

Universidad Rafael Landívar.

Facultad De Ingeniería.

Licenciatura En Ingeniería En Informática Y Sistemas.

Laboratorio de Pensamiento Computacional, Sección 19.

Docente: Inga. Cindy García.

PROYECTO 02 (INCISO A)

“SISTEMA DE GESTIÓN DE ESTACIONAMIENTO”

Estudiante: Cucul Tut, Erik Carlos Omar

Carné: 1292625

7 de mayo de 2025

I. ACCIONES DEL PROGRAMA

El jugador debe elegir una de las siguientes acciones para intentar sobrevivir hasta llegar a su destino:

1. Ingresar un vehículo manualmente:

Se registra un vehículo con los datos obligatorios.

- **Tipo:** Moto, Carro estándar o SUV.
- **Placa:** 6 caracteres alfanuméricos en mayúsculas.
- **Marca:** Texto libre.
- **Color:** Texto libre.
- **Hora de entrada:** Entero entre 5 y 17.
 - **Asignación de espacio:**
 - Muestra matriz con:
 - Códigos para espacios libres del tipo requerido.
 - "X" para ocupados/tipos no compatibles
 - Valida que el código ingresado exista y esté libre.

2. Ingresar un lote de vehículos:

Genera 5 vehículos con datos aleatorios

- **Tipo:** Moto, Carro estándar o SUV.
- **Placa:** 6 caracteres alfanuméricos en mayúsculas.
- **Marca:** Honda, Nissan, Hyundai, Toyota y Kia.
- **Color:** Rojo, Azul, Negro, Gris y Blanco.
- **Hora de entrada:** Entero entre 5 y 17.
 - **Registro automático:**
 - Para cada vehículo:
 - Busca el primer espacio libre del tipo generado.
 - Si existe:
 - Asigna el espacio.
 - Registra placa y ubicación.
 - Si no: Omite el vehículo.
 - Muestra matriz con:
 - Códigos para espacios libres del tipo requerido.
 - "X" para ocupados/tipos no compatibles

3. Buscar un vehículo:

Busca por placa exacta.

- **Devuelve:**
 - Todos los datos del vehículo
 - Datos de la ubicación.
 - Hora de ingreso

4. Retirar un vehículo:

Requiere código de parqueo válido y ocupado

- **Calcula:**

- Tiempo de estadía "aleatorio 1 a [24 - horaEntrada]".
- **Monto** "precioPorHora \times tiempo":
 - Procesa pago – efectivo o tarjeta
 - Libera el espacio y actualiza matriz

5. Mostrar estado general del parqueo:

- **Proporciona:**

- Conteo numérico por tipo.
- Representación matricial con "X" para los espacios ocupados y códigos para los espacios libres.

6. Generar reporte de vehículos estacionados:

- Filtra vehículos por:

- **Tipo:** Moto, Carro estándar o SUV.
- Rango horario de ingreso.
 - **Lista:**
 - Placas, marcas, ubicaciones.
 - Horas de entradas

7. Fin del programa:

El programa termina cuando esta condición:

- Opción 7:
 - Finaliza la ejecución del sistema.

II. DATOS DE ENTRADA

1. Menú principal:

Cada iteración del sistema solicita.

- (1) Ingresar vehículo manual.
- (2) Ingresar lote de vehículos.
- (3) Buscar vehículo.
- (4) Retirar vehículo.
- (5) Mostrar estado del parqueo.
- (6) Generar reporte.
- (7) Salir.
- Tipo de datos: Número entero (Int / Integer).

2. Configuración inicial:

Si el sistema está iniciando, se le pedirá al usuario que ingrese.

- Cantidad de pisos del estacionamiento
- Cantidad de espacios por piso.
- Precio por hora de estacionamiento.
- Cantidad de espacios para motos.
- Cantidad de espacios para SUV's.
- Tipo de datos: Número entero (Int / Integer).

3. Ingreso manual de Vehículo:

Si el usuario elige registrar un vehículo, se le pedirá:

- Tipo de vehículo.
- Placa del vehículo.
- Marca del vehículo.
- Color del vehículo.
- Código de estacionamiento.
- Tipo de datos: Cadena de caracteres (String).
- Hora de entrada.
- Tipo de dato: Número entero (Int / Integer).

