

## **Parqueadero - Documentación**

### **Descripción General**

El sistema de gestión de parqueadero implementado en C++ está diseñado para manejar eficientemente el control de vehículos que ingresan, estacionan y salen de un parqueadero con capacidad de 12 espacios. La estructura utiliza una lista doblemente enlazada circular para gestionar los espacios, garantizando flexibilidad y eficiencia en las operaciones. La persistencia de datos se realiza mediante archivos TXT para asegurar que el estado del parqueadero se mantenga entre ejecuciones del programa.

### **1. Funcionalidades**

#### **1.1. Estacionar Auto**

- Se estaciona si:
  - o El auto esté permitido (placa registrada en el sistema). Si no, se rechaza el ingreso.
  - o El espacio seleccionado esté libre. Si está ocupado, se solicita otro espacio, el auto solo puede estacionarse en un espacio a la vez
- Si no hay espacios disponibles, muestra un mensaje de que el parqueadero está lleno.
- Agrega los datos del auto\_permitido al espacio correspondiente y registra su fecha y hora de ingreso
- Actualiza el estado del parqueadero en la lista doblemente enlazada y en el archivo parqueadero.txt

#### **1.2. Retirar Auto**

- Libera un espacio ocupado según la placa:
  - o Marca el espacio como libre en la lista doblemente enlazada.
  - o Elimina los datos del auto\_permitido del espacio correspondiente, y registra la fecha y hora del retiro.
  - o Actualiza los archivos parqueadero.txt y espacio\_parqueadero.txt.

#### **1.3. Mostrar Estado del Parqueadero**

- Visualiza el estado del parqueadero en formato gráfico con colores:
  - o Rojo (\033[31m): Espacios ocupados.
  - o Color por defecto (\033[0m): Espacios libres.
- La visualización se actualiza cada vez que se estaciona o retira un vehículo.

#### **1.4. Registrar Autos Permitidos**

- Actualiza el archivo autos\_permitidos.txt

#### **1.5. Eliminar Autos Permitidos**

- Elimina según la placa un auto registrado del sistema.
- Actualiza el archivo autos\_permitidos.txt.

#### **1.6. Historial de Estacionamiento**

- Registra todas las entradas y salidas en historial\_estacionamiento.txt, incluyendo:  
Según la placa del auto los datos de auto\_permitido se registra el o los id del Espacio ocupado con su fecha y hora de ingreso, y fecha y hora de salida.

### **2. Componentes del Sistema**

#### **2.1 Menú**

- 1, Mostrar estado del parqueadero
2. Estacionar
3. Retirar auto
4. Registrar auto permitido
5. Eliminar auto Permitido
6. Mostrar lista de autos permitidos
7. Mostrar Historial de almacenamiento
8. Salir

- La navegación es mediante teclas de flecha para subir y bajar entre opciones

#### **2.2. Parqueadero.h y Parqueadero.cpp**

Clase principal que gestiona la lógica del parqueadero.

Atributos:

- Lista doblemente enlazada circular que representa los espacios.
- Lista de autos permitidos.

Métodos clave:

1. estacionarAuto():

o Verifica las condiciones de estacionamiento (auto permitido, espacio libre).

o Actualiza el nodo correspondiente y los archivos relacionados.

2. retirarAuto():

o Libera el nodo correspondiente y actualiza los archivos.

3. mostrarEstado():

o Visualiza el estado del parqueadero con colores y formato gráfico.

4. guardarDatos() y cargarDatos():

o Persistencia de datos en los archivos parqueadero.txt y espacio\_parqueadero.txt.

### **2.3. EspacioParqueadero.h y EspacioParqueadero.cpp**

Clase que representa un espacio de estacionamiento.

Estructura:

```
struct Nodo {  
  
    string id; // Identificador (01-12)  
  
    bool ocupado; // Estado del espacio  
  
    Nodo* siguiente;  
  
    Nodo* anterior;  
  
};
```

Métodos:

- ocuparEspacio(): Asigna un auto y un propietario al espacio, y registra fecha y hora de entrada
- liberarEspacio(): Limpia el espacio y lo marca como libre, registra la fecha y hora de salida
- obtenerEstado(): Devuelve el estado del espacio como texto.

### **2.4. AutoPermitido.h y AutoPermitido.cpp**

Clase para manejar datos del auto.

Atributos:

- Placa, marca, color, Nombre, cédula, correo

Estos son los datos del auto: placa, marca, color

Y estos los datos de su propietario: nombre, cédula

Métodos:

- Métodos para obtener y establecer datos.

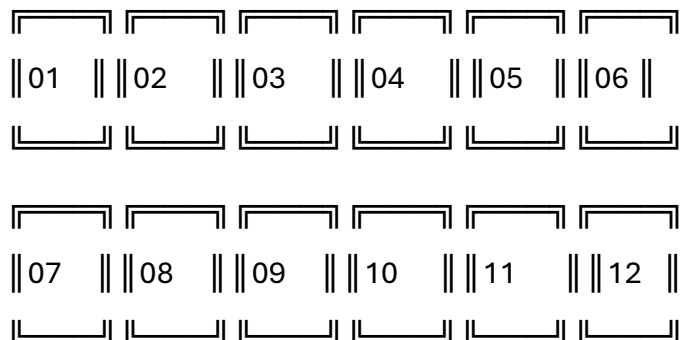
### 3. Persistencia de Datos

Archivos:

1. autos\_permitidos.txt: Lista de autos permitidos con datos del auto y su propietario.
2. parqueadero.txt: Estado general de los espacios. Según cada id-identificador, hay boolean para ver el estado ocupado del espacio
3. espacio\_parqueadero.txt: Datos detallados de cada espacio ocupado, en el que se muestra el id-identificador del espacio y los datos del auto\_permitido que ocupado el lugar, así como la fecha y hora de estacionamiento de ingreso y salida, se actualiza cuando un nuevo auto se estaciona, en caso de estar vacío
4. historial\_estacionamiento.txt: Registro de entradas y salidas de un id-identificador de estacionamiento según los datos del auto\_permitido, los datos del auto\_permitido son únicos y se mantienen, mientras que los datos de cuando se estaciona el auto son acumulativos.

### 4. Estructura Visual del Parqueadero

El parqueadero se muestra así:



- Rojo (\033[31m): Espacios ocupados.
- Color por defecto (\033[0m): Espacios libres.

Se actualiza dinámicamente tras cada operación.