Sistema de Parqueadero

Parking Center S.A.S.

Logotipo, nombre de la empresa

El contenido generado por IA puede ser incorrecto.

Presentación técnica para el cliente .

Juan Alejandro Tamayo Manzano.

Javier Steven Solís Ruiz.

Johan Sebastián Jojoa Meneses.

Sergio Andrés Bustos Mondragón.

Técnico en Programación de Software – SENA.

Fecha de solicitud: 29/Agosto/2025

Fecha de entrega: 15/Septiembre/2025



Parking Center S.A.S.

Asunto: Desarrollo de Prototipo para Sistema de Registro y Cobro de Parqueadero

Introducción:

La empresa Parking Center S.A.S., encargada de operar varios parqueaderos en centros comerciales y zonas empresariales, desea desarrollar un sistema básico de control para registrar el ingreso y salida de vehículos, y calcular el cobro de acuerdo con el tiempo de permanencia. Este sistema será usado inicialmente como simulador de caja para capacitar a nuevos empleados y formar aprendices técnicos en desarrollo de software.

En esta presentación técnica les explicaremos desde como fue el análisis,hasta el paso a paso de todo esto



Planteamiento del problema:

Esta empresa, requiere un sistema de control para sus parqueaderos, contando con detalles simples pero importantes como pedir la placa, el tipo de vehículo hallar su hora de entrada, su hora de salida,etc. Todo esto para un sistema optimizado y útil de alto nivel para sus parqueaderos..

Requerimientos técnicos del sistema:

1. Datos de los vehículos

o Cada vehículo se representa con una tupla: (placa, tipo, hora\_entrada)

o Todos los vehículos activos se guardan en una lista

o Al salir, se calcula el valor y se guarda en el historial

2. Tipos de vehículos y tarifas

o Moto: $1.000 por hora

o Carro: $2.000 por hora

o Camioneta: $2.500 por hora

3. Menú interactivo con bucles y condicionales

o Registrar ingreso

o Registrar salida

o Consultar parqueadero actual

o Ver total recaudado

o Salir

4. Fórmulas matemáticas

o Calcular horas de permanencia y multiplicar por tarifa

o Simular el ingreso/salida con horas ficticias (número entero entre 0 y 24)

5. F-strings

o Mostrar recibos claros: placa, tipo, horas, tarifa por hora, total a pagar.

Restricciones adicionales:

• No se permite registrar dos veces la misma placa sin haber salido.

• El sistema debe mostrar mensajes claros de error.

• Si se registra una salida con placa no existente, debe advertirlo.

Atentamente,

Javier Ortega

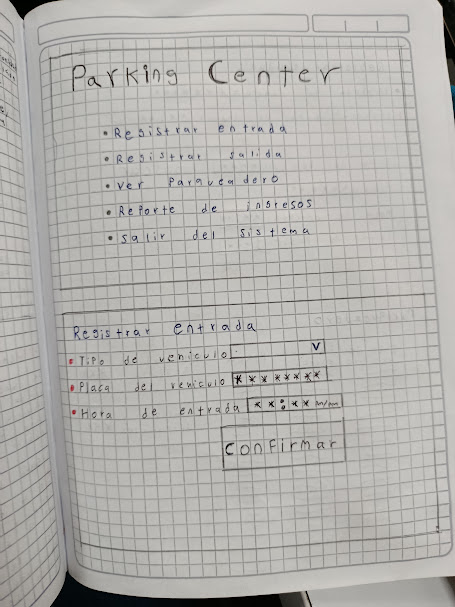
Coordinador de Operaciones Parking Center S.A.S.

PROCESO POR PARTE DEL EQUIPO DE DESARROLLADORES.

-------------------------------------------------------1. Mockups -----------------------------------------------

Para empezar a desarrollar para la empresa Parking Center S.A.S,empezamos por una base solida de Mockups; para todos juntos determinar según los requerimientos y detalles oficiales del cliente el diseño completo del aplicativo para sus parqueaderos; empezando por avances pequeños, para luego corregir y cambiar los detalles técnicos y decorativos para el cliente, y mejorar el desarrollo del aplicativo…

PRIMEROS AVANCES: Aquellos hechos desde el inicio para tener una base solida en el desarrollo



Primero hacemos el boceto general, el menú que muestre todas las opciones disponibles

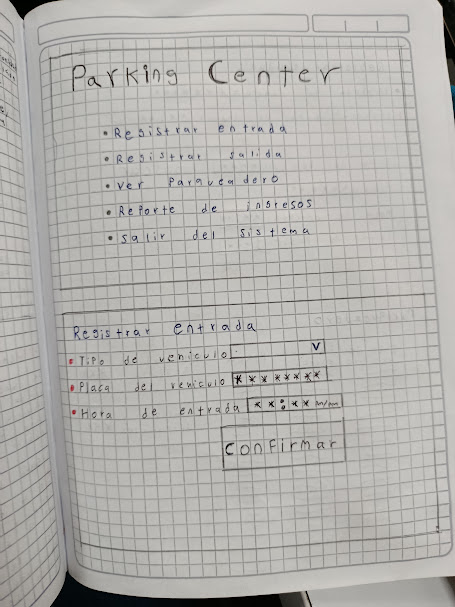
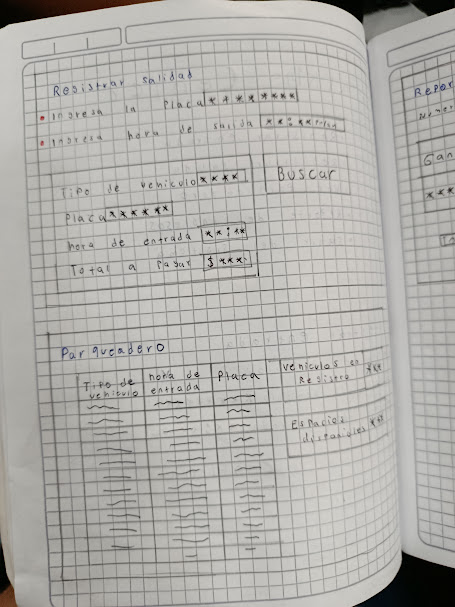
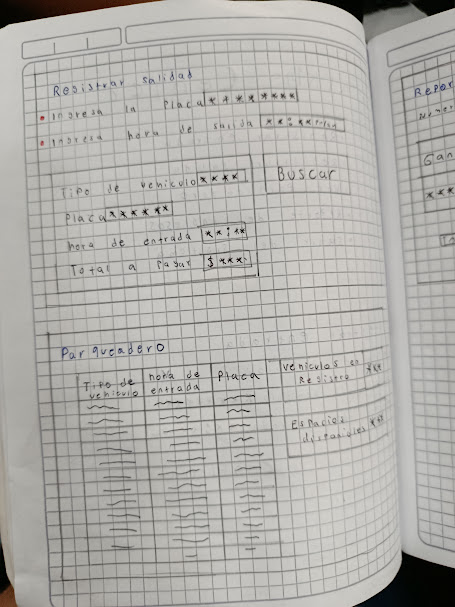


