

	<p>UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p>Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 1</p>

## 4 INFORME DE LABORATORIO

(formato estudiante)

INFORMACIÓN BÁSICA					
ASIGNATURA:	PROGRAMACIÓN DE WEB 2				
TÍTULO DE LA PRÁCTICA:	JAVASCRIPT				
NÚMERO DE PRÁCTICA:	4	AÑO LECTIVO:	2024-B	NRO. SEMESTRE:	Número de semestre
FECHA DE PRESENTACIÓN	12/10/2024	HORA DE PRESENTACIÓN			
INTEGRANTE (s) GIOVANI MENDOZA CONTRERAS				NOTA (0-20)	Nota colocada por el docente
DOCENTE(s): LINO JOSE PINTO OPPE					

RESULTADOS Y PRUEBAS
<p><b>I. EJERCICIOS RESUELTOS:</b></p> <p>Primero la forma en que realice el laboratorio fue dividir en clases la boleta y el registro para una mejor forma de manejo de los datos a la hora de querer mostrar clientes femeninos o masculinos, el tipo de pasaje si es Económico, Ejecutivo o primera clase de esta manera se hace más sencillo poder manejar los datos bajo mi punto de vista.</p> <ol style="list-style-type: none"> <li>boleta.py en esta clase estan los principales atributos de boleta estan tipo de cliente , cantidad de pasajes ,género del cliente tipo de servicio además se calcula el importe que se realiza por cada compra de boletas, también se muestra las boletas el importe bruto , monto de descuento y importe neto.</li> </ol>

```
boleta.py x registro_ventas.py main.py
boleta.py > ...
2 C:\Users\giovani mendoza\Desktop\lab04\boleta.py
3 def __init__(self, tipo_cliente, cantidad_pasajes, genero_cliente, tipo_servicio):
6     self.genero_cliente = genero_cliente
7     self.tipo_servicio = tipo_servicio
8     self.importe_bruto = 0
9     self.monto_descuento = 0
10    self.importe_netto = 0
11    self.calcular_importes()
12
13    def calcular_importes(self):
14        precios = {1: 70, 2: 140, 3: 280}
15        descuento = 0
16        precio = precios.get(self.tipo_servicio, 0)
17
18        if self.cantidad_pasajes == 1:
19            descuento = 0
20        elif 2 <= self.cantidad_pasajes <= 5:
21            descuento = 0.05
22        elif 6 <= self.cantidad_pasajes <= 10:
23            descuento = 0.12
24        elif self.cantidad_pasajes >= 11:
25            descuento = 0.15
26
27        self.importe_bruto = self.cantidad_pasajes * precio
28        self.monto_descuento = self.importe_bruto * descuento
29        self.importe_netto = self.importe_bruto - self.monto_descuento
30
```

```
2 class Boleta:
13     def calcular_importes(self):
27         self.importe_bruto = self.cantidad_pasajes * precio
28         self.monto_descuento = self.importe_bruto * descuento
29         self.importe_netto = self.importe_bruto - self.monto_descuento
30
31     def mostrar_boleta(self):
32         print(f"Importe Bruto: ${self.importe_bruto:.2f}")
33         print(f"Monto Descuento: ${self.monto_descuento:.2f}")
34         print(f"Importe Neto: ${self.importe_netto:.2f}")
35
36     def get_tipo_cliente(self):
37         return self.tipo_cliente
38
39     def get_genero_cliente(self):
40         return self.genero_cliente
41
42     def get_importe_netto(self):
43         return self.importe_netto
44
```

