

_5class_poo/class_gestion_empleados.py

```

1  """SISTEMA DE GESTIÓN DE EMPLEADOS
2  Supongamos que estás construyendo un sistema de gestión de empleados
3  para una empresa. Crea un sistema de clases que maneje la información de
4  los empleados, incluyendo empleados a tiempo completo y empleados a
5  tiempo parcial.
6      - Clase base: `Empleado`
7          - Atributos: nombre, apellido, salario base
8          - Clase derivada: `EmpleadoTiempoCompleto` (hereda de `Empleado`)
9          - Atributo adicional: bono anual
10         - Clase derivada: `EmpleadoTiempoParcial` (hereda de `Empleado`)
11         - Atributo adicional: horas trabajadas por semana
12     Resuelve el problema creando instancias de estas clases y calculando los
13     salarios totales para diferentes tipos de empleados."""
14
15 class Empleado:
16     def __init__(self,nombre,apellido,salario_base=1200):
17         self.nombre=nombre
18         self.apellido=apellido
19         self.salario_base=salario_base
20         print(f"Empleado{nombre} {apellido}. Con salario base {salario_base}")
21
22 class EmpleadoTiempoCompleto(Empleado):
23     def __init__(self, nombre,apellido,salario_base,bono_anual):
24         super().__init__(nombre,apellido,salario_base)
25         self.bono_anual=bono_anual
26         print(f"Empleado completo {nombre} {apellido} Con salario base {salario_base} mas
27         bono anual {bono_anual}")
28
29 class EmpleadoTiempoParcial(Empleado):
30     def __init__(self,nombre,apellido,horas_semana,salario_base):
31         super().__init__(self,nombre,apellido,salario_base)
32         self.horas_semana=horas_semana
33         precio_hora=7.40
34         resultado=horas_semana*precio_hora
35         print(f"Empleado parcial {nombre} {apellido}. Con salario base {salario_base} con
36         {horas_semana}={resultado}")
37
38 Empleado("Paco", "Cucamono",1200)
39 EmpleadoTiempoParcial("Pepe","MuertoHambre",15,1200)
40 EmpleadoTiempoCompleto("Jesus", "Bolloleche",1200)
41
42 class Empleado:
43     def __init__(self, nombre, apellido, salario_base):
44         self.nombre = nombre
45         self.apellido = apellido
46         self.salario_base = salario_base
47
48     def calcular_salario(self):
49         return self.salario_base
50
51 class EmpleadoTiempoCompleto(Empleado):
52     def __init__(self, nombre, apellido, salario_base,bono_anual):
53         super().__init__(self,nombre, apellido, salario_base)

```

```
51         self.bono_anual = bono_anual
52
53     def calcular_salario(self):
54         return super().calcular_salario() + self.bono_anual
55
56 class EmpleadoTiempoParcial(Empleado):
57     def __init__(self, nombre, apellido, salario_base, horas_semana):
58         super().__init__(nombre, apellido, salario_base)
59         self.horas_semana = horas_semana
60
61     def calcular_salario(self):
62         pago_por_hora = 7.40
63         salario_semanal = self.horas_semana * pago_por_hora
64         return salario_semanal * 4 # Suponiendo 4 semanas en un mes
65
66 # Crear instancias y calcular salarios
67 empleado1 = Empleado("Paco", "Cucamono", 1200)
68 empleado2 = EmpleadoTiempoCompleto("Pepe", "MuertoHambre", 500)
69 empleado3 = EmpleadoTiempoParcial("Jesus", "Bolloleche", 0, 15)
70
71 print(f"Salario total de {empleado1.nombre} {empleado1.apellido}:
72     ${empleado1.calcular_salario()}")
73 print(f"Salario total de {empleado2.nombre} {empleado2.apellido}:
74     ${empleado2.calcular_salario()}")
75 print(f"Salario total de {empleado3.nombre} {empleado3.apellido}:
76     ${empleado3.calcular_salario()}")
```