Imagen que contiene Interfaz de usuario gráfica

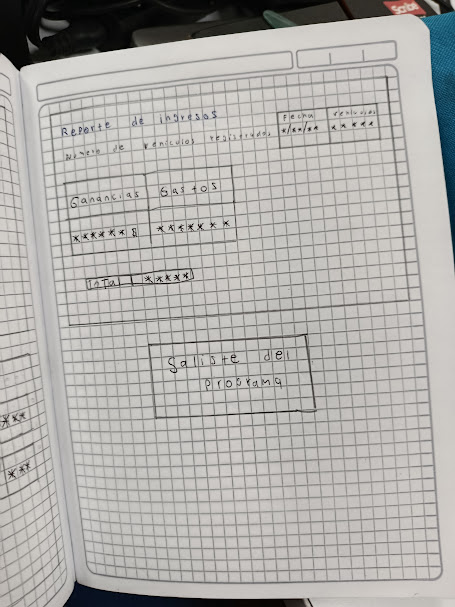
El contenido generado por IA puede ser incorrecto.Para luego hacer el boceto de registrar la entrada; La primera opción que nos pidió el cliente

Luego el de registrar salida La segunda opción que nos pidió el cliente..

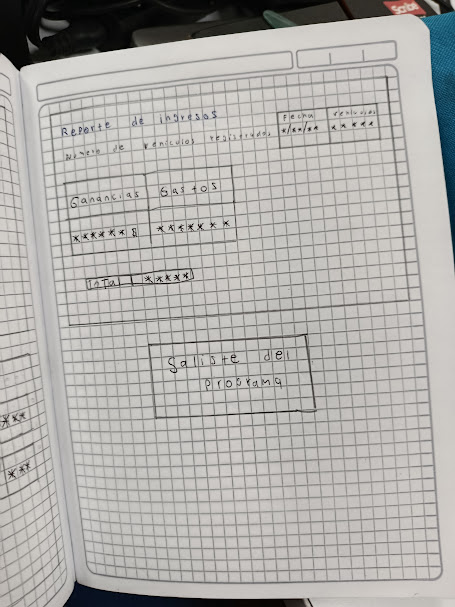
Junto al recibo de pago que recibe el usuario al salirse del parqueadero



Luego consultar parqueadero; La tercera opción que nos pidió el cliente para poder saber cuantos vehículos y cuales hay



Luego reporte de ingresos; La cuarta opción que nos pidió el cliente, para ver sus ganancias del día

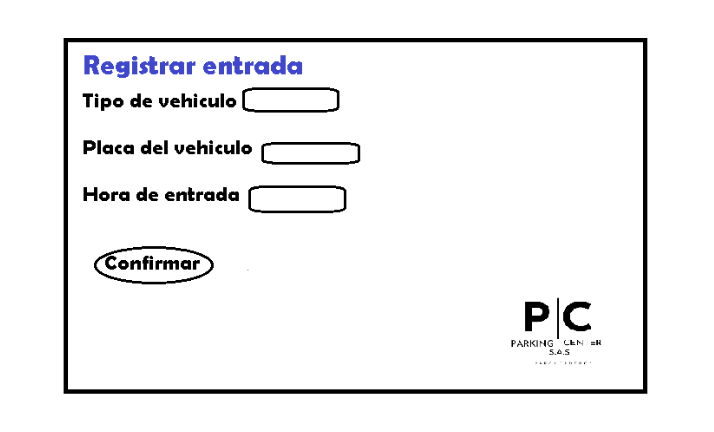


Luego finalmente salir del programa; La quinta opción que nos pidió el usuario, para que una vez elegida una opción cualquiera de las anteriores, o no, salir del sistema

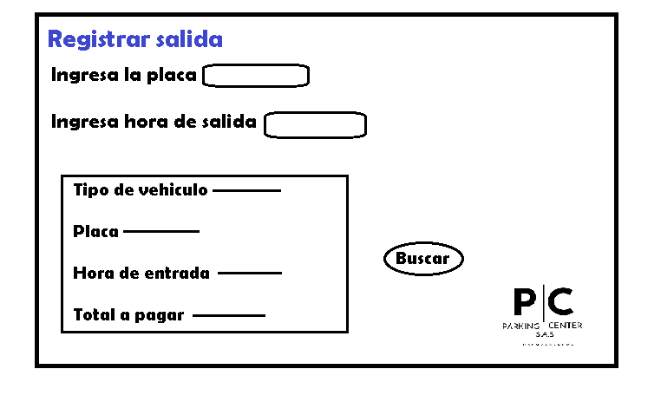
AVANCES FINALES A ENTREGAR: Aquellos finales para luego empezar con el planeamiento final; diagrama de flujo, la codificación, la entrega final



El menú que muestra todas las opciones disponibles



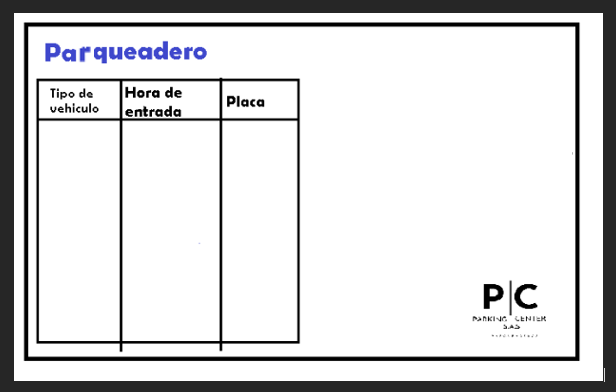
La primera opción; Registrar entrada con sus respectivos datos que pidió el cliente.



La segunda opción; Registrar salida con sus respectivos datos y el recibo de pago para el usuario

Imagen que contiene Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.