4. Retiro de Vehículo:

Si el usuario quiere retirar un vehículo, se le pedirá:

- Código de estacionamiento.
- Método de pago.
- Tipo de datos: Cadena de caracteres (String).
- Monto entregado.
- Tipo de dato: Número decimal (Float).

5. Para buscar un vehículo:

Si el usuario necesita encontrar un vehículo, se le pedirá:

- Placa del vehículo.
- Tipo de dato: Cadena de caracteres (String).

6. Para generar reporte:

Si el usuario solicita un reporte, se le pedirá:

- Criterio de filtrado.
- Tipo de dato: Número entero (Int / Integer).

III. VARIABLES

1. Configuración del sistema:

- **Cantidad de pisos:** Almacena el número total de pisos del estacionamiento.
 - Tipo de dato: Número entero (Int / Integer).
 - Valor inicial: Sin asignar.
- **Espacios por piso:** Guarda cuántos espacios de parqueo hay en cada piso.
 - Tipo de dato: Número entero (Int / Integer).
 - Valor inicial: Sin asignar.
- **Precio por hora:** Registra el costo por hora de estacionamiento.
 - Tipo de dato: Número decimal (Float).
 - Valor inicial: Sin asignar.
- **Espacios para motos:** Cantidad de lugares reservados para motos.
 - Tipo de dato: Número entero (Int / Integer).
 - Valor inicial: Sin asignar.
- **Espacios para SUV:** Cantidad de lugares para vehículos SUV.
 - Tipo de dato: Número entero (Int / Integer).
 - Valor inicial: Sin asignar.

2. Gestión de Vehículos:

- **Lista de vehículos:** Registro de todos los vehículos estacionados.
 - Tipo de dato: List<Vehículo> - Lista de objetos.
 - Valor inicial: Vacía.
- **Vehículo actual:** Almacena temporalmente el vehículo en proceso.
 - Tipo de dato: Vehículo – Objeto.
 - Valor inicial: Sin asignar.

3. Temporales de Operación:

- **Código de espacio:** Identificador de ubicación.
 - Tipo de dato: Cadena de caracteres (String).
 - Valor inicial: Vacío.
- **Monto calculado:** Total a pagar al retirar vehículo.
 - Tipo de dato: Número decimal (Float).
 - Valor inicial: 0.00.
- **Tiempo de estadía:** Duración en horas.
 - Tipo de dato: Número entero (Int / Integer).
 - Valor inicial: Sin calcular.

4. Control del sistema:

- **Opción del menú:** Acción seleccionada por usuario.
 - Tipo de dato: Número entero (Int / Integer).
 - Valor inicial: 0.
- **Generador aleatorio:** Para datos/tiempos aleatorios.
 - Tipo de dato: Random.
 - Valor inicial: Nueva instancia.

5. Estructuras de Datos:

- **Vehículo:** Almacena toda la información de un vehículo estacionado.
 - Tipo de dato: Clase personalizada (Vehículo).
 - Contiene:
 - Tipo (string).
 - Placa (string).
 - Marca (string).
 - Color (string).
 - HoraEntrada (int).
 - CodigoParqueo (string).
- **Estacionamiento:** Representa cada espacio físico del parqueo.
 - Tipo de dato: Clase personalizada (Estacionamiento).
 - Contiene:
 - Codigo (string).
 - Tipo (string).
 - Disponible (bool).

6. Estados del sistema:

- **Placa Válida:** Verificación de formato.
 - Tipo de dato: bool.
 - Valor inicial: false.
- **Espacio libre:** Confirmación de disponibilidad.
 - Tipo de dato: bool.
 - Valor inicial: false.

7. Variables por evento o funcionalidad:

- **Color:** Color del vehículo registrado.
 - Tipo de dato: String.
 - Valor inicial: Sin asignar.
- **CodigoParqueo:** Código del espacio asignado al vehículo.
 - Tipo de dato: String
 - Valor inicial: Sin asignar.
- **MarcasDisponibles:** Marcas posibles para generación aleatoria.
 - Tipo de dato: Lista de String
 - Contiene: ["Honda", "Nissan", "Hyundai", "Toyota", "Kia"]
- **ColoresDisponibles:** Colores posibles para vehículos generados.
 - Tipo de dato: Lista de String
 - Contiene: ["Rojo", "Azul", "Negro", "Gris", "Blanco"]
- **ListaLoteVehiculos:** Almacena temporalmente el lote generado.
 - Tipo de dato: Lista<Vehículo>
 - Contiene: Vacía.
- **ResultadoBusqueda:** Almacena el vehículo encontrado por búsqueda.
 - Tipo de dato: Vehículo
 - Valor inicial: Vacía.
- **FiltroTipoVehiculo:** Tipo de vehículo seleccionado para el reporte.
 - Tipo de dato: String
 - Valor inicial: Sin asignar.
- **RangoHoralInicio:** Hora inicial para el filtro del reporte.
 - Tipo de dato: Entero.
 - Valor inicial: 0.
- **RangoHoraFin:** Hora final para el filtro del reporte.
 - Tipo de dato: Entero
 - Valor inicial: 0

