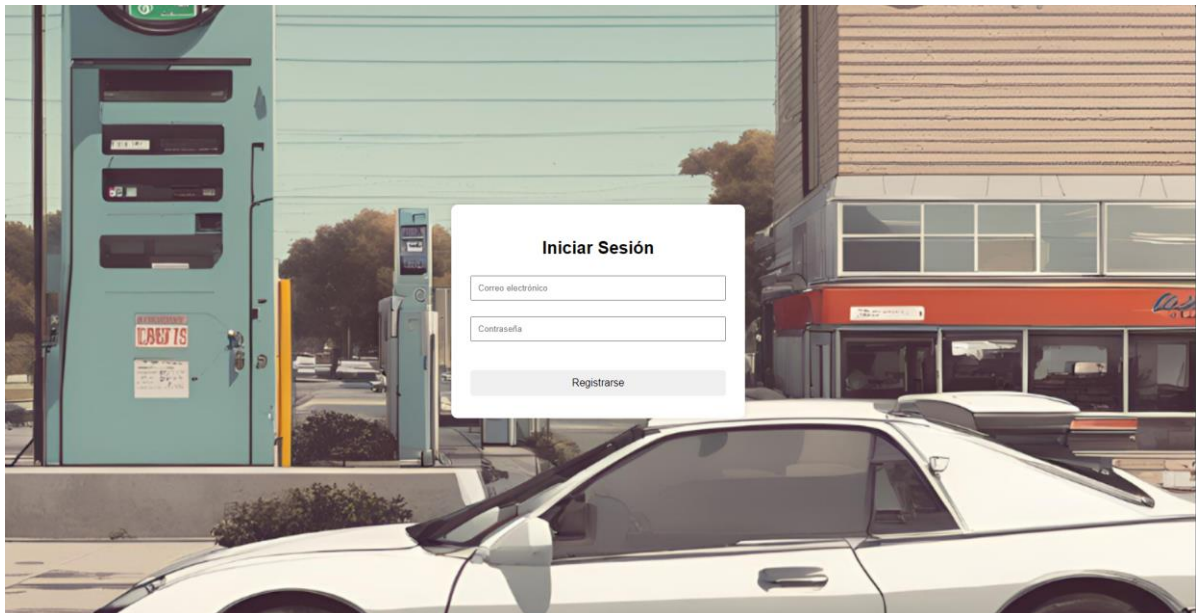


Login:



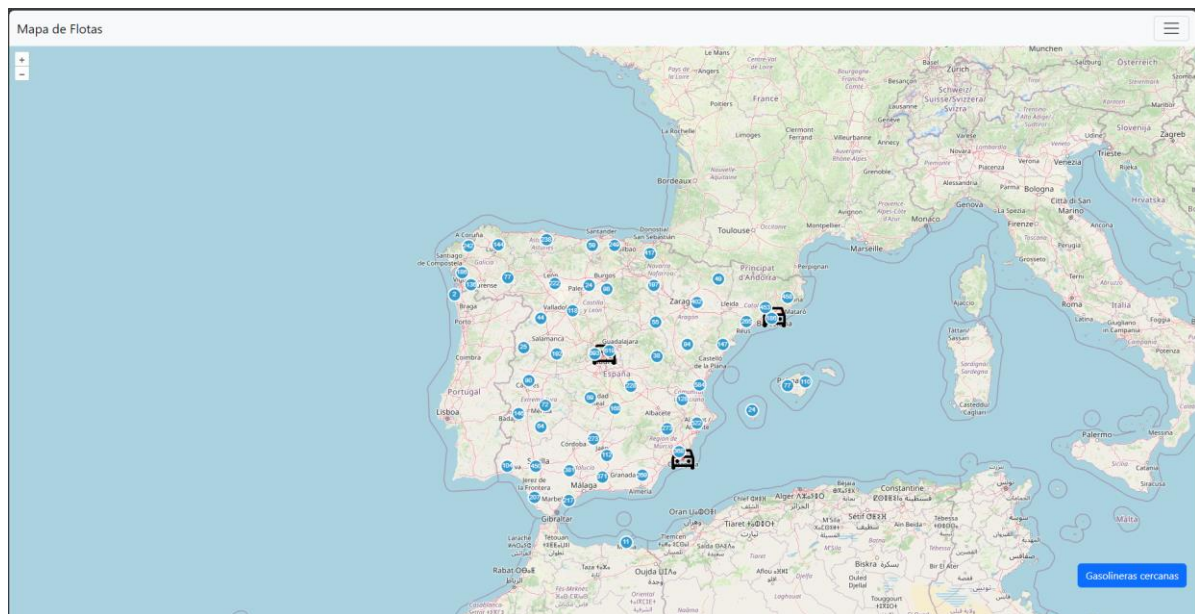
Ventana sencilla con una imagen extraída de canva. Se puede iniciar sesión con Google, con los campos introducidos o con la contraseña.

Registro:



Ventana que sigue el mismo estilo que el Login. Se puede crear el usuario.

Mapa:



Ventana principal de la aplicación, donde se puede ampliar para ver la ubicación exacta de los coches y de las gasolineras. Se puede sacar las gasolineras cercanas a estos coches. Y por ultimo, se puede abrir una barra lateral para cargar datos de coches, acceder a la tabla con todos los coches del usuario o cerrar la sesión.

### Vehiculos:

Crear

Editar

Eliminar

Search

<input type="checkbox"/>	ID Coche	Latitud	Longitud	Marca	Modelo	Año	Autonomía Total	Autonomía Restante	ID Usuario	Matricula
<input type="checkbox"/>	1	41.3851	2.1734	Chevrolet	Cruze	2005	350	324	H66yf0EEQkbJjic0O1pHRXOLf2	WCS1231
<input type="checkbox"/>	2	40.4168	-3.7038	Honda	CR-V	2019	450	250	H66yf0EEQkbJjic0O1pHRXOLf2	DEF456
<input type="checkbox"/>	1011	37.60512	-0.98623	BMW	3 Series	2002	200	120	H66yf0EEQkbJjic0O1pHRXOLf2	ERW3425

Showing 1 to 3 of 3 rows

Ventana que refleja todos los vehiculos del usuario, en esta ventana se puede buscar registros, seleccionar los campos a visualizar y acceder a los modales para crear o editar nuevos vehiculos, además de que si existen muchos registros se establecerá una paginación en la tabla.

### **4.2.3 Diseño de la Aplicación.**

El diseño de la aplicación se basa en la simplicidad y la usabilidad. Se busca que el usuario pueda realizar sus tareas de manera intuitiva y eficiente, sin distracciones innecesarias. Se ha priorizado la claridad en la presentación de la información y la facilidad de navegación entre las diferentes secciones de la aplicación.

## **4.3 Implementación:**

### **4.3.1 Entorno de desarrollo.**

El desarrollo de la aplicación se llevó a cabo en un entorno profesional utilizando herramientas y tecnologías avanzadas para garantizar la eficiencia y la calidad del software. A continuación, se detallan las principales herramientas y tecnologías utilizadas durante el proceso de desarrollo:

#### Git:

Git se utilizó como sistema de control de versiones para mantener un historial de cambios en el código fuente. Esto permitió un trabajo colaborativo eficiente, la gestión de ramas de desarrollo y la reversión de cambios si fuera necesario.

#### Visual Studio Professional 2022:

Visual Studio es un entorno de desarrollo integrado (IDE) altamente avanzado que proporciona una amplia gama de herramientas y características para el desarrollo de software. En este caso, se utilizó Visual Studio Professional 2022 para escribir, depurar y compilar el código de la aplicación. Las funcionalidades avanzadas de depuración, el soporte integrado para múltiples lenguajes de programación y las herramientas de administración de proyectos fueron fundamentales durante todo el ciclo de desarrollo.