2.registro.py aqui se esta creando una lista donde se almacenará los datos de boleta y contadores para los elementos que nos pide mostrar el ejercicio

```
registro_ventas.py > ...
1 class RegistroVentas:
2     def __init__(self):
3         self.boletas = []
4         self.total_clientes_masculinos = 0
5         self.total_ventas_entre_70_y_500 = 0
6         self.total_ventas_femeninas_entre_140_y_1000 = 0
7         self.acumulado_ventas = 0
8         self.acumulado_importe_netto_tipo_1 = 0
9         self.contador_clientes_tipo_1 = 0
10
11     def registrar_boleta(self, boleta):
12         self.boletas.append(boleta)
13         self.acumulado_ventas += boleta.get_importe_netto()
14         if boleta.get_tipo_cliente() == 1:
15             self.acumulado_importe_netto_tipo_1 += boleta.get_importe_netto()
16             self.contador_clientes_tipo_1 += 1
17         if boleta.get_genero_cliente().lower() == 'm':
18             self.total_clientes_masculinos += 1
19         if 70 <= boleta.get_importe_netto() <= 500:
20             self.total_ventas_entre_70_y_500 += 1
21         if boleta.get_genero_cliente().lower() == 'f' and 140 <= boleta.get_importe_netto() <= 1000:
22             self.total_ventas_femeninas_entre_140_y_1000 += 1
23
24     def mostrar_resultados(self):
25         print("Resultados:")
26         print(f"Cantidad de clientes de género masculino: {self.total_clientes_masculinos}")
27         print(f"Cantidad de ventas con Importe Neto >= 70 y <= 500: {self.total_ventas_entre_70_y_500}")
28         print(f"Cantidad de ventas de clientes femeninos con Importe Neto >= 140 y <= 1000: {self.total_ventas_femeninas_entre_140_y_1000}")
29         print(f"Acumulado del Importe de Ventas: ${self.acumulado_ventas:.2f}")
30         print(f"Acumulado del Importe Neto de clientes tipo 1: ${self.acumulado_importe_netto_tipo_1:.2f}")
31         if self.contador_clientes_tipo_1 > 0:
32             promedio_importe_netto_tipo_1 = self.acumulado_importe_netto_tipo_1 / self.contador_clientes_tipo_1
33             print(f"Promedio del Importe Neto de clientes tipo 1: ${promedio_importe_netto_tipo_1:.2f}")
34         else:
35             print("No se ingresaron datos de clientes tipo 1.")
```

3.main.py aqui se realiza la validacion del ingreso de los datos si son ingresados incorrectamente se le comunicará al usuario que ingrese correctamente los datos si es correcto se le dirá que ingresó correctamente en la opción 2 que nos pide que mostremos los resultados también se mostrará

```

boleta.py  registro_ventas.py  main.py  x
main.py > ...
1  from boleta import Boleta
2  from registro_ventas import RegistroVentas
3
4  # Función principal
5  def empresa_transporte():
6      registro_ventas = RegistroVentas()
7      while True:
8          print("Seleccione una opción: ")
9          print("1. Registrar venta de pasajes")
10         print("2. Mostrar resultados")
11         print("3. Salir")
12         opcion = int(input())
13
14         if opcion == 1:
15             # Ingreso y validación de datos
16             tipo_cliente = int(input("Ingrese el tipo de cliente (1 o 2): "))
17             while tipo_cliente not in [1, 2]:
18                 tipo_cliente = int(input("Tipo de cliente inválido. Ingrese el tipo de cliente (1 o 2): "))
19
20             cantidad_pasajes = int(input("Ingrese la cantidad de pasajes: "))
21             while cantidad_pasajes <= 0:
22                 cantidad_pasajes = int(input("Cantidad de pasajes inválida. Ingrese la cantidad de pasajes: "))
23
24             genero_cliente = input("Ingrese el género del cliente (M/m/F/f): ").lower()
25             while genero_cliente not in ['m', 'f']:
26                 genero_cliente = input("Género inválido. Ingrese el género del cliente (M/m/F/f): ").lower()
27
28
29         tipo_servicio = int(input("Ingrese el tipo de servicio (1-Económica, 2-Ejecutiva, 3-Primera clase): "))
30         while tipo_servicio not in [1, 2, 3]:
31             tipo_servicio = int(input("Tipo de servicio inválido. Ingrese el tipo de servicio (1-Económica, 2-Ejecutiva, 3-Primera clase): "))
32
33         # Crear boleta y añadir al registro
34         boleta = Boleta(tipo_cliente, cantidad_pasajes, genero_cliente, tipo_servicio)
35         registro_ventas.registrar_boleta(boleta)
36
37         # Mostrar la boleta del cliente
38         boleta.mostrar_boleta()
39
40     elif opcion == 2:
41         # Mostrar resultados
42         registro_ventas.mostrar_resultados()
43
44     elif opcion == 3:
45         print("Saliendo del sistema...")
46         break
47
48     else:
49         print("Opción no válida. Intente de nuevo.")
50
51 # Ejecutar el programa
52 if __name__ == "__main__":
53     empresa_transporte()
54

```

	<p style="text-align: center;">UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p style="text-align: center;"><b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 5</p>

## II. PRUEBAS

Los resultados al realizar el ingreso de los datos son los siguientes: por ejemplo si colocamos mal los datos no avanzara el programa hasta que se ingresen los datos correctos como se puede ver en la imagen solo acepta los datos que restringimos como el género el tipo de servicio etc además estamos mostrando el importe bruto el monto de descuento y el importe neto.

```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS

3. Salir
1
Ingrese el tipo de cliente (1 o 2): 6
Tipo de cliente inválido. Ingrese el tipo de cliente (1 o 2): 1
Ingrese la cantidad de pasajes: -5
Cantidad de pasajes inválida. Ingrese la cantidad de pasajes: 5
Ingrese el género del cliente (M/m/F/f): 1
Género inválido. Ingrese el género del cliente (M/m/F/f): m
Ingrese el tipo de servicio (1-Económica, 2-Ejecutiva, 3-Primera clase): 6
Tipo de servicio inválido. Ingrese el tipo de servicio (1-Económica, 2-Ejecutiva, 3-Primera clase): 3
Importe Bruto: $1400.00
Monto Descuento: $70.00
Importe Neto: $1330.00
Seleccione una opción:
1. Registrar venta de pasajes
2. Mostrar resultados
3. Salir

```

EL punto 2 nos pide que mostremos algunos datos la cantidad de clientes masculinos el importe que se encuentra entre el rango de  $\geq 70$  y  $\leq 500$  y así con todos los datos además si ponemos 3 se sale del sistema de boletas y así culminando la ejecución

```

Resultados:
Cantidad de clientes de género masculino: 2
Cantidad de ventas con Importe Neto  $\geq 70$  y  $\leq 500$ : 1
Cantidad de ventas de clientes femeninos con Importe Neto  $\geq 140$  y  $\leq 1000$ : 0
Acumulado del Importe de Ventas: $2926.00
Acumulado del Importe Neto de clientes tipo 1: $1596.00
Promedio del Importe Neto de clientes tipo 1: $798.00
Seleccione una opción:
1. Registrar venta de pasajes
2. Mostrar resultados
3. Salir
3
Saliendo del sistema...
PS C:\Users\giovani mendoza\Desktop\lab04>

```

### III. CUESTIONARIO:

CONCLUSIONES
<p>El uso de clases en Python es esencial para estructurar programas de manera organizada, eficiente y escalable. Las clases permiten agrupar datos (atributos) y comportamientos (métodos) relacionados en una única entidad, facilitando el reuso del código, la modularidad, y la legibilidad. Esto es particularmente importante en proyectos grandes o con múltiples componentes, ya que las clases permiten crear objetos con sus propias propiedades y responsabilidades, lo que mejora la mantención y extensibilidad del software. Además, promueven el enfoque de la programación orientada a objetos (POO), que ayuda a modelar problemas del mundo real de una forma más intuitiva y manejable.</p>

METODOLOGÍA DE TRABAJO

REFERENCIAS Y BIBLIOGRAFÍA					
	Contenido y demostración	Puntos	Checklist	Estudiante	Profesor
1. GitHub	Hay enlace URL activo del directorio para el laboratorio hacia su repositorio GitHub con código fuente terminado y fácil de revisar.	2	x	2	
2. Commits	Hay capturas de pantalla de los commits más importantes con sus explicaciones detalladas. (El profesor puede preguntar para refrendar calificación).	4	x	4	
3. Código fuente	Hay porciones de código fuente importantes con numeración y explicaciones detalladas de sus funciones.	2	x	2	

	<p style="text-align: center;"><b>UNIVERSIDAD NACIONAL DE SAN AGUSTIN</b>  <b>FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS</b>  <b>ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</b></p>	
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		
Aprobación: 2022/03/01	Código: GUIA-PRLE-001	Página: 7

4. Ejecución	Se incluyen ejecuciones/pruebas del código fuente explicadas gradualmente.	2	x	2	
5. Pregunta	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación).	2	x	2	
6. Fechas	Las fechas de modificación del código fuente están dentro de los plazos de fecha de entrega establecidos.	2	x	2	
7. Ortografía	El documento no muestra errores ortográficos.	2	x	2	
8. Madurez	El Informe muestra de manera general una evolución de la madurez del código fuente, explicaciones puntuales pero precisas y un acabado impecable. (El profesor puede preguntar para refrendar calificación).	4	x	3	
<b>TOTAL</b>		20		18	