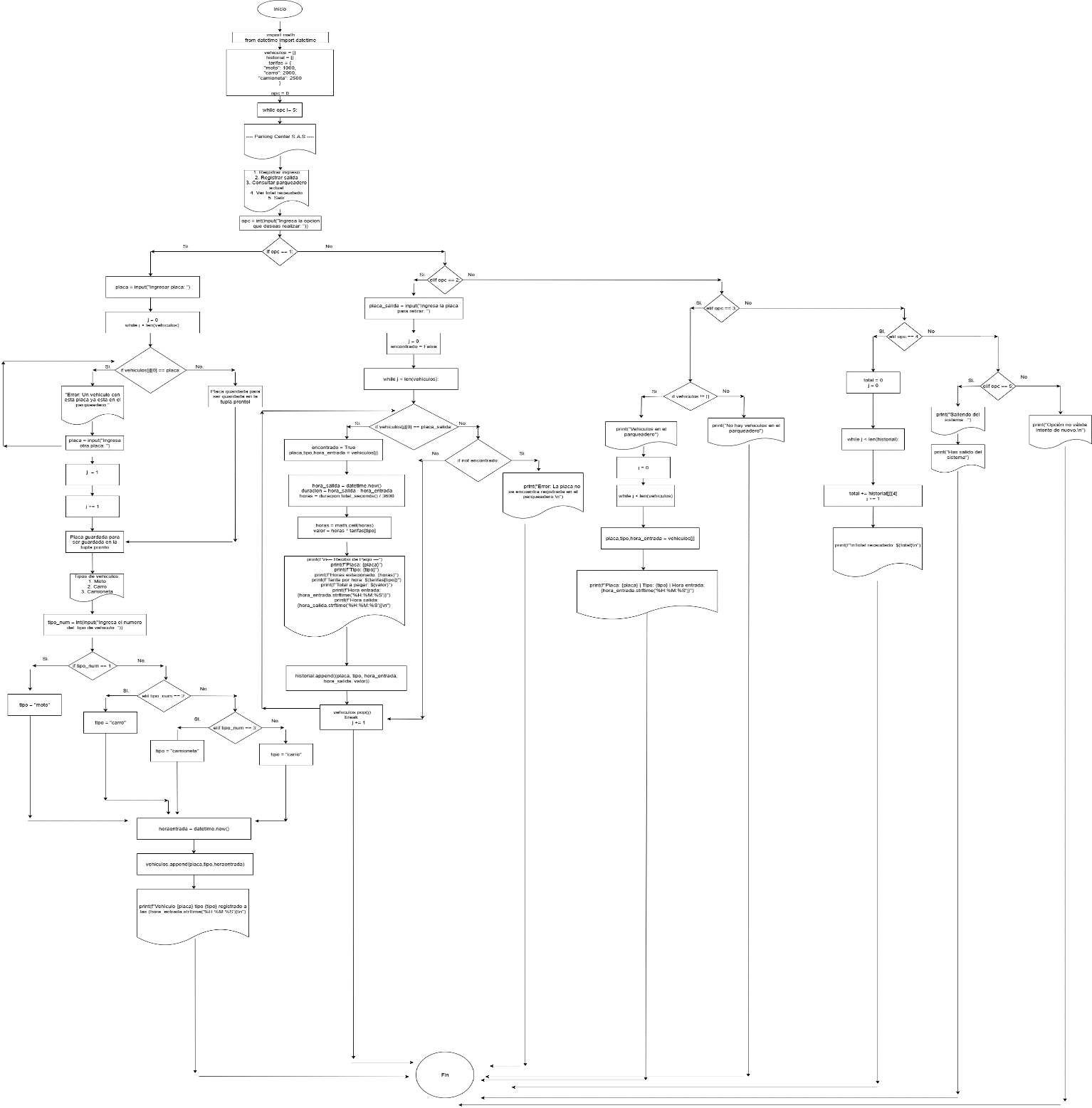
La tercera opción; Consultar parqueadero donde muestra los datos que se ven en el mockup



La cuarta opción; Reporte de ingreso, donde se le mostrara como Fecha: Hoy, y ganancias las que se haya hecho en ese día