IV. CONDICIONES Y CALCULOS

1. Configuración inicial :

- **Pisos del estacionamiento:**
 - Fórmula: $\text{pisos} = \text{valor_ingresado}$.
- **Espacios por piso:**
 - Fórmula: $\text{espaciosPorPiso} = \text{valor_ingresado}$.
- **Precio por hora:**
 - Fórmula: $\text{precioPorHora} = \text{valor_ingresado}$.

2. Validaciones de entradas:

- **Placa de vehículo:**
 - Fórmula: $\text{longitud}(\text{placa}) == 6 \ \& \ \text{solo_letras_numeros}(\text{placa})$.
- **Hora de entrada:**
 - Fórmula: $\text{horaEntrada} \geq 5 \ \& \ \text{horaEntrada} \leq 17$.
- **Asignación de espacio:**
 - Fórmula: $\text{espacio.tipo} == \text{vehiculo.tipo} \ \& \ \text{espacio.disponible} == \text{true}$.

3. Cálculos principales:

- **Tiempo de estadía:**
 - Fórmula: $\text{tiempoEstadia} = \text{random.Next}(1, 25 - \text{horaEntrada})$.
- **Monto a pagar:**
 - Fórmula: $\text{montoTotal} = \text{tiempoEstadia} * \text{precioPorHora}$.
- **Cambio en efectivo:**
 - Fórmula: $\text{cambio} = \text{montoEntregado} - \text{montoTotal}$.
 - Desglose:
 - Billetes: Q100, Q50, Q20, Q10, Q5, Q1
 - Monedas: Q0.50, Q0.25, Q0.10, Q0.05, Q0.01
- **Validación clave:**
 - Placa:
 - Fórmula: $\text{longitud} = 6$ caracteres, solo letras/números.
 - Espacios:
 - Fórmula: $\text{disponible} = \text{true} \ \& \ \text{tipo coincide}$.
 - Pago:
 - Fórmula: $\text{montoEntregado} \geq \text{montoTotal}$.

4. Gestión de espacios:

- **Ocupación:**
 - Al ingresar:
 - Fórmula: espacio.disponible = false.
 - Al retirar:
 - Fórmula: espacio.disponible = true.
- **Verificación de capacidad:**
 - Fórmula: espacios_disponibles(tipo) > 0.

5. Reportes:

- **Filtrado por hora:**
 - Fórmula: horaEntrada >= hora_inicio & horaEntrada <= hora_fin.
- **Filtrado por tipo:**
 - Fórmula: vehículo.tipo == tipo_seleccionado.

