



Facultad de ingeniería

Programa de ingeniería industrial

Curso: Algoritmos y Programación

Semestre 2025-1

**MANUAL DE USUARIO
SISTEMA DE GESTIÓN DE PARQUEADERO
“LUXURY PARKING”**



Tabla de Contenido

1. **Introducción**
2. **Especificación de Requisitos**
 - 2.1 Requisitos Funcionales
 - 2.2 Requisitos No Funcionales
3. **Opciones del Sistema**
 - 3.1 Registrar Usuario
 - 3.2 Registrar Ingreso de Vehículo
 - 3.3 Retirar Vehículo y Calcular el Costo
 - 3.4 Ingreso como Administrador
 - 3.5 Cerrar Día y Generar CSV
 - 3.6 Salir del Programa
4. **Beneficios del Sistema**
5. **Consideraciones Visuales**
6. **Equipo de Desarrollo**

1. Introducción

El presente proyecto consiste en la creación de un sistema de gestión para un parqueadero que presta sus servicios únicamente a automóviles. Mediante el uso del lenguaje de programación Python y la plataforma Google Colab, se busca desarrollar un sistema que permita registrar vehículos, asignar celdas de parqueo, controlar ingresos y salidas, calcular tarifas por tiempo de permanencia y generar informes administrativos en formato CSV. Este sistema es fácil de usar y permite la administración eficiente de hasta 50 espacios disponibles, garantizando la continuidad del servicio en una jornada diaria de 6:00 am a 12:00 pm.

2. Especificación de Requisitos

2.1 Requisitos funcionales

- Registro de usuario: permite registrar nombre, apellido, documento y placa del vehículo.
- Registro de ingreso: valida datos del usuario y asigna celda libre.
- Retiro de vehículo: calcula valor por permanencia.
- Administración: acceso a estadísticas y reportes.
- Cierre de jornada: genera reporte CSV y finaliza el sistema.
- Salida del sistema: permite cerrar sin generar reporte.

2.2 Requisitos no funcionales

- Rendimiento: debe manejar registros sin retrasos.
- Usabilidad: interfaz intuitiva en consola.

- Seguridad: acceso restringido y protección de datos.
- Compatibilidad: adaptable a distintos entornos de parqueo.

3. Opciones del Sistema

3.1. Registrar Usuario

Permite registrar la información de un nuevo usuario y su vehículo.

Paso a paso:

1. Seleccionar la opción 1 en el menú.
2. Ingresar nombre (mínimo 3 letras, sin números).
3. Ingresar apellido (mínimo 3 letras, sin números).
4. Ingresar documento (entre 3 y 15 dígitos, solo números).
5. Ingresar la placa del vehículo (ej: ABC123).
6. Si la información es válida, se guarda en la lista de usuarios.

```
opcion_input = input('Ingrese la opción deseada: ')
if not opcion_input.isdigit():
    print("Debe ingresar un número.")
    continue
opcion = int(opcion_input)

if opcion == 1:
    print('Bienvenido al sistema de registro de usuario.')
    errores_totales = []

    nombre = input('Ingrese su nombre: ')
    errores_totales += validar_nombre(nombre)

    apellido = input('Ingrese su apellido: ')
    errores_totales += validar_apellido(apellido)

    documento = input('Ingrese su documento de identidad: ')
    errores_totales += validar_documento(documento)

    placa = input('Ingrese la placa de su vehículo: ').upper()
    errores_totales += validar_placa(placa)
```

3.2. Registrar Ingreso de Vehículo

Registra la entrada de un vehículo y asigna una celda de parqueo.

Paso a paso:

1. Seleccionar la opción 2.
2. Ingresar la placa.
3. Verificación de registro.
4. Asignación automática de la primera celda libre.
5. Registro de hora de ingreso.
6. Generación de factura parcial.

```

elif opcion == 2:
    print('Bienvenido al sistema de ingreso de vehículos.')
    if not Espacios_disponibles:
        print('El parqueadero está lleno.')
        continue

    placa_ingresar = input('Ingrese la placa del vehículo: ').upper()
    usuario_encontrado = None
    for usuario in Usuarios:
        if usuario[3] == placa_ingresar:
            usuario_encontrado = usuario
            break

    if usuario_encontrado:
        hora_ingreso = datetime.now()
        celda_asignada = Espacios_disponibles.pop(0)
        vehiculo = [usuario_encontrado[0], usuario_encontrado[3], hora_ingreso, celda_asignada]
        Vehiculos_en_parqueo.append(vehiculo)

        registro = {
            "Nombre": usuario_encontrado[0],
            "Placa": usuario_encontrado[3],
            "Hora_ingreso": hora_ingreso,
            "Hora_salida": None,
            "Total_pagado": None,

```

3.3 Retirar Vehículo y Calcular el Costo

Permite retirar un vehículo y calcular el valor a pagar.

