

▼ Actividad 2

```

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

```

```
1 import re
2 class Verificador:
3     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}$"
10        if re.match(patron, self.nomina) != None:
11            return True
12        else:
13            return False
14
15    def verificarNombre(self):
16        #patron = "[A-Za-záéíóúÁÉÍÓÚñ]+\s[A-Za-záéíóúÁÉÍÓÚñ]+\s([A-Za-záéíóúÁÉÍÓÚñ]+\s)?[A-Za-záéíóúÁÉÍÓÚñ]+$"
17        patron = "^[A-Za-záéíóúÁÉÍÓÚñ]+\s){1,2}[A-Za-záéíóúÁÉÍÓÚñ]+\s[A-Za-záéíóúÁÉÍÓÚñ]+$"
18        if re.match(patron, self.nombre) != None:
19            return True
20        else:
21            return False
22
23    def verificarEdad(self):
24        patron = "^((\d?\d)|1([0-2][0-9]))$"
25        if re.match(patron, str(self.edad)) != None:
26            return True
27        else:
28            return False
```

```
1 def valida_datos(nomina, nombre, edad):
2     v = Verificador(nomina, nombre, edad)
3     if v.verificarNomina() == True and v.verificarNombre() == True and v.verificarEdad() == True:
4         return True
5     else:
6         print()
7         print("¿Es válida la nomina?", v.verificarNomina())
8         print("¿Es válido el nombre?", v.verificarNombre())
9         print("¿Es válida la edad?", v.verificarEdad())
10        print("Introduce datos válidos")
11        return False
12
13 def main():
14     es_valida = False
15     while es_valida != True:
```