

Tarea 6: Tarea 6 Programa Cobro y Facturación de parqueo.

Proyectos de Computación Aplicados a la Ingeniería Electrónica.*

Cindy Melissa Gatica Arriola, 201709692¹, **

¹Facultad de Ingeniería, Universidad de San Carlos,
Edificio T1, Ciudad Universitaria, Zona 12, Guatemala.

Se realiza un programa el cual permite calcular el precio que un cliente debe pagar por el tiempo que permaneció en el parqueo. Para ello, se presenta un menú, en el cual el cliente debe seleccionar un número de las opciones mostradas. Entre las opciones encontramos: Ingresar datos de facturación, generación de la factura, historial de datos, borrar y salir del menu. En el ingreso de datos, el cliente debe ingresar datos como su nombre, nit, placa, hora de entrada, hora de salida y posteriormente, se calcula el monto a pagar.

I. OBJETIVOS

A. Generales

- Crear un programa que permita calcular el pago por la estadía en el estacionamiento y guardar sus datos de facturación.

B. Específicos

- * Calcular el monto total a pagar por las horas dentro del parqueo.
- * Guardar la información del servicio en la base de datos y en el archivo de texto.
- * Crear un archivo de texto, almacenar información y eliminar.
- * Implementar la lógica del programa en Python y Octave.
- * Permitir la sincronización con la base de datos PostgreSQL.

II. CÓDIGO EN OCTAVE

El código para calcular el IMC de la persona, así como la creación del archivo de texto:

```
1 % Función para mostrar el menú
2 function mostrar_menu()
3 disp("\n--- Menú de Estacionamiento ---");
4 disp("1. Ingresar datos del cliente");
5 disp("2. Imprimir Factura");
6 disp("3. Ver historial de datos");
7 disp("4. Borrar datos");
8 disp("5. Salir");
9 endfunction
10
11 % Función para guardar información en un archivo
12 function guardar_factura(factura)
13 ruta_archivo = "facturas.txt";
14 fid = fopen(ruta_archivo, "a");
15 if fid != -1
16 fprintf(fid, "%s\n", factura);
17 fclose(fid);
18 else
19 disp("Error al escribir en el archivo.");
20 end
21 endfunction
22
23 % Función para leer datos desde un archivo
24 function imprimir_factura()
25 ruta_archivo = "facturas.txt";
26 if exist(ruta_archivo, "file")
27 fid = fopen(ruta_archivo, "r");
28 while ~feof(fid)
29 linea = fgetl(fid);
30 disp(linea);
31 end
32 fclose(fid);
33 else
34 disp("No se han generado facturas aún.");
35 end
36 endfunction
37
38 % Función para borrar el archivo de facturas
39 function borrar_datos()
40 ruta_archivo = "facturas.txt";
41 if exist(ruta_archivo, "file")
42 unlink(ruta_archivo);
43 disp("Todos los datos han sido borrados.");
44 else
45 disp("El archivo no existe.");
46 end
47 endfunction
48
49 % Función para calcular el monto total
50 function monto = calcular_monto(hora_entrada,
51 hora_salida)
52 tiempo_estadia = etime(hora_salida,
53 hora_entrada) / 3600; % Convierte segundos a horas
54 if tiempo_estadia <= 1
55 monto = 15.00;
56 else
57 monto = 15.00 + (tiempo_estadia - 1) *
58 20.00;
59 end
60 endfunction
```

* Proyectos de Computación Aplicados a la Ingeniería Electrónica.

** e-mail: 2787947930101@ingenieria.usac.edu.gt

```

59 % Función para ingresar datos del cliente
60 function [nombre, nit, placa, hora_entrada,
61   hora_salida] = ingresar_datos_cliente()
62 nombre = input("Nombre del cliente: ", "s");
63 nit = input("NIT del cliente sin guiones: ", "s");
64 placa = input("Placa del vehículo en
65   Mayúsculas: ", "s");
66
67 disp("Introduce la hora en formato [YYYY MM DD
68   HH MM SS]");
69 hora_entrada = input("Hora de entrada: ");
70 hora_salida = input("Hora de salida: ");
71 endfunction
72
73 % Función principal
74 function ejecutar_programa()
75   while true
76     mostrar_menu();
77     opcion = input("\nSeleccione una opción: ",
78       "s");
79
80     switch opcion
81       case "1"
82         [nombre, nit, placa, hora_entrada,
83           hora_salida] = ingresar_datos_cliente();
84         monto = calcular_monto(hora_entrada,
85           hora_salida);
86         tiempo = etime(hora_salida, hora_entrada)
87           / 3600;
88         factura = sprintf("Nombre: %s, Placa: %s
89           , Tiempo: %.2f horas, Monto: Q%.2f", nombre,
90           placa, tiempo, monto);
91         disp(factura);
92         guardar_factura(factura);
93
94       case "2"
95         imprimir_factura();
96
97       case "3"
98         imprimir_factura(); % Usa el mismo
99         archivo como "historial"
100
101       case "4"
102         borrar_datos();
103
104       case "5"
105         disp("Gracias por utilizar el servicio
106           de estacionamiento.");
107         break;
108
109       otherwise
110         disp("Opción no válida. Intenta
111           nuevamente.");
112       end
113     end
114   endfunction
115
116 % Ejecutar el programa
117 clear;
118 ejecutar_programa();

