

# Tarea 6: Tarea 6 Programa Cobro y Facturación de parqueo. Proyectos de Computación Aplicados a la Ingeniería Electrónica.\*

Cindy Melissa Gatica Arriola, 201709692<sup>1</sup>,\*\*

<sup>1</sup>Facultad 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 numero de las opciones mostradas. Entre las opciones encontramos: Ingresar datos de facturación, generacion 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 % Funci n para mostrar el men
2 function mostrar_menu()
3     disp("\n--- Men 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 % Funci n para guardar informaci n 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 % Funci n 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 a n.");
35     end
36 endfunction
37
38 % Funci n 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 % Funci n para calcular el monto total
50 function monto = calcular_monto(hora_entrada,
    hora_salida)
51     tiempo_estadia = etime(hora_salida,
    hora_entrada) / 3600; % Convierte segundos a
    horas
52     if tiempo_estadia <= 1
53         monto = 15.00;
54     else
55         monto = 15.00 + (tiempo_estadia - 1) *
    20.00;
56     end
57 endfunction
58
```

\* Proyectos de Computación Aplicados a la Ingeniería Electrónica.  
\*\* e-mail: 2787947930101@ingenieria.usac.edu.gt

```

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

```

Listing 1: IMC

```

>> imcOctave
Seleccione una opcion:
1. Ingresar datos
2. Mostrar datos
3. Guardar
4. Leer
5. Borrar
6. Salir
Ingrese su eleccion: 1
Ingrese el nombre del usuario: Cindy
Ingrese el peso del usuario en kilogramos: 45
Ingrese la altura del usuario en metros: 1.60
Informaci3n guardada correctamente.

Seleccione una opcion:
1. Ingresar datos
2. Mostrar datos
3. Guardar
4. Leer
5. Borrar
6. Salir
Ingrese su eleccion: 2
El 3ndice de Masa Corporal de Cindy es de 17.58
Cindy se encuentra en la categor3a: Bajo Peso

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

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

```

Figura 1: Informaci3n 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 eleccion: 2
El índice de Masa Corporal de Cindy es de 17.58
Cindy se encuentra en la categoría: Bajo Peso

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

Seleccione una opcion:
1. Ingresar datos
2. Mostrar datos
3. Guardar
4. Leer
5. Borrar
6. Salir
Ingrese su eleccion: 4
Leyendo el archivo...
-----

Nombre: Cindy

Peso: 45.00 [kg]

Altura: 1.60 [m]

Indice de Masa Corporal (IMC): 17.58

Categoría: Bajo Peso

-----

Cerrando archivo...

```

Figura 2: Información desplegada en terminal

#### IV. CÓDIGO EN PYTHON

Se reemplazó 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 Men  Principal
8         !")
9         print("Seleccione una opci n:")
10        print("1. Ingresar datos de facturaci n
11        ")
12        print("2. Generaci n de factura")
13        print("3. Historial de datos")
14        print("4. Borrado de datos")
15        print("5. Salir")

```

```

14
15
16     try:
17         opcion = int(input("Ingrese su
18         opci n: "))
19
20         # Validar rango de opci n
21         if opcion < 1 or opcion > 5:
22             print("Entrada inv lida. Por
23             favor, ingrese un n mero entre 1 y 6.\n")
24             continue
25
26         # Procesar las opciones
27         if opcion == 1: # ingreso de datos
28             try:
29                 ingresoDatos()
30             except ValueError as e:
31                 print(f"Error: {e}")
32
33         elif opcion == 2: #Mostrar datos
34             try:
35                 generacion_factura()
36             except ValueError as e:
37                 print(f"Error: {e}")
38
39         elif opcion == 3: # Guardando el
40         archivo
41             try:
42                 except Exception as e:
43                     print(f"Error al guardar los
44                     datos: {e}")
45
46         elif opcion == 4: #Visualizar datos
47         del archivo txt
48             try:
49                 print("Leyendo el archivo
50                 ...\n")
51
52                 f = open("imcpy.txt", "r")
53                 print(f.read())
54                 f.close()
55
56             except FileNotFoundError: #
57             Manejo espec fico si el archivo no existe
58                 print("El archivo 'imcpy.txt
59                 ' no existe. No hay datos para mostrar.")
60             except Exception as e:
61                 print(f"Error al visualizar
62                 los datos: {e}")
63
64         elif opcion == 5: # opcion Borrar
65             try:
66                 if os.path.exists("imcpy.txt
67                 "):
68                     os.remove("imcpy.txt")
69                     print("El archivo fue
70                     eliminado correctamente")
71                 else:
72                     print("El archivo no
73                     existe")
74
75             except Exception as e:
76                 print(f"Error al visualizar
77                 los datos: {e}")
78
79         elif opcion == 6: # opcion Salir
80             print(" Gracias por usar el
81             programa!\n")
82
83         except ValueError:

```

