

## **A) Reporte de Visión**

### **Descripción general:**

El software de gestión para el parqueadero Smartparking de la Universidad de Antioquia será una aplicación de consola desarrollada en Python, enfocada en facilitar la administración de usuarios y vehículos, el cálculo de cobros y la generación de reportes administrativos. Su diseño armonizado permitirá un flujo de trabajo ágil entre los turnos de atención, garantizando la disponibilidad y correcta facturación de las celdas de parqueo.

### **Objetivos:**

- Automatizar el registro de ingreso y salida de vehículos.
- Calcular y registrar cobros de manera precisa según el tiempo de estancia.
- Generar facturas visualizables en pantalla y exportables en formato CSV.
- Proporcionar reportes administrativos para análisis de ocupación, facturación y estadística de uso.
- Garantizar la seguridad y consistencia de la información de usuarios y transacciones.

### **Beneficios:**

- Eficiencia operacional: Reducción de tiempo y errores en el proceso manual de registro y cobro.
- Trazabilidad: Historial digital completo de ingresos, retiros y facturación.
- Disponibilidad de datos: Reportes en tiempo real y exportación de resultados para análisis financiero.
- Escalabilidad: Estructura de software que permite futuras ampliaciones (por ejemplo, acceso web).

## **B) Especificación de Requisitos**

### **1. Requisitos Funcionales**

#### **Registro de usuarios:**

- Capturar nombre y apellido con validación de longitud mínima de 3 letras y sin números.
- Capturar documento con longitud entre 3 y 15 dígitos, solo números.
- Registrar placa con formato de 3 letras seguidas de 3 números.

#### **Ingreso de vehículo:**

- Verificar que el usuario esté registrado.
- Comprobar disponibilidad de celdas (50 espacios en total).
- Asignar automáticamente la celda libre más baja.
- Generar recibo con hora y minuto de ingreso.

#### **Retiro de vehículo:**

- Verificar usuario y placa registrados.
- Calcular tiempo estacionado (horas completas y cuartos de hora).
- Aplicar tarifa de 7.000 COP por hora y 1.500 COP por cada cuarto de hora.
- Garantizar pago mínimo de 7.000 COP.
- Mostrar recibo de salida con desglose de cobros.

#### **Módulo Administrador:**

- Autenticación con usuario y contraseña.
- Reporte de total de vehículos registrados.
- Reporte de vehículos retirados y pendientes.
- Informe del total de pagos recaudados.
- Cálculo de tiempo promedio de estancia.
- Identificación de vehículo con tiempo máximo y mínimo.

- Estado de ocupación por celda.
- Exportación de datos a CSV al finalizar la jornada.

## **2. Requisitos No Funcionales:**

### **Usabilidad:**

- Interfaz de consola clara y amigable, menús numerados.
- Mensajes de error descriptivos y ayudas de validación.

### **Rendimiento:**

- Operaciones de registro y consulta en tiempo  $O(1)$  o cercano.
- Respuesta inmediata ( $< 1s$ ) en equipos estándar.

### **Seguridad:**

- Almacenamiento seguro de credenciales (contraseñas).
- Manejo adecuado de excepciones.

### **Fiabilidad y Disponibilidad:**

- Recuperación ante fallos: persistencia en CSV al cierre.
- Respaldo automático de datos al iniciar y cerrar sesión.

### **Compatibilidad:**

- Funciona en cualquier terminal con Python.

## **C) Librerías**

Para el desarrollo del trabajo se buscaron cierto tipo de librerías que nos facilitará la resolución del trabajo. Estas librerías nos permiten expandir la información, ofreciéndonos a detalle una perspectiva mas profunda y detallada de la información permitiéndonos plasmar el objetivo que se desea conseguir.

- **Numpy;** Esta librería nos ayuda en cuanto a la ejecución de cálculos numéricos, constituye un soporte en cuanto a problemas que requieran operaciones avanzadas de nivel, así como arreglos de matrices de datos. Su uso en términos técnicos resulta indispensable, por su eficiencia en cuanto al análisis cuantitativo que requiera el ejercicio.
- **Pandas;** Su uso proporciona un análisis tanto en la exploración y la modelación de la información, abarcando estructuras de datos de alto rendimiento. Su especialidad se centra en datos tabulares, facilitando la agrupación, y agregación de la información de conjuntos grandes de datos.
- **Ploty;** Es una librería la cual abarca la creación de gráficos interactivos, permitiendo una mecánica de despliegue de líneas las cuales nos ilustran mejor la situación del problema.