6. Condiciones de terminación:

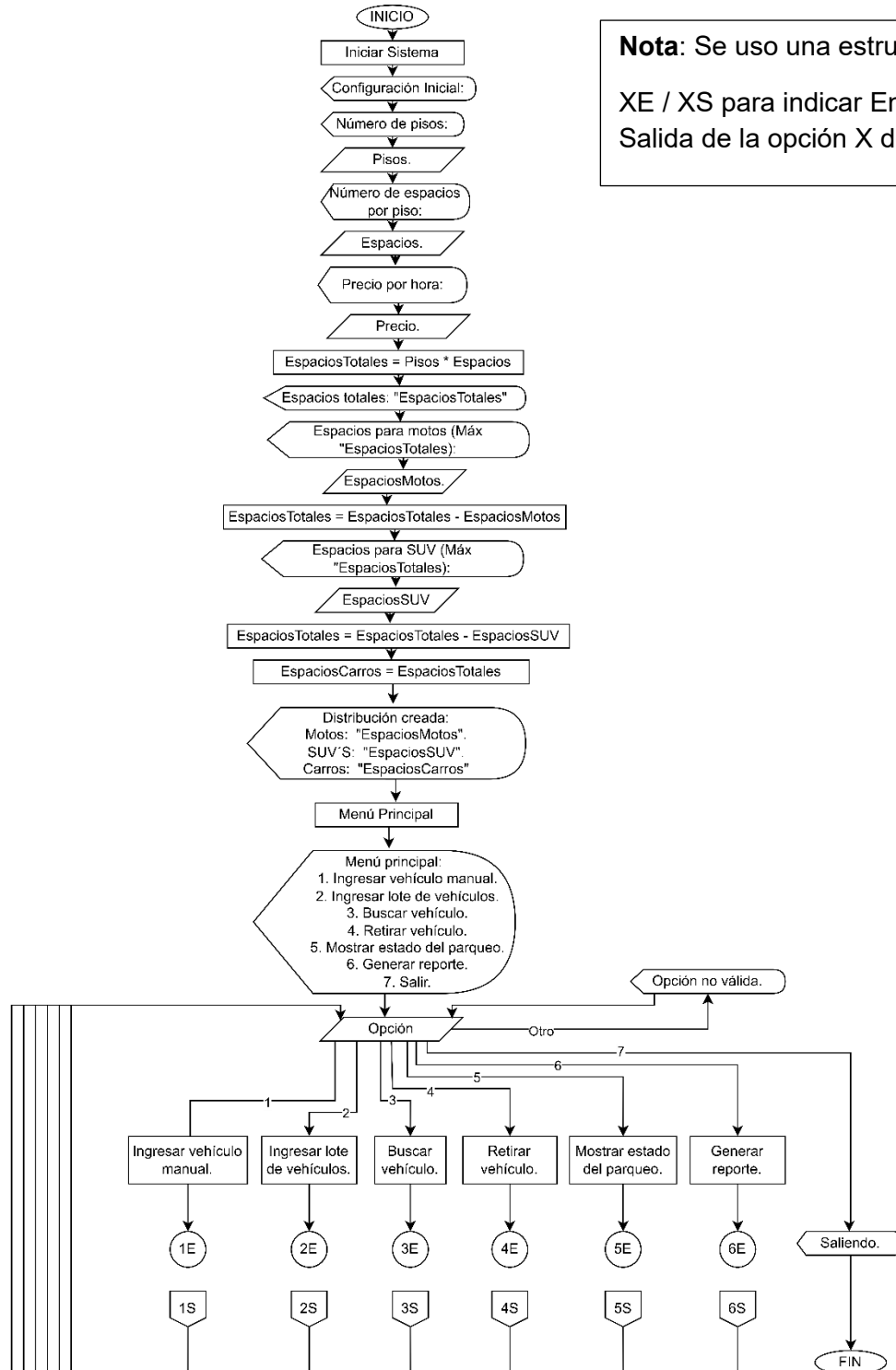
- **Salida del sistema:**
 - Fórmula: opcion_menu == 7.