```

69         print("Entrada inv lida. Por favor,
           ingrese un n mero entero.\n")
70
71 #Funcion para la opcion 1
72 def ingresoDatos():
73     nombre = input("Ingrese el nombre del
74     usuario: ").strip()
75     if not nombre or nombre.isdigit():
76         raise ValueError("El nombre no puede ser
77         un n mero o estar vac o.")
78
79     nit_usuario = int(input("Ingrese el NIT del
80     sin guiones ni espacios: "))
81     if nit_usuario <= 0:
82         raise ValueError("El NIT no puede ser un
83         n mero negativo o estar vac o.")
84
85     No_placa = input("Ingrese la placa del carro
86     : ").strip()
87     if not No_placa or No_placa.isdigit():
88         raise ValueError("Error en el ingreso de
89         la placa. Por favor, intente de nuevo.")
90
91     hora_entrada = float(input("Ingrese la hora
92     de entrada al estacionamiento como Hora.
93     Minutos: "))
94     if hora_entrada <= 0:
95         raise ValueError("La hora de entrada no
96         puede ser un n mero negativo o estar vac o
97         .")
98
99     hora_salida = float(input("Ingrese la hora
100     de salida del estacionamiento Hora.Minutos:
101     "))
102     if hora_salida <= 0:
103         raise ValueError("La hora de salida no
104         puede ser un n mero negativo o estar vac o
105         .")
106
107     print("Informaci n guardada correctamente.\n")
108
109     return nombre, nit_usuario, No_placa,
110     hora_entrada, hora_salida
111
112 #funcion para calcular el monto a pagar
113 def total_pago(hora_entrada, hora_salida):
114     tiempo_estancia = hora_salida - hora_entrada
115
116     # Redondear hacia arriba para contabilizar
117     fracciones de hora como hora completa
118     horas_totales = math.ceil(tiempo_estancia)
119     pago_total = 0
120     # Calcular el pago
121     if horas_totales == 1:
122         # Primera hora tiene tarifa fija
123         pago_total = 15.00
124         return pago_total
125     else:
126         # Primera hora + horas adicionales
127         # La primera hora cuesta Q15, las
128         siguientes Q20
129         pago_total = 15.00 + (horas_totales - 1)
130         * 20.00
131         return pago_total
132
133 def generacion_factura(nombre, nit_usuario,
134     No_placa, hora_entrada, hora_salida,
135     tiempo_estancia, pago_total):
136     print("Resumen de la transacci n")

```

```

117     with open("factura.txt", "a") as f:
118         f.write("
119         -----\n")
120         f.write(f"Nombre: {nombre}\n")
121         f.write(f"N mero de Placa: {No_placa}\n")
122         f.write(f"")
123         f.write(f"Hora de entrada: {hora_entrada
124         } [h.m]\n")
125         f.write(f"Hora de salida: {hora_salida}
126         [h.m]\n")
127         f.write(f"Tiempo total en el parqueo: {
128         tiempo_estancia}\n")
129         f.write(f"Total a cancelar: {pago_total
130         }\n")
131         f.write("
132         -----\n")
133     print("Documento creado y guardado
134     correctamente.\n")

```

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_francisco
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: 17.578124999999996
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: 17.578124999999996, 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

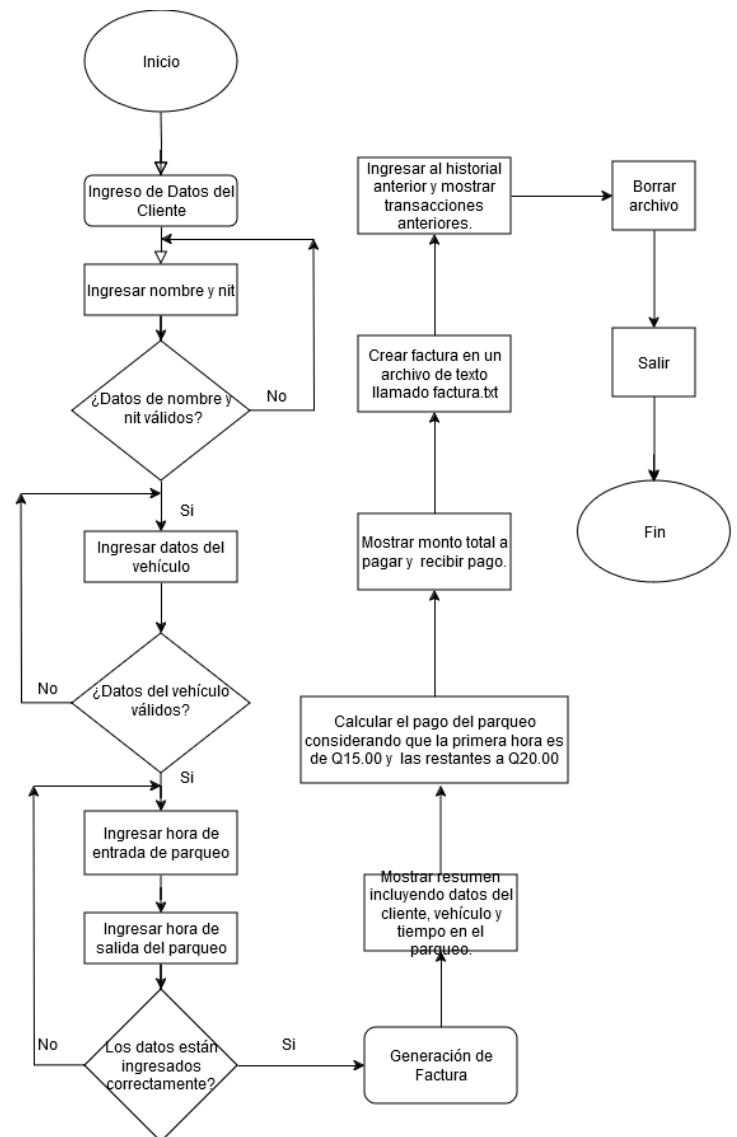


Figura 4: Diagrama de flujo y cobro

## VII. LINK DE GITHUB

Se ha subido un documento a GitHub en el repositorio privado Lenguajes aplicados IE. Presionar aquí o bien seguir el siguiente link <https://github.com/CINDYGAT/Proyectos-aplicados-a-IE/tree/91e31a0065a6427d75309ebdcf8159af3e30e874/Tarea%206%20-%20corto%202%20->