

Sistema de matriculación vehicular.

SIVEM

Grupo 4

Escuela Politécnica Nacional

Ingeniería en Software

Programación 1

20/07/2024



Índice

1. **Resumen**
2. **Introducción**
3. **Objetivos**
 - 3.1. Objetivo general
 - 3.2. Objetivos específicos
4. **Justificación**
5. **Desarrollo del sistema**
 - 5.1. Tecnologías utilizadas
 - 5.2. Estructura general del sistema
 - 5.2.1. Módulo de Login
 - 5.2.2. Menú principal
 - 5.2.3. Registro de vehículo
 - 5.2.4. Búsqueda de vehículo
 - 5.2.5. Lista general
 - 5.2.6. Cierre de sesión
 - 5.3. Validaciones implementadas
 - 5.4. Manejo de archivos
6. **Resultados**
7. **Conclusiones**
8. **Recomendaciones**
9. **Glosario de términos**

1. Resumen

El sistema de matriculación vehicular SIVEM, es un sistema informático desarrollado en lenguaje C, orientado a facilitar la gestión de vehículos mediante funcionalidades clave: el registro del vehículo, el cálculo automatizado de matrícula según normativas vigentes y el seguimiento de las revisiones técnicas obligatorias.

2. Introducción

SIVEM implementa un sistema de gestión vehicular para el registro del vehículo y el calculo del valor a pagar para la matriculación vehicular, destinado para el uso de los trabajadores a cargo de dicho proceso como para el publico en general.

3. Objetivos

3.1 Objetivo general:

Diseñar e implementar un sistema de gestión vehicular que permita registrar vehículos, calcular el monto correspondiente al pago de la matrícula según normativa vigente y verificar el cumplimiento de las tres revisiones técnicas obligatorias anuales.

3.2 Objetivos específicos:

- Crear un módulo que permita el registro y la validación de los datos correspondientes al vehículo, así como los datos del usuario.
- Aplicar normativo vigente para el calculo del valor a pagar por la matriculación vehicular.
- Generar un archivo de texto sobre el valor a pagar de la matricula.

4. Justificación

El proceso de matriculación vehicular, a menudo manual o con sistemas desactualizados, puede ser propenso a errores y lentitud. Este proyecto aborda la necesidad de un sistema automatizado que agilice estas operaciones, garantizando precisión y eficiencia en la gestión de la información vehicular. El desarrollo se enfocó en crear una solución robusta y bien estructurada que sirva como una base para futuros sistemas más complejos.

5. Desarrollo del sistema.

5.1 Tecnologías utilizadas

El presente proyecto fue diseñado en el IDE Zinjai, el cual permite ejecutar el lenguaje de programación C, aplicando conocimientos aprendidos en el semestre.

5.2 Estructura general del sistema.

El sistema esta estructurado por una serie de módulos:

5.2.1 Modulo de login.

El módulo de login es el punto de entrada del sistema y se encarga de gestionar el acceso de los usuarios. Proporciona las siguientes funcionalidades principales:

- **Inicio de sesión** (login).
- **Registro de nuevo usuario.**
- **Salir**

1. Registro de usuario

Cuando un usuario selecciona la opción de registrarse, el sistema solicita ingresar un nombre de usuario y una contraseña. Se aplican las siguientes restricciones:

- El nombre de usuario debe tener hasta 25 caracteres.
- La contraseña debe tener un máximo de 15 caracteres.

Si alguno de los campos excede estos límites, el sistema muestra un mensaje de error y retorna al menú principal después de una breve espera.

2. Inicio de sesión

Si el usuario selecciona la opción de iniciar sesión, el sistema solicita sus credenciales previamente registradas. Se verifica que los datos coincidan exactamente con los almacenados.

- Si las credenciales son correctas, se otorga el acceso al menú principal.
- Si los datos no coinciden, se muestra un mensaje de “usuario o contraseña incorrectos” y se retorna al menú.

3. Salida del sistema

La tercera opción permite cerrar completamente el programa.

```
printf("\n--- Inicio de Sesion ---\n");  
  
limpiar_buffer(); // Limpiar el buffer de entrada antes de pedir el nombre de usuario  
  
printf("Usuario: ");  
  
len = strlen(nombre_input);  
if (len > 0 && nombre_input[len - 1] == '\n') {  
    nombre_input[len - 1] = '\0';  
} else {  
    limpiar_buffer();  
}  
  
if (strlen(nombre_input) > MAX_LONGITUD_USUARIO) {  
    printf("\nError: El nombre de usuario excede el maximo de %d caracteres.\n", MAX_LONGITUD_USUARIO);  
    esperar_3segundos();  
    return 0;  
}
```

Figura 1: Fragmento del código que pide y valida las credenciales del usuario.

5.2.2 Menu principal

El menú principal del sistema se muestra inmediatamente después de que el usuario inicia sesión correctamente. Desde esta interfaz, el usuario puede acceder a las distintas funcionalidades disponibles del sistema.

El sistema valida que el usuario ingrese un carácter numérico válido y que la opción seleccionada esté dentro del rango permitido. En caso de que el usuario ingrese una letra, símbolo u opción no disponible, se muestra un mensaje de error y se solicita nuevamente una entrada válida.

```

//declaramos la funcion mostrarMenu

void mostrarMenu() {
    printf("\n=== SISTEMA SIVEM ===\n");
    printf("1. Registrar vehiculo\n");
    printf("2. Buscar vehiculo\n");
    printf("3. Lista de vehiculos\n");
    printf("4. Salir\n");
    printf("Seleccione opcion: ");
}

```

Figura 2: Fragmento de código que muestra el menú en pantalla.

```

void procesarOpcion(int opcion) {    //recibe una opcion numerica
    switch(opcion) { //evaluar el valor de la opcion
    case 1:
        limpiar_pantalla();
        limpiar_buffer();
        registrarVehiculo();
        break;
    case 2:
        limpiar_pantalla();
        limpiar_buffer();
        buscarVehiculoPorPlaca();
        break;
    case 3:
        limpiar_pantalla();
        limpiar_buffer();
        listaVehiculos();
        break;
    case 4:
        printf("\nSaliendo del programa.\n");
        break;
    default:
        limpiar_pantalla();
        limpiar_buffer();
        printf("\nOpcion no valida.Ingrese otra opcion.\n");
    }
}

```

Figura 3: Función creada para procesar la opción que el usuario ingrese.

5.2.3 Registro de vehículo

El módulo de registro vehicular permite al usuario ingresar todos los datos necesarios para matricular un vehículo en el sistema. Este módulo se encarga de capturar, validar y almacenar la información correspondiente, además de calcular automáticamente el valor a pagar por la matrícula.

Datos requeridos

Al ejecutar este módulo, el sistema solicita al usuario los siguientes campos:

- **Número de cédula del propietario** (exactamente 10 dígitos numéricos).
- **Año de fabricación del vehículo** (entre 1990 y 2025).
- **Placa del vehículo** (formato: tres letras, un guion, cuatro números, ej. *ABC-1234*).
- **Avalúo del vehículo** (valor máximo permitido: \$500,000).
- **Tipo de vehículo** (selección entre: particular, motocicleta, transporte público).
- **Cantidad de revisiones técnicas realizadas** (deben ser 3 para continuar).
- **Presencia de multas** (sí o no).

Validaciones implementadas

Este módulo incluye múltiples validaciones para asegurar la integridad de los datos ingresados:

- Solo se aceptan **números** para cédula, año, avalúo y revisiones.
- Se verifica que la placa cumpla con el formato exacto.
- Se limita la entrada del avalúo a un valor razonable.
- Se impide continuar con el proceso si las **revisiones técnicas** no están completas ($\neq 3$).
- Se verifica que no haya errores de formato o tipo de dato.

Si alguno de los datos es inválido, el sistema muestra un mensaje de error y solicita que se reingrese la información correctamente.

Cálculo de la matrícula

Una vez validados los datos, el sistema calcula el valor de la matrícula aplicando las siguientes reglas:

- A partir del avalúo del vehículo, se aplica un porcentaje base.
- Si hay **multas**, se suman como recargos.

```

// Solicita al usuario una cedula valida
void pedirCedula(char cedula[]) {
    int valido = 0;
    while (!valido) {
        printf("Ingrese cedula (10 digitos): ");
        scanf("%s", cedula);
        limpiar_buffer();
        if (strlen(cedula) != 10 || !esNumerica(cedula)) {
            limpiar_pantalla();
            printf("Error: la cedula debe tener exactamente 10 digitos numericos.\n");
        } else {
            valido = 1;
        }
    }
}

```

Figura 4: Parte del modulo de registro. FUncion que valida si la cedula ingresada tiene exactamente los diez dígitos numéricos.

5.2.4 Búsqueda de vehículo

Este módulo permite al usuario buscar un vehículo previamente registrado mediante el ingreso de su número de placa. Validando que el formato ingresado sea el mismo que se utilizó para el previo registro. Permitiendo la opción de imprimir el comprobante del vehículo que ha sido buscado.

```

// Busca un vehiculo por placa
void buscarVehiculoPorPlaca() {
    char placa[10];
    int encontrado = 0;

    printf("\n--- Buscar Vehiculo por Placa ---\n");
    printf("Ingrese la placa (formato ABC-1234): ");
    scanf("%9s", placa);
    limpiar_buffer();
    convertirMayusculas(placa);

    for (int i = 0; i < total; i++) {
        if (strcmp(lista[i].placa, placa) == 0) {
            printf("\nVehiculo encontrado:\n");
            mostrarComprobante(lista[i]);
            generarComprobante(lista[i]);
            encontrado = 1;
            break;
        }
    }

    if (!encontrado) {
        printf("Vehiculo con placa %s no encontrado.\n", placa);
    }
}

```

Figura 5: FUncion que realiza la búsqueda y muestra la información del vehículo.

5.2.5 Lista general

El módulo de lista general permite visualizar en pantalla la información completa de todos los vehículos registrados en el sistema hasta el momento.

Al seleccionar esta opción desde el menú principal, el sistema accede al archivo donde se guardan los datos (vehiculos.txt) y recorre línea por línea cada vehículo registrado.

Al finalizar la visualización de la lista, el sistema ofrece la opción de regresar al menú principal o finalizar el programa.

```
void listaVehiculos() {
    if (total == 0) {
        printf("\nNo hay vehiculos registrados.\n");
        return; // Sale de la función si la lista está vacía
    }

    printf("\nVehiculos Registrados \n");
    for (int i = 0; i < total; i++) { //Recorre todos los vehiculos registrados para procesarlos uno a uno.
        printf("%d.-||", i + 1);
        printf("Fecha: %s ||", lista[i].fechaHora);
        printf("Placa: %s ||", lista[i].placa);
        printf("Año: %d ||", lista[i].anio);
        printf("Tipo: %s ||", lista[i].tipo);
        printf("Avaluo: $%.2f\n", lista[i].avaluo);
        printf("-----\n");
    }
}
```

Figura 6: Función que muestra la lista general de los vehículos.

5.2.6 Salida

Este módulo permite al usuario cerrar la sesión actual y regresar al menú de inicio (login).

5.3 Validaciones implementadas

Durante el desarrollo del sistema, se implementaron múltiples validaciones para prevenir errores y garantizar una correcta ejecución de las funcionalidades.

Validaciones por módulo

1. Login y Registro de usuario

- Verificación de longitud máxima de usuario (**25 caracteres**) y contraseña (**15 caracteres**).
- Rechazo de credenciales si no coinciden con las almacenadas.
- Manejo de errores al ingresar datos incorrectos.

2. Menú principal

- Validación de que la opción ingresada sea un número válido dentro del rango de opciones.
- Rechazo de caracteres no numéricos o fuera de rango.
- Mensajes de error claros ante una entrada inválida.

3. Registro de vehículo

- **Cédula:** validación de que tenga exactamente 10 dígitos numéricos.

- **Año de fabricación:** solo se aceptan valores entre **1990 y 2025**.
- **Placa:** debe seguir el formato **AAA-1234** (tres letras, guion, cuatro números).
- **Avalúo:** solo se permiten números y hasta un máximo de **\$500,000**.
- **Tipo de vehículo:** selección válida entre las opciones predefinidas.
- **Revisiones técnicas:** se exige que haya **exactamente 3 revisiones**.
- **Multas:** ingreso válido de "sí" o "no".

4. Búsqueda de vehículo

- Validación del formato de placa ingresado.
- Mensaje de error si no se encuentra la placa registrada.
- Insensibilidad a mayúsculas/minúsculas en la búsqueda.

5. Lista general

- Verificación del archivo.
- Mensaje de error si el archivo se encuentra vacío.

6. Comprobantes

- Validación de datos antes de generar y guardar el archivo .txt.
- Verificación de escritura exitosa del comprobante.

5.3 Manejo de archivos.

El sistema utiliza archivos de texto para guardar y recuperar la información registrada por los usuarios.

Archivos utilizados

- **usuarios.txt:** almacena los usuarios y contraseñas registrados en el sistema.
- **vehiculos.txt:** contiene los datos de todos los vehículos registrados.
- **comprobante.txt:** se genera cuando se completa el registro de un vehículo y se calcula el valor de la matrícula. Contiene un comprobante imprimible. Se genera únicamente cuando el usuario lo desee.

```

// Guarda los vehiculos en un archivo
void guardarVehiculos() {
    FILE *archivo = fopen("vehiculos.txt", "w");
    if (archivo == NULL) {
        printf("Error al guardar en archivo.\n");
        return;
    }

    for (int i = 0; i < total; i++) {
        fprintf(archivo, "%s,%s,%s,%d,%d,%.2f,%.2f,%d,%s\n",
            lista[i].cedula,
            lista[i].placa,
            lista[i].tipo,
            lista[i].anio,
            lista[i].revisiones,
            lista[i].avaluo,
            lista[i].costo,
            lista[i].multas,
            lista[i].fechaHora);
    }

    fclose(archivo);
}

```

Figura 7: Función que guarda la información de los vehículos registrados en un archivo txt.

6. Resultados

El sistema integral de matriculación vehicular fue desarrollado y probado satisfactoriamente, cumpliendo con todos los requisitos funcionales y técnicos planteados al inicio del proyecto.

Durante la ejecución, el sistema despliega las opciones principales del menú, permitiendo al usuario acceder a cada módulo de manera fluida e intuitiva. Se logró:

- Registro correcto de usuarios y vehículos, con validaciones que garantizan la integridad de los datos.
- Cálculo automático del valor a pagar por la matrícula, considerando recargos, descuentos y multas.
- Generación de comprobantes visualizables en pantalla y exportables en archivos .txt.
- Búsqueda eficiente de vehículos mediante número de placa con formato validado.
- Visualización completa de los vehículos registrados en el sistema.
- Manejo seguro del acceso mediante el módulo de login y cierre de sesión.
- Almacenamiento persistente de la información usando archivos de texto, asegurando la disponibilidad de datos entre sesiones.

```
=== SISTEMA DE ACCESO ===  
1. Iniciar sesion  
2. Registrar nuevo usuario  
3. Salir  
Seleccione opcion :
```

Figura 8: Imagen del sistema que permite al usuario registrarse o iniciar sesio.

```
=== SISTEMA SIVEM ===  
1. Registrar vehiculo  
2. Buscar vehiculo  
3. Lista de vehiculos  
4. Salir  
Seleccione opcion:
```

Figura 9: Menu principal para el acceso a las funciones del sistema

```
Vehiculo registrado.  
===== COMPROBANTE DE MATRICULA =====  
Fecha de registro: 2025-07-20 16:15  
Placa: ABC-2024  
Tipo: Vehiculo particular  
Anio: 2024  
Avaluo: $200000.00  
Valor a pagar: $3018.00  
=====
```

Figura 10: Información mostrada al finalizar el registro del vehículo.

```

--- Buscar Vehiculo por Placa ---
Ingresa la placa (formato ABC-1234): ABC-2024

Vehiculo encontrado:
===== COMPROBANTE DE MATRICULA =====
Fecha de registro: 2025-07-20 16:15
Placa: ABC-2024
Tipo: Vehiculo particular
Año: 2024
Avaluo: $200000.00
Valor a pagar: $3018.00
=====
Desea generar el comprobante en archivo? (s/n):

```

Figura 11: Información mostrada al realizar la búsqueda del vehículo.

```

Vehiculos Registrados
1.-||Fecha: 2025-07-19 20:58 ||Placa: PDX-7845 ||Año: 2024 ||Tipo: Transporte publico-comercial ||Avaluo: $254564.00
-----
2.-||Fecha: 2025-07-19 21:03 ||Placa: PDX-1527 ||Año: 2024 ||Tipo: Transporte publico-comercial ||Avaluo: $200000.00
-----
3.-||Fecha: 2025-07-19 21:13 ||Placa: AAA-1111 ||Año: 2000 ||Tipo: Vehiculo particular ||Avaluo: $163.20
-----
4.-||Fecha: 2025-07-20 16:15 ||Placa: ABC-2024 ||Año: 2024 ||Tipo: Vehiculo particular ||Avaluo: $200000.00
-----
Desea volver al menu? (S o N):

```

Figura 12: Lista general de los vehículos registrados.

```

----- COMPROBANTE DE MATRICULA -----
Fecha de registro: 2025-07-20 16:15
Placa: ABC-2024
Tipo: Vehiculo particular
Año: 2024
Avaluo: $200000.00
Valor a pagar: $3018.00
-----

```

Figura 13: Comprobante generado en archivo txt.

7. Conclusiones

El sistema SIVEM desarrollado cumple satisfactoriamente con todas las funcionalidades establecidas en los requisitos del proyecto, logrando simular el proceso integral de matriculación vehicular de manera eficiente y segura.

A través de este proyecto, se aplicaron correctamente los conocimientos adquiridos durante el semestre, tales como estructuras de control, manejo de archivos, validaciones de datos, funciones, y principios de buen código.

Además, el sistema demostró ser robusto frente a entradas inválidas y errores comunes, gracias a las validaciones implementadas, lo que garantiza una experiencia de usuario confiable.

En resumen, este proyecto no solo valida la comprensión técnica adquirida, sino que también ofrece una solución práctica y funcional que puede servir de base para desarrollos futuros relacionados con la gestión vehicular.

8. Recomendaciones

- Profundizar en el uso de control de versiones con Git y GitHub, comprendiendo buenas prácticas, ya que esto mejora la colaboración y el control del desarrollo del código.
- Aplicar consistentemente los principios de Clean Code, como nombres significativos, funciones pequeñas y legibles, y separación de responsabilidades, para facilitar el mantenimiento y la comprensión del código fuente.
- Implementar validaciones exhaustivas para las entradas del usuario, considerando todos los posibles errores, formatos inválidos y casos límite, con el fin de aumentar la robustez y confiabilidad del sistema.
- Documentar mejor el código y el proceso de desarrollo, incluyendo pruebas realizadas, decisiones técnicas y posibles mejoras para versiones futuras.

9. Glosario

Término	Definición
Avalúo	Valor económico estimado del vehículo, generalmente determinado por la autoridad competente.
Placa	Código alfanumérico único asignado a un vehículo para su identificación legal.
Revisión técnica vehicular	Inspección obligatoria para verificar que el vehículo cumple con las condiciones mecánicas y ambientales requeridas por ley.
Tipo de vehículo	Clasificación del vehículo según su uso: particular, motocicleta o transporte público/comercial.
Multa	Sanción económica impuesta por una infracción de tránsito. Debe pagarse antes de matricular el vehículo.

Número de cédula	Documento de identidad del propietario del vehículo, requerido para registro.
Matrícula	Documento legal que acredita que el vehículo está registrado y habilitado para circular.
Comprobante	Documento que muestra el detalle del trámite realizado, como el valor de la matrícula, información del vehículo y confirmación del registro.
Login / Inicio de sesión	Proceso mediante el cual el usuario accede al sistema usando su nombre de usuario y contraseña.
Registro de usuario	Proceso de creación de una cuenta nueva en el sistema mediante credenciales únicas.