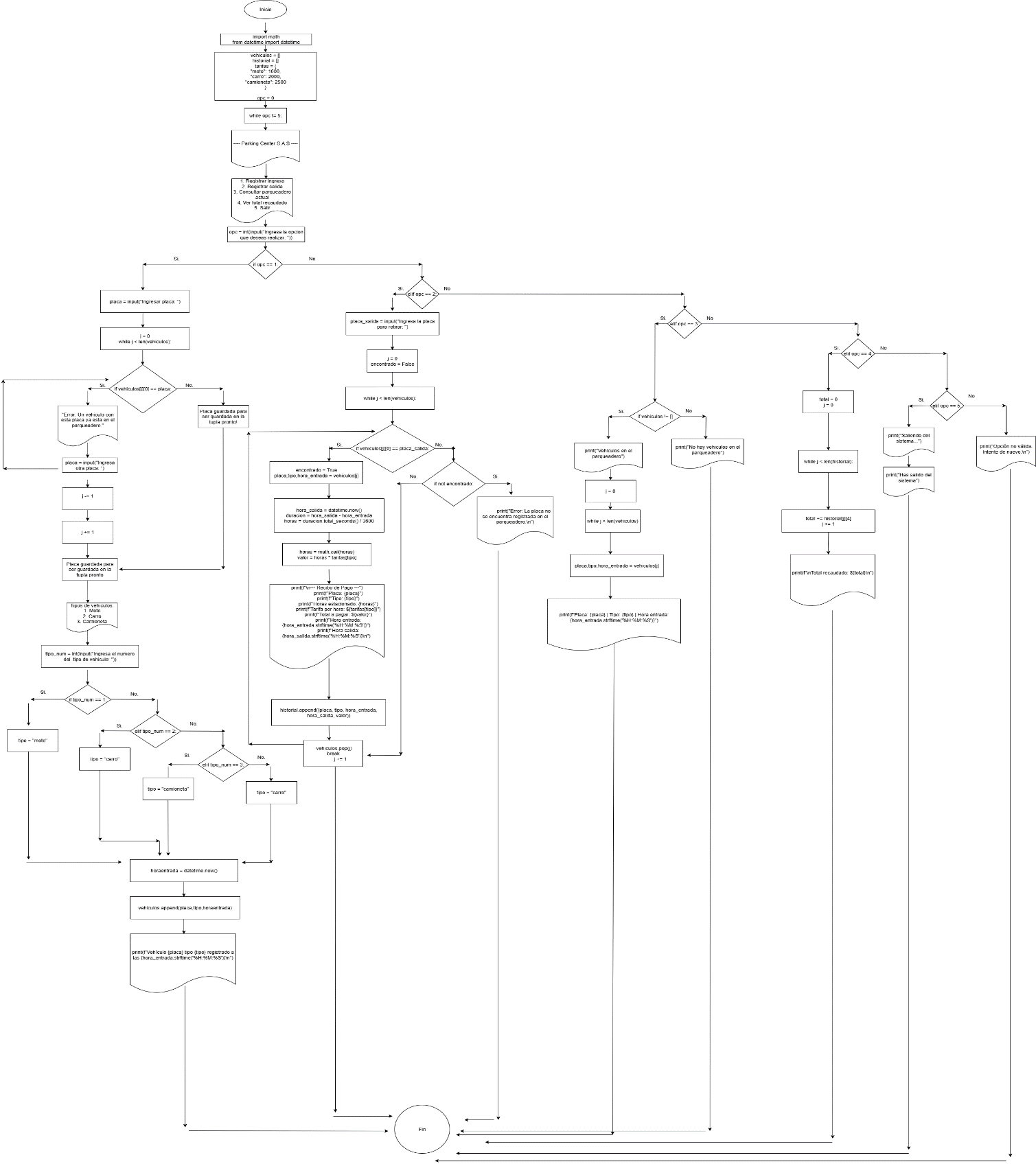


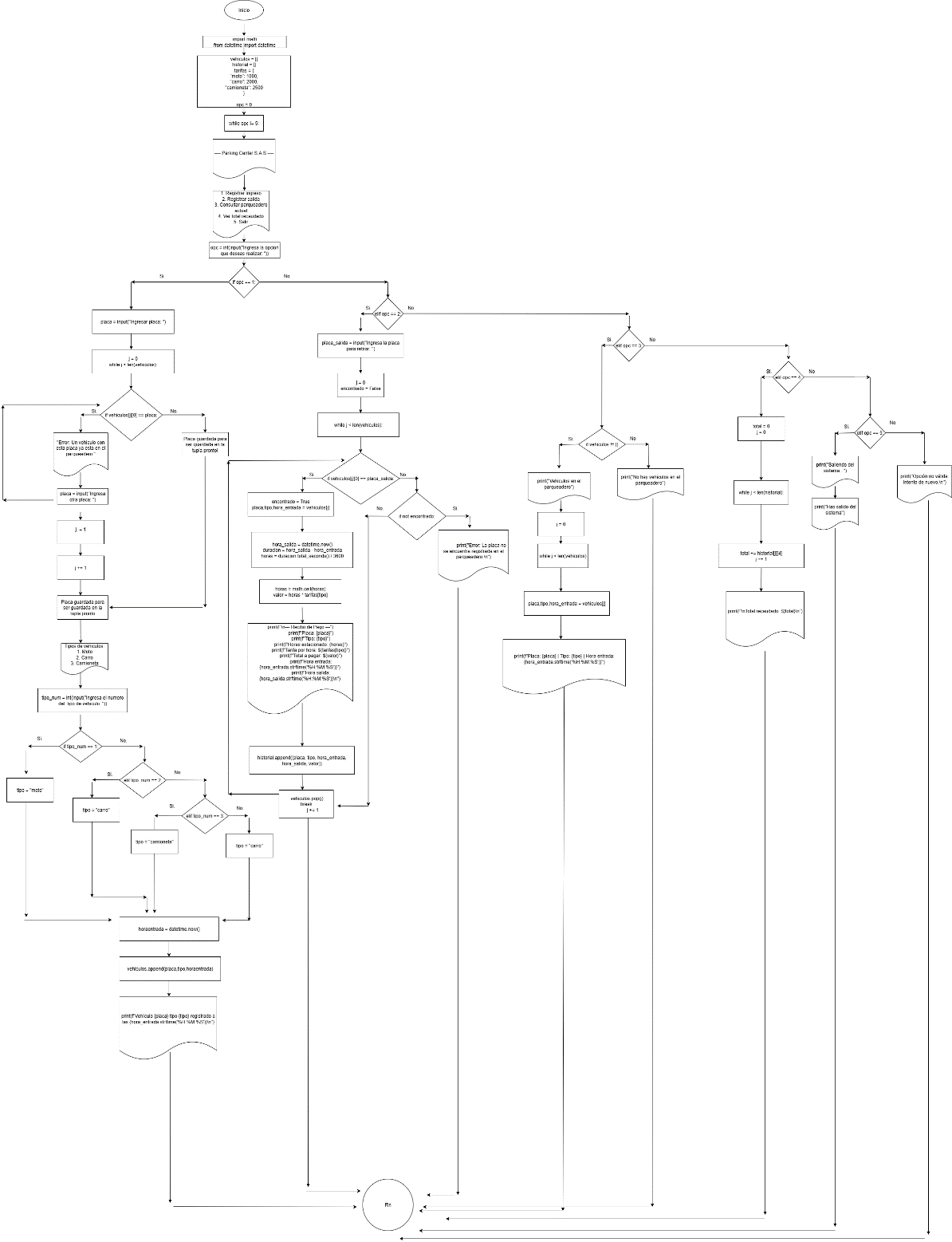
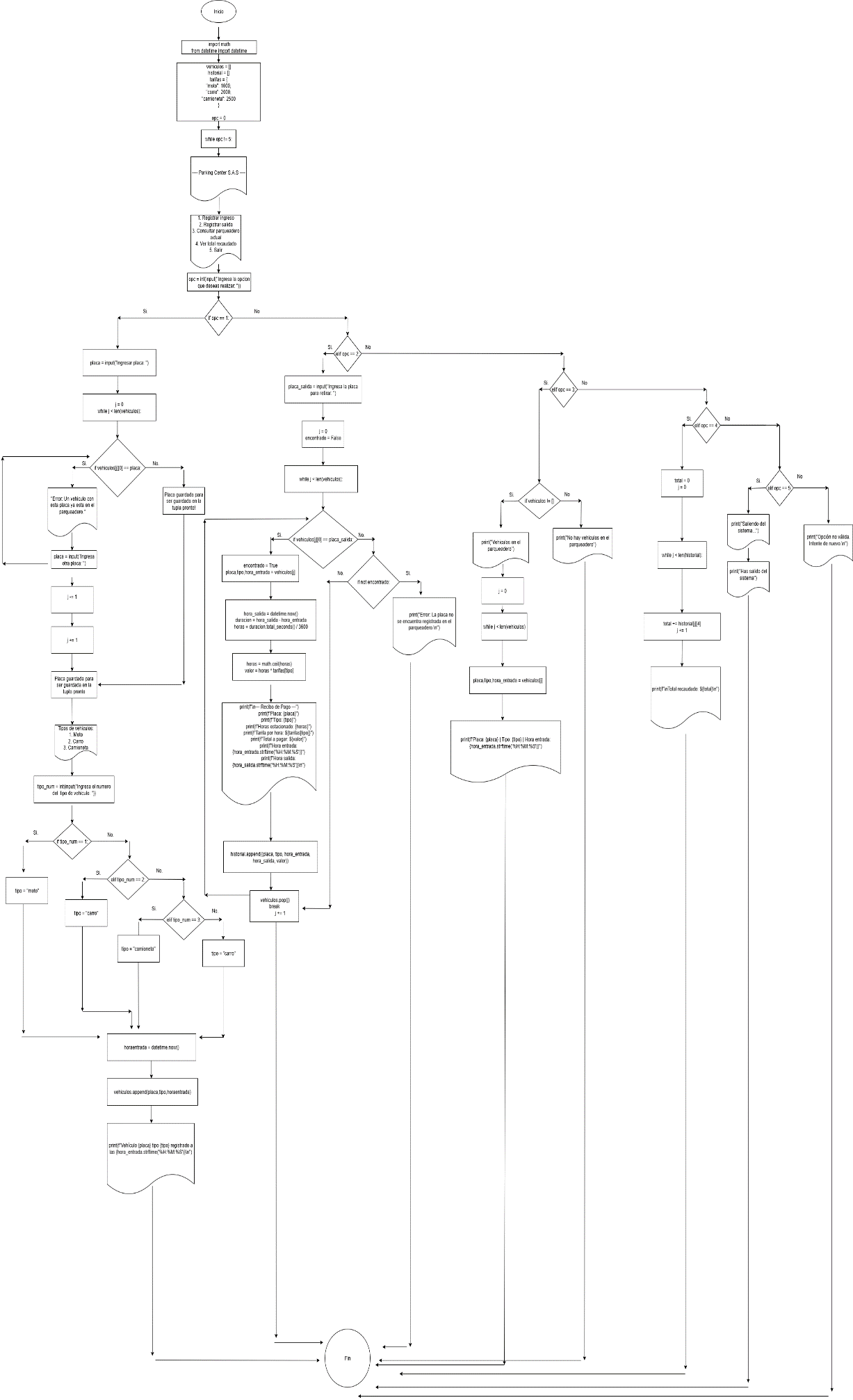
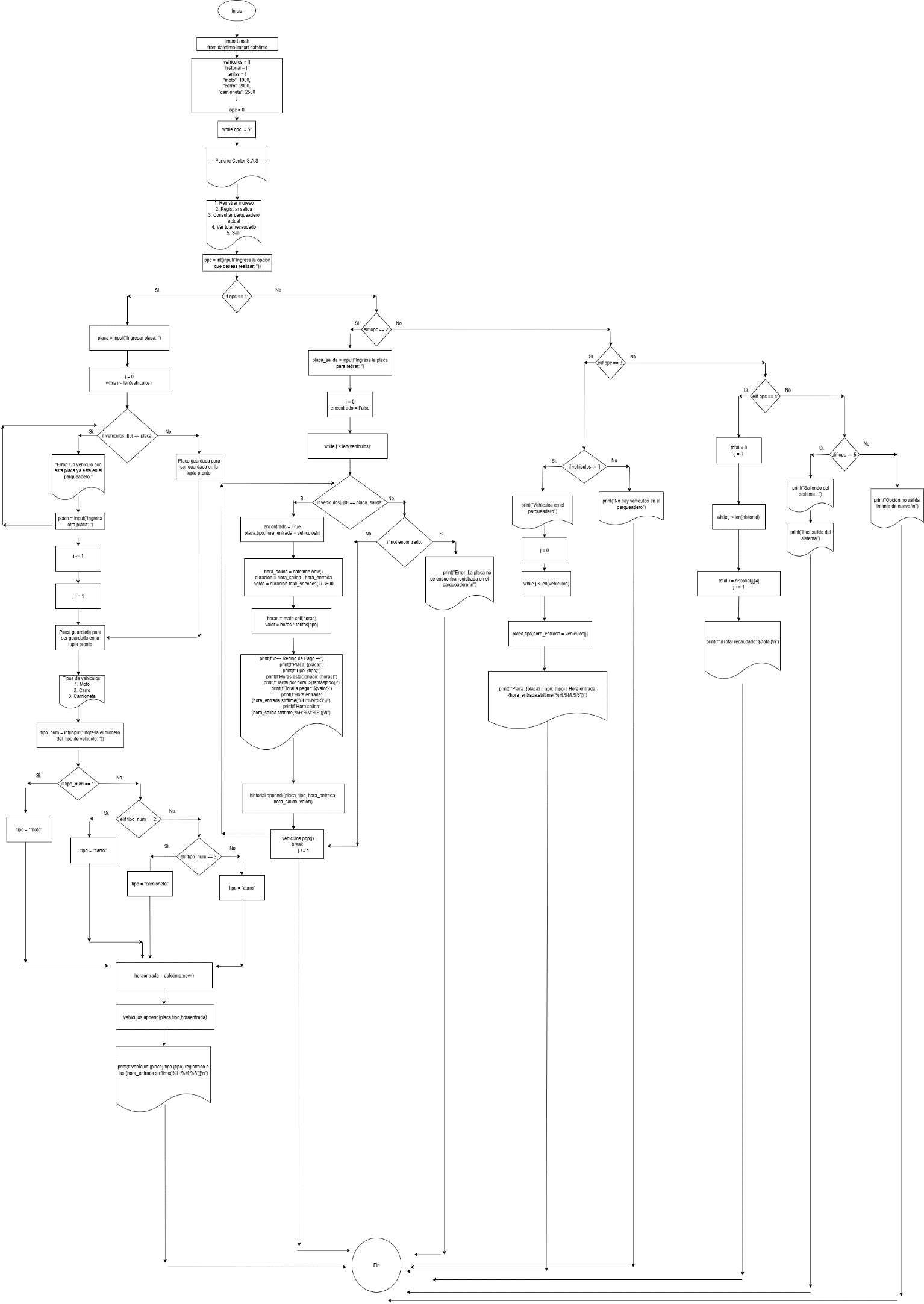
