

INFORME DE LABORATORIO

INFORMACIÓN BÁSICA					
ASIGNATURA:	Programación Web 2				
TÍTULO DE LA PRÁCTICA:	Python				
NÚMERO DE PRÁCTICA:	4	AÑO LECTIVO:	2024	NRO. SEMESTRE:	4
FECHA DE PRESENTACIÓN	11/10/2024	HORA DE PRESENTACIÓN	11/59/59		
Hecho por: - Mathias Dario Davila Flores				NOTA (0-20)	
DOCENTE(s): - Lino Pinto Oppe					

RESULTADOS Y PRUEBAS
<p>I. PREGUNTAS PLANTEADAS:</p> <p>1. <u>Novedades en el código expuesto:</u></p> <ul style="list-style-type: none">• Uso de do-while: En Python, no existe una estructura do-while como tal, pero puedes simular su comportamiento utilizando un while normal con una lógica que garantice que el bloque de código se ejecute al menos una vez, para realizar esto se hace uso de un TRUE que en mi caso del código se evidencia al darle ese valor TRUE a una variable, para la modificación correspondiente.• función upper() La función upper() en Python se utiliza para convertir un string en mayúsculas. En el caso de la validación del género (genero_cliente), se permite al usuario ingresar los chars de 'M', 'm', 'F' o 'f', y se normaliza la entrada a mayúsculas para simplificar la comparación• print f El uso de la función print con formato en Python se realiza usando la sintaxis f-string, lo que hace posible la inserción de variables dentro de las cadenas con el uso de llaves { } permitiendo que el código sea más legible

- uso de .2f

El modificador .2f dentro de los f-strings se utiliza para formatear los números en un formato de punto flotante con exactamente 2 decimales. Resultando útil a la hora de trabajar con valores monetarios. En el caso del ejercicio , se utiliza para mostrar el importe bruto, el descuento, el importe neto, etc con dos decimales:

II. Desarrollo del código

Variables empleadas para la salida de datos en la opción 1 y 2 así como la creación de la variable de verificar_menu para el manejo del bucle general

```
ejercicio.py 7 ...
1  verificar_menu = True
2  genero_m = 0 # Cantidad de clientes masculinos
3  importe_bruto = 0
4  importe_netto = 0
5  monto_descuento = 0
6  tipo_cliente_1 = 0 # Contador de clientes de tipo 1
7  acumulado_importe_netto_tipo_1 = 0
8  acumulado_importe_ventas = 0 # Total de ventas
9  ventas_femeninas = 0 # Ventas de clientes femeninos con importes entre 140 y 1000
10 ventas_rango = 0 # Ventas con importes entre 70 y 500
11 tipo_cliente = 0
12
```

Do-while general para la creacion del menu, asi como el manejo de entradas no disponibles

```
13 while verificar_menu:
14     print("\nMENU DE OPCIONES\n[1] REGISTRAR VENTA DE PASAJE\n[2] REPORTAR VENTAS\n[3] SALIR")
15     opcion = int(input("Seleccione una opción: "))
16
17     if opcion == 1:
```

```
elif opcion == 2:
```

```
elif opcion == 3:
    verificar_menu = False
```

```
else:
    print("ERROR: Opción no válida. Intente de nuevo.")
```

Do-while para validación de entradas de datos del cliente, mostrando mensaje de error en cada caso que se ingrese un dato no disponible y volviendo a hacer la pregunta pero mostrando el mensaje de error

```
while True:
    # Validar tipo de cliente
    tipo_cliente = int(input("Ingrese tipo de cliente (1 o 2): "))
    while tipo_cliente != 1 and tipo_cliente != 2:
        print("Error. Ingrese una opcion valida")
        tipo_cliente = int(input("Ingrese tipo de cliente (1 o 2): "))

    # Validar cantidad de pasajes
    numero_pasajes = int(input("Ingrese cantidad de pasajes: "))
    while numero_pasajes <= 0:
        print("Error. Ingrese una opcion valida")
        numero_pasajes = int(input("Ingrese cantidad de pasajes: "))

    # Validar género del cliente
    genero_cliente = input("Ingrese género del cliente (M/F): ").upper()
    while genero_cliente != 'M' and genero_cliente != 'F':
        print("Error. Ingrese una opcion valida")
        genero_cliente = input("Ingrese género del cliente (M/F): ").upper()

    # Validar tipo de servicio
    tipo_servicio = int(input("Ingrese tipo de servicio (1: Económico, 2: Ejecutivo, 3: Primera Clase): "))
    while tipo_servicio != 1 and tipo_servicio != 2 and tipo_servicio != 3:
        print("Error. Ingrese una opcion valida")
        tipo_servicio = int(input("Ingrese tipo de servicio (1: Económico, 2: Ejecutivo, 3: Primera Clase): "))
```

Precios y descuentos según la entrada de datos del cliente

```
# Precios
if tipo_servicio == 1:
    precio = 70.00
elif tipo_servicio == 2:
    precio = 140.00
elif tipo_servicio == 3:
    precio = 280.00

# Descuentos segun el numero de pasajes
if numero_pasajes == 1:
    desc = 0.00
elif 2 <= numero_pasajes <= 5:
    desc = 0.05
elif 6 <= numero_pasajes <= 10:
    desc = 0.12
else:
    desc = 0.15
```

Calcular importes

```
# Cálculos de importes
importe_bruto = numero_pasajes * precio
monto_descuento = importe_bruto * desc
importe_netto = importe_bruto - monto_descuento
```

Según los datos ingresados por el cliente, si cumplen con ciertas condiciones se agrega los valores respectivos a las variables creadas al inicio, para su posterior impresión en la opción 2

```
# Actualización de datos
acumulado_importe_ventas += importe_netto

if genero_cliente == 'M':
    genero_m += 1
if genero_cliente == 'F' and 140 <= importe_netto <= 1000:
    ventas_femeninas += 1
if 70 <= importe_netto <= 500:
    ventas_rango += 1
if tipo_cliente == 1:
    tipo_cliente_1 += 1
    acumulado_importe_netto_tipo_1 += importe_netto
```

Impresión de datos, por cliente respecto a su datos ingresados anteriormente y creación de otro menú con solo dos opciones

```
# Mostrar resultados de la venta
print("\nResumen de la venta:")
print(f"Importe Bruto: ${importe_bruto:.2f}")
print(f"Descuento aplicado: ${monto_descuento:.2f}")
print(f"Importe Neto: ${importe_netto:.2f}\n")

# Menu para ver si desea realizar otro movimiento o salir de la opcion 1 (registrar pasaje de venta)
print("\nMENU DE OPCIONES\n[1] REGISTRAR OTRA VENTA\n[2] SALIR")
otra_opcion = int(input("Seleccione una opción: "))
if otra_opcion == 2:
    break # Salir al menú principal
```

Opción 2, manejo de todas las salidas de datos con el uso de f-string para un mejor manejo así mismo , manejar la última impresión debido a la división entre cero

```
elif opcion == 2:
    # Mostrar reporte de ventas
    print("\nReporte de Ventas:")
    print(f"Cantidad de clientes de género masculino: {genero_m}")
    print(f"Cantidad de ventas con Importe Neto entre $70 y $500: {ventas_rango}")
    print(f"Cantidad de ventas de clientes femeninos con Importe Neto entre $140 y $1000: {ventas_femeninas}")
    print(f"Acumulado total de Importe de Ventas: ${acumulado_importe_ventas:.2f}")
    print(f"Acumulado de Importe Neto de clientes de tipo 1: ${acumulado_importe_netos_tipo_1:.2f}")

    # Calcular promedio de Importe Neto de clientes tipo 1
    if tipo_cliente_1 > 0:
        promedio_importe_netos_tipo_1 = acumulado_importe_netos_tipo_1 / tipo_cliente_1
    else:
        promedio_importe_netos_tipo_1 = 0
    print(f"Promedio de Importe Neto de clientes de tipo 1: ${promedio_importe_netos_tipo_1:.2f}")
```

Opción 3, cambio de la variable que se usa para el bucle general para acabar con el ciclo true y en caso se elija en el menú principal una opción diferente a las propuestas se manda ese mensaje de error y de nuevo el menú principal

```
elif opcion == 3:
    verificar_menu = False

else:
    print("ERROR: Opción no válida. Intente de nuevo.")
```

EJECUCIÓN DEL CÓDIGO:

- MENU

```
MENU DE OPCIONES
[1] REGISTRAR VENTA DE PASAJE
[2] REPORTAR VENTAS
[3] SALIR
Seleccione una opción:
```

- OPCIÓN 1

MENU DE OPCIONES

[1] REGISTRAR VENTA DE PASAJE

[2] REPORTAR VENTAS

[3] SALIR

Seleccione una opción: 1

Ingrese tipo de cliente (1 o 2): 1

Ingrese cantidad de pasajes: 5

Ingrese género del cliente (M/F): f

Ingrese tipo de servicio (1: Económico, 2: Ejecutivo, 3: Primera Clase): 2

Resumen de la venta:

Importe Bruto: \$700.00

Descuento aplicado: \$35.00

Importe Neto: \$665.00

MENU DE OPCIONES

[1] REGISTRAR OTRA VENTA

[2] SALIR

Seleccione una opción:

- OPCIÓN 2

MENU DE OPCIONES

[1] REGISTRAR VENTA DE PASAJE

[2] REPORTAR VENTAS

[3] SALIR

Seleccione una opción: 2

Reporte de Ventas:

Cantidad de clientes de género masculino: 0

Cantidad de ventas con Importe Neto entre \$70 y \$500: 0

Cantidad de ventas de clientes femeninos con Importe Neto entre \$140 y \$1000: 1

Acumulado total de Importe de Ventas: \$665.00

Acumulado de Importe Neto de clientes de tipo 1: \$665.00

Promedio de Importe Neto de clientes de tipo 1: \$665.00

- OPCION 3

```
MENU DE OPCIONES
```

```
[1] REGISTRAR VENTA DE PASAJE
```

```
[2] REPORTAR VENTAS
```


```
[3] SALIR
```

```
Seleccione una opción: 3
```

```
PS C:\Users\AORUS\OneDrive\Documentos\UNSA 2023\2024-B\PWEB2\labs aca resolver\lab 4>
```


COMMITTS SUBIDOS:



arreglando todo el codigo (espacios, etc)

 mathiasddf committed 39 minutes ago


45282cf  

contenido de la opcion 2 (impresion de datos asi mismo un pequeño manejo sobre la ultima impresion respecto a la division entre cero)

 mathiasddf committed 40 minutes ago


fce1202  

contenido para el manejo de la pcion 1 (uso de do-while, validaciones de entradas de datos, calculos, actu de datos para la opcion 2, resultados de la venta y menu de dos opciones

 mathiasddf committed 42 minutes ago


9ff83af  


Creacion del bucle general para el manejo del menu y la creacion del menu

 mathiasddf committed 44 minutes ago


9402da2  


creacion de variables para el manejo de datos numericos para la opcion 1 y 2 del menu asi como la creacion de la variable que verifique el menu para manejar el bucle general

 mathiasddf committed 46 minutes ago

f95ef68  

creacion de carpeta y documento para el ejercicio

 mathiasddf committed 49 minutes ago

56903e6  

REFERENCIAS Y BIBLIOGRAFÍA

Colocare las referencias utilizadas para el desarrollo de la práctica en formato IEEE

Tabla : Rúbrica para contenido del Informe y evidencias

Contenido y demostración		Puntos	Checklist	Estudiante	Profesor
1. GitHub	Repositorio se pudo clonar y se evidencia la estructura adecuada para revisar los entregables. (Se descontará puntos por error u observación)	4	X	3	
2. Commits	Hay porciones de código fuente asociado a los commits planificados con explicaciones detalladas. (El profesor puede preguntar para refrendar calificación).	4	X	3	
3. Ejecución	Se incluyen comandos para ejecuciones y pruebas del código fuente explicadas gradualmente que permitirían replicar el proyecto. (Se descontará puntos por cada omisión)	4	X	3	
4. Pregunta	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación).	2	X	1	
7. Ortografía	El documento no muestra errores ortográficos. (Se descontará puntos por error encontrado)	2	X	2	
8. Madurez	El Informe muestra de manera general una evolución de la madurez del código fuente con explicaciones puntuales pero precisas, agregando diagramas generados a partir del código fuente y refleja un acabado impecable. (El profesor puede preguntar para refrendar calificación).	4	X	2	
Total		20		14	