

✓ Actividad integradora 2: Creación de objetos y validando con expresiones regulares

- Juan Pérez
- A0888888

```

1 class Empleado:
2     def __init__(self, nomina, nombre, edad):
3         self.nomina = nomina
4         self.nombre = nombre
5         self.edad = edad
6
7     def pagoMensual(self):
8         pass
9
10    def mostrar(self):
11        print("Nómina: " + self.nomina + ", Nombre: " + self.nombre + ", Edad: " + str(self.edad))
12
13 class Asalariado(Empleado):
14     def __init__(self, nomina, nombre, edad, salarioMensual):
15         self.__salarioMensual = salarioMensual
16         super().__init__(nomina, nombre, edad)
17
18     def getsalario(self):
19         return self.__salarioMensual
20
21     def setsalario(self, salarioMensual):
22         self.__salarioMensual = salarioMensual
23
24     def pagoMensual(self, beneficios):
25         return self.__salarioMensual + beneficios
26
27     def mostrar(self):
28         print("Nómina: " + self.nomina + ", Nombre: " + self.nombre)
29         print("Edad: " + str(self.edad) + ", Salario mensual: " + str(self.getsalario()))
30
31 class PorHora(Empleado):
32     def __init__(self, nomina, nombre, edad, horasTrabajadas, pagoHora):
33         self.__horasTrabajadas = horasTrabajadas
34         self.__pagoHora = pagoHora
35         super().__init__(nomina, nombre, edad)
36
37     def gethorasTrabajadas(self):
38         return self.__horasTrabajadas
39
40     def getpagoHora(self):
41         return self.__pagoHora
42
43     def sethorasTrabajadas(self, horasTrabajadas):
44         self.__horasTrabajadas = horasTrabajadas
45
46     def setpagoHora(self, pagoHora):
47         self.__pagoHora = pagoHora
48
49     def pagoMensual(self, beneficios):
50         return self.__horasTrabajadas * self.__pagoHora + beneficios
51
52     def mostrar(self):
53         print("Nómina: " + self.nomina + ", Nombre: " + self.nombre + ", Edad: " + str(self.edad))
54         print("Horas trabajadas: " + str(self.gethorasTrabajadas()) + ", Pago por hora: " + str(self.getpagoHora))
55
56 import re
57 class Verificador:
58     def __init__(self, nomina, nombre, edad):

```

```

4     self.nomina = nomina
5     self.nombre = nombre
6     self.edad = edad
7
8     def verificarNomina(self):
9         patron = "^L00\d{6}$"
10        if re.match(patron, self.nomina) != None:
11            return True
12        else:
13            return False
14
15    def verificarNombre(self):
16        patron = "^[A-Za-záéíóúÁÉÍÓÚñÑ]{1,2}[A-Za-záéíóúÁÉÍÓÚñÑ]+\s[A-Za-záéíóúÁÉÍÓÚñÑ]+$"
17        if re.match(patron, self.nombre) != None:
18            return True
19        else:
20            return False
21
22    def verificarEdad(self):
23        patron = "(1[89])|([2-5]\d)|(6[0-5])"
24        if re.match(patron, self.edad) != None:
25            return True
26        else:
27            return False

```

1 # Regresa verdadero si los tres atributos de instancia son válidos

```

2 def valida_datos(nomina, nombre, edad):
3     v = Verificador(nomina, nombre, edad)
4     if v.verificarNomina() == True and v.verificarNombre() == True and v.verificarEdad() == True:
5         return True
6     else:
7         print()
8         print("Es válida la nómina?: ", v.verificarNomina())
9         print("Es válido el nombre?: ", v.verificarNombre())
10        print("Es válida la edad?: ", v.verificarEdad())
11        print()
12        print("INTRODUCE DATOS VÁLIDOS")
13        return False

```

```

1 def main():
2     es_valido = False
3     while es_valido != True:
4         print()
5         nomina = input("Dame la nómina: ")
6         nombre = input("Dame el nombre: ")
7         edad = input("Dame la edad: ")
8         es_valido = valida_datos(nomina, nombre, edad)
9         print("\nEMPLEADO")
10        e = Empleado(nomina, nombre, edad)
11        e.mostrar()
12        print("\nEMPLEADO ASALARIADO")
13        salario = float(input("Dame el salario mensual: "))
14        a = Asalariado(nomina, nombre, edad, salario)
15        a.mostrar()
16        print("\nEMPLEADO POR HORA")
17        horas = int(input("Dame las horas trabajadas: "))
18        pago = float(input("Dame el pago por hora: "))
19        h = PorHora(nomina, nombre, edad, horas, pago)
20        h.mostrar()
21
22 main()

```



```

Dame la nómina: L00614578
Dame el nombre: Juan Pérez
Dame la edad: 70

```

```
Es válida la nómina?: True
Es válido el nombre?: False
Es válida la edad?: False
```

INTRODUCE DATOS VÁLIDOS

```
Dame la nómina: L006383
Dame el nombre: Juan Pérez
Dame la edad: 75
```

```
Es válida la nómina?: False
Es válido el nombre?: False
Es válida la edad?: False
```

INTRODUCE DATOS VÁLIDOS

```
Dame la nómina: L00613535
Dame el nombre: Juan López Pérez
Dame la edad: 25
```

EMPLEADO

Nómina: L00613535, Nombre: Juan López Pérez, Edad: 25

EMPLEADO ASALARIADO

```
Dame el salario mensual: 30000
Nómina: L00613535, Nombre: Juan López Pérez
Edad: 25, Salario mensual: 30000.0
```

EMPLEADO POR HORA

```
Dame las horas trabajadas: 15
Dame el pago por hora: 300
Nómina: L00613535, Nombre: Juan López Pérez, Edad: 25
Horas trabajadas: 15, Pago por hora: 300.0
```