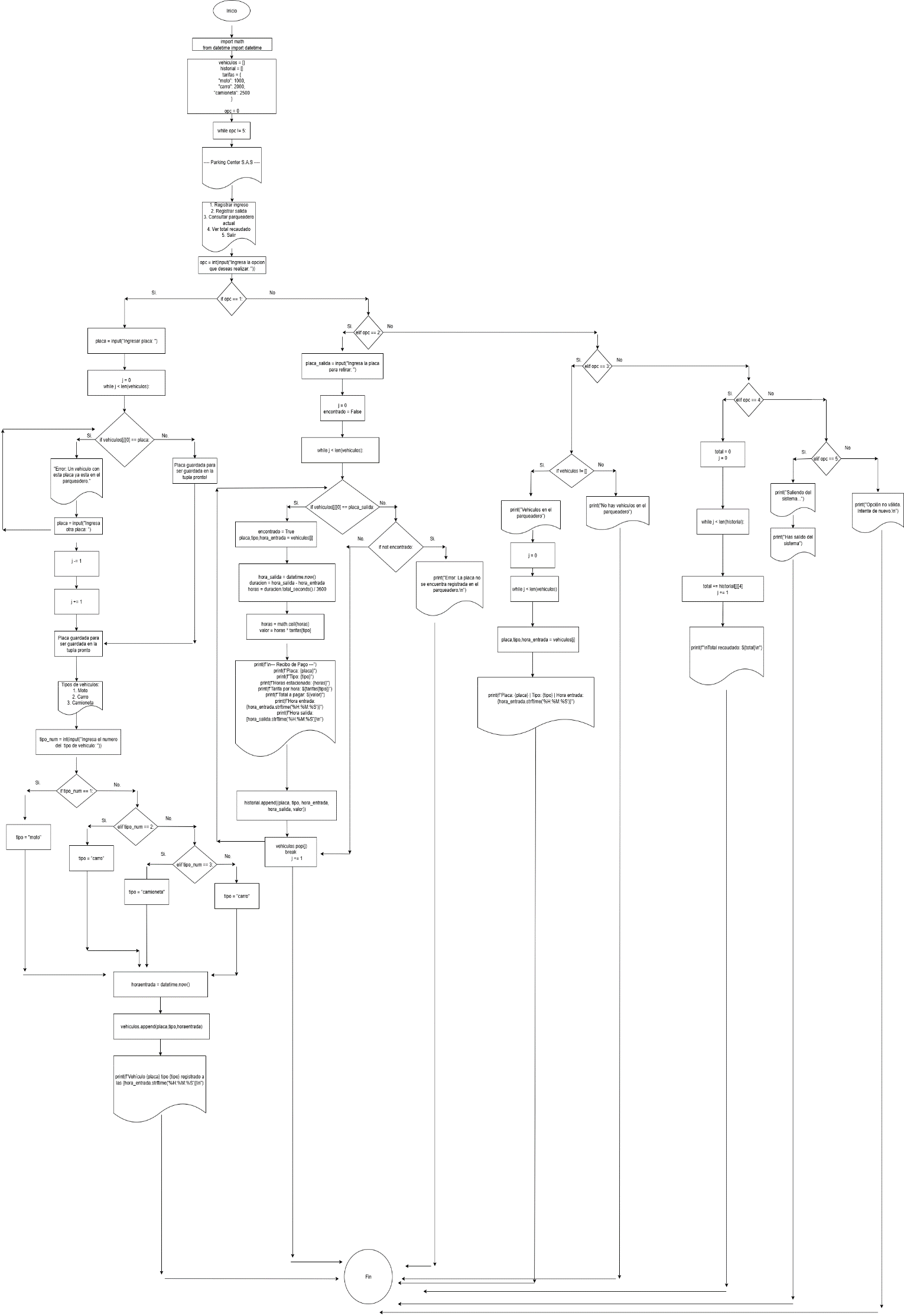
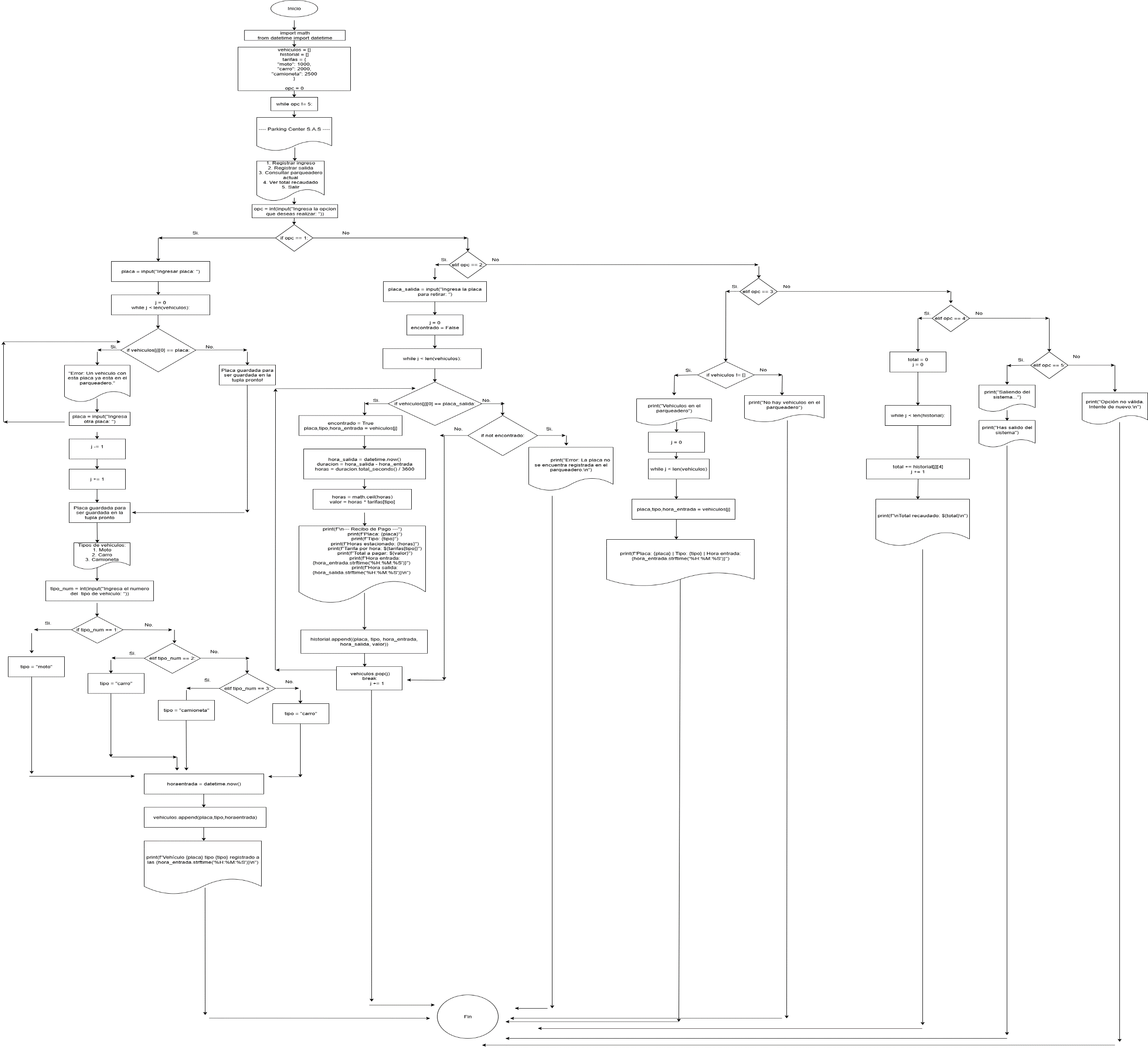
La quinta y ultima opción; Salir del programa, donde se le mostrara al usuario el mensaje que se ve en el mockup.. Siendo así la primera parte del desarrollo; los mockups,bien diseñados y mejorados para la entrega hacia el cliente