```

Listing 1: IMC

```

>> imcOctave
Seleccione una opción:
1. Ingresar datos
2. Mostrar datos
3. Guardar
4. Leer
5. Borrar
6. Salir
Ingrrese su elección: 1
Ingrrese el nombre del usuario: Cindy
Ingrrese el peso del usuario en kilogramos: 45
Ingrrese la altura del usuario en metros: 1.60
Información guardada correctamente.

Seleccione una opción:
1. Ingresar datos
2. Mostrar datos
3. Guardar
4. Leer
5. Borrar
6. Salir
Ingrrese su elección: 2
El Índice de Masa Corporal de Cindy es de 17.58
Cindy se encuentra en la categoría: Bajo Peso

Seleccione una opción:
1. Ingresar datos
2. Mostrar datos
3. Guardar
4. Leer
5. Borrar
6. Salir
Ingrrese su elección: 3
;Documento creado y guardado correctamente!

Seleccione una opción:
1. Ingresar datos
2. Mostrar datos
3. Guardar
4. Leer
5. Borrar
6. Salir

```

Figura 1: Información desplegada en terminal

III. RESULTADOS DEL PROGRAMA

Los resultados en terminal del programa son los siguientes

```

1. Ingresar datos
2. Mostrar datos
3. Guardar
4. Leer
5. Borrar
6. Salir
Ingrese su elección: 2
El índice de Masa Corporal de Cindy es de 17.58
Cindy se encuentra en la categoría: Bajo Peso

Seleccione una opción:
1. Ingresar datos
2. Mostrar datos
3. Guardar
4. Leer
5. Borrar
6. Salir
Ingrese su elección: 3
¡Documento creado y guardado correctamente!

Seleccione una opción:
1. Ingresar datos
2. Mostrar datos
3. Guardar
4. Leer
5. Borrar
6. Salir
Ingrese su elección: 4
Leyendo el archivo...
-----
Nombre: Cindy
Peso: 45.00 [kg]
Altura: 1.60 [m]
Índice de Masa Corporal (IMC): 17.58
Categoría: Bajo Peso
-----
Cerrando archivo...

```

```

14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
try:
    opcion = int(input("Ingrese su opción: "))

    # Validar rango de opción
    if opcion < 1 or opcion > 5:
        print("Entrada inválida. Por favor, ingrese un número entre 1 y 6.\n")
        continue

    # Procesar las opciones
    if opcion == 1: # ingreso de datos
        try:
            ingresoDatos()
        except ValueError as e:
            print(f"Error: {e}")

    elif opcion == 2: #Mostrar datos
        try:
            generacion_factura()
        except ValueError as e:
            print(f"Error: {e}")

    elif opcion == 3: # Guardando el archivo
        try:
            except Exception as e:
                print(f"Error al guardar los datos: {e}")

    elif opcion == 4: #Visualizar datos del archivo txt
        try:
            print("Leyendo el archivo ...\\n")
            f = open("imcpy.txt", "r")
            print(f.read())
            f.close()

            except FileNotFoundError: #
                Manejo específico si el archivo no existe
                print("El archivo 'imcpy.txt' no existe. No hay datos para mostrar.")
            except Exception as e:
                print(f"Error al visualizar los datos: {e}")

    elif opcion == 5: # opción Borrar
        try:
            if os.path.exists("imcpy.txt"):
                os.remove("imcpy.txt")
                print("El archivo fue eliminado correctamente")
            else:
                print("El archivo no existe")

            except Exception as e:
                print(f"Error al visualizar los datos: {e}")

    elif opcion == 6: # opción Salir
        print(" Gracias por usar el programa!\n")

    except ValueError:

```

Figura 2: Información desplegada en terminal

IV. CÓDIGO EN PYTHON

Se reimplementó el código anterior para el procesamiento de señales en Python.

```

1 import os
2 import math
3
4 def menu():
5     opcion = 0
6     while opcion != 5:
7         print("\n Bienvenido al Menú Principal !")
8         print(" Seleccione una opción:")
9         print("1. Ingresar datos de facturación ")
"")
10        print("2. Generación de factura")
11        print("3. Historial de datos")
12        print("4. Borrado de datos")
13        print("5. Salir")

