

	<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>

## INFORME DE LABORATORIO

### (formato estudiante)

INFORMACIÓN BÁSICA					
ASIGNATURA:	Programación Web 2				
TÍTULO DE LA PRÁCTICA:	Ejercicio Python Transporte Aéreo				
NÚMERO DE PRÁCTICA:	Lab 04	AÑO LECTIVO:	2024	NRO. SEMESTRE:	2024 - B
FECHA DE PRESENTACIÓN	12/10/2024	HORA DE PRESENTACIÓN			
INTEGRANTE (s): Caracela Challco Anthony Jeffry				NOTA:	
DOCENTE(s): Ing. Lino Pinto Oppe					
LINK: Al apartado del ejercicio en Github <a href="https://github.com/LINOPINTO2023/PWII-2024B/tree/master1/Lab04/Lab04_Caracela_Challco_Antony_Jeffry_PWEB2">https://github.com/LINOPINTO2023/PWII-2024B/tree/master1/Lab04/Lab04_Caracela_Challco_Antony_Jeffry_PWEB2</a>					

SOLUCIÓN Y RESULTADOS
<p><b>I. SOLUCIÓN DE EJERCICIOS/PROBLEMAS</b></p> <p>Yo en lo personal me inspire en el paradigma de POO que lo aprendí en Fundamentos de la Programación 2 y se me hizo un camino más fácil de llevar lo que planteé fue crear una clase clientes que me permita registrar sus datos con métodos propios y que los almacene en atributos de la misma clase y almacenarlos en una lista y así poder usar los datos ya después cree una función para el registro de datos, otra para mostrar los resultados y el while true para el menú, a continuación explico las herramientas que use:</p> <p><b>1. Clase (class)</b></p> <p>Una clase es un modelo o plantilla que define las características y comportamientos de un conjunto de objetos. En programación orientada a objetos (OOP), las clases permiten crear objetos que comparten propiedades y métodos similares. Cada objeto creado a partir de una clase se llama una instancia.</p>

	<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: 2</p>

## 2. Métodos

Los métodos son funciones definidas dentro de una clase que actúan sobre las instancias (objetos) de esa clase. En Python, el primer parámetro de un método es siempre **self**, que representa al objeto actual. Esto permite acceder a los atributos y otros métodos del objeto.

## 3. while True (Bucle Infinito)

El while True crea un bucle infinito que se repetirá hasta que se cumpla una condición que lo interrumpa. Esto es útil cuando no se sabe exactamente cuántas veces se ejecutará el bucle y se desea que continúe hasta que el usuario decida detenerlo.

## 4. input()

La función input() se utiliza para recibir entradas del usuario desde la consola. Permite que el programa sea interactivo y reciba datos directamente del usuario.

## 5. Listas (list)

Las listas son estructuras de datos que pueden almacenar múltiples elementos. En Python, las listas pueden contener elementos de cualquier tipo (números, cadenas, objetos) y son dinámicas, lo que significa que pueden crecer o reducirse según sea necesario.

### CLASE CLIENTE

Le agregue un dato más que es nombre del titular que es importante para ver quien es el dueño de las entradas, el tipo de cliente, cantidad de pasajes, género del titular y el tipo de servicio.

Los métodos para calcular precio, si merece descuento o no y los importes

```
# Definir la clase Cliente
class Cliente:
    def __init__(self, nombre, tipo_cliente, cantidad_pasajes, genero, tipo_servicio):
        self.nombre = nombre
        self.tipo_cliente = tipo_cliente
        self.cantidad_pasajes = cantidad_pasajes
        self.genero = genero
        self.tipo_servicio = tipo_servicio

    def obtenerPrecio(self):
        # Precios según el tipo de servicio
        if self.tipo_servicio == 1:
            return 70.0 # Económica
        elif self.tipo_servicio == 2:
            return 140.0 # Ejecutiva
        elif self.tipo_servicio == 3:
            return 280.0 # Primera clase
        return 0.0

    def obtener_descuento(self):
        # Descuento según la cantidad de pasajes
        if self.cantidad_pasajes == 1:
            return 0.0
        elif 2 <= self.cantidad_pasajes <= 5:
            return 0.05 # 5%
        elif 6 <= self.cantidad_pasajes <= 10:
            return 0.12 # 12%
        elif self.cantidad_pasajes >= 11:
            return 0.15 # 15%
        return 0.0

    def calcular_importe_bruto(self):
        return self.cantidad_pasajes * self.obtenerPrecio()

    def calcular_monto_descuento(self):
        return self.calcular_importe_bruto() * self.obtener_descuento()

    def calcular_importe_netto(self):
        return self.calcular_importe_bruto() - self.calcular_monto_descuento()
```

### FUNCIÓN INGRESAR DATOS

Se crea la lista para almacenar los diferentes clientes y la función para obtener los datos del registro de clientes que tiene un while true para que registre las veces que queramos más y más clientes

```
# Lista para almacenar clientes
clientes = []

# Función para ingresar datos
def ingresar_datos():
    while True:
        nombre = input("Ingrese el nombre del titular: ")
        tipo_cliente = int(input("Ingrese el tipo de cliente (1 o 2): "))
        cantidad_pasajes = int(input("Ingrese la cantidad de pasajeros: "))
        genero = input("Ingrese el género del cliente (M/F): ").upper()
        tipo_servicio = int(input("Ingrese el tipo de servicio (1-Económica, 2-Ejecutiva, 3-Primera clase): "))

        # Crear un nuevo objeto Cliente
        cliente = Cliente(nombre, tipo_cliente, cantidad_pasajes, genero, tipo_servicio)
        clientes.append(cliente)

        # Mostrar los resultados para el cliente
        print(f"Importe Bruto: {cliente.calcular_importe_bruto():.2f}")
        print(f"Monto de Descuento: {cliente.calcular_monto_descuento():.2f}")
        print(f"Importe Neto: {cliente.calcular_importe_neto():.2f}")

        # Preguntar si el usuario desea agregar otro cliente
        continuar = input("¿Desea agregar otro cliente? (s/n): ").lower()
        if continuar != 's':
            break
```

## **FUNCIÓN PARA MOSTRAR DATOS DE VENTAS**

Inicializamos los contadores para que nos permita almacenar los datos de la lista de clientes, respondiendo a las preguntas necesarias del problema como:

- Cantidad de clientes de género masculino.
- Cantidad de clientes de género femenino.
- Cantidad de ventas cuyo Importe Neto sea  $\geq 70$  y  $\leq 500$
- Cantidad de ventas de clientes de género femenino cuyo Importe Neto sea  $\geq 140$  y  $\leq 1000$
- El acumulado del Importe de Ventas.
- El acumulado del Importe Neto de clientes, de tipo 1.
- Promedio de Importe Neto, de clientes, de tipo 1.

```
# Función para mostrar los resultados de ventas
def mostrar_resultados():
    total_ventas = 0
    total_clientes_masculinos = 0
    total_clientes_femeninos = 0
    ventas_rango_70_500 = 0
    ventas_femeninas_rango_140_1000 = 0
    acumulado_tipo_1 = 0
    total_importe_neto_tipo_1 = 0
    cantidad_tipo_1 = 0

    for cliente in clientes:
        importe_neto = cliente.calcular_importe_neto()
        total_ventas += importe_neto

        if cliente.genero == 'M':
            total_clientes_masculinos += 1

        if cliente.genero == 'F':
            total_clientes_femeninos += 1

        if 70 <= importe_neto <= 500:
            ventas_rango_70_500 += 1

        if cliente.genero == 'F' and 140 <= importe_neto <= 1000:
            ventas_femeninas_rango_140_1000 += 1

        if cliente.tipo_cliente == 1:
            acumulado_tipo_1 += importe_neto
            total_importe_neto_tipo_1 += 1
            cantidad_tipo_1 += 1

    print(f"Total de ventas: {total_ventas:.2f}")
    print(f"Cantidad de clientes masculinos: {total_clientes_masculinos}")
    print(f"Cantidad de clientes femeninos: {total_clientes_femeninos}")
    print(f"Cantidad de ventas con importe neto entre 70 y 500: {ventas_rango_70_500}")
    print(f"Cantidad de ventas femeninas con importe neto entre 140 y 1000: {ventas_femeninas_rango_140_1000}")
    print(f"Acumulado de importe neto de clientes de tipo 1: {acumulado_tipo_1:.2f}")

    if cantidad_tipo_1 > 0:
        promedio_tipo_1 = acumulado_tipo_1 / cantidad_tipo_1
        print(f"Promedio de importe neto de clientes de tipo 1: {promedio_tipo_1:.2f}")
    else:
        print("No hay clientes de tipo 1.")
```

## MENÚ PRINCIPAL











Usamos un while true para que elija solo una de las 3 opciones y no otro número x

```
# Menú principal
print("Bienvenido al registro de Transporte Aéreo")
while True:
    print("MENÚ DE OPCIONES")
    print("[1]Registrar Venta de Pasaje\n[2]Reportar Ventas\n[3]Salir")

    opcion = input("Seleccione una opción: ")

    if opcion == '1':
        ingresar_datos()
    elif opcion == '2':
        mostrar_resultados()
    elif opcion == '3':
        print("Saliendo del programa.")
        break
    else:
        print("Opción no válida. Intente de nuevo.")
```

