



## UD 6. Programación orientada a objetos

### EJERCICIOS

Desarrollo web en entorno servidor  
2do CFGS DAW

Autor: Vanessa Tarí Costa  
[vanessa.tari@ceedcv.es](mailto:vanessa.tari@ceedcv.es)  
2020/2021

## Licencia



**Reconocimiento – NoComercial – CompartirIgual (by-nc-sa):** No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

## Nomenclatura

A lo largo de este tema se utilizarán distintos símbolos para distinguir elementos importantes dentro del contenido. Estos símbolos son:



Importante



Atención



Interesante

## ