**Gestión de Camiones y Cargas**

**1. Página de Portada**

**Título**:  
**Sistema de Gestión de Camiones y Cargas**

**Autores**:  
juan sebastian cabrera y Johan David guerra tique  
universidad Antonio nariño

**Fecha**:  
22/11/24

**2. Página de Título**

**Título Completo**:  
**Manual del Usuario para el Sistema de Gestión de Camiones y Cargas**

**Versión del Software**:  
Versión 5

**Autores**:  
juan sebastian cabrera y Johan David guerra tique  
**Institución**:  
universidad Antonio nariño

**3. Formato F277 - Dirección Nacional de Derechos de Autor**

(Se debe adjuntar el formato F277-Dirección Nacional de Derechos de Autor.pdf completado con los datos del software.)

* **Título del software**: Gestión de Camiones y Cargas
* **Versión del software**: 5
* **Resumen funcional del software**: Este sistema permite registrar, consultar y gestionar camiones con diferentes capacidades de carga y consumo de gasolina además de tener almacenada la información actual del camion según su placa.
* **Autores del software**: juan sebastian cabrera y Johan David guerra tique

**4. Prefacio**

**4.1. Documentos relacionados**

Este manual complementa el código fuente del software. Consulte el archivo README.md incluido en el repositorio para instrucciones básicas.

**4.2. Cómo navegar por esta guía**

INDICE

1. Introducción al sistema y sus funciones principales
2. Requisitos previos para el uso.
3. Instrucciones de instalación y configuración.
4. Guía para el uso de funciones principales.
5. Documentación técnica (diagramas y diseño del software).
6. Preguntas frecuentes (FAQ).

**5. Introducción**

**5.1. Descripción del sistema y su finalidad**

El Sistema de Gestión de Camiones y Cargas permite a los usuarios:

* Registrar camiones con matrícula, capacidad de carga, consumo de gasolina y carga actual.
* Consultar información de camiones por matrícula.
* Modificar la carga de camiones ya sea para disminuir o aumentar la carga.
* Identificar camiones óptimos según su consumo y capacidad dependiendo la necesidad de la gestión de la empresa y/o entidad.
* Verificar disponibilidad de camiones para transportar cargas .

**5.2. Novedades de la última versión**

**Bitácora de Cambios:**

* **Versión 1**: no retorna el menú para

continuar con los siguientes puntos . en las validaciones

.Capacidad de carga (kg), Consumo de gasolina

(galones/km),Carga actual (kg) que tenga una validación de

registrar solo números positivos, y en la matricula una validación

de placa 3 alfabetos y tres números que me deje ingresar.

**Versión 2**: se le agrego el uso de estructuras

**Versión 3**: se le agrego un validador para asegurarse que la carga ingresada no supere la capacidad del camion

**Versión 4**: se arreglo el problema del which para que el menú no se repita indefinidamente

**Versión 5**: se opservaron los siguientes errores No utlizar funsiones "nullptr". Falta hacer una validación de matricula de placa y que si ingreso un numero igual que retorne y diga "matricula repetida".(No deja ejecutar el código) Cuando declaras Camion\* mejorCamion directamente en el cuerpo de case 5, esta variable entra en un ámbito compartido con todo el switch, lo cual causa problemas cuando el flujo salta a otro case (como case 0 o default) los cuales fueron solucionados

* Observaciones: Se corrigieron problemas de validación de matrícula y se mejoró el manejo de cargas validando la capacidad de carga del camion y el problema del swhit mejoro con otro case para uno salirse a voluntad del el menú .

**6. Requisitos Previos**

**6.1. Conocimientos mínimos del usuario**

* Uso básico de sistemas operativos.
* Conocimiento básico de c++
* Usos de estructuras
* Manejo de la línea de comandos.
* Manejo de medios tecologicos y editores de programas (c++)

**6.2. Requisitos técnicos previos**

**a. Capacidades mínimas del equipo:**

* Procesador: Intel i3 o .
* RAM: 2 GB.
* Almacenamiento: 2 g de espacio libre.

**b. Software asociado necesario:**

* Compilador compatible con C++ online .
* Sistema operativo: Windows 11.

**c. Mecanismo de acceso:**  
El programa se ejecuta mediante un copilador y editor de códigos de c++ online accesible desde cualquier dispositivo (pc) con requerimientos funcionales básicos .

**7. Instalación y Configuración**

1. Descargue el código fuente desde el repositorio en github o copiar y pegar desde ahí a el editor de c++ de su preferencia .
2. Instale un compilador compatible o asegurarse de que el copilador online sea el correcto en el lenguaje de programación de c++ de preferencia la ultima versión posible que es las 23.
3. Compile el código

Copiar código

100% code

1. Ejecute el programa con:

Opción ejecutar

#include <iostream>

using namespace std;

struct Camion {

string matricula;

double capacidadCarga;

double consumoGasolina;

double cargaActual;

};

int main() {

Camion camiones[4];

int pmenu;

bool salir = false;

int camionesRegistrados = 0; // Llevar control de cuántos camiones se han registrado

while (!salir) { // Bucle para repetir el menú

cout << "\n=== MENU PRINCIPAL ===" << endl;

cout << "1. Registrar camiones" << endl;

cout << "2. Consultar datos de un camión por matrícula" << endl;

cout << "3. Cargar o descargar un camión" << endl;

cout << "5. Camión con menor consumo y capacidad suficiente para una carga dada" << endl;

cout << "6. Informar cuando no haya camión disponible para una carga dada" << endl;

cout << "0. Salir" << endl;

cout << "Seleccione una opción: ";

cin >> pmenu;

// Declaración de variables para uso compartido en los `case`

string matriculaBuscar;

double cargaNecesaria;

int mejorIndice = -1;

switch (pmenu) {

case 1: {

// Registro de camiones

if (camionesRegistrados >= 4) {

cout << "Ya se han registrado 4 camiones. No es posible agregar más." << endl;

break;

}

cout << "\nIngrese los datos del camión " << camionesRegistrados + 1 << ":" << endl;

// Validación de matrícula única

string nuevaMatricula;

bool matriculaRepetida;

do {

matriculaRepetida = false;

cout << "Matrícula: ";

cin >> nuevaMatricula;

for (int i = 0; i < camionesRegistrados; i++) {

if (camiones[i].matricula == nuevaMatricula) {

matriculaRepetida = true;

cout << "Error: Matrícula repetida. Intente con otra." << endl;

break;

}

}

} while (matriculaRepetida);

camiones[camionesRegistrados].matricula = nuevaMatricula;

// Validación de la capacidad de carga

do {

cout << "Capacidad de carga (kg): ";

cin >> camiones[camionesRegistrados].capacidadCarga;

if (camiones[camionesRegistrados].capacidadCarga <= 0) {

cout << "Error: La capacidad de carga debe ser un número positivo. Intente de nuevo." << endl;

}

} while (camiones[camionesRegistrados].capacidadCarga <= 0);

// Validación del consumo de gasolina

do {

cout << "Consumo de gasolina (galones/km): ";

cin >> camiones[camionesRegistrados].consumoGasolina;

if (camiones[camionesRegistrados].consumoGasolina <= 0) {

cout << "Error: El consumo de gasolina debe ser un número positivo. Intente de nuevo." << endl;

}

} while (camiones[camionesRegistrados].consumoGasolina <= 0);

// Validación de la carga actual

do {

cout << "Carga actual (kg): ";

cin >> camiones[camionesRegistrados].cargaActual;

if (camiones[camionesRegistrados].cargaActual < 0) {

cout << "Error: La carga actual no puede ser negativa. Intente de nuevo." << endl;

} else if (camiones[camionesRegistrados].cargaActual > camiones[camionesRegistrados].capacidadCarga) {

cout << "Error: La carga actual no puede exceder la capacidad máxima del camión. Intente de nuevo." << endl;

}

} while (camiones[camionesRegistrados].cargaActual < 0 || camiones[camionesRegistrados].cargaActual > camiones[camionesRegistrados].capacidadCarga);

cout << "Camión registrado exitosamente.\n" << endl;

camionesRegistrados++;

break;

}

case 2: {

// Consultar datos de un camión por matrícula

cout << "\nIngrese la matrícula del camión a buscar: ";

cin >> matriculaBuscar;

bool encontrado = false;

for (int i = 0; i < camionesRegistrados; i++) {

if (camiones[i].matricula == matriculaBuscar) {

cout << "\nDatos del camión con matrícula " << matriculaBuscar << ":" << endl;

cout << "Capacidad de carga: " << camiones[i].capacidadCarga << " kg" << endl;

cout << "Consumo de gasolina: " << camiones[i].consumoGasolina << " galones/km" << endl;

cout << "Carga actual: " << camiones[i].cargaActual << " kg" << endl;

encontrado = true;

break;