--------------------------------------------- 2. Diagrama de flujo ----------------------------------------------

Donde se evidencia el funcionamiento del programa, empezando con la codificación del programa primeramente con el diagrama de flujo para poder analizar y resolver futuros problemas a la hora de codificar..

Donde se evidencia cada de las 5 opciones que dio el cliente, y que debajo de estas todas las funciones que tienen

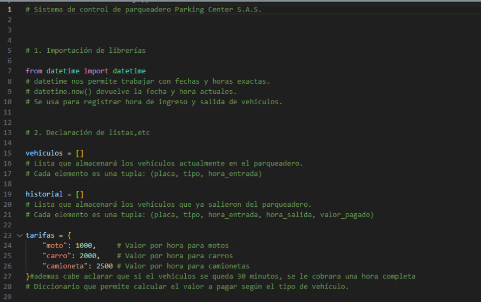
Opción 1 Opción 2 Opción 3 Opción 4 Opción 5 Filtro



Y abajo se evidencia el proceso que conlleva cada opcional opción 1 entrar vehículos, la 2 eliminarlos, la 3 consultar el parqueadero, la 4 ver las ganancias, la 5 salir, y el filtro por si puso un numero diferente a 1,2,3,4,5

------------------------------------------- 3. Codificación y su estructura -----------------------------------

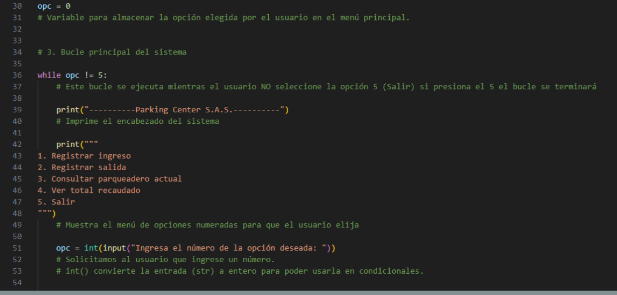
Lo seguido de la lógica; El diagrama de flujo, donde pensamos como hacerlo, como iba ser y planear un código optimizado final para codificarlo



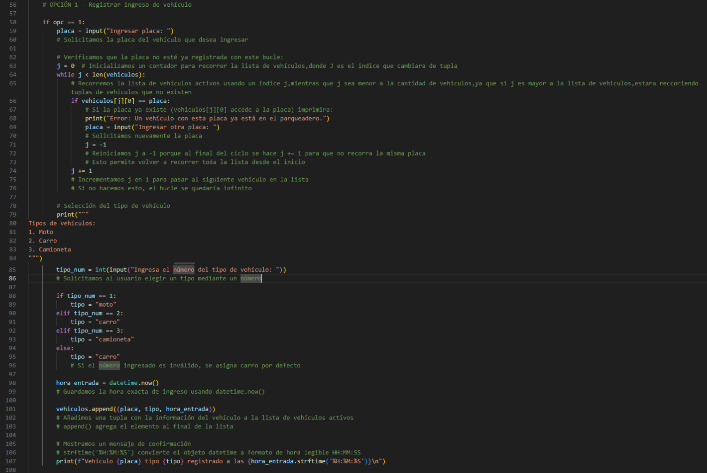
Donde primero se empieza por las variables mas importantes; La lista de vehículos activos que almacenara cada vehículo en una tupla con sus datos; Placa,tipo,y la hora de entrada

Y de seguido, la lista contraria, la lista de vehículos inactivos que almacenara cada vehículo en una tupla con sus datos; Placa,tipo,y la hora de entrada, y hora de salida y valor que pago de los que ya se fueron y pagaron del parqueadero

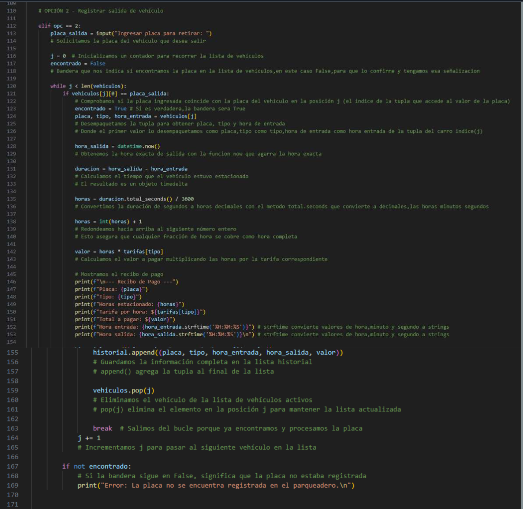
Y seguido las tarifas, el precio que se cobra por hora a las motos,carros,y camionetas

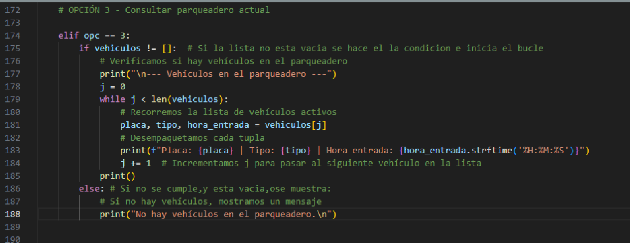


Luego empezamos por el MENÚ, creamos la variable opc = 0,que será la variable que usaremos en el bucle si el usuario escribe del uno al 5 o si escribe otra fuera de este rango para las condicionales en el bucle, y claramente el funcionamiento del bucle.

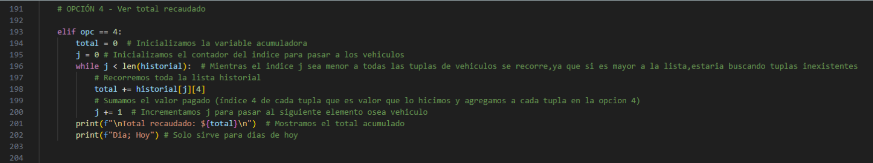
Y el bucle empieza, mientras no sea igual a 5 el bucle correrá, e imprimirá luego las opciones que el usuario tiene por elegir y hacer, donde luego se le pide el numero según su opción elegida

Para luego empezar con la OPCION 1,con las condicionales, pidiéndole la placa para ingresar si la opción es 1,donde habrá un índice contador j = 0 que recorrerá cada vehículo con el j+ = 1 y reiniciara con el j -= 1,donde entra un bucle anidado, que es mientras j sea menor a todos los vehículos que hay,por que si no, el índice j analizara tuplas no existentes, dónde luego si la posición [0] del vehículo (ósea la placa) es igual a la que ingreso al usuario dará error ya que son repetidas, y luego se le pide otra placa nuevamente hasta que ingrese una que no este… Pasado el tiempo de dar la placa, se le pide el tipo de vehiculo,donde se asigna una variable a cada tipo dependiendo de la que ingreso el usuario una vez mostrados los tipos… Luego se halla la hora de entrada con datetime,que saca la hora exacta en la que esa variable este corriendo, y luego se añade a la lista de vehículos, la tupla del carro con su placa, tipo y hora de entrada, y se imprime luego lo que hizo…

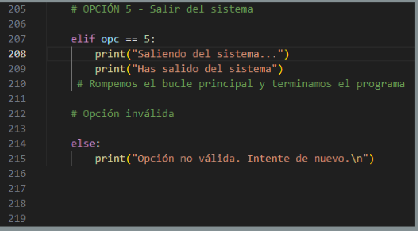


Para luego, ir a la OPCION 2,donde empezamos pidiendo la placa para retirar, y volvemos con nuestra variable j = 0 indexativa para recorrer cada tupla o vehículo una bandera = false, para que Python interprete que el vehículo aun no ha sido encontrado, y empezamos con otro bucle anidado con el mientras j sea menor a todos los vehículos, como dijimos para que no cuente vehículos que no existen, y si la posición [0] ósea la placa del vehículo de índice J,es igual a la de placa de salida que dio el usuario significa que si esta en el parqueadero, por lo tanto se pone la bandera = True para que Python interprete que el vehículo ya fue encontrado, y luego desempaquetamos la tupla para sacar todos sus valores que tiene; placa,tipo,hora\_entrada,y hallamos la hora de salida nuevamente con datatime,y hacemos un calculo simple para calcular las horas que estuvo en el parqueadero que se redondea luego junto con mathceil (proveniente de la librería math) para que redondee si la hora tiene minutos como decimales, y luego simplemente se calcula todo y se muestra el recibo y se añade ese vehículo al historial con el índice J y se borra de la lista de vehículos, por que ya no esta activo ese vehículo se rompe el bucle, y pasa al siguiente, pero si luego de todo eso, aún no ha sido encontrado, imprimirá error

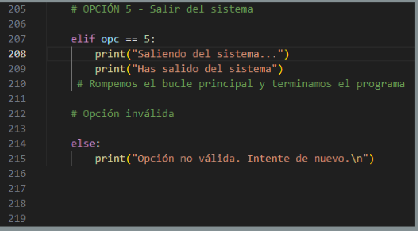
Luego vamos a la OPCION 3,donde se empieza con el condicional si la lista de vehículos no esta vacía empieza el bucle, pero si no, imprime que no hay vehículos, donde si no esta vacia,empieza con la variable indexativa J para recorrer cada vehículo y el bucle anidado, donde es otra vez mientras J sea menor a todos los vehículos que estén activos para que no cuente vehículos que realmente no existen, y luego se desempaqueta la tupla con ese índice J del vehiculo,para imprimir cada vehículo en el que esta con su placa, tipo y hora de entrada para que se diferencie cada uno



Luego vamos a la OPCION 4,donde empezamos con la variable acumulativa 0 para luego hallar el total de dinero ganado, luego el índice J nuevamente para cada vehículo que ya salió y pago, y otra vez el bucle que mientras J sea menor a todos los vehículos de el bucle para no contar vehículos no existentes, y luego la posición 4,que es el valor que pago cada vehículo inactivo (ósea que salió del parqueadero y pago su tarifa) que hicimos en el recibo de pago en la opción 2 de registrar salida, y a todo eso se lo sumamos al total, para que el índice J vaya a cada tupla de vehiculo,agarrando el valor que pago y sumándoselo a la variable total hasta que llegue al ultimo vehículo y muestre todo lo ganado, a excepción de que si no hay vehículos inactivos, ósea que no han pagado, no pasara nada, solo mostrara 0



Luego la ultima; OPCION 5,en que se rompe el bucle principal ya que es while opc != 5,ósea que se rompe, y muestra que se salió del sistema



Luego si no puso del rango 1 a 5,se imprime opción no valida..

Siendo así la estructura final del código y siendo optimizado por muchos intentos, con la estructura mas clara y fácil de entender posible, siendo un código fácil de maniobrar, manejar y experimentar en la terminal, para los fines del cliente pedidos; Sistema de control de parqueadero

---------------------------------------------- 4. Análisis y mejoras ---------------------------------------------

En esta parte, se analizo el problema paso a paso; desde los posibles mejoramientos o problemas para el código y diagrama de flujo para mejorarlo y optimizarlo para la entrega final… Dando mejores pasos como mejor documentación, mejor estética en el diagrama de flujo para mayor legibilidad,etc…

---------------------------------------------------- 5. Presentación ---------------------------------------------



En esta parte, se hizo la respectiva presentación para aclarar como fue nuestra experiencia, hablando de manera un poco mas cómoda y no tan técnica y formal el análisis y desarrollo del Sistema de control de parqueadero; siendo una presentación desde un punto de vista analítica sobre el problema, no sobre contar como fue, sino el análisis que hicimos a este y contarlo…

--------------------------------------------------- 6. Presentación técnica -----------------------------------

Ya luego de hacer la presentación analítica y menos técnica, se hizo esta; La presentación técnica y completa de como fue todo el proceso del desarrollo; Desde el análisis (lo que se presento en la anterior presentación),el proceso; de como fue el código, el diagrama, de cuál fue el paso a paso,etc,las simulaciones de los 5 ejercicios, y luego la conclusión...

--------------------------------------- 7. Simulación de 5 ingresos y 5 salidas -----------------------------

-------------------------------------------------ENTRADAS-------------------------------------------------------

Texto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

---------------------------------------------------- SALIDAS------------------------------------------------------

Texto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

--------------------------- Consultar parqueadero actual ----------------------------------------

Texto

El contenido generado por IA puede ser incorrecto.

------------------------------- Ver total recaudado ---------------------------------------------------

Texto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

------------------------------------------- Salir del sistema -----------------------------------------------

Texto

El contenido generado por IA puede ser incorrecto.

------------------------------- Opción invalida ------------------------------------------------------------------

Texto

El contenido generado por IA puede ser incorrecto.

(Se hizo borrando la terminal ya que al poner 5 se pausa todo el sistema

--------------------------------------------------------8. Conclusión -------------------------------------------

El sistema de parqueadero desarrollado en Python permite registrar ingresos y salidas de vehículos, calcular de forma automática el tiempo y el valor a pagar, consultar los vehículos en el parqueadero y ver el total recaudado. Su uso de listas, tuplas, datetime y math.ceil() asegura un funcionamiento correcto y sin errores, ofreciendo una solución práctica y eficiente para la gestión básica de un parqueadero para la empresa Parking Center S.A.S..

Concluyendo también que con este proyecto aprendimos a manejar listas, tuplas y el módulo datetime en Python. También entendimos cómo aplicar la lógica de condicionales y cálculos para resolver un problema real…

GRACIAS POR SU ATENCION…

Atte: Equipo de programadores

Juan Alejandro Tamayo

Sebastián Jojoa Meneses

Javier Steven Solís Ruiz

Sergio Andrés Bustos