V. ALGORITMO

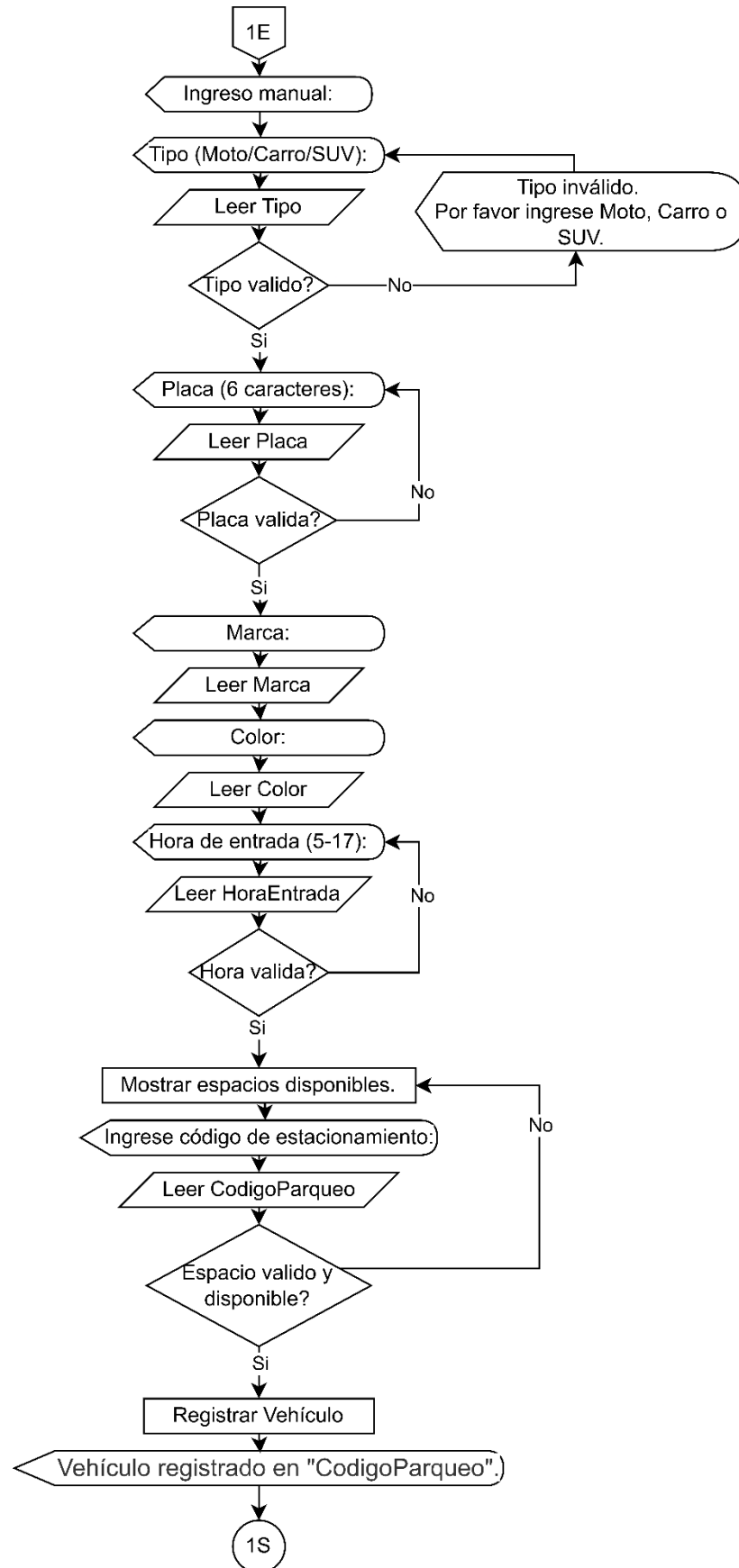
Enlace:

<https://drive.google.com/file/d/1r0xbmh4sArciPQDpSA3wgffvFHBjXoFa/view?usp=sharing>

