

Universidad Rafael Landívar.  
Facultad de Ingeniería.  
Ingeniería en Informática y Sistemas.  
Laboratorio de Pensamiento Computacional,  
Sección 04  
Docente: Ing. Andrés Sebastián Gálvez.

**PROYECTO 02 (PARTE A)**  
**“Sistema de gestión de estacionamiento”**

Estudiante: Lopez Higueros, Diego Alessandro  
Carné: 1187525

Guatemala, 5 de mayo de 2025

## **Análisis y Algoritmo**

### **Acciones del Programa:**

1. Ingresar un vehículo manualmente.
2. Ingresar un lote de vehículos.
3. Encontrar un vehículo.
4. Retirar un vehículo.
5. Salir del programa.

### **Datos de Entrada:**

1. Marca del vehículo. Tipo de dato – Cadena de Texto.  
Ejemplo: “Honda”, “Mazda”, “Toyota”, etc.
2. Color del vehículo. Tipo de dato – Cadena de Texto.  
Ejemplo: “Rojo”, “Azul”, “Blanco”.
3. Placa del vehículo de nomenclatura guatemalteca. Tipo de dato – Cadena de Texto.  
Ejemplo: “ABC123”, “730KHL”, etc.
4. Tipo de vehículo. Tipo de dato – Cadena de Texto.  
Valores posibles: “Moto”, “Sedan”, “SUV”.
5. Hora de entrada. Tipo de dato – Numero Entero.  
Valores posibles: Un rango dentro de 6 AM y 20 PM.

## **Variables:**

1. marcaVehiculo - String: almacenará la marca del vehículo.
2. colorVehiculo - String: almacenará el color del vehículo.
3. placaVehiculo – String: almacenará la placa del vehículo.
4. tipoVehiculo – String: almacenará el tipo de vehículo.
5. horaEntrada – Int: almacenará la hora de entrada del vehículo.
6. matrizVehiculos – Matriz: matriz bidimensional que representará el estado del parqueo, es decir, los espacios disponibles.
7. parqueoDisponible – Bool: identificará si hay un espacio disponible para el tipo de vehículo.
8. montoAPagar – Float: almacenará el monto que deberá pagar el cliente dependiendo del tiempo de estadía, utilizaremos float y no double por ahorro de memoria.
9. tipoDePago – String: identificará el método a pagar ingresado por el cliente.
10. dineroIngresado – Float: almacenará el dinero ingresado por el cliente que está pagando.
11. vuelto – Float: almacenará el monto que se devuelve al cliente.
12. ctdEstacionamientosPiso – Int: almacenará la cantidad de estacionamientos disponibles por piso.
13. ctdEstacionamientosAlPublico – Int: almacenará la cantidad de estacionamientos abiertos al público.
14. ctdEstacionamientosMoto – Int: almacenará la cantidad de estacionamientos para motocicletas.
15. ctdEstacionamientosSUV – Int: almacenará la cantidad de estacionamientos para SUV.
16. o – Int: variable de opción para switch.
17. espacio – String: variable para cambiar los puestos de la matriz para que sea entendible el mapa de estacionamientos.

## Condiciones y Cálculos:

### 1. Restricciones:

- La hora de entrada debe de ser un valor dentro de 6 a 20.
- La placa debe de ser una cadena de 6 caracteres máximo (los primeros tres caracteres deben de ser tres letras mayúsculas seguidas de tres números del 0 al 9).
- El código del estacionamiento debe ser válido según el tipo de vehículo.