#### SQL Server:

SQL Server se empleó como sistema de gestión de bases de datos relacional para el almacenamiento y la manipulación de datos. Proporciona un entorno robusto y escalable

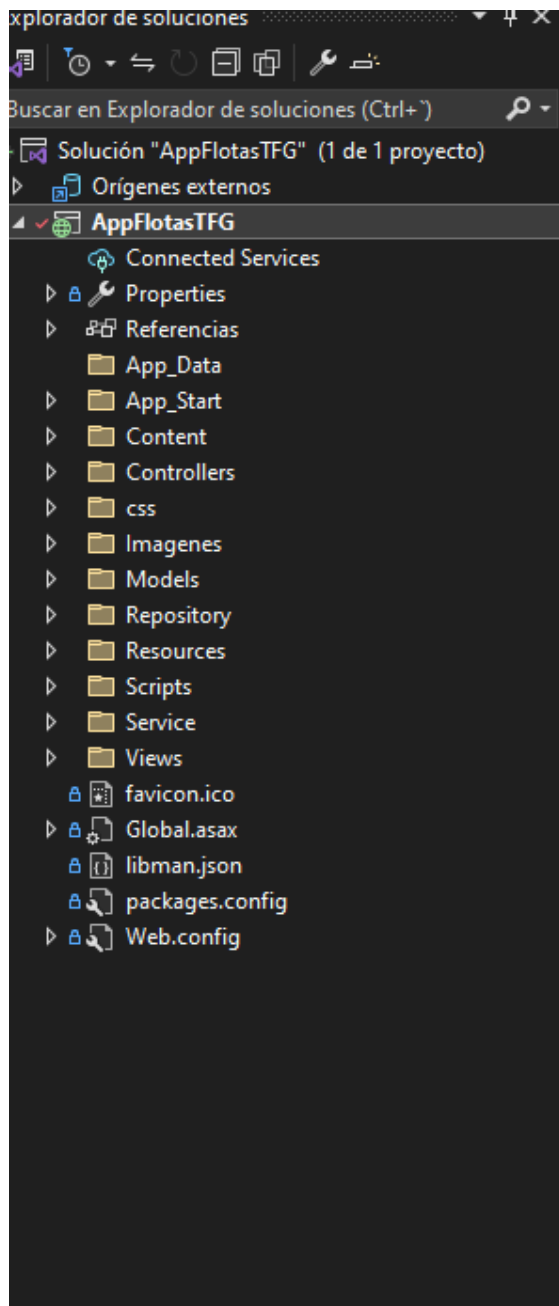
para la gestión de grandes volúmenes de datos, así como herramientas avanzadas para la administración y optimización de bases de datos.

#### Firebase:

Firebase se utilizó como plataforma para el desarrollo de aplicaciones móviles y web, proporcionando una amplia gama de servicios en la nube, incluido Firebase Authentication. Estos servicios fueron fundamentales para la autenticación de usuarios, el almacenamiento de datos en tiempo real y la comunicación en tiempo real entre la aplicación y el servidor.

#### **4.3.2 Estructura del código.**

La estructura del código es la siguiente



La aplicación viene con las carpetas y archivos cargados por defecto, pero a esta solución he añadido distintas carpetas.

- Controllers:

Esta carpeta contiene archivos de controladores, que son responsables de manejar las solicitudes HTTP entrantes y coordinar las acciones necesarias para satisfacer esas

solicitudes. Los controladores pueden contener métodos para procesar formularios, recuperar datos de la base de datos y responder con las vistas adecuadas.

- Css:

Aquí se almacenan archivos CSS que contienen reglas de estilo para dar formato y diseño a las páginas web de la aplicación. Estos archivos pueden incluir estilos generales, así como estilos específicos para componentes individuales de la interfaz de usuario.

- Imágenes:

Esta carpeta contiene imágenes utilizadas en la interfaz de usuario de la aplicación. Las imágenes pueden incluir logotipos, iconos, gráficos y cualquier otro elemento visual necesario para mejorar la experiencia del usuario.

- Models:

En esta carpeta se encuentran los archivos de modelos, que definen la estructura de los datos utilizados en la aplicación. Los modelos representan entidades de dominio como usuarios, vehículos, tipos de vehículos, etc., y suelen estar relacionados con las tablas de la base de datos.

- Repository:

La carpeta Repository contiene archivos que implementan el patrón Repository o similar para abstraer el acceso a datos de la aplicación. Estos archivos pueden contener métodos para realizar operaciones CRUD en la base de datos, como crear, leer, actualizar y eliminar registros.

- Scripts:

Aquí se almacenan archivos JavaScript que contienen lógica del lado del cliente para mejorar la interactividad de las páginas web de la aplicación. Los scripts pueden

incluir funciones para validar formularios, realizar animaciones, manipular el DOM y realizar solicitudes AJAX al servidor.

- Service:

La carpeta Service contiene archivos que implementan la lógica de negocio de la aplicación. Estos archivos pueden contener clases y funciones que realizan operaciones complejas que no pertenecen directamente a los controladores ni a los modelos, como cálculos, procesamiento de datos y coordinación de acciones entre diferentes partes de la aplicación.

- View:

En esta carpeta se encuentran archivos de vistas, que son archivos HTML que definen la estructura y el contenido de las páginas web generadas por la aplicación. Las vistas suelen contener código HTML junto con etiquetas y directivas de servidor para mostrar datos dinámicos proporcionados por los controladores

#### **4.3.3 Cuestiones de diseño e implementación reseñables.**

Durante la implementación, surgieron diversas cuestiones de diseño e implementación que requirieron atención especial. Estas incluyeron la gestión de la seguridad y la privacidad de los datos, la optimización del rendimiento de la aplicación y la integración con sistemas externos.

Las mayores dudas ocurrieron para la forma en la que hacer el login, y la creación y edición de vehículos. Tuve muchas dudas entre crear una ventana nueva para el formulario, o no hacerlo, lo mismo paso para mostrar los datos de los coches en el mapa. Pero al final por rendimiento preferí cargarlo en modales que aumentaría la rapidez de la aplicación.

#### **4.4 Pruebas.**

Las pruebas relacionadas con los inicios de sesión o registro de usuarios están realizadas en el manual de Usuario.

Pruebas sobre creación de vehículos:

- No se rellena alguno de los campos del modal

## Nuevo Vehículo



<p>Marca</p> <div> <div>Selecciona una Marca</div> <div>▼</div> </div> <p>Por favor, rellene este campo.</p>	<p>Modelo</p> <div> <div>Seleccione una Marca</div> <div>▼</div> </div> <p>Por favor, rellene este campo.</p>	<p>Año</p> <div> <div>Elige un Año</div> <div>▼</div> </div>
<p>Latitud</p> <div> <input type="text"/> </div> <p>Por favor, rellene este campo.</p>	<p>Longitud</p> <div> <input type="text"/> </div> <p>Por favor, rellene este campo.</p>	
<p>Autonomía Total</p> <div> <input type="text"/> </div> <p>Por favor, rellene este campo.</p>	<p>Autonomía Restante</p> <div> <input type="text"/> </div> <p>Por favor, rellene este campo.</p>	<p>Matrícula</p> <div> <input type="text"/> </div> <p>Por favor, rellene este campo.</p>

Cerrar Agregar

- Se rellenan los campos del modal y se agrega el coche

localhost:44394 dice

Coche creado correctamente.

Aceptar

## Nuevo Vehículo

BMW

▼

3 Series

▼

2002

▼

37.60512

-0.98623

200

120

ERW3425

Cerrar Agregar

Crear

Edit

Eliminar

Search

▼

	ID Coche	Latitud	Longitud	Marca	Modelo	Año	Autonomía Total	Autonomía Restante	ID Usuario	Matrícula
<input type="checkbox"/>	1	41.3851	2.1734	Chevrolet	Cruze	2005	350	324	H66yOEEQkbJjlc0O1iPhRX0LtF2	WCS1231
<input type="checkbox"/>	2	40.4168	-3.7038	Honda	CR-V	2019	450	250	H66yOEEQkbJjlc0O1iPhRX0LtF2	DEF456
<input type="checkbox"/>	1011	37.60512	-0.98623	BMW	3 Series	2002	200	120	H66yOEEQkbJjlc0O1iPhRX0LtF2	ERW3425

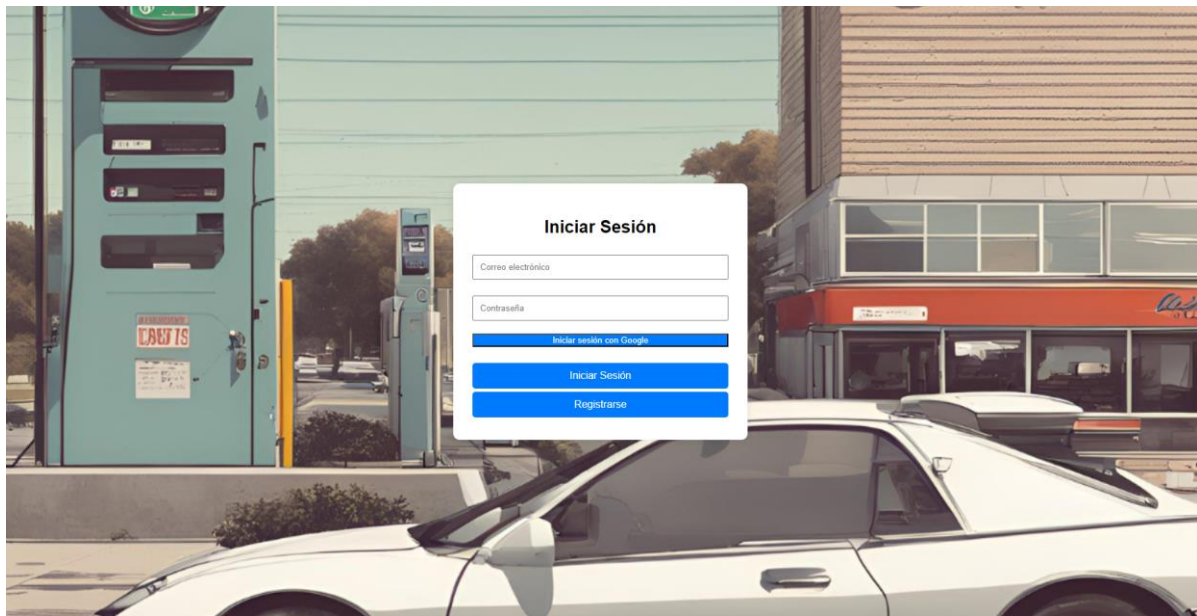
- Al editar ocurrirá lo mismo que cuando se crea un nuevo vehículo.