Paso a paso:

1. Seleccionar la opción 3.
2. Ingresar la placa.
3. Cálculo del tiempo total.
4. Cálculo: \$7000/hora o fracción (cada 15 min = \$1500).
5. Mínimo a pagar: \$7000.
6. Se muestra tiempo y costo, se libera celda.

```

elif opcion == 3:
    print('Bienvenido al sistema de retiro de vehículo.')
    placa_retirar = input('Ingrese la placa del vehículo a retirar: ').upper()
    vehiculo_encontrado = None
    for vehiculo in Vehiculos_en_parqueo:
        if vehiculo[1] == placa_retirar:
            vehiculo_encontrado = vehiculo
            break

    if vehiculo_encontrado:
        hora_salida = datetime.now()
        hora_ingreso = vehiculo_encontrado[2]
        tiempo_total_min = (hora_salida - hora_ingreso).total_seconds() / 60

        if tiempo_total_min <= 0:
            print("Error: la hora de salida debe ser mayor que la de ingreso.")
        else:
            horas = int(tiempo_total_min) // 60
            minutos_restantes = int(tiempo_total_min) % 60
            cuartos_hora = minutos_restantes // 15

            cobro_horas = horas * 7000
            cobro_cuartos = cuartos_hora * 1500
            total_pagar = max(7000, cobro_horas + cobro_cuartos)

```

3.4. Ingreso como Administrador

Acceso a estadísticas del parqueadero.

Paso a paso:

1. Seleccionar la opción 4.
2. Ingresar usuario y contraseña (admin / admin123).
3. Acceder a:
 - Vehículos registrados y retirados.
 - Total recaudado.
 - Tiempos promedio, máximo y mínimo.
 - Lista de usuarios y estado de celdas.

```
elif opcion == 4:
    print('Ingreso como administrador.')
    usuario = input("Usuario: ")
    contraseña = input("Contraseña: ")
    if usuario in Credenciales_admin and Credenciales_admin[usuario] == contraseña:
        print('Acceso concedido.')

        total_registrados = len(Historico_vehiculos)
        total_retirados = sum(1 for r in Historico_vehiculos if r["Hora_salida"] is not None)
        total_en_parqueo = total_registrados - total_retirados
        total_pagado = sum(r["Total_pagado"] for r in Historico_vehiculos if r["Total_pagado"] is not None)

        tiempos = []
        for r in Historico_vehiculos:
            if r["Hora_salida"] is not None:
                tiempo = (r["Hora_salida"] - r["Hora_ingreso"]).total_seconds() / 60
                tiempos.append((r["Placa"], tiempo))

        tiempo_prom = sum(t for _, t in tiempos) / len(tiempos) if tiempos else 0
        if tiempos:
            max_tiempo = max(tiempos, key=lambda x: x[1])
            min_tiempo = min(tiempos, key=lambda x: x[1])
        else:
            max_tiempo = min_tiempo = ("N/A", 0)
```

3.5. Cerrar Día y Generar CSV

Finaliza jornada y exporta informe.

Paso a paso:

1. Seleccionar opción 5.
2. Se genera 'registro_dia.csv' con:
 - Nombre, Apellido, Placa, Hora ingreso/salida, Total pagado, Celda.
3. El sistema finaliza su ejecución.

```

elif opcion == 5:
    print("Cerrando el día. Generando archivo CSV...")

    with open('registro_dia.csv', 'w', newline='') as archivo_csv:
        writer = csv.writer(archivo_csv)
        writer.writerow(['Nombre', 'Placa', 'Hora de Ingreso', 'Hora de Salida', 'Total Pagado', 'Celda'])

        for registro in Historico_vehiculos:
            hora_ingreso = registro["Hora_ingreso"].strftime('%Y-%m-%d %H:%M:%S')
            hora_salida = registro["Hora_salida"].strftime('%Y-%m-%d %H:%M:%S') if registro["Hora_salida"] else "En parqueadero"
            total = registro["Total_pagado"] if registro["Total_pagado"] is not None else "N/A"
            writer.writerow([registro["Nombre"], registro["Placa"], hora_ingreso, hora_salida, total, registro["Celda"]])

    print("Archivo CSV generado como 'registro_dia.csv'. ¡Hasta mañana!")
    break

```

3.6. Salir del Programa

Termina el sistema sin generar reporte.

