



Concesionaria Premium Quality

Accede a la demo en <https://premium-quality.alejoide.com>

Descripción del Proyecto

Premium Quality es un proyecto realizado para la materia EDI II - Desarrollo de páginas web, para el Instituto Superior de Formación Docente y Técnica N° 93.

Se requería el desarrollo de una página web para una concesionaria de autos, que permita a los usuarios consultar el catálogo de vehículos disponibles, obtener información de cada vehículo, y contactar a la concesionaria para realizar consultas o agendar visitas.

El proyecto incluye las siguientes funcionalidades principales:

- **Landing Page:** Una página de inicio atractiva que presenta la concesionaria y sus servicios.
- **Catálogo de Vehículos:** Una sección donde los usuarios pueden explorar los vehículos disponibles
- **Ofertas Especiales:** Una sección dedicada a promociones y ofertas especiales en vehículos seleccionados.
- **Financiamiento:** Información sobre opciones de financiamiento disponibles para la compra de vehículos.
- **Formulario de Contacto:** Un formulario para que los usuarios puedan enviar consultas o solicitar información adicional.
En él se pueden adjuntar el vehículo de interés y el plan de financiamiento deseado.
- **Diseño Responsivo:** La página está diseñada para ser accesible y funcional en dispositivos móviles y de escritorio.

Tecnologías Utilizadas

Frontend



- **HTML:** Para la estructura de la página web.
- **CSS:** Para el diseño y estilo visual de la página.
- **JavaScript:** Para agregar interactividad y funcionalidades dinámicas.
- **Astro:** Como framework para construir la página web de manera eficiente.
- **TypeScript:** Para mejorar la calidad del código JavaScript con tipado estático.

Backend



- **Python:** Lenguaje de programación utilizado para el desarrollo del backend.
- **FastAPI:** Framework web para construir APIs rápidas y eficientes.
- **SQLite:** Base de datos ligera para almacenar la información de vehículos y consultas de usuarios.

Despliegue



- **Docker:** Para la contenedorización de la aplicación y facilitar su despliegue.
- **Git Hub Actions:** Para la integración continua y despliegue automático.
- **Servidor Self-Hosted:** La aplicación se despliega en un servidor propio, sin utilizar servicios en la nube.
- **NGINX:** Servidor web utilizado como proxy inverso para manejar las solicitudes HTTP y servir la aplicación.

Estructura del Proyecto

```
.
├── deploy -> Contiene los archivos necesarios para el despliegue de la aplicación
│   ├── config
│   │   ├── backend
│   │   ├── frontend
│   │   └── nginx
│   └── docs -> Documentación del proyecto
├── src -> Código fuente de la aplicación
│   ├── backend -> Código del backend desarrollado con FastAPI
│   │   ├── controllers -> Lógica de controladores de la aplicación
│   │   ├── core -> Configuraciones y funcionalidades centrales
│   │   ├── data -> Manejo de datos y acceso a la base de datos
│   │   ├── models -> Definición de modelos de datos
│   │   ├── routers -> Rutas y endpoints de la API
│   │   ├── static -> Archivos estáticos servidos por el backend
│   │   │   ├── data -> Datos estáticos
│   │   │   └── tests -> Archivos de prueba
│   │   └── templates -> Plantillas HTML para renderizado del lado del servidor
│   ├── frontend -> Código del frontend desarrollado con Astro
│   │   ├── node_modules -> Dependencias del proyecto
│   │   ├── public -> Archivos públicos accesibles directamente
│   │   └── src -> Código fuente del frontend
│   │       ├── assets -> Recursos estáticos
│   │       │   ├── images -> Imágenes y gráficos
│   │       │   └── videos -> Videos promocionales
│   │       ├── components -> Componentes reutilizables de la interfaz de usuario
│   │       ├── core -> Configuraciones y funcionalidades centrales
│   │       ├── data -> Datos estáticos y de configuración
│   │       ├── layouts -> Plantillas de diseño reutilizables
│   │       ├── lib -> Librerías y utilidades
│   │       ├── pages -> Páginas principales del sitio web
│   │       │   ├── api -> Endpoints del frontend
│   │       │   └── cars -> Páginas de catálogo de vehículos
│   │       └── types -> Definiciones de tipos TypeScript
│   └── tests -> Pruebas unitarias y de integración
```