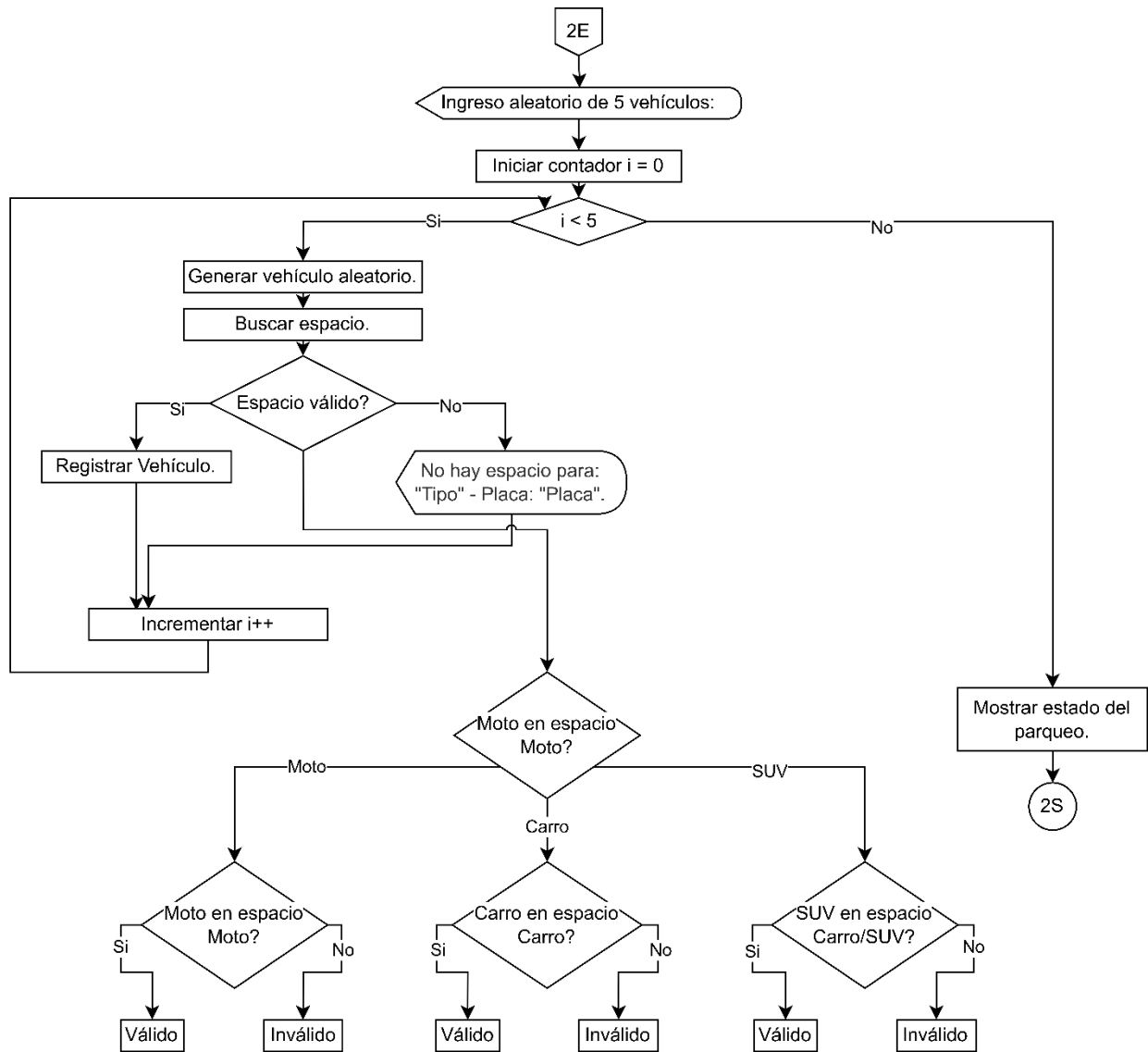
Diagrama de flujo, Parte A:



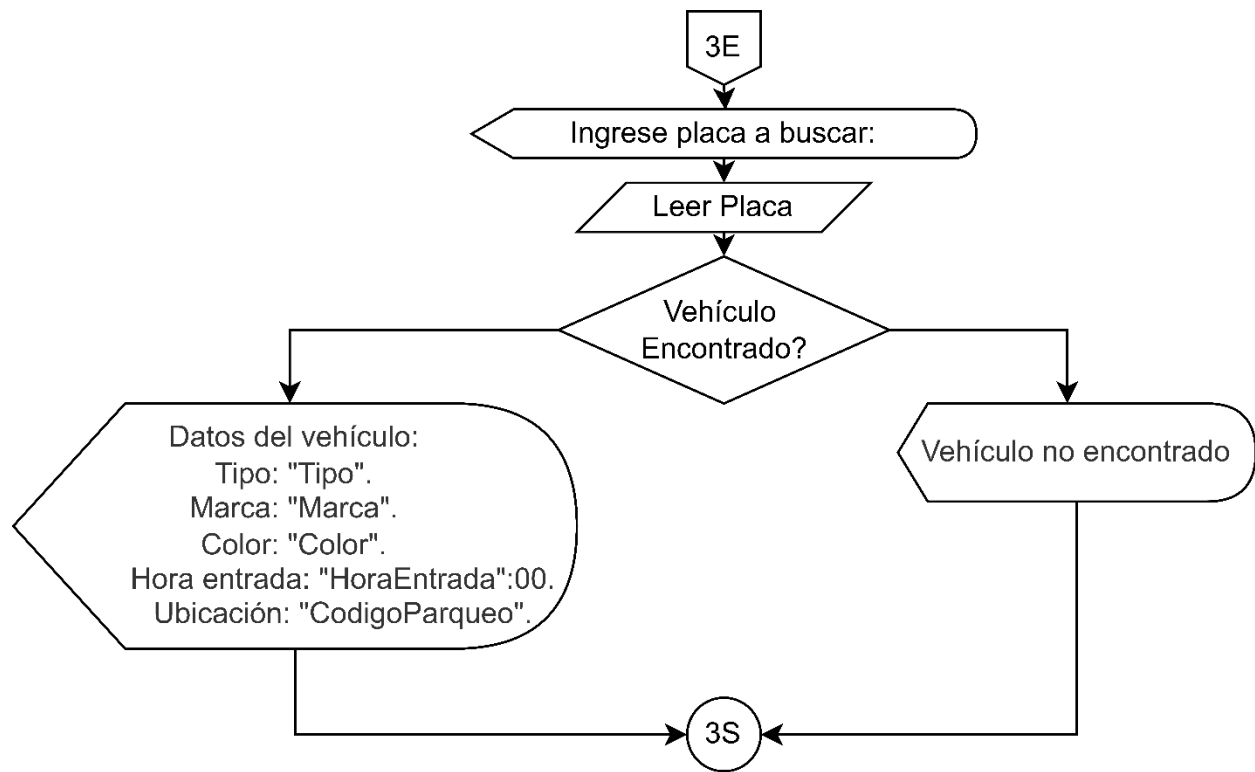
1:



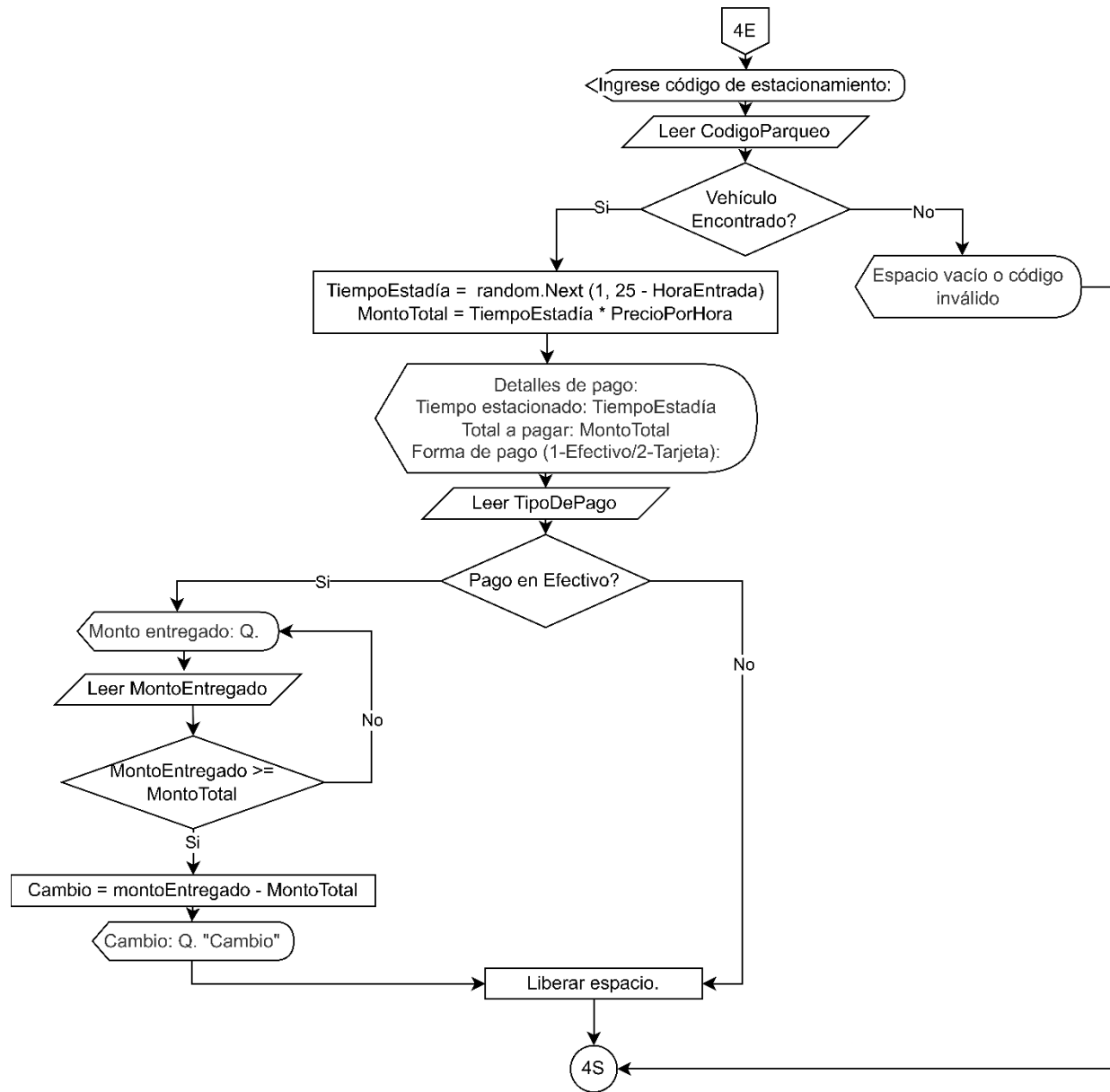
2:



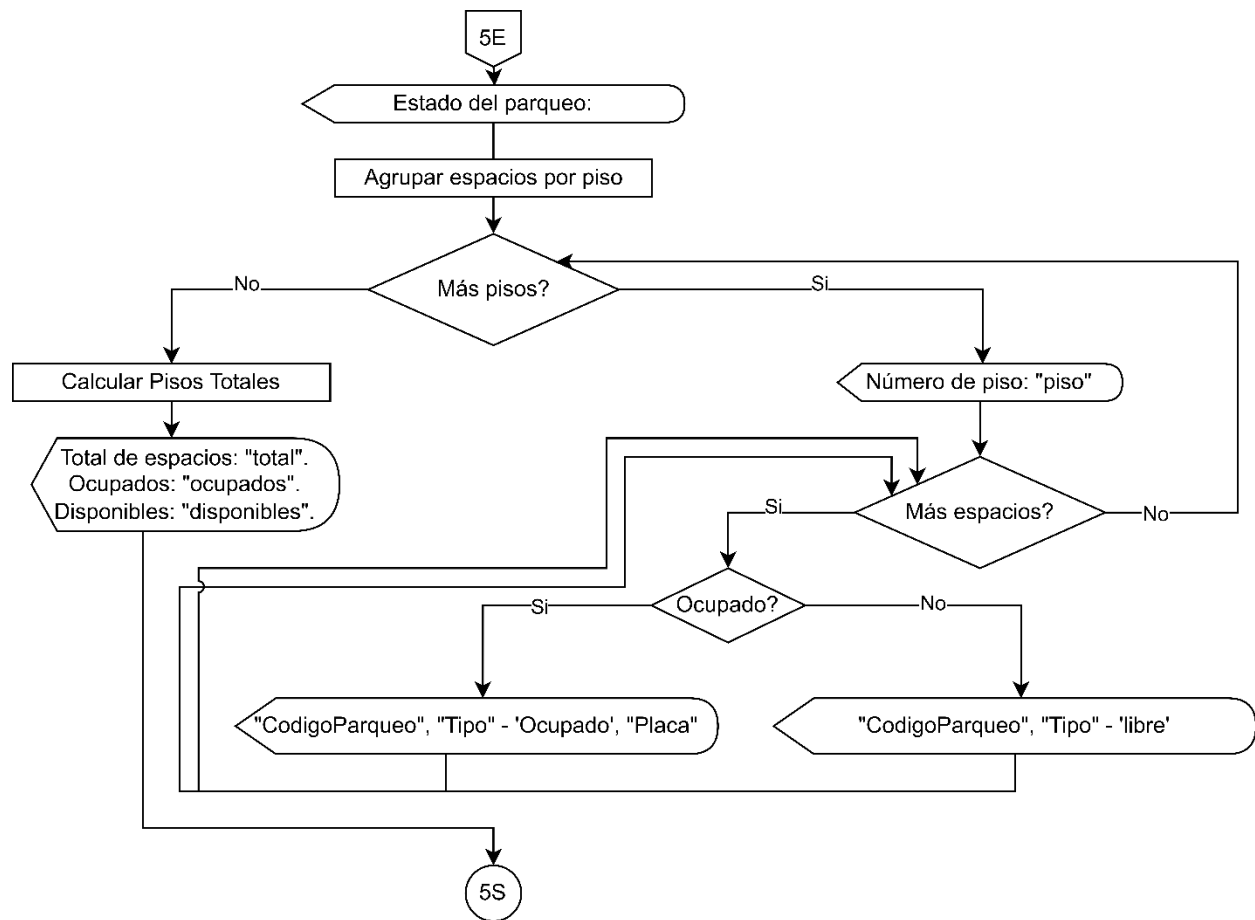
3:



4:



5:



6:

