

Nombre:	Registro de entrada de vehículos	código
Parámetros:	- placa(cadena): Placa del vehículo.	cpp\ncin >>vehiculo.placa;\n
	- horaEntrada(cadena): Hora de entrada (HH (Hora estándar del este/hora estándar del este).	cpp\ncin>>vehiculo.horaEntrada;\n
	- nombrePropietario(cadena): Nombre del propietario.	cpp\ncin >> propietario.nombre;\n
	- cedulaPropietario(cadena): Cédula del propietario (solo números).	cpp\ndo {\n cout << "Ingrese la cédula...";\n cin >> propietario.cedula;\n if (!validarCedula(proprietario.cedula)) {...}\n } while (!validarCedula(...));\n
	- correoPropietario(cadena): Correo del propietario.	cpp\ncin >> propietario.correo;\n
	- tipoVehiculo(cadena): Tipo de vehículo (carro/moto).	cpp\ncin >> vehiculo.tipoVehiculo;\n
	- colorVehiculo(cadena): Color del vehículo.	cpp\ncin >> vehiculo.color;\n
Retorno	- Número de puesto asignado (int).	cpp\nfor (int i = 0; i < TOTAL_PUESTOS; i++) {\n if (!parqueadero[i].ocupado) { parqueadero[i].numero = i + 1; }\n}
	- Mensaje: "Carro ingresado correctamente" o "No hay puestos disponibles"	cpp\ncout << "Vehículo con placa " << vehiculo.placa << " ha sido ingresado en el puesto " << parqueadero[i].numero << ".\n";\n
Descripción:	Esta función registra la entrada de un vehículo en el parqueadero. Se ingresarán los datos del vehículo y el propietario. Si la placa es válida y hay puestos disponibles, se asigna un puesto libre. Si no hay puestos, se informa al usuario.	cpp\nVehiculo vehiculo;\nProprietario propietario;\nbool puestoEncontrado = false;\nfor (int i = 0; i < TOTAL_PUESTOS; i++) { ... }\n

Nombre:	Dar salida a un carro del parqueadero	Código
Parámetros:	- <b>placa</b> (string): Número de la placa del vehículo. - <b>tarifa</b> (float): Tarifa por hora.	string placa, float tarifa
Retorno:	- <b>costo total</b> (float): Monto a pagar calculado en función del tiempo que el carro estuvo en el parqueadero. - <b>mensaje de salida</b> (string): Mensaje indicando que el carro con placa "xxx" debe pagar el monto calculado.	float costoTotal, string mensajeSalida
Descripción:	Registra la salida de un carro del parqueadero y calcula un monto a pagar según el tiempo que estuvo en el estacionamiento.	<pre> int minutosEntrada = convertirHoraEnMinutos(parqueadero[i].vehiculo.horaEntrada); int minutosSalida = convertirHoraEnMinutos(horaSalida); int duracion = minutosSalida - minutosEntrada; int horas = duracion / 60; int minutos = duracion % 60; totalPagar = (horas + (minutos &gt; 0 ? 1 : 0)) * tarifaPorHora;           </pre>

### REQUERIMIENTO 3

Nombre:	Consultar la cantidad de puestos disponibles	Código
Parámetros:	Ninguno	Ninguno
Retorno:	- <b>puestos disponibles</b> (int): Cantidad de puestos disponibles. - <b>mensaje</b> (string): "Hay x puestos disponibles."	int disponibles, string mensaje
Descripción:	Informa cuántos de los 40 puestos están disponibles.	<pre> int disponibles = 0; for (int i = 0; i &lt; TOTAL_PUESTOS; i++) {     if (!parqueadero[i].ocupado) {         disponibles++;     } } cout &lt;&lt; "Hay " &lt;&lt; disponibles &lt;&lt; " puestos disponibles." &lt;&lt; endl;           </pre>

#### REQUERIMIENTO 4

Nombre	Avanzar el reloj del parqueadero	Códigos del nuevo código proporcionado
<b>Parámetros</b>	<p>- <b>horasAvanzar</b> (int): Cantidad de horas que se desea avanzar el reloj. Valor entre 1 y 12.</p>	<pre>int TOTAL_PUESTOS = 40; double tarifaPorHora = 5.0; struct Propietario {...}; struct Vehiculo {...}; struct Puesto {...};</pre>
<b>Retorno</b>	<p>- <b>mensaje</b> (string): Mensaje indicando cuántas horas ha avanzado el reloj y cuál es la hora actual tras el avance.</p> <p>- <b>error</b> (string): Mensaje de error si el valor ingresado excede el horario permitido (si se intenta avanzar más de 12 horas).</p>	<pre>bool validarCedula(string cedula); bool validarPlacaCarro(string placa); bool validarPlacaMoto(string placa); int convertirHoraEnMinutos(string hora);</pre>
<b>Descripción</b>	<p>Permite avanzar el reloj de la aplicación del parqueadero en una cantidad determinada de horas. Si el número de horas ingresado es válido (entre 1 y 12), el sistema avanza el reloj y muestra el estado actualizado. Si el número de horas es inválido, devuelve un mensaje de error.</p>	<pre>if (opcion == '1') {...} if (opcion == '2') {...} if (opcion == '3') {...} if (opcion == '4') {...} if (opcion == '5') {...}</pre>