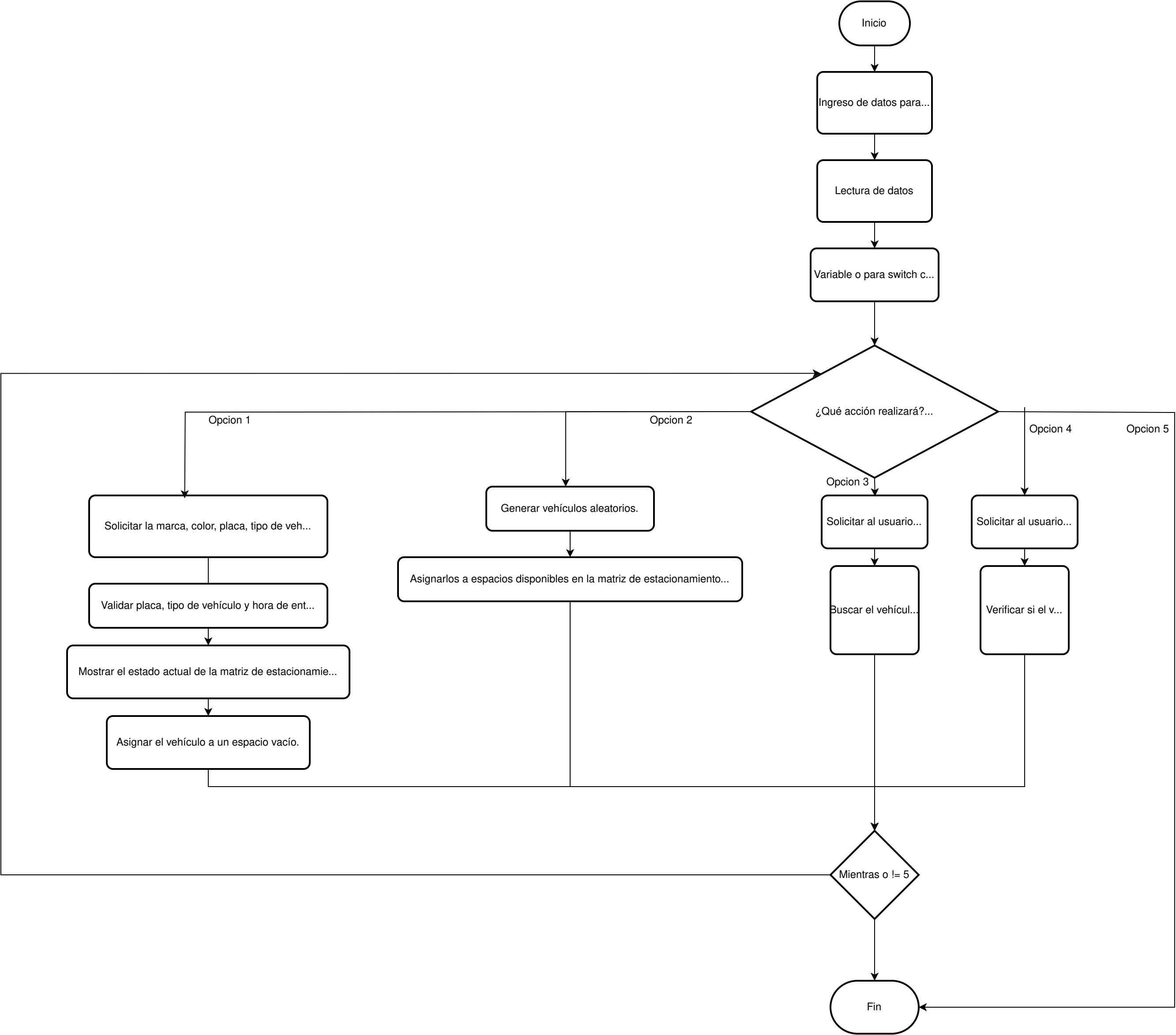
### 2. Cálculos:

- Monto total para pagar:

Se debe calcular según el tiempo de estancia, y se aplican tarifas definidas:

- **0–1 hora:** Costo de Q0 (cortesía).
- **2–4 horas:** Costo de Q15.
- **5–7 horas:** Costo de Q45.
- **8–12 horas:** Costo de Q60.
- **12–24 horas:** Costo de Q150.
- Vuelto:  $\text{dineroIngresado} - \text{montoAPagar}$
- 24 – Hora de entrada.

### 3. Diagramas de flujo:



Pago
float montoAPagar string tipoDePago float dineroIngresado float vuelto
funcion CalcularVuelto() funcion RealizarPago()

Parqueo
string[,] matrizVehiculos
procedimiento InicializrMatriz () procedimiento BuscarEspacio() procedimiento MostrarEstado() procedimiento AsignarEspacio()

Vehiculo
string marcaVehiclo string colorVehiculo string placaVehiculo string tipoVehiculo int horaEntrada = 0
procedimiento RegistrarVehiculo() procedimiento HoraDeEntradaValida() procedimiento PlacasValidas() procedimiento TipoVehiculoValido()

Estacionamiento
string codigoEstacionamiento bool estado = false string tipoEstacionamiento
procedimiento CambiarEstado()