Esta librería proporciona una base mediante la cual se realizan cálculos, y análisis de datos, la cual nos permite la exploración visual de datos, generando una mayor comprensión en la resolución del análisis.

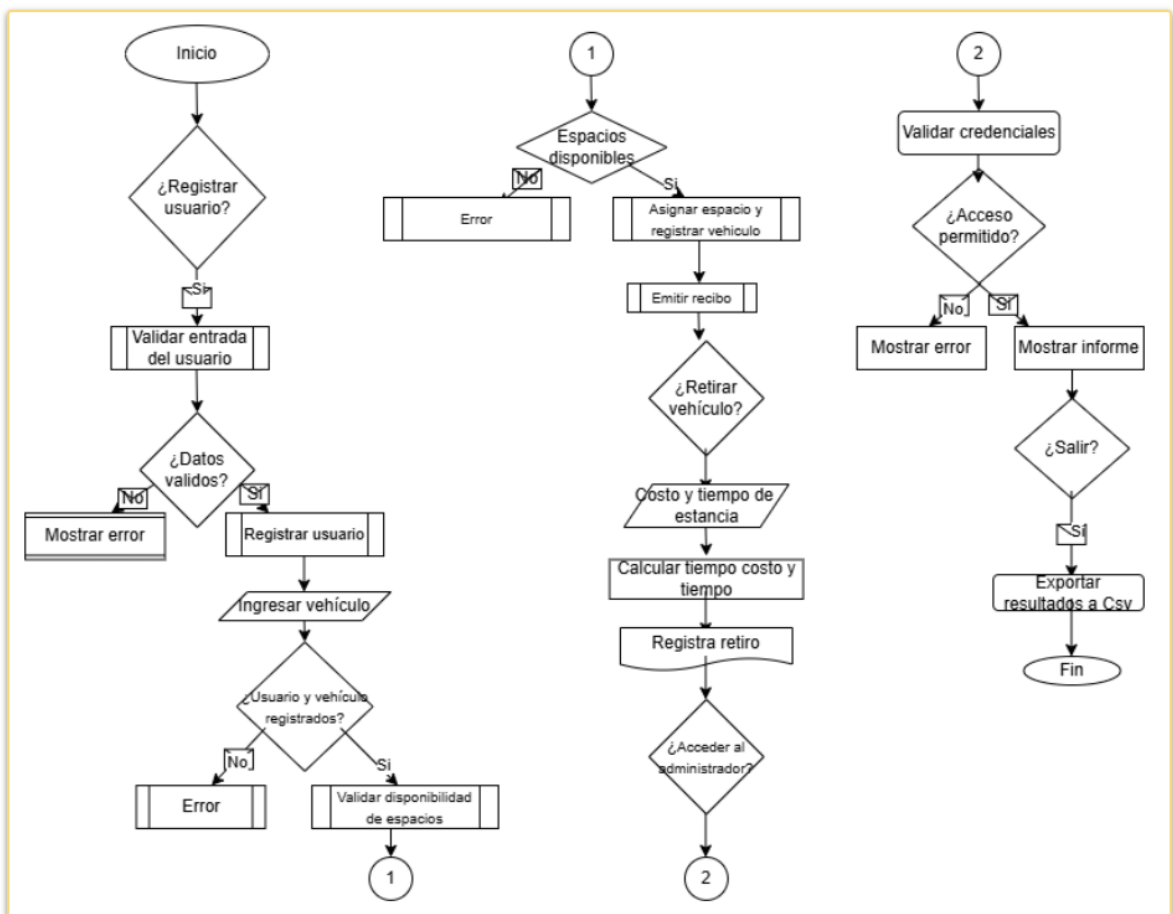
- **Date time;** Su función se basa en la gestión y precisión del tiempo. Es una librería común de Python la cual nos ofrece operaciones temporales como manipular fechas y horas, así como la gestión de zonas horarias. El trabajo en si requiere un algoritmo el cual nos permita la medición del tiempo, por lo cual consideraría que es una librería que nos ayudaría en cálculos de diferencias y un seguimiento preciso de eventos en el tiempo.
- **Tensor Flow;** Es una plataforma de código abierto la cual es clave en la resolución de cálculos numéricos, ya que facilita operaciones complejas, además permite un entorno de aprendizaje sistemático impulsando diversas especializaciones.

Estas librerías nos ayudaran y nos brindaran una ayuda en cuanto a la comprensión del abordaje del problema.

#### **D) Visual:**



## E) Algoritmo:



### • Pseudocódigo del problema:

INICIAR SISTEMA

MIENTRAS el sistema esté activo:

MOSTRAR menú principal

LEER opción del usuario

SI opción es REGISTRAR USUARIO:

SOLICITAR datos del usuario

VALIDAR datos

SI datos válidos:

GUARDAR usuario

SI opción es INGRESAR VEHÍCULO:

VERIFICAR que el usuario esté registrado

VALIDAR disponibilidad de espacio

SI disponible:

ASIGNAR celda

REGISTRAR hora de ingreso

GUARDAR ingreso

SI opción es RETIRAR VEHÍCULO:

BUSCAR vehículo ingresado

CALCULAR tiempo de parqueo

CALCULAR valor a pagar

LIBERAR celda

GUARDAR retiro y pago

SI opción es ADMINISTRADOR:

VERIFICAR usuario y contraseña

SI acceso válido:

MOSTRAR reportes administrativos

SI opción es SALIR:

PREGUNTAR si desea exportar datos

SI sí:

EXPORTAR reportes a CSV

CERRAR sistema

FINALIZAR SISTEMA

## **F) Estructura de datos:**

Para desarrollar el software del sistema del parqueadero, es necesario abarcar distintas estructuras y métodos de Python que ayuden en la resolución del problema. Las siguientes estructuras de datos son las siguientes:

- **Diccionario de usuarios:**

Esta estructura es tipo documento, la cual nos permite un almacenamiento de la información de los usuarios, además de que agiliza una búsqueda rápida en cuanto a la información suministrada, facilitando una validación al momento de ingresar al parqueadero.

Los datos que se almacenan serian, documento, nombre, apellido, placa, espacio asignado y hora de ingreso.

- **Lista celda o parqueadero:**

Una lista de diccionarios la cual representa las 50 posiciones del parqueadero es fundamental ya que nos ayuda a buscar espacios libres de una forma más eficiente.

La información en este inciso seria, estado de ocupación, documento del usuario que la ocupa, hora y minuto de ingreso.

- **Historial o lista de vehículos retirados:**

Su función es generar reportes sobre todos los vehículos retirados calculado variables como datos de tiempo, pago y espacio utilizado.

Se utilizarán datos como, documento del usuario, placa del vehículo, tiempo de estadía, valor cancelado, y celda o espacio utilizado.

- **Diccionario administrador:**

Su función es validar un usuario y contraseña para acceder de una manera rápida al usuario del administrador.

Los datos que almacena este ítem serian, nombre de usuario, contraseña asociada.

- **Lista de usuario ingresados en el parqueadero:**

Mantiene un orden en cuanto a la separación de los usuarios que tienen una estadía de su vehículo en el parqueadero, de los que ya salieron, permitiendo tener información como tiempos máximos, mínimos, y recaudaciones.



Datos que almacena serían, documento, placa, celda asignada, hora de ingreso y estadía.

### **G) Presentación de Resultados del Módulo Administrador**

- **Menú principal de administración:** Lista numerada con cada reporte.
- **Panel resumen:** Mostrar al ingresar totales clave (vehículos totales, retirados, pendientes y recaudación).
- **Tablas en consola:** Listar datos de usuarios y ocupación de celdas con formato tabulado.
- **Indicadores estadísticos:** Tiempo promedio, máximo y mínimo resaltado con texto e iconos ASCII.
- **Exportación:** Opción para generar CSV con todos los datos de la jornada.

