

```
RFC: empleado_o3, Apellidos: Plata, Nombres: Oswaldo
Ingresos: 600.0
Sueldo Neto: 594.0
RFC: empleado_o3, Apellidos: Gómez, Nombres: Ana
Sueldo Neto: 178.0
PS C:\Users\Alumno\oswa> []
```

```
from abc import ABC, abstractmethod

class Empleado(ABC):
    def __init__(self, rfc, apellidos, nombres):
        self._rfc = rfc
        self._apellidos = apellidos
        self._nombres = nombres
    @abstractmethod
    def calcular_ingresos(self):
        pass
    @abstractmethod
    def calcular_descuento(self):
        pass
    @abstractmethod
    def calcular_sueldo_neto(self):
        pass
    def mostrar_info(self):
        return f"RFC: {self._rfc}, Apellidos: {self._apellidos}, Nombres: {self._nombres}"

class EmpleadoVendedor(Empleado):
    def __init__(self, rfc, apellidos, nombres, monto_vendido, tasa_comision):
        super().__init__(rfc, apellidos, nombres)
        self._monto_vendido = monto_vendido
        self._tasa_comision = tasa_comision
    def calcular_ingresos(self):
        return self._monto_vendido * self._tasa_comision
    def calcular_bonificacion(self):
        ingresos = self.calcular_ingresos()
        if self._monto_vendido < 1000:
            return 0
        elif 1000 <= self._monto_vendido <= 5000:
            return 0.05 * ingresos
        else:
            return 0.10 * ingresos
    def calcular_descuento(self):
        ingresos = self.calcular_ingresos()
        if ingresos < 1000:
```

```

        return 0.11 * ingresos
    else:
        return 0.15 * ingresos

    def calcular_sueldo_neto(self):
        ingresos = self.calcular_ingresos()
        bonificacion = self.calcular_bonificacion()
        descuento = self.calcular_descuento()
        return ingresos + bonificacion - descuento

class EmpleadoPermanente(Empleado):
    def __init__(self, rfc, apellidos, nombres, sueldo_base,
numero_seguro_social):
        super().__init__(rfc, apellidos, nombres)
        if sueldo_base < 150:
            raise ValueError("No puede ser menor a 150")
        self._sueldo_base = sueldo_base
        self._numero_seguro_social = numero_seguro_social

    def calcular_ingresos(self):
        return self._sueldo_base

    def calcular_descuento(self):
        return 0.11 * self._sueldo_base

    def calcular_sueldo_neto(self):
        ingresos = self.calcular_ingresos()
        descuento = self.calcular_descuento()
        return ingresos - descuento

def main():
    try:
        vendedor = EmpleadoVendedor('RFC001', 'Pérez', 'Juan', 6000, 0.1)
        print(vendedor.mostrar_info())
        print(f"Ingresos: {vendedor.calcular_ingresos()}")
        print(f"Suelo Neto: {vendedor.calcular_sueldo_neto()}")

        permanente = EmpleadoPermanente('RFC002', 'Gómez', 'Ana', 200,
'SS12345')
        print(permanente.mostrar_info())
        print(f"Suelo Neto: {permanente.calcular_sueldo_neto()}")

    except ValueError as e:
        print(e)

```

## Examen PARADIGMAS oswaldo

```
if __name__ == "__main__":  
    main()
```