Paso a paso:

1. Seleccionar opción 6.
2. Confirmación visual de salida.
3. Programa cerrado.

```

elif opcion == 6:
    print("Saliendo del programa. ¡Hasta luego!")
    break

else:
    print("Debe ingresar una opción válida.")

```

4. Beneficios

Automatización de procesos: Facilita el registro, control y seguimiento de vehículos, eliminando la necesidad de llevar registros manuales.

Asignación eficiente de espacios: Garantiza un uso óptimo de las celdas disponibles mediante asignación automática del primer espacio libre.

Cálculo preciso de tarifas: El sistema realiza el cálculo automático del valor a pagar según el tiempo de permanencia, asegurando precisión y transparencia.

Generación de reportes administrativos: Permite generar informes detallados en formato CSV con los datos del día, facilitando la toma de decisiones y el control financiero.

Acceso a estadísticas en tiempo real: Los administradores pueden consultar datos clave como recaudación total, promedio de estancia, ocupación de celdas y más, en cualquier momento.

Facilidad de uso: La interfaz por menú es intuitiva y amigable, permitiendo una experiencia fluida tanto para operadores como para usuarios sin necesidad de capacitación previa.

Escalabilidad y adaptabilidad: El sistema está diseñado para ser flexible y adaptarse a diferentes contextos de operación de parqueaderos, con posibilidad de expandirse o integrarse con futuras versiones más visuales.

Seguridad de los datos: El registro y procesamiento de información se realiza de manera estructurada y segura, cumpliendo con principios básicos de confidencialidad y fiabilidad.

5. Consideraciones Visuales

La versión inicial del sistema está desarrollada en Google Colab con entradas y salidas por consola. Se proyecta en futuras versiones una evolución hacia interfaces gráficas para aplicaciones de escritorio o plataformas web. Actualmente, el diseño se centra en la simplicidad, legibilidad y facilidad de navegación mediante menús numerados.

Estas serían unas imágenes de como se ve el programa

```
-----
*** MENÚ DE OPCIONES ***
1. Registrar usuario.
2. Registrar ingreso de vehículo.
3. Retirar vehículo y calcular el costo.
4. Ingreso como administrador.
5. Cerrar día y generar CSV.
6. Salir del programa.
Ingrese la opción deseada: 1
Bienvenido al sistema de registro de usuario.
Ingrese su nombre: Juan
Ingrese su apellido: Marin
Ingrese su documento de identidad: 1234
Ingrese la placa de su vehículo: Pab100
Juan, su usuario se ha registrado con éxito.
*** MENÚ DE OPCIONES ***
1. Registrar usuario.
2. Registrar ingreso de vehículo.
3. Retirar vehículo y calcular el costo.
4. Ingreso como administrador.
5. Cerrar día y generar CSV.
6. Salir del programa.
Ingrese la opción deseada: 2
Bienvenido al sistema de ingreso de vehículos.
Ingrese la placa del vehículo: pab100
Vehículo ingresado correctamente en la celda P1
```

***** FACTURA *****

Hora de ingreso: 2025-06-27 17:48:10

Usuario: Juan

Placa: PAB100

Celda asignada: P1

	Nombre	Placa	Hora de Ingreso	Hora de Salida	Total Pagado	Celda
0	Juan	PAB100	2025-06-27 17:48:10	2025-06-27 17:49:19	7000.0	P1

6. Equipo de Desarrollo

Este proyecto fue desarrollado por los estudiantes de Ingeniería Industrial:

- Juan Pablo Marín Duque
- David Fernando Báez Coronado
- Santiago Osorio Pérez

Como parte del proyecto académico de Algoritmos y Programación.