```

```

    print("Entrada inv lida. Por favor,
ingrese un n mero entero.\n")

#Funcion para la opcion 1
def ingresoDatos():
    nombre = input("Ingrese el nombre del
usuario: ").strip()
    if not nombre or nombre.isdigit():
        raise ValueError("El nombre no puede ser
un n mero o estar vac o.")

    nit_usuario = int(input("Ingrese el NIT del
sin guiones ni espacios: "))
    if nit_usuario <= 0:
        raise ValueError("El NIT no puede ser un
n mero negativo o estar vac o.")

    No_placa = input("Ingrese la placa del carro
: ").strip()
    if not No_placa or No_placa.isdigit():
        raise ValueError("Error en el ingreso de
la placa. Por favor, intente de nuevo.")

    hora_entrada = float(input("Ingrese la hora
de entrada al estacionamiento como Hora.
Minutos: "))
    if hora_entrada <= 0:
        raise ValueError("La hora de entrada no
puede ser un n mero negativo o estar vac o
.")

    hora_salida = float(input("Ingrese la hora
de salida del estacionamiento Hora.Minutos:
"))
    if hora_salida <= 0:
        raise ValueError("La hora de salida no
puede ser un n mero negativo o estar vac o
.")

    print("Informaci n guardada correctamente.\n")

    return nombre, nit_usuario, No_placa,
hora_entrada, hora_salida

#funcion para calcular el monto a pagar
def total_pago(hora_entrada, hora_salida):
    tiempo_estancia = hora_salida - hora_entrada

    # Redondear hacia arriba para contabilizar
    # fracciones de hora como hora completa
    horas_totales = math.ceil(tiempo_estancia)
    pago_total = 0
    # Calcular el pago
    if horas_totales == 1:
        # Primera hora tiene tarifa fija
        pago_total = 15.00
        return pago_total
    else:
        # Primera hora + horas adicionales
        # La primera hora cuesta Q15, las
        # siguientes Q20
        pago_total = 15.00 + (horas_totales - 1) *
        20.00
        return pago_total

def generacion_factura(nombre, nit_usuario,
No_placa, hora_entrada, hora_salida,
tiempo_estancia, pago_total):
    print("Resumen de la transacci n")
    print("Nombre: {} ".format(nombre))
    print("N mero de Placa: {} ".format(No_placa))
    print("Hora de entrada: {} ".format(hora_entrada))
    print("Hora de salida: {} ".format(hora_salida))
    print("Tiempo total en el parqueo: {} ".format(tiempo_estancia))
    print("Total a cancelar: {} ".format(pago_total))

    with open("factura.txt", "a") as f:
        f.write("\n")
        f.write("Nombre: {} \n".format(nombre))
        f.write("N mero de Placa: {} \n".format(No_placa))
        f.write("Hora de entrada: {} \n".format(hora_entrada))
        f.write("Hora de salida: {} \n".format(hora_salida))
        f.write("Tiempo total en el parqueo: {} \n".format(tiempo_estancia))
        f.write("Total a cancelar: {} \n".format(pago_total))
        f.write("\n")
    print("Documento creado y guardado
correctamente.\n")

```

Listing 2: Programa de cobro y facturación de parqueo

V. RESULTADOS DEL PROGRAMA

Los resultados del cálculo del índice corporal de masa reimplementado en Python son

```

PS C:\Users\Melissa A> & "C:/Users/Melissa A/AppData/Local/Pro
IE/Tarea 5 -Corto 1-/IMC_FranciscoCuxil/python/imc_franciscod
1. Calcular IMC y mostrar resultados
2. Leer informaci n del archivo
3. Borrar informaci n del archivo
4. Salir del programa
Ingres e la opci n deseada: 1
Ingres e su nombre: cindy
Ingres e su peso en kilogramos: 45
Ingres e su altura en metros: 1.60
IMC de cindy: 17.578124999999996
Categor a: Bajo peso
1. Calcular IMC y mostrar resultados
2. Leer informaci n del archivo
3. Borrar informaci n del archivo
4. Salir del programa
Ingres e la opci n deseada: 2
Nombre: cindy, IMC: 17.578124999999996, Categor a: Bajo peso
1. Calcular IMC y mostrar resultados
2. Leer informaci n del archivo
3. Borrar informaci n del archivo
4. Salir del programa
Ingres e la opci n deseada: 3
Informaci n borrada correctamente.
1. Calcular IMC y mostrar resultados
2. Leer informaci n del archivo
3. Borrar informaci n del archivo
4. Salir del programa
Ingres e la opci n deseada: 4
¡Gracias por usar el programa!
PS C:\Users\Melissa A>

```

Figura 3: Resultado en terminal del programa

Listing 2: Programa de cobro y facturacion de parqueo

V. RESULTADOS DEL PROGRAMA

Los resultados del calculo del índice corporal de masa reimplementado en Python son

```
PS C:\Users\Melissa A> & "C:/Users/Melissa A/AppData/Local/Pro  
IE/Tarea 5 -Corto 1-/IMC_FranciscoCuxil/python/imc_franciscod  
1. Calcular IMC y mostrar resultados  
2. Leer información del archivo  
3. Borrar información del archivo  
4. Salir del programa  
Ingrese la opción deseada: 1  
Ingrese su nombre: cindy  
Ingrese su peso en kilogramos: 45  
Ingrese su altura en metros: 1.60  
IMC de cindy: 17.57812499999996  
Categoría: Bajo peso  
1. Calcular IMC y mostrar resultados  
2. Leer información del archivo  
3. Borrar información del archivo  
4. Salir del programa  
Ingrese la opción deseada: 2  
Nombre: cindy, IMC: 17.57812499999996, Categoría: Bajo peso  
  
1. Calcular IMC y mostrar resultados  
2. Leer información del archivo  
3. Borrar información del archivo  
4. Salir del programa  
Ingrese la opción deseada: 3  
Información borrada correctamente.  
1. Calcular IMC y mostrar resultados  
2. Leer información del archivo  
3. Borrar información del archivo  
4. Salir del programa  
Ingrese la opción deseada: 4  
¡Gracias por usar el programa!  
PS C:\Users\Melissa A>
```

Figura 3: Resultado en terminal del programa

VI. DIAGRAMA DE FLUJO

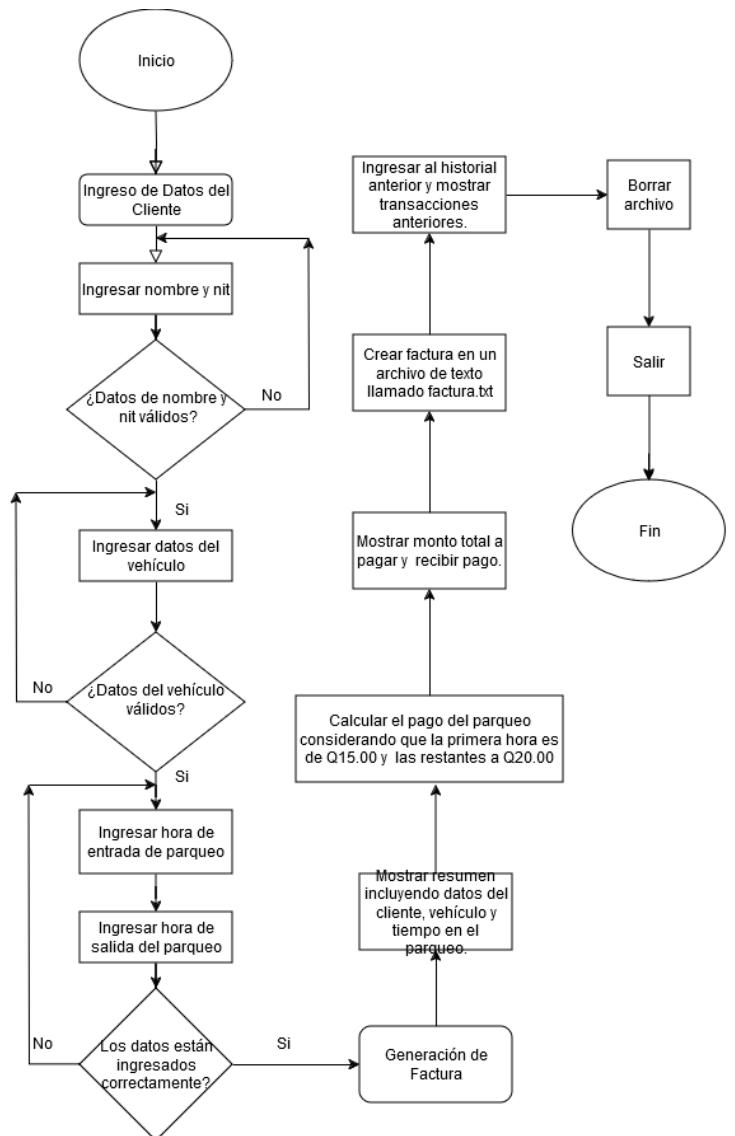


Figura 4: Diagrama de flujo y cobro

VII. LINK DE GITHUB

Se ha subido un documento a GitHub en el repositorio privado Lenguajes aplicados IE. Presionar aqui o bien seguir el siguiente link <https://github.com/CINDYGAT/Proyectos-aplicados-a-IE/tree/91e31a0065a6427d75309ebdcf8159af3e30e874/Tarea%20-%20corto%20-%20->