}

}

if (!encontrado) {

cout << "No se encontró ningún camión con la matrícula " << matriculaBuscar << "." << endl;

}

break;

}

case 5: {

// Camión con menor consumo y capacidad suficiente para una carga dada

if (camionesRegistrados == 0) {

cout << "No hay camiones registrados." << endl;

break;

}

cout << "\nIngrese la carga necesaria (kg): ";

cin >> cargaNecesaria;

if (cargaNecesaria <= 0) {

cout << "Error: La carga debe ser un número positivo." << endl;

break;

}

for (int i = 0; i < camionesRegistrados; i++) {

if (camiones[i].capacidadCarga - camiones[i].cargaActual >= cargaNecesaria) {

if (mejorIndice == -1 || camiones[i].consumoGasolina < camiones[mejorIndice].consumoGasolina) {

mejorIndice = i;

}

}

}

if (mejorIndice != -1) {

cout << "El camión con menor consumo adecuado para la carga de " << cargaNecesaria << " kg es:" << endl;

cout << "Matrícula: " << camiones[mejorIndice].matricula << endl;

cout << "Consumo: " << camiones[mejorIndice].consumoGasolina << " galones/km" << endl;

} else {

cout << "No hay camiones con capacidad suficiente para la carga de " << cargaNecesaria << " kg." << endl;

}

break;

}

case 6: {

// Informar cuando no haya camión disponible para una carga dada

if (camionesRegistrados == 0) {

cout << "No hay camiones registrados." << endl;

break;

}

cout << "\nIngrese la carga necesaria (kg): ";

cin >> cargaNecesaria;

if (cargaNecesaria <= 0) {

cout << "Error: La carga debe ser un número positivo." << endl;

break;

}

bool hayCamionDisponible = false;

for (int i = 0; i < camionesRegistrados; i++) {

if (camiones[i].capacidadCarga - camiones[i].cargaActual >= cargaNecesaria) {

hayCamionDisponible = true;

break;

}

}

if (!hayCamionDisponible) {

cout << "No hay camiones disponibles para transportar una carga de " << cargaNecesaria << " kg." << endl;

} else {

cout << "Sí hay camiones disponibles para esta carga." << endl;

}

break;

}

case 0:

// Salir del programa

salir = true;

cout << "Saliendo del programa..." << endl;

break;

default:

cout << "Opción no válida. Intente de nuevo." << endl;

break;

}

}

return 0;

}

**8. Guía de Uso del Sistema**

**8.1. Funciones principales**

**1. Registrar camiones**  
Permite ingresar los datos de matrícula, capacidad de carga, consumo y carga actual.

**2. Consultar camión por matrícula**  
Muestra los detalles de un camión registrado.

**3. Cargar/Descargar camión**  
Permite modificar la carga actual de un camión según sea necesario.

**5. Buscar camión óptimo**  
Identifica el camión con menor consumo que pueda transportar una carga específica.

**6. Verificar disponibilidad**  
Informa si no hay camiones disponibles para transportar una carga dada.

**9. Preguntas Frecuentes (FAQ)**

**P: ¿Qué pasa si ingreso una matrícula repetida?**  
R: El sistema notificará que la matrícula ya existe y pedirá ingresar una nueva.

**P: ¿Qué sucede si intento exceder la capacidad de carga de un camión?**  
R: Se mostrará un mensaje de error indicando que la carga excede el límite permitido y pide otra carga hasta que la que se ingrese sea una permitida.

**P: ¿Es posible registrar más de 4 camiones?**  
R: No, solo admite hasta 4 camiones simultáneamente.

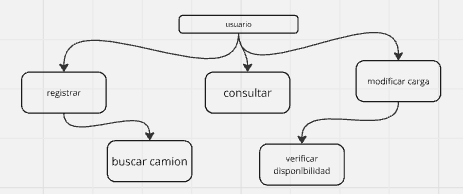
**10. Documentación Técnica**

**10.1. Diagrama Entidad-Relación**

|  |
| --- |
| camion |
| * Matricula * Capacidad de carga * Consumo del camion * Carga actual |

