

Tarea 6: Cálculo Tarifa de Parqueo en Python y Octave

Marianne Nicté, Rodríguez Canek, 202000656^{1,*}

¹Facultad de Ingeniería, Departamento de Electronica,
Universidad de San Carlos, Ciudad Universitaria, Zona 12, Guatemala.

INTRODUCCIÓN

En la actualidad, la gestión eficiente de los parqueos es fundamental para asegurar un flujo adecuado de vehículos y un sistema de facturación adecuado. Este proyecto tiene como objetivo desarrollar un programa que permita gestionar el parqueo de vehículos, calcular el tiempo de estancia y generar automáticamente las facturas correspondientes. Utilizando tecnologías como PostgreSQL para la base de datos, y Python para el desarrollo del sistema, se logra una solución automatizada que facilita tanto la gestión como la facturación.

I. EXPLICACIÓN DEL CÓDIGO

El código que se describe a continuación está diseñado para interactuar con el sistema de base de datos PostgreSQL, donde se almacenan los datos relacionados con los vehículos y sus períodos de parqueo. El código se encarga de realizar las siguientes tareas: El código desarrollado en Octave realiza las siguientes funciones:

1. Ingreso de los datos del usuario, como el nombre, NIT y placa del vehículo.
2. Cálculo del tiempo de parqueo y la tarifa a cobrar en base a este.
3. Generación de la factura en formato texto y almacenamiento en un archivo de texto.
4. Inserción de los datos en una base de datos PostgreSQL.
5. Consulta de las facturas previas almacenadas en el archivo de texto y la base de datos.

A continuación, se presenta el fragmento de código que realiza el cálculo del tiempo de parqueo y genera la factura.

```
% Función para calcular el monto a pagar
function monto = calcular_monto(hora_entrada, hora_salida)
    tiempo_estadia = (hora_salida - hora_entrada) * 24; % Convertir a horas
    if tiempo_estadia <= 1
        monto = 15.00;
    else
        monto = 15.00 + 20.00 * (tiempo_estadia - 1);
    end
end
```

Figura 1: Código de calculo de monto de factura en octave.

```
# Validar y solicitar hora de entrada
while True:
    hora_entrada = input("Hora de entrada (HH:MM): ")
    if validar_hora(hora_entrada):
        break
    else:
        print("Formato de hora no válido. Use HH:MM.")

# Validar y solicitar hora de salida
while True:
    hora_salida = input("Hora de salida (HH:MM): ")
    if validar_hora(hora_salida):
        break
    else:
        print("Formato de hora no válido. Use HH:MM.")

# Calcular el monto a pagar
horas, total = calcular_monto(hora_entrada, hora_salida)

# Mostrar resumen
print("\nResumen de la transacción:")
print(f"Nombre del cliente: {nombre}")
print(f"NIT: {nit}")
print(f"Placa del vehículo: {placa}")
print(f"Hora de entrada: {hora_entrada}")
print(f"Hora de salida: {hora_salida}")
print(f"Tiempo en el parqueo: {horas} horas")
print(f"Monto total a pagar: Q{total:.2f}")

# Guardar en la base de datos PostgreSQL
guardar_factura(nombre, nit, placa, hora_entrada, hora_salida, horas, total)
```

Figura 2: Código de calculo de monto de factura en python.

El sistema también realiza la inserción de los datos en la base de datos PostgreSQL mediante el uso de la librería psycopg2. El siguiente fragmento muestra cómo se realiza la inserción de los datos:

II. UTILIDAD DEL CÓDIGO

Este programa es útil para:

- Monitorear el estado de salud relacionado con el peso corporal.
- Registrar datos históricos de peso y altura para analizar tendencias a lo largo del tiempo.
- Facilitar el cálculo del IMC de manera rápida y precisa, sin necesidad de realizar operaciones manuales.

III. FUNCIONAMIENTO DEL SISTEMA

El sistema permite al usuario ingresar los datos de entrada mediante un menú interactivo. Tras completar los datos necesarios, el sistema calcula el tiempo de parqueo en horas y minutos, genera una factura que se muestra en pantalla y se guarda tanto en un archivo de texto como en la base de datos. Al final de cada transacción, el programa genera una factura con los detalles del parqueo y el monto a pagar. La información de la factura también se almacena en un archivo de texto para futuras consultas.

* e-mail: 3243383091703@ingenieria.usac.edu.gt

El sistema desarrollado permite gestionar de manera eficiente el parqueo de vehículos, calculando automáticamente las tarifas y generando facturas. La integración con una base de datos PostgreSQL facilita el almacenamiento y consulta de los datos, mientras que el archivo de texto actúa como un respaldo adicional para las facturas generadas. Este sistema mejora la gestión de parqueos y la experiencia del usuario al automatizar procesos clave como el cálculo de tarifas y la generación de facturas. Al intentar realizar el mismo proceso en octave se complico, así que únicamente se dejó implementado en python

IV. CONCLUSIONES

- El proyecto logra optimizar la administración de los parqueos al automatizar tareas clave como el cálculo del tiempo de estancia, la tarifa correspondiente y la generación de facturas. Esto permite reducir errores manuales y mejorar la eficiencia en el control de ingresos y salidas de vehículos.
- El sistema combina el uso de Python y PostgreSQL, lo que permite tanto el almacenamiento seguro de

datos en una base de datos como la creación de respaldos en archivos de texto. Esta integración fortalece la confiabilidad del sistema y facilita la consulta de facturas previas.

- Aunque se planteó inicialmente desarrollar la solución en Octave y python en el archivo de texto "facturas.txt", la implementación final se consolidó únicamente en Octave, debido a complicaciones con el rastreo de ruta con python, pero se guardó correctamente en la base de datos de PostgreSQL. Esto refleja la importancia de seleccionar herramientas adecuadas al proyecto y la necesidad de adaptarse a las limitaciones técnicas encontradas.

V. REPOSITORIO EN GITHUB

El código fuente de este proyecto, junto con ejemplos y otros recursos, está disponible en el siguiente enlace: <https://github.com/Marianne8934/Tareas-y-proyectos> Este repositorio contiene el código original, las mejoras realizadas y las gráficas generadas por los programas descritos en este documento.