

Universidad Americana

Facultad de Ingeniería y Arquitectura



Algoritmos y Estructuras de Datos

Grupo 4

Manejo de Funciones, Clases y Paquetes

Realizado por:

Joaquín Alberto Pérez Zúñiga

Docente:

MSc. César Marín López

Código:

1. main.py:

```
>_ nu      main.py M X
semana-3 > sesion-2 > main.py > ...
1  """
2  Ejemplo de paquetes y módulos.
3  """
4
5  from empleado import Empleado  # pylint: disable=E0401
6
7
8  def main() -> None:
9      """
10     Ejecución del programa.
11     """
12
13     empleados = []
14     n = int(input("\nIngrese la cantidad de empleados: "))
15
16     print("\nIngrese los datos de los empleados:")
17     for _ in range(n):
18         nombre = input("Nombre: ")
19         salario_bruto = float(input("Salario Bruto: "))
20         empleados.append(Empleado(nombre, salario_bruto))
21
22     print("\nDatos del Empleado:\n")
23     for emp in empleados:
24         print(emp.nombre)
25         print(emp.calcular_salario_neto())
26
27
28     if __name__ == "__main__":
29         main()
30
```

2. empleado.py:

```
semana-3 > ejercicios_funciones-oop > ej6 > factura.py > ...
1  """
2  Definición de la clase Factura.
3  """
4
5  from typing import Union
6  from clientes import (· · # type: ignore # pylint: disable=E0401
7  ····Cliente,
8  ····ClienteOro,
9  ····ClientePlatino
10 )
11
12 from producto import Producto· · # type: ignore # pylint: disable=E0401
13
14
15 class Factura:
16     """
17     Representa una factura.
18     """
19
20     def __init__(
21         self,
22         codigo: str,
23         cliente: Union[Cliente, ClienteOro, ClientePlatino],
24         productos: list[Producto]
25     ):
26         self.__codigo = codigo
27         self.__cliente = cliente
28         self.__productos = productos
29
30     @property
31     def codigo(self) -> str:
32         """
33         Getter para el atributo protegido __codigo.
34         """
35
36         return self.__codigo
37
```

> nu

main.py M

factura.py M X

salario_neto.py

semana-3 > ejercicios_funciones-oop > ej6 > factura.py > ...

```
15 class Factura:
38     ... @property
39     ... def cliente(self) -> Union[Cliente, ClienteOro, ClientePlatino]:
40     ...     """
41     ...     Getter para el atributo protegido __cliente.
42     ...     """
43
44     ...     return self.__cliente
45
46     ... @property
47     ... def productos(self) -> list[Producto]:
48     ...     """
49     ...     Getter para el atributo protegido __productos.
50     ...     """
51
52     ...     return self.__productos
53
54     ... def get_producto_str(self) -> str:
55     ...     """
56     ...     Convierte la lista de productos a un string para imprimir.
57     ...     """
58
59     ...     return "\n".join(str(p) for p in self.productos)
60
61     ... def calcular_total(self) -> float:
62     ...     """
63     ...     Suma todos los montos de todos los productos en
64     ...     self.productos y le aplica el descuento del cliente.
65     ...     """
66
67     ...     return sum(
68     ...         p.precio * p.cantidad for p in self.productos
69     ...     ) * self.cliente.descuento
70
```

> nu

main.py M

factura.py M X

salario_net.py

semana-3 > ejercicios_funciones-oop > ej6 > factura.py > ...

```
15 class Factura:
71     def generar_reporte(self) -> str:
72         """
73         Organiza los datos de la factura para imprimirlos.
74         """
75
76         total = self.calcular_total()
77         subtotal = total * (1 - self.cliente.descuento + 1)
78         reporte = ""
79
80         reporte += "=" * 75
81         reporte += f"\nFACTURA #{self.codigo}\n"
82         reporte += "=" * 75
83
84         reporte += "\nDATOS DEL CLIENTE:\n"
85         reporte += "-" * 75
86         reporte += f"\nNombre: {self.cliente.nombre}"
87         reporte += f"\nNúmero de teléfono: {self.cliente.telefono}"
88         reporte += f"\nDescuento aplicable: {(1 - self.cliente.descuento) * 100:.0f}%\n"
89
90         reporte += "-" * 75
91         reporte += "\nPRODUCTOS:\n"
92         reporte += "-" * 75
93         reporte += "\n" + self.get_producto_str()
94
95         reporte += "\n" + "=" * 75
96         reporte += f"\nSUBTOTAL: C${subtotal:.2f}"
97         reporte += f"\nDESCUENTO: {(1 - self.cliente.descuento) * 100:.0f}%"
98         reporte += f"\nTOTAL A PAGAR: C${total:.2f}\n"
99         reporte += "=" * 75
100        reporte += "\n"
101
102        return reporte
103
```

3. salario_neto.py:

```
> nu      main.py M      factura.py M      salario_neto.py X
semana-3 > sesion-2 > salario_neto.py > ...
1  """
2  Definición de la función salario_neto()
3  """
4
5  def salario_neto(salario_bruto: float) -> float:
6      """
7      ... Calcula el salario neto de un empleado.
8      ... """
9
10     ... inss = salario_bruto * 0.07
11     ... ir = (salario_bruto - inss) * 0.15
12
13     ... return salario_bruto - inss - ir
14
```

Output:

```
> nu      ×  main.py M
● C:\dev\algo-structs> python semana-3\ejercicios_funciones-oop\ej6\main.py

=====
FACTURA #001
=====
DATOS DEL CLIENTE:
-----
Nombre: Juancito Pérez
Número de teléfono: 7945-9316
Descuento aplicable: 0%
-----
PRODUCTOS:
-----
Código: 100, Nombre: Calculadora, Cantidad: 2, Precio: C$123.40
Código: 101, Nombre: Laptop, Cantidad: 1, Precio: C$2345.60
Código: 102, Nombre: Borrador, Cantidad: 3, Precio: C$45.60
=====
SUBTOTAL: C$2729.20
DESCUENTO: 0%
TOTAL A PAGAR: C$2729.20
=====

=====
FACTURA #002
=====
DATOS DEL CLIENTE:
-----
Nombre: Juanita X
Número de teléfono: 1243-6549
Descuento aplicable: 15%
-----
PRODUCTOS:
-----
Código: 103, Nombre: iPhone, Cantidad: 1, Precio: C$3456.70
Código: 104, Nombre: Galaxy S24, Cantidad: 1, Precio: C$2345.60
Código: 105, Nombre: Galaxy Buds, Cantidad: 2, Precio: C$1234.50
=====
SUBTOTAL: C$8085.20
DESCUENTO: 15%
TOTAL A PAGAR: C$7030.60
=====
```