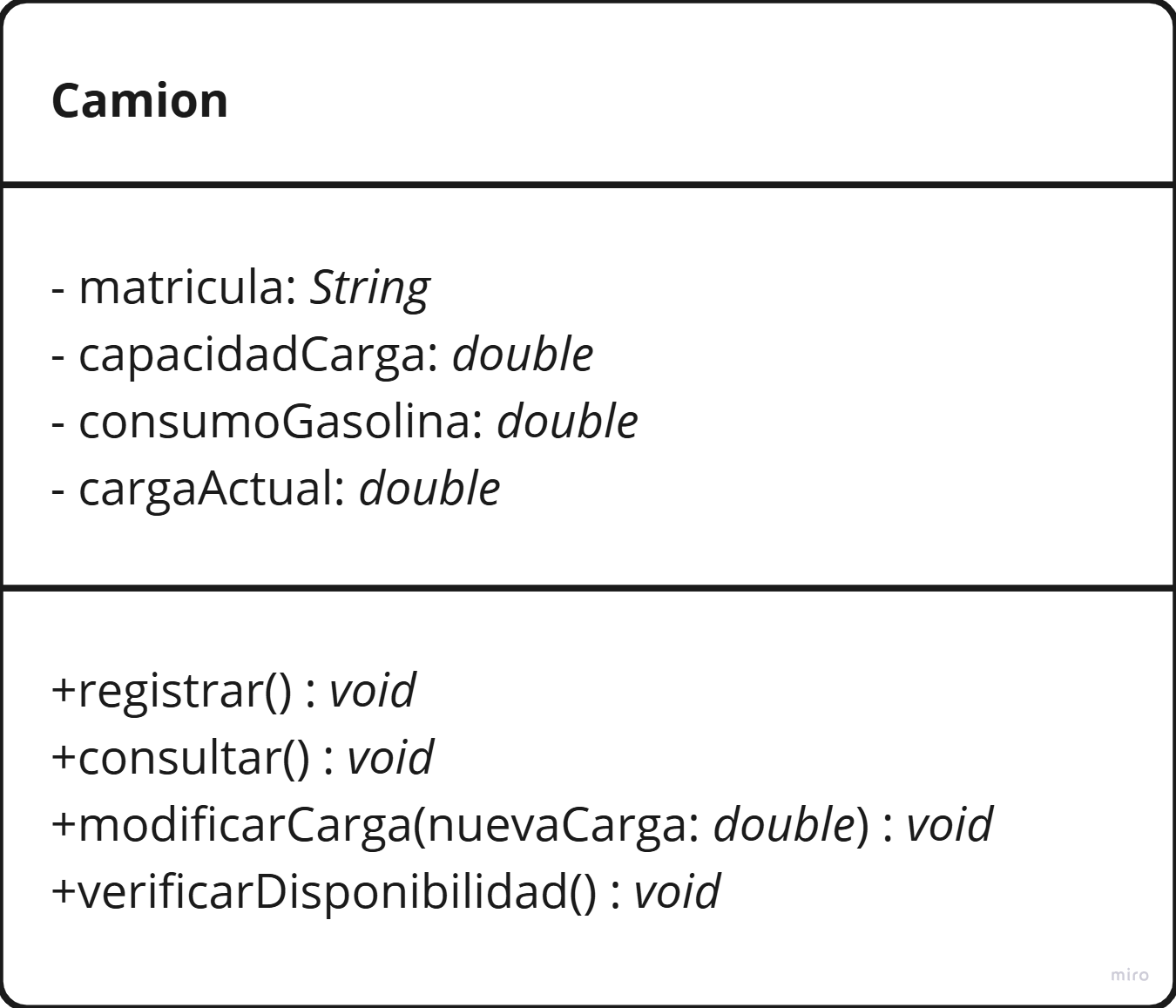
**10.2. Diagrama de Casos de Uso**

Incluye casos como registro, consulta, modificación y verificación de disponibilidad.



**10.3. Diagrama de Clases**

* Clase Camion con atributos: matricula, capacidadCarga, consumoGasolina, cargaActual.



**11. Datos de Contacto**

**Autor principal:** juan sebastian cabrera y Johan David guerra tique  
**Correo electrónico:** [jcabrera38@uan.edu.co](mailto:jcabrera38@uan.edu.co) [jguerra69@gmail.com](mailto:jguerra69@gmail.com)   
**Teléfono:3203247349**