## Commits

Commits on Oct 12, 2024		
Agregando más datos de mostrar datos	AnthonyCaracela committed now	ccebc5c  
Commits on Oct 11, 2024		
Merge branch 'master1' of <a href="https://github.com/LINOPINTO2023/PWII-2024B">https://github.com/LINOPINTO2023/PWII-2024B</a> into master1	AnthonyCaracela committed 14 hours ago	8bdecc8  
Informe lab2	AnthonyCaracela committed 14 hours ago	6ca3f6d  
Merge branch 'master1' of <a href="https://github.com/LINOPINTO2023/PWII-2024B">https://github.com/LINOPINTO2023/PWII-2024B</a> into master1	AnthonyCaracela committed 16 hours ago	5cbac1e  
Primera del registro con objetos y listas	AnthonyCaracela committed 16 hours ago	87aFe6f  

## Una pequeña prueba del registro

```
PS C:\Users\Lenovo\Documents\Visual Studio Code\Proyectos-PYTHON> Python TransporteAereo.py
Bienvenido al registro de Transporte Aéreo
MENÚ DE OPCIONES
[1]Registrar Venta de Pasaje
[2]Reportar Ventas
[3]Salir
Seleccione una opción: 1
Ingrese el nombre del titular: Mario
Ingrese el tipo de cliente (1 o 2): 1
Ingrese la cantidad de pasajes: 3
Ingrese el género del cliente (M/F): m
Ingrese el tipo de servicio (1-Económica, 2-Ejecutiva, 3-Primera clase): 2
Importe Bruto: 420.00
Monto de Descuento: 21.00
Importe Neto: 399.00
¿Desea agregar otro cliente? (s/n): s
Ingrese el nombre del titular: Fiorela
Ingrese el tipo de cliente (1 o 2): 2
Ingrese la cantidad de pasajes: 5
Ingrese el género del cliente (M/F): f
Ingrese el tipo de servicio (1-Económica, 2-Ejecutiva, 3-Primera clase): 1
Importe Bruto: 350.00
Monto de Descuento: 17.50
Importe Neto: 332.50
¿Desea agregar otro cliente? (s/n): n
MENÚ DE OPCIONES
[1]Registrar Venta de Pasaje
[2]Reportar Ventas
[3]Salir
Seleccione una opción: 2
Total de ventas: 731.50
Cantidad de clientes masculinos: 1
Cantidad de clientes masculinos: 1
Cantidad de ventas con importe neto entre 70 y 500: 2
Cantidad de ventas femeninas con importe neto entre 140 y 1000: 1
Acumulado de importe neto de clientes de tipo 1: 399.00
Promedio de importe neto de clientes de tipo 1: 399.00
MENÚ DE OPCIONES
[1]Registrar Venta de Pasaje
[2]Reportar Ventas
[3]Salir
Seleccione una opción: 3
Saliendo del programa.
PS C:\Users\Lenovo\Documents\Visual Studio Code\Proyectos-PYTHON> |
```

	<p 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: 8

### **RÚBRICA PARA EL CONTENIDO DEL INFORME Y DEMOSTRACIÓN**

El alumno debe marcar o dejar en blanco en celdas de la columna Checklist si cumplió con el ítem correspondiente.

Si un alumno supera la fecha de entrega, su calificación será sobre la nota mínima aprobada, siempre y cuando cumpla con todos los ítems.

El alumno debe autocalificarse en la columna Estudiante de acuerdo a la siguiente tabla:

<b>Contenido y demostración</b>		<b>Puntos</b>	<b>Checklist</b>	<b>Estudiante</b>	<b>Profesor</b>
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		2	
3. Código fuente	Hay porciones de código fuente importantes con numeración y explicaciones detalladas de sus funciones.	2		2	
4. Ejecución	Se incluyen ejecuciones/pruebas del código fuente explicadas gradualmente.	2		2	
5. Pregunta	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación).	2		2	
6. Fechas	Las fechas de modificación del código fuente están dentro de los plazos de fecha de entrega establecidos.	2		2	
7. Ortografía	El documento no muestra errores ortográficos.	2		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		2	
<b>TOTAL</b>		<b>20</b>		<b>16</b>	

Tabla 2: Rúbrica para contenido del Informe y demostración