

Sistema de Gestión Vehicular

Módulo: 06_Vehiculo_CRUD – Parte III

1 06_Vehiculo_CRUD_ParteIII

1.1 Descripción General

El módulo **06_Vehiculo_CRUD – Parte III** implementa el CRUD completo del componente Vehículo dentro del Sistema de Matriculación. En esta fase el sistema permite realizar todas las operaciones fundamentales sobre la tabla **vehiculo** en la base de datos: crear registros, listar, consultar detalle, actualizar y eliminar.

La solución se desarrolla en Python con **Flask** y MySQL, aplicando programación orientada a objetos mediante la clase *Vehiculo*. La interfaz se presenta en HTML utilizando **Bootstrap**, lo que mejora la experiencia del usuario y la organización visual del sistema.

1.2 Objetivo

Implementar un CRUD completo para el manejo de vehículos, permitiendo:

- Crear nuevos vehículos (INSERT).
- Listar vehículos registrados (SELECT).
- Consultar el detalle de un vehículo (SELECT por ID).
- Actualizar información de un vehículo (UPDATE).
- Eliminar un vehículo (DELETE), respetando restricciones de integridad referencial.

1.3 Conexión a Base de Datos

La aplicación establece conexión hacia MySQL utilizando parámetros de configuración del sistema (servidor, usuario, contraseña y base de datos). Se configura el juego de caracteres en UTF-8 para el manejo correcto de información.

1.4 Operaciones CRUD Implementadas

1.4.1. Listado de Vehículos (Read)

El método `get_list()` genera el listado general de vehículos, consultando la tabla *vehiculo* y relacionándola con *marca* y *color*. Se presentan campos como: placa, marca, color, año y avalúo, incluyendo botones de acción para:

- Detalle
- Editar
- Borrar

:contentReference[oaicite:2]index=2

1.4.2. Formulario Nuevo / Editar (Create / Update)

El método `get_form()` permite cargar un formulario Bootstrap para:

- Registrar un nuevo vehículo (modo *new*).
- Editar un vehículo existente (modo *update*).

El formulario incluye campos: placa, marca, motor, chasis, combustible, año, color, foto y avalúo. Además, utiliza listas desplegables cargadas desde base de datos para marca y color, y un combo dinámico para año. :contentReference[oaicite:3]index=3

1.4.3. Guardar Información (INSERT / UPDATE)

La aplicación procesa peticiones **POST** al presionar el botón *Guardar*. Dependiendo del tipo de operación:

- **INSERT:** registra un nuevo vehículo en la base de datos.
- **UPDATE:** actualiza los datos del vehículo seleccionado mediante su ID.

Luego de guardar, el sistema devuelve un mensaje de confirmación y permite regresar al listado. :contentReference[oaicite:4]index=4

1.4.4. Detalle del Vehículo (Read)

El método `get_detail_vehiculo()` muestra toda la información del vehículo seleccionado, incluyendo su imagen y el cálculo del valor de matrícula (10% del avalúo). :contentReference[oaicite:5]index=5

1.4.5. Eliminar Vehículo (Delete) con Control de Integridad

El método `delete_vehiculo()` elimina el registro únicamente si no existen matrículas asociadas al vehículo. Esto evita errores de integridad referencial y asegura consistencia en la base de datos. :contentReference[oaicite:6]index=6

1.5 Manejo de Navegación y Operaciones

Las acciones del sistema se ejecutan por enlaces que codifican la operación y el ID del registro, permitiendo llamar funcionalidades como:

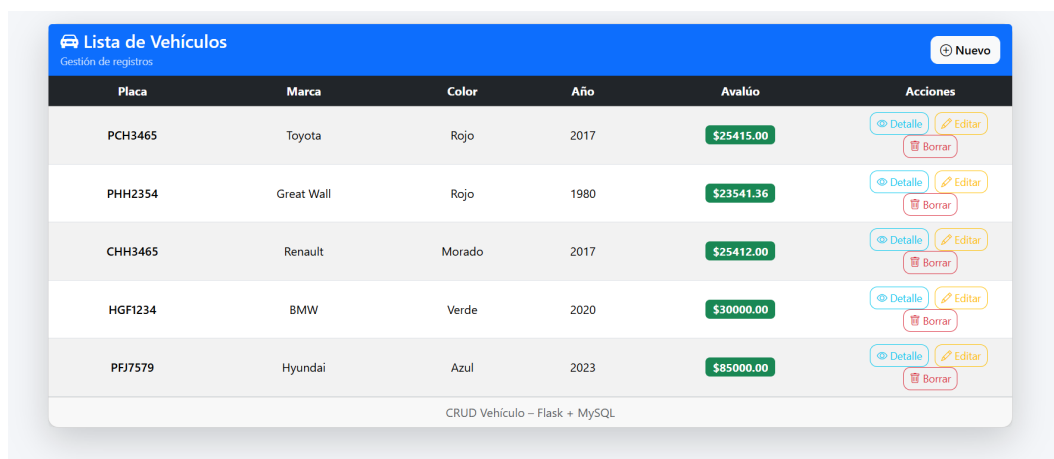
- Nuevo
- Editar
- Detalle
- Borrar

Esto permite mantener el control del flujo del CRUD de manera ordenada y segura dentro del sistema.

1.6 Resultados Esperados

Al ejecutar el módulo, el sistema debe permitir:

- Visualizar el listado completo de vehículos.
- Registrar vehículos nuevos mediante formulario.
- Modificar registros existentes.
- Consultar el detalle de un vehículo.
- Eliminar registros, respetando la relación con matrículas.



The screenshot displays a web interface titled "Lista de Vehículos" with a subtitle "Gestión de registros". It features a table with columns for "Placa", "Marca", "Color", "Año", "Avalúo", and "Acciones". The "Avalúo" column shows values in green boxes. The "Acciones" column contains links for "Detalle", "Editar", and "Borrar". A "Nuevo" button is located in the top right corner. The footer text reads "CRUD Vehículo - Flask + MySQL".

Placa	Marca	Color	Año	Avalúo	Acciones
PCH3465	Toyota	Rojo	2017	\$25415.00	Detalle Editar Borrar
PHH2354	Great Wall	Rojo	1980	\$23541.36	Detalle Editar Borrar
CHH3465	Renault	Morado	2017	\$25412.00	Detalle Editar Borrar
HGF1234	BMW	Verde	2020	\$30000.00	Detalle Editar Borrar
PFJ7579	Hyundai	Azul	2023	\$85000.00	Detalle Editar Borrar

Figura 1: Listado de vehículos con acciones CRUD

Nuevo Vehículo
Formulario de registro

← Regresar

Placa:

Marca:

Color:

Motor:

Chasis:

Combustible: ☐ Gasolina ☐ Diesel ☐ Eléctrico

Año:

Foto: Sin archivos seleccionados

Avalúo:


Actual: Sin foto

GUARDAR Cancelar

Figura 2: Formulario para nuevo y actualización de vehículo

Datos del Vehículo

← Regresar



Placa	CHH3465
Marca	Renault
Motor	edcdede
Chasis	Dedede
Combustible	Gasolina
Año	2017
Color	Morado
Avalúo	\$25412.00 USD
Valor Matrícula	\$2541.20 USD

← Regresar

Figura 3: Detalle del vehículo con cálculo de matrícula

1.7 Importancia del Módulo

Este módulo completa la funcionalidad del componente Vehículo, implementando el CRUD real con base de datos y una interfaz amigable. Además, incorpora control de restricciones (vehículos con matrículas asociadas no pueden eliminarse), lo cual garantiza integridad y confiabilidad en el sistema.