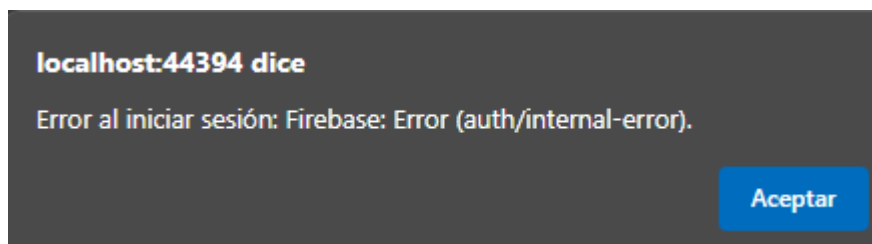
## 5 Manuales de usuario

### 5.1 Manual de usuario

La Aplicación empezara con un login, en el que podremos iniciar sesión manualmente, acceder a la ventana para registrar un usuario o iniciar sesión con Google.



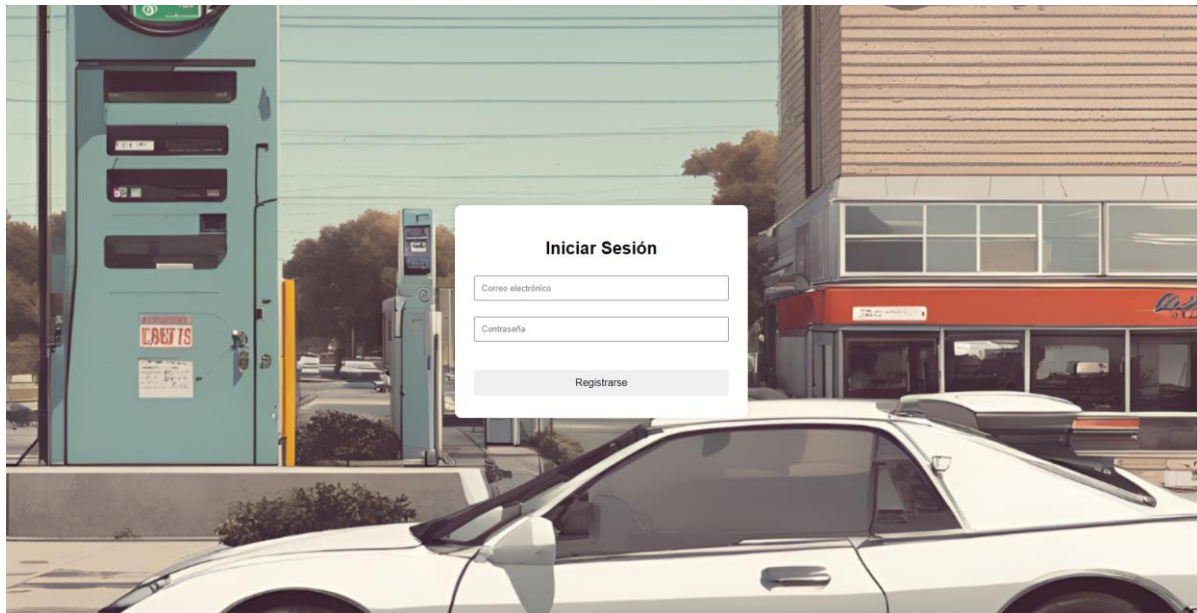
Si queremos iniciar sesión con una cuenta que no este creada en el firebase, o que si este creada pero la contraseña es incorrecta nos saltara el siguiente alert.



Si por ejemplo introducimos como correo, un correo no valido, por ejemplo, que no acabe ni en @gmail.com, @educa.jcyl.es..., nos saltara el siguiente alert



Pasando a la ventana de registro, pulsando al botón Registrarse accederemos a la siguiente ventana



En esta ventana podremos introducir un correo y una contraseña para que se cree en la base de datos, para los valores de estos campos también tendremos alert si se escribe un correo no valido, también si la contraseña tiene menos de 6 caracteres

**localhost:44394 dice**

La contraseña debe tener al menos 6 caracteres

Aceptar

Una vez se introduzcan los campos de manera correcta nos saldrá el siguiente alert, informando de que se ha creado el usuario correctamente

**localhost:44394 dice**

usuario creado

Aceptar

Y otro alert confirmando que se ha enviado un correo de verificación a la cuenta escrita

**localhost:44394 dice**

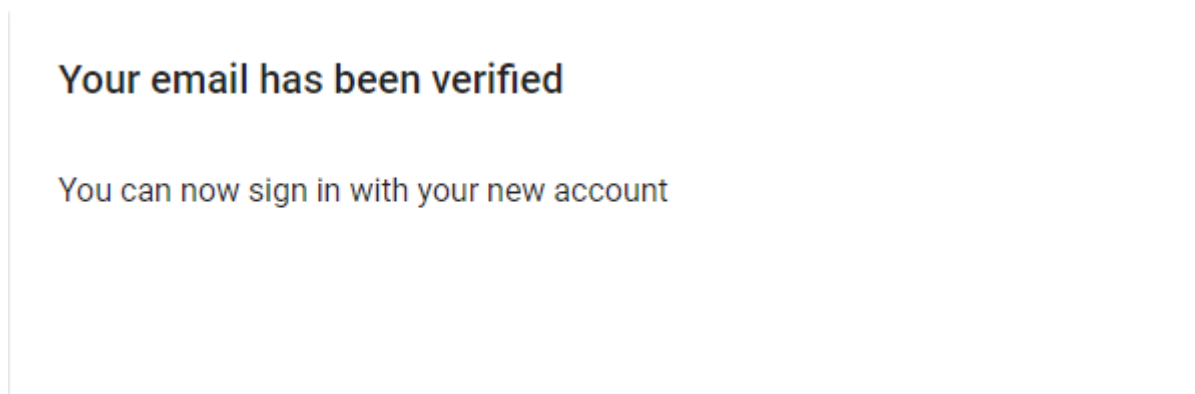
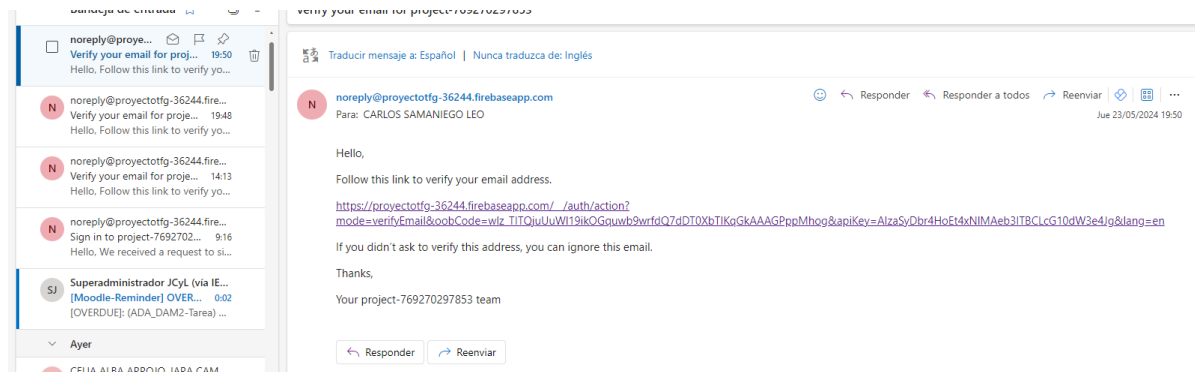
Se ha enviado el correo de verificación

Aceptar

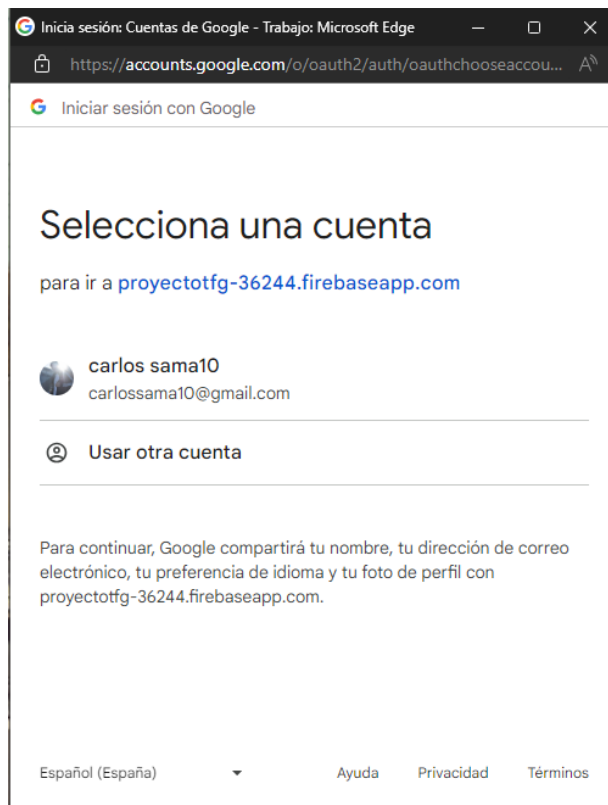
Una vez aceptados estos alerts, volveremos al login dando a la flecha de al lado de la recarga, si intentamos iniciar sesión con esta cuenta creada sin haber confirmado el email, no nos entrará a la app y nos saldrá el siguiente alert.



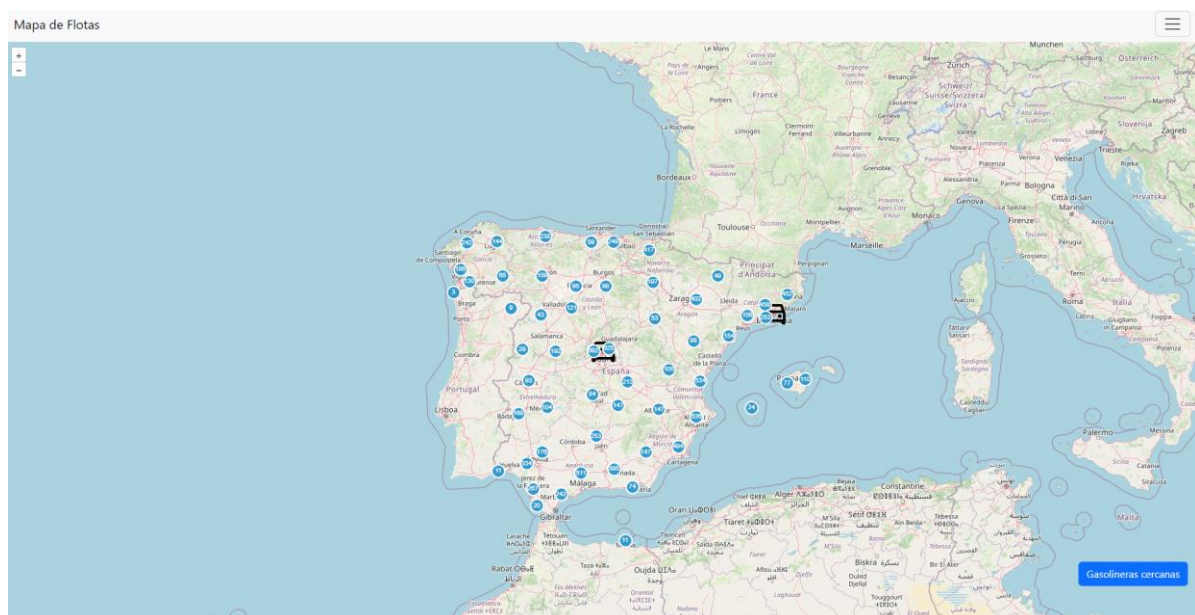
Entonces procedemos a acceder al correo y aceptar el link que se nos enviará el cual nos redirigirá a una ventana confirmando la verificación de este correo



Una vez hecho todo esto podremos acceder con la cuenta creada a la aplicación. Otra de las opciones que hay para logarse es hacerlo a través de Google, pinchando en el botón de inicio de sesión con Google nos aparecerá la siguiente ventana en la que, seleccionando una cuenta, si esa cuenta no esta creada en firebase se creara con una contraseña aleatoria, pero si esta creada solamente iniciara sesión. Cuando se crea a través de Google se establece una contraseña aleatoria.

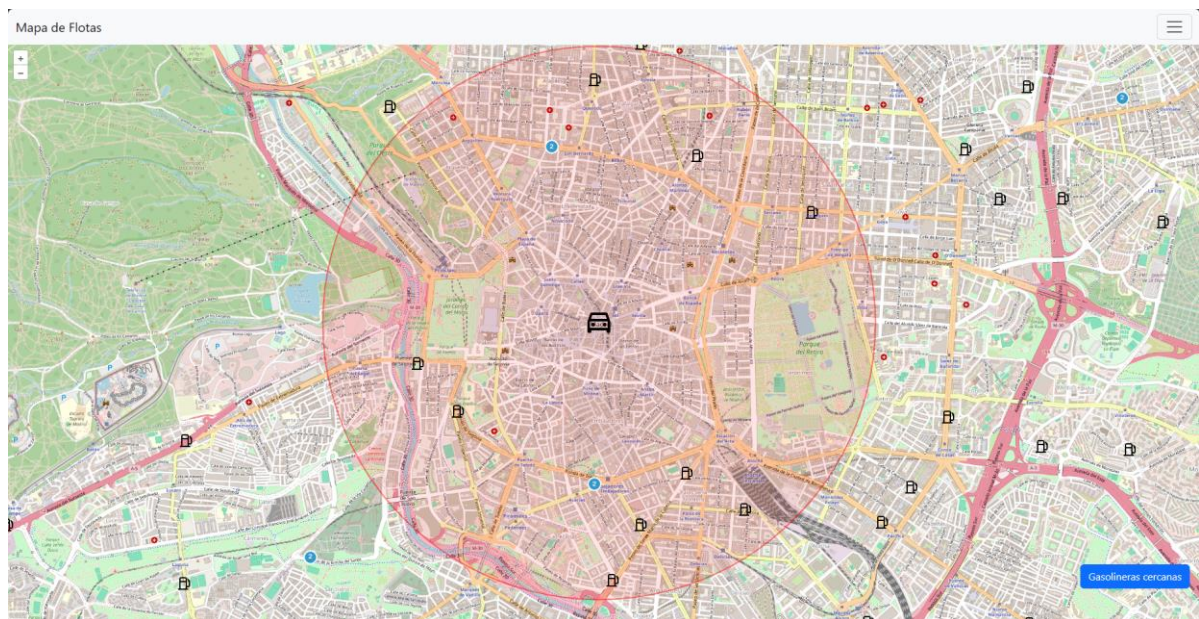


Una vez dentro, dependiendo con que cuenta nos hayamos logado, guardaremos un id u otro para cargar en el siguiente mapa los coches asociados a ese id.



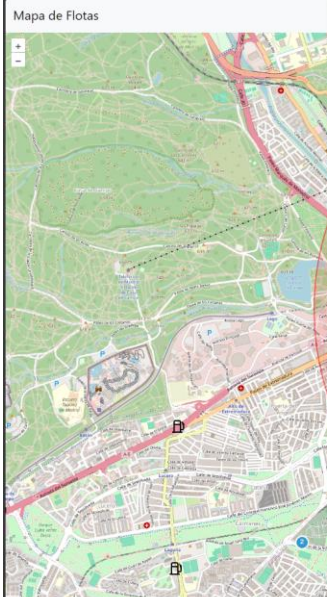


En este mapa podemos observar la ubicación de los coches de la cuenta que se ha logado y todas las gasolineras del api del gobierno español. Una vez ampliamos el mapa por ejemplo a un vehículo observamos que cada coche tiene un radio de 3 kilómetros alrededor suyo y que las gasolineras van mostrando con un icono la ubicación exacta en la que se encuentran.



Este radio es el que usaremos para marcar las gasolineras cercanas a ese coche a las cuales podremos planificar su ruta. Para ver todas las gasolineras cercanas pulsaremos el botón de Gasolineras cercanas situado en la parte inferior derecha de la ventana y son saldrá la siguiente tabla con los datos de las gasolineras (Principalmente se cargaran las gasolineras de todos los vehículos del mapa, pero luego tendremos un select para elegir el coche del que queremos ver sus gasolineras cercanas) Aquí está un ejemplo para ver todos los vehículos:


Mapa de Flotas



Gasolineras dentro del radio

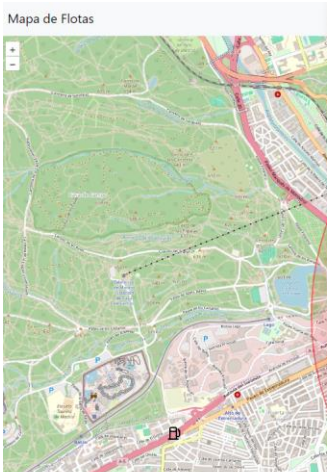
Seleccione un vehículo

Localidad	Dirección	Precio Gasolina 95	Precio Gasóleo A	Ruta
BARCELONA	CALLE DOCTOR AIGUADER, 50	No disponible	No disponible	<a href="#">Ruta</a>
BARCELONA	CALLE TALLAT 15 ESQUINA CIUDAD DE GRANADA,	1,549	1,439	<a href="#">Ruta</a>
BARCELONA	AVENIDA BOGATELL, 49	1,779	1,589	<a href="#">Ruta</a>
BARCELONA	AVENIDA DEL LITORAL, 49	1,789	1,645	<a href="#">Ruta</a>
BARCELONA	CALLE PUJADES, 154	1,799	1,629	<a href="#">Ruta</a>
BARCELONA	Almogavares, 66	1,719	1,479	<a href="#">Ruta</a>
BARCELONA	CALLE PERE IV, 79	1,528	1,417	<a href="#">Ruta</a>
BARCELONA	CL ALMOGAVARES 159 - 161	1,769	1,679	<a href="#">Ruta</a>
BARCELONA	CALLE BALMES, 288	1,799	1,609	<a href="#">Ruta</a>
BARCELONA	CALLE BADAJOZ, 108	1,485	1,365	<a href="#">Ruta</a>
BARCELONA	AVINGUDA DEL PARALELO, 132	1,669	1,479	<a href="#">Ruta</a>
BARCELONA	RONDA SAN PABLO, 39	1,739	1,595	<a href="#">Ruta</a>
BARCELONA	CALLE CASANOVA, 89	1,799	1,609	<a href="#">Ruta</a>
BARCELONA	CASANOVA, 23	1,779	1,579	<a href="#">Ruta</a>
BARCELONA	AVENIDA PARALELO, 179	1,775	1,586	<a href="#">Ruta</a>
BARCELONA	CALLE COMTE D'URIGELL, 219	1,799	1,609	<a href="#">Ruta</a>
BARCELONA	CALLE ROGER DE FLOR, 58	1,789	1,544	<a href="#">Ruta</a>



Gasolineras cercanas


Mapa de Flotas



Gasolineras dentro del radio

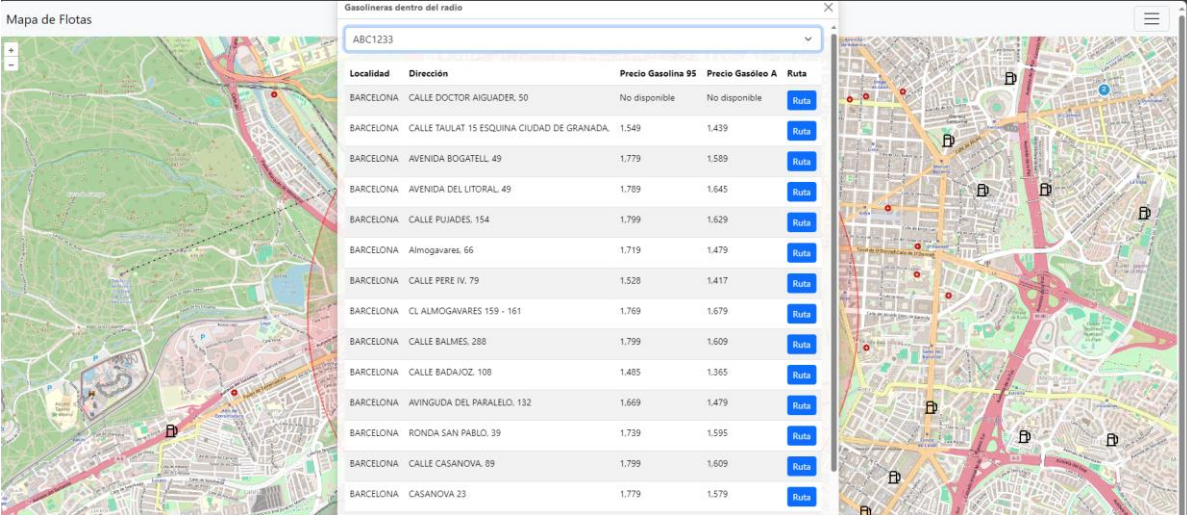
Seleccione un vehículo

Localidad	Dirección	Precio Gasolina 95	Precio Gasóleo A	Ruta
BARCELONA	CALLE LEGALITAT, 59	1,789	1,589	<a href="#">Ruta</a>
MADRID	PASEO MORET, 7	1,799	1,639	<a href="#">Ruta</a>
MADRID	CALLE SEGOVIA C/V A PASEO VIRGEN DEL PUERTO, S/N	1,789	1,619	<a href="#">Ruta</a>
MADRID	CL PASEO ACACIAS, 8	1,789	1,539	<a href="#">Ruta</a>
MADRID	RONDA SEGOVIA, 37	1,756	1,594	<a href="#">Ruta</a>
MADRID	CALLE ALBERTO AGUILERA, 9	1,799	1,549	<a href="#">Ruta</a>
MADRID	CALLE ALBERTO AGUILERA, 18	1,779	1,549	<a href="#">Ruta</a>
MADRID	CALLE BRAVO MURILLO, 012	1,765	1,595	<a href="#">Ruta</a>
MADRID	BARRIO CENTRO-EMBAJADORES, 83	1,749	1,549	<a href="#">Ruta</a>
MADRID	GLORIETA EMBAJADORES, 0	1,789	1,619	<a href="#">Ruta</a>
MADRID	PASEO SANTA MARIA DE LA CABEZA, 90	1,749	1,549	<a href="#">Ruta</a>
MADRID	CL ANCORA 39	1,779	1,535	<a href="#">Ruta</a>
MADRID	PASEO SANTA MARIA DE LA CABEZA, 18	1,789	1,539	<a href="#">Ruta</a>
MADRID	CL PLAZA LEGAZPI, 9	1,789	1,579	<a href="#">Ruta</a>
MADRID	AVENIDA CIUDAD DE BARCELONA, 61	1,769	1,589	<a href="#">Ruta</a>



Y aquí otro ejemplo para ver solo las gasolineras del vehículo que queremos





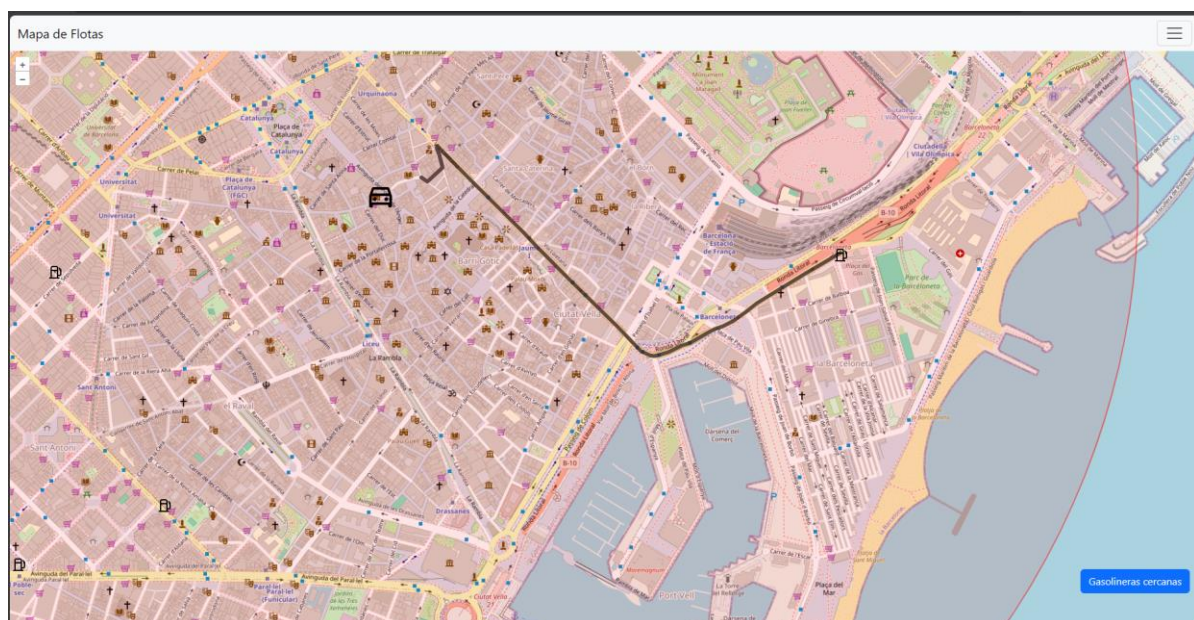
Mapa de Flotas

Gasolineras dentro del radio

ABC1233

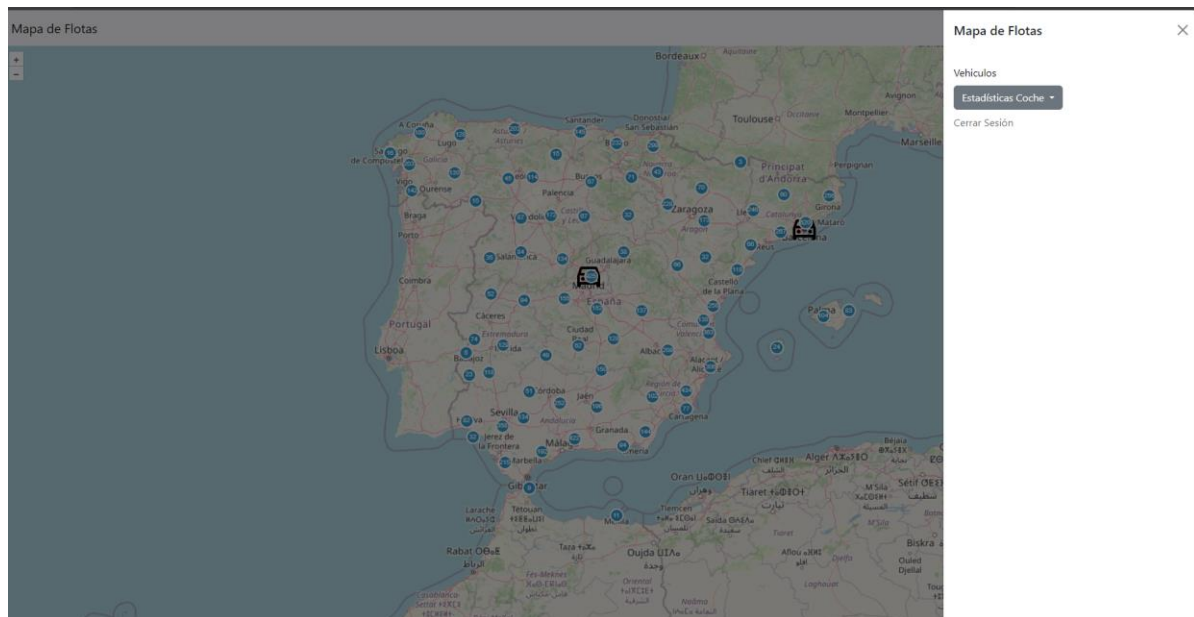
Localidad	Dirección	Precio Gasolina 95	Precio Gasóleo A	Ruta
BARCELONA	CALLE DOCTOR AIGUADER, 50	No disponible	No disponible	<a href="#">Ruta</a>
BARCELONA	CALLE TAULAT 15 ESQUINA CIUDAD DE GRANADA	1,549	1,439	<a href="#">Ruta</a>
BARCELONA	AVENIDA BOGATELL, 49	1,779	1,589	<a href="#">Ruta</a>
BARCELONA	AVENIDA DEL LITORAL, 49	1,789	1,645	<a href="#">Ruta</a>
BARCELONA	CALLE PUJADES, 154	1,799	1,629	<a href="#">Ruta</a>
BARCELONA	Almogavares, 66	1,719	1,479	<a href="#">Ruta</a>
BARCELONA	CALLE PERE IV, 79	1,528	1,417	<a href="#">Ruta</a>
BARCELONA	CL. ALMOGAVARES 159 - 161	1,769	1,679	<a href="#">Ruta</a>
BARCELONA	CALLE BALMES, 288	1,799	1,609	<a href="#">Ruta</a>
BARCELONA	CALLE BADAJOZ, 108	1,485	1,365	<a href="#">Ruta</a>
BARCELONA	AVINGUDA DEL PARALELO, 132	1,669	1,479	<a href="#">Ruta</a>
BARCELONA	RONDA SAN PABLO, 39	1,739	1,595	<a href="#">Ruta</a>
BARCELONA	CALLE CASANOVA, 89	1,799	1,609	<a href="#">Ruta</a>
BARCELONA	CASANOVA, 23	1,779	1,579	<a href="#">Ruta</a>

Como se puede observar tendremos un botón ruta, con este botón lo que conseguiremos es que se ampliar el mapa a la ruta del coche a la gasolinera cercana, marcando esta ruta entre ambos puntos



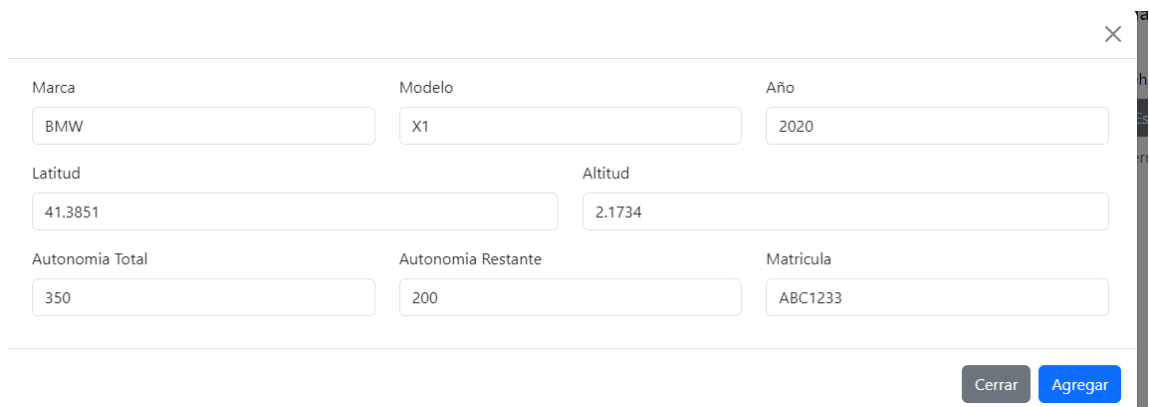
Otra de las opciones que tenemos en esta ventana es abrir un menú lateral clicando sobre el botón del navbar en la derecha





En este menú lateral encontraremos 3 opciones:

- Clicando a vehículos accederemos a la ventana con todos los vehículos asociados a la persona logada.
- En estadísticas coche encontramos un select para seleccionar uno de los vehículos y sacar un modal con los datos del coche seleccionado.



Marca	Modelo	Año
BMW	X1	2020
Latitud	Altitud	
41.3851	2.1734	
Autonomía Total	Autonomía Restante	Matrícula
350	200	ABC1233

Cerrar Agregar

- En cerrar sesión, volveremos al login cerrando correctamente la sesión de firebase, porque si intentamos volver al login por otro medio como dando a la flecha para volver al lado del símbolo de recarga, nos volverá a redirigir instantáneamente a la ventana del mapa.

En cuanto a la ventana de vehículos encontramos la siguiente tabla

Crear

Edit

Eliminar

9

Search

<input type="checkbox"/>	ID Coche	Latitud	Longitud	Marca	Modelo	Año	Autonomía Total	Autonomía Restante	ID Usuario	Matricula
<input type="checkbox"/>	1	41.3851	2.1734	BMW	X1	2020	350	200	H66yfoEEQkbJjcc001iPhRX0Ltf2	ABC1233
<input type="checkbox"/>	2	40.4168	-3.7038	Honda	CR-V	2019	450	250	H66yfoEEQkbJjcc001iPhRX0Ltf2	DEF456

Showing 1 to 2 of 2 rows

En esta tabla, podemos encontrar tablas específicas escribiendo el dato que queramos en el buscador de arriba a la izquierda, además a la derecha de este buscador encontramos un botón que nos permitirá seleccionar los campos que queremos o no queremos visualizar en la tabla

<div> <div>Search</div> <div></div> </div>									
<input type="checkbox"/>	Latitud	Longitud	Marca	Autonomía Total	ID Usuario	<div> <div> <input type="checkbox"/> ID Coche           <input checked="" type="checkbox"/> Latitud           <input checked="" type="checkbox"/> Longitud           <input type="checkbox"/> Marca           <input type="checkbox"/> Modelo           <input type="checkbox"/> Año           <input checked="" type="checkbox"/> Autonomía Total           <input type="checkbox"/> Autonomía Restante           <input checked="" type="checkbox"/> ID Usuario           <input checked="" type="checkbox"/> Matrícula         </div> </div>			
<input type="checkbox"/>	41.3851	2.1734	BMW	350	H66yfoEEQkbJjcc001iPhRX0Ltf2				
<input type="checkbox"/>	40.4168	-3.7038	Honda	450	H66yfoEEQkbJjcc001iPhRX0Ltf2				

Showing 1 to 2 of 2 rows

Arriba a la derecha de la ventana se dispone de un botón para regresar al mapa.

Otras funciones de esta tabla son crear, editar y eliminar coches, principalmente solo vendrá activo el botón de crear el cual nos mostrara el siguiente modal para crear un vehículo.

### Nuevo Vehículo ✕

Marca

Modelo

Año

Latitud

Longitud

Autonomía Total

Autonomía Restante

Matrícula

Cerrar

Agregar

Cuando vamos a seleccionar un modelo nos requiere que seleccionemos una marca y una vez seleccionada esta marca se cargaran los modelos correspondientes a la Marca seleccionada. Como no se dispone de sensores se introduce la latitud y longitud a mano.

Si queremos editar o eliminar en la tabla se debe de seleccionar alguno de los recuadros situados en la primera columna de esta tabla.

Crear

Editar

Eliminar

Search

☐

Latitud

☐

Longitud

☐

Marca

☐

Autonomía Total

☐

ID Usuario

☐

Matrícula

☒

41.3851

☐

2.1734

☐

BMW

☐

350

☐

H66yf0EEQkbJjC001PhRX0Lf2

☐

ABC1233

☐

40.4168

☐

-3.7038

☐

Honda

☐

450

☐

H66yf0EEQkbJjC001PhRX0Lf2

☐

DEF456

Showing 1 to 2 of 2 rows

pág. 35

Esto permite editar la fila seleccionada, abriendo el siguiente modal con los datos de este registro

### Nuevo Vehículo

Marca

Modelo

Año

BMW

3 Series

2020

Latitud

Longitud

41.3851

2.1734

Autonomía Total

Autonomía Restante

Matrícula

350

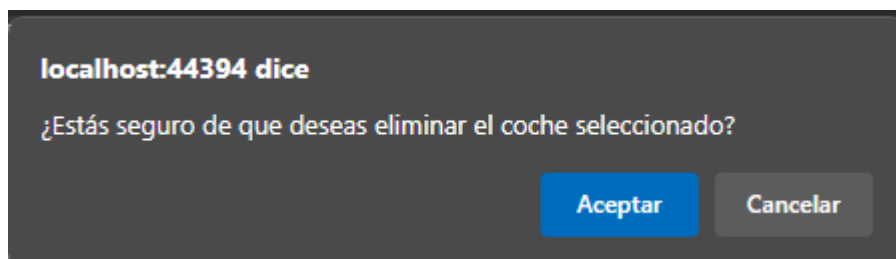
200

ABC1233

Cerrar

Agregar

Si por ejemplo queremos eliminar, se podrán seleccionar todas las filas que se quiera, pudiendo eliminar la cantidad de registro que se desee, en el caso de seleccionar más de una fila, el botón de editar dejara de funcionar. Si queremos eliminar deberemos de confirmar un mensaje que nos aparecerá en la ventana.



Por último, si la tabla contiene una cantidad superior a x registros, se establece una paginación de los registros.

Esto serian todas las funcionalidades de la aplicación.

## **5.2 Manual de instalación**

### Plataforma Tecnológica:

La aplicación se despliega en un entorno local utilizando un servidor local o una máquina virtual que cumpla con los requisitos de hardware y software necesarios.

### Requisitos de Hardware:

Sistema operativo compatible (por ejemplo, Windows, Linux).

Recursos de memoria y almacenamiento adecuados para ejecutar la aplicación de manera óptima.

### Requisitos de Software:

Visual Studio Professional 2022 o versión posterior.

SQL Server para la gestión de la base de datos.

Git para el control de versiones.

Firebase para la autenticación de usuarios.

### Instrucciones de Instalación:

#### Configuración del Entorno:

Instala Visual Studio Professional 2022 en tu sistema.

Configura un servidor local o crea una máquina virtual con el sistema operativo compatible.

Asegúrate de tener los permisos necesarios para instalar software en el servidor local o la máquina virtual.

### Despliegue del Backend:

Clona el repositorio del backend de la aplicación desde Git.

Abre el proyecto en Visual Studio y compílalo para asegurarte de que no haya errores.

Publica el proyecto en el servidor local o la máquina virtual utilizando Visual Studio.

Configura la aplicación en el servidor web para que esté disponible para recibir solicitudes HTTP.

### Despliegue del Frontend:

Clona el repositorio del frontend de la aplicación desde Git.

Copia los archivos estáticos del frontend en el servidor web, asegurando que estén accesibles desde el navegador.

#### Configuración de la Base de Datos:

Instala y configura SQL Server en el servidor local o la máquina virtual.

Crea una nueva base de datos para la aplicación y asegúrate de tener los permisos necesarios para acceder a ella.

Ejecuta los scripts de creación de tablas y datos iniciales proporcionados con la aplicación para configurar la base de datos.

#### Integración con APIs Externas:

Configura la integración con la API del Gobierno Español para las Gasolineras, asegurando que la aplicación pueda acceder a los datos actualizados sobre las estaciones de servicio en España.

#### Puesta en Marcha:

Realiza pruebas de funcionalidad para verificar que todas las funcionalidades de la aplicación funcionen según lo esperado.

Realiza pruebas de rendimiento y seguridad para evaluar la capacidad de la aplicación para manejar cargas de trabajo esperadas y asegurar la protección de los datos del usuario.

## **6 Conclusiones y posibles ampliaciones**

En conclusión, considero que es una aplicación muy funcional, en la cual obviamente habría que introducir mejoras como las que ahora comentaré, pero hasta el momento ya me ha parecido muy útil. Creo que recoge muchas funcionalidades que no había visto nunca y me parecen bastante interesantes.

Ha sido un proceso duro, pero entretenido, con mucho aprendizaje y cada vez aumentando mas y mas las ganas de aprender.

#### Posibles Ampliaciones

##### **Expansión Geográfica**

Ampliar la cobertura de la aplicación a otros países integrando datos locales de gasolineras y rutas.

Añadir soporte multilenguaje para una mayor accesibilidad.

### **Mantenimiento Preventivo**

Incorporar módulos de mantenimiento preventivo que avisen sobre revisiones y servicios necesarios.

Implementar alertas automáticas para revisiones y servicios técnicos.

### **Mejoras en la Interfaz de Usuario**

Utilizar tecnologías modernas para una experiencia de usuario más dinámica y personalizada.

### **Optimización del Rendimiento**

Migrar la infraestructura a la nube para mejorar la escalabilidad.

Optimizar las consultas a la base de datos para aumentar la velocidad.

### **Integración con Otros Sistemas**

Integrar la aplicación con sistemas ERP para una gestión más completa y automatizada.

Conectar con sistemas de gestión de mantenimiento computarizado (CMMS) para una mejor planificación y ejecución de tareas de mantenimiento.

### **Monitoreo en Tiempo Real con Sensores**

Implementar el uso de sensores para controlar en tiempo real la ubicación de los vehículos en intervalos de 15 segundos, mejorando la precisión y actualidad de los datos de localización.

## **7 Bibliografía**

- GMV. (s.f.). *Manuales de la empresa*. GMV.

- Tutora de prácticas. (s.f.). *Ayuda ofrecida durante las prácticas*.
- OpenAI. (2023). *ChatGPT*. Recuperado de <https://chatgpt.com>
- Stack Overflow. (s.f.). Recuperado de <https://stackoverflow.com/>
- YouTube. (2023, 10 de abril). *Introduction to Bootstrap* [Video]. Recuperado de [https://www.youtube.com/watch?v=2n5\\_9boyGWw](https://www.youtube.com/watch?v=2n5_9boyGWw)
- YouTube. (2023, 15 de mayo). *Advanced Bootstrap Techniques* [Video]. Recuperado de [https://www.youtube.com/watch?v=y9xTe\\_zQdY](https://www.youtube.com/watch?v=y9xTe_zQdY)
- Google. (2023). *Bootstrap - Google Search*. Recuperado de <https://www.google.com/search?client=firefox-b-d&q=bootstrap>
- Canva. (2023). *Canva - Diseño gráfico y editor de fotos* [Software]. Recuperado de <https://www.canva.com>

## 8 Anexos

Script de la base de datos – control\_de\_flotas

USE [master]

GO

```
/****** Object: Database [control_de_flotas] Script Date: 23/05/2024 22:27:55
*****/
```

```
CREATE DATABASE [control_de_flotas]
```

```
CONTAINMENT = NONE
```

```
ON PRIMARY
```

```
( NAME = N'control_de_flotas', FILENAME = N'C:\Users\trcas1\control_de_flotas.mdf' ,
  SIZE = 8192KB , MAXSIZE = UNLIMITED, FILEGROWTH = 65536KB )
```

```
LOG ON
```

```
( NAME = N'control_de_flotas_log', FILENAME =
  N'C:\Users\trcas1\control_de_flotas_log.ldf' , SIZE = 8192KB , MAXSIZE = 2048GB ,
  FILEGROWTH = 65536KB )
```

```
WITH CATALOG_COLLATION = DATABASE_DEFAULT, LEDGER = OFF
```

GO

```
IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))
```

```
begin
```

```
EXEC [control_de_flotas].[dbo].[sp_fulltext_database] @action = 'enable'
```

```
end
```



GO

```
ALTER DATABASE [control_de_flotas] SET ANSI_NULL_DEFAULT OFF
```

GO

```
ALTER DATABASE [control_de_flotas] SET ANSI_NULLS OFF
```

GO

```
ALTER DATABASE [control_de_flotas] SET ANSI_PADDING OFF
```

GO

```
ALTER DATABASE [control_de_flotas] SET ANSI_WARNINGS OFF
```

GO

```
ALTER DATABASE [control_de_flotas] SET ARITHABORT OFF
```

GO

```
ALTER DATABASE [control_de_flotas] SET AUTO_CLOSE ON
```

GO

```
ALTER DATABASE [control_de_flotas] SET AUTO_SHRINK OFF
```

GO

```
ALTER DATABASE [control_de_flotas] SET AUTO_UPDATE_STATISTICS ON
```

GO

```
ALTER DATABASE [control_de_flotas] SET CURSOR_CLOSE_ON_COMMIT OFF
```

GO

```
ALTER DATABASE [control_de_flotas] SET CURSOR_DEFAULT GLOBAL
```

GO

```
ALTER DATABASE [control_de_flotas] SET CONCAT_NULL_YIELDS_NULL OFF
```

GO

```
ALTER DATABASE [control_de_flotas] SET NUMERIC_ROUNDABORT OFF
```

GO

```
ALTER DATABASE [control_de_flotas] SET QUOTED_IDENTIFIER OFF
```

GO

```
ALTER DATABASE [control_de_flotas] SET RECURSIVE_TRIGGERS OFF  
GO
```

```
ALTER DATABASE [control_de_flotas] SET ENABLE_BROKER  
GO
```

```
ALTER DATABASE [control_de_flotas] SET AUTO_UPDATE_STATISTICS_ASYNC OFF  
GO
```

```
ALTER DATABASE [control_de_flotas] SET DATE_CORRELATION_OPTIMIZATION OFF  
GO
```

```
ALTER DATABASE [control_de_flotas] SET TRUSTWORTHY OFF  
GO
```

```
ALTER DATABASE [control_de_flotas] SET ALLOW_SNAPSHOT_ISOLATION OFF  
GO
```

```
ALTER DATABASE [control_de_flotas] SET PARAMETERIZATION SIMPLE  
GO
```

```
ALTER DATABASE [control_de_flotas] SET READ_COMMITTED_SNAPSHOT OFF  
GO
```

```
ALTER DATABASE [control_de_flotas] SET HONOR_BROKER_PRIORITY OFF  
GO
```

```
ALTER DATABASE [control_de_flotas] SET RECOVERY SIMPLE  
GO
```

```
ALTER DATABASE [control_de_flotas] SET MULTI_USER  
GO
```

```
ALTER DATABASE [control_de_flotas] SET PAGE_VERIFY CHECKSUM  
GO
```

```
ALTER DATABASE [control_de_flotas] SET DB_CHAINING OFF  
GO
```

```
ALTER DATABASE [control_de_flotas] SET FILESTREAM( NON_TRANSACTED_ACCESS = OFF )  
GO
```

```
ALTER DATABASE [control_de_flotas] SET TARGET_RECOVERY_TIME = 60 SECONDS
GO
```

```
ALTER DATABASE [control_de_flotas] SET DELAYED_DURABILITY = DISABLED
GO
```

```
ALTER DATABASE [control_de_flotas] SET ACCELERATED_DATABASE_RECOVERY = OFF
GO
```

```
ALTER DATABASE [control_de_flotas] SET QUERY_STORE = ON
GO
```

```
ALTER DATABASE [control_de_flotas] SET QUERY_STORE (OPERATION_MODE = READ_WRITE,
CLEANUP_POLICY = (STALE_QUERY_THRESHOLD_DAYS = 30), DATA_FLUSH_INTERVAL_SECONDS = 900,
INTERVAL_LENGTH_MINUTES = 60, MAX_STORAGE_SIZE_MB = 1000, QUERY_CAPTURE_MODE = AUTO,
SIZE_BASED_CLEANUP_MODE = AUTO, MAX_PLANS_PER_QUERY = 200, WAIT_STATS_CAPTURE_MODE =
ON)
GO
```

```
ALTER DATABASE [control_de_flotas] SET READ_WRITE
GO
```

Script tabla coches

```
USE [control_de_flotas]
GO
```

```
/***** Object: Table [dbo].[Coches]    Script Date: 23/05/2024 22:28:50 *****/
SET ANSI_NULLS ON
GO
```

```
SET QUOTED_IDENTIFIER ON
GO
```

```
CREATE TABLE [dbo].[Coches](
    [idCoche] [int] IDENTITY(1,1) NOT NULL,
    [Latitud] [varchar](100) NOT NULL,
    [Longitud] [varchar](100) NOT NULL,
    [Marca] [varchar](100) NOT NULL,
```

```
[Modelo] [varchar](100) NOT NULL,  
[Año] [int] NOT NULL,  
[AutonomiaTotal] [int] NOT NULL,  
[AutonomiaRestante] [int] NOT NULL,  
[idUsuario] [varchar](100) NOT NULL,  
[matricula] [varchar](20) NOT NULL,  
PRIMARY KEY CLUSTERED  
(  
    [idCoche] ASC  
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,  
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON  
[PRIMARY],  
UNIQUE NONCLUSTERED  
(  
    [matricula] ASC  
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,  
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON  
[PRIMARY]  
) ON [PRIMARY]  
GO
```

Script tabla TiposCoches(referido a marcas y modelos)

```
USE [control_de_flotas]  
GO
```

```
/***** Object: Table [dbo].[TiposCoches]    Script Date: 23/05/2024 22:29:29 *****/
```

```
SET ANSI_NULLS ON  
GO
```

```
SET QUOTED_IDENTIFIER ON  
GO
```

```
CREATE TABLE [dbo].[TiposCoches](  
    [id] [int] IDENTITY(1,1) NOT NULL,  
    [marca] [varchar](255) NULL,  
    [modelo] [varchar](255) NULL,  
PRIMARY KEY CLUSTERED  
(  
    [id] ASC
```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO
```

Un anexo a esta memoria es el empleo del archivo web.config, esencial para cargar la connectionString que se asocia a la base de datos que se utiliza en los repository para acceder directamente a esta base de datos.

```
<?xml version="1.0" encoding="utf-8"?>
<!--
  Para obtener más información acerca de cómo configurar la aplicación ASP.NET,
  visite
  https://go.microsoft.com/fwlink/?LinkId=301880
-->
<configuration>
  <appSettings>
    <add key="webpages:Version" value="3.0.0.0" />
    <add key="webpages:Enabled" value="false" />
    <add key="ClientValidationEnabled" value="true" />
    <add key="UnobtrusiveJavaScriptEnabled" value="true" />
  </appSettings>
  <system.web>
    <compilation debug="true" targetFramework="4.7.2" />
    <httpRuntime targetFramework="4.7.2" />
  </system.web>
  <runtime>
    <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
      <dependentAssembly>
        <assemblyIdentity name="Antlr3.Runtime"
publicKeyToken="eb42632606e9261f" />
        <bindingRedirect oldVersion="0.0.0.0-3.5.0.2" newVersion="3.5.0.2" />
      </dependentAssembly>
      <dependentAssembly>
        <assemblyIdentity name="Microsoft.Web.Infrastructure"
publicKeyToken="31bf3856ad364e35" />
        <bindingRedirect oldVersion="0.0.0.0-2.0.1.0" newVersion="2.0.1.0" />
      </dependentAssembly>
    </assemblyBinding>
  </runtime>
</configuration>
```

```

    <assemblyIdentity                                name="Newtonsoft.Json"
publicKeyToken="30ad4fe6b2a6aeed" />
    <bindingRedirect oldVersion="0.0.0.0-12.0.0.0" newVersion="12.0.0.0" />
</dependentAssembly>
<dependentAssembly>
    <assemblyIdentity                                name="System.Web.Optimization"
publicKeyToken="31bf3856ad364e35" />
    <bindingRedirect oldVersion="1.0.0.0-1.1.0.0" newVersion="1.1.0.0" />
</dependentAssembly>
<dependentAssembly>
    <assemblyIdentity name="WebGrease" publicKeyToken="31bf3856ad364e35" />
    <bindingRedirect                                oldVersion="0.0.0.0-1.6.5135.21930"
newVersion="1.6.5135.21930" />
</dependentAssembly>
<dependentAssembly>
    <assemblyIdentity                                name="System.Web.Helpers"
publicKeyToken="31bf3856ad364e35" />
    <bindingRedirect oldVersion="1.0.0.0-3.0.0.0" newVersion="3.0.0.0" />
</dependentAssembly>
<dependentAssembly>
    <assemblyIdentity                                name="System.Web.WebPages"
publicKeyToken="31bf3856ad364e35" />
    <bindingRedirect oldVersion="1.0.0.0-3.0.0.0" newVersion="3.0.0.0" />
</dependentAssembly>
<dependentAssembly>
    <assemblyIdentity                                name="System.Web.Mvc"
publicKeyToken="31bf3856ad364e35" />
    <bindingRedirect oldVersion="1.0.0.0-5.2.9.0" newVersion="5.2.9.0" />
</dependentAssembly>
</assemblyBinding>
</runtime>
<system.codedom>
    <compilers>
        <compiler                                language="c#;cs;csharp"                extension=".cs"
type="Microsoft.CodeDom.Providers.DotNetCompilerPlatform.CSharpCodeProvider,
Microsoft.CodeDom.Providers.DotNetCompilerPlatform,                Version=2.0.1.0,
Culture=neutral,                PublicKeyToken=31bf3856ad364e35"                warningLevel="4"
compilerOptions="/langversion:default /nowarn:1659;1699;1701" />
        <compiler                                language="vb;vbs;visualbasic;vbscript"                extension=".vb"
type="Microsoft.CodeDom.Providers.DotNetCompilerPlatform.VBCodeProvider,
Microsoft.CodeDom.Providers.DotNetCompilerPlatform,                Version=2.0.1.0,

```

```
Culture=neutral,          PublicKeyToken=31bf3856ad364e35"          warningLevel="4"
compilerOptions="/langversion:default          /nowarn:41008
/define:_MYTYPE=\"Web\"; /optionInfer+" />
</compilers>
</system.codedom>
<connectionStrings>
  <add          name="Flotas"          connectionString="Data
Source=(localdb)\ServerFlotas;Initial          Catalog=control_de_flotas;Integrated
Security=True;" />
</connectionStrings>
</configuration>
```