

Ivan Martinez  
20 de junio de 2024

# APEXAUTO

## Proyecto web

**Resumen:** ApexAuto es una aplicación web desarrollada para permitir los usuarios buscar autos por diversas características como marca, color, fecha y precio. Utiliza HTML, CSS, JavaScript, Font Awesome, Python con Flask y MySQL. Esta documentación detalla el diseño del sistema, las tecnologías utilizadas y proporciona guías para la configuración y el despliegue

## Indice

1. Introducción
2. Descripción del Sistema
3. Tecnologías Utilizadas
4. Diseño Detallado
5. Proceso de Desarrollo
6. Configuración y Despliegue
7. Anexos

## Introducción

**Objetivo del Documento:** Proporcionar una guía detallada sobre el diseño de las tecnologías y los procedimientos de configuración y despliegue del proyecto APEXAUTO

**Alcance:** Este documento cubre el desarrollo de la aplicación web usando HTML, CSS, JavaScript, Python con Flask y MySQL, y proporciona instrucciones detalladas para su configuración y despliegue local.

**Visión General del Sistema:** La aplicación web permite a los usuarios buscar y filtrar autos utilizando una interfaz sencilla y fácil de usar, La arquitectura esta compuesta por un frontend en HTML/CSS/JavaScript y un backend en Python con Flask, con una base de datos MySQL para almacenamiento de datos

## Descripción del Sistema

### Arquitectura del sistema:

- **Frontend:** HTML, CSS (Tailwind CSS), JavaScript (Font Awesome para iconos).
- **Backend:** Python con Flask.
- **Base de Datos:** MySQL.

### Componentes Principales:

1. **Interfaz de Usuario (UI):** Construida con HTML, CSS y JavaScript
2. **API Backend:** Implementada con Flask
3. **Base de Datos:** MySQL para el almacenamiento de datos.

## Tecnologías Utilizadas

### Front-end:

- **HTML5:** Para la estructura de las paginas web
- **CSS3:** Para el estilo y el diseño con Tailwind CSS
- **JavaScript:** Para la interactividad, incluyendo Font Awesome para iconos

### Back-End:

- **Python:** Lenguaje de Programación
- **Flask:** Freamwork web ligero para Python
- **Jinja2:** Motor de plantillas usando con Flask para renderizar HTML dinámicamente.

### Base de Datos:

- **MySQL:** Base de datos relacional utilizada para almacenar la información de autos.

## Diseño Detallado

### Diagrama de Arquitectura:

- El frontend envía solicitudes el backend de Flask.
- Flask procesa las solicitudes y se comunica con la base de datos MySQL

- Flask devuelve las respuestas al frontend.

### **Modelo de Autos:**

#### **• Tabla de Autos:**

- id (INT, PK)
- Marca (VARCHAR)
- Modelo (VARCHAR)
- Precio (INT)
- Fecha (DATE)
- Color (VARCHAR)
- Img (VARCHAR)

## **Proceso de Desarrollo**

**Metodología:** Desarrollo iterativo basado en objetos específicos

## **Configuración y Despliegue**

### **Requisitos del Sistema:**

- **Backend:** Python 3.x, Flask.
- **Base de Datos:** MySQL 5.7+
- **Frontend:** Navegador moderno compatible con HTML5, CSS3 y JavaScript.

### **Instrucciones de Configuración:**

#### **1. Descargar el código fuente**

#### **2. Configurar entorno virtual:**

```
python3 -m venv venv  
source venv/bin/activate  
pip install -r requirements.txt
```

- #### **3. Configura la base de datos:** Desde la terminal o MySQL Workbench importa la base de datos que esta en GitHub

## **Proceso de Despliegue:**

### **1. Iniciar la aplicación Python:**

```
python3 apex.py
```

## **Anexos**

### **Glosario:**

- **HTML:** Lenguaje de marcado para la creación de paginas web
- **CSS:** Lenguaje de estilo para la presentación de documentos HTML
- **JavaScript:** Lenguaje de programación para la creación de contenido web dinámico.
- **Flask:** Micro framework web para Python.
- **MySQL:** Sistema de gestión de bases de datos relacional.

### **Referencias:**

- Documentación de Flask
- Documentación de Tailwind CSS
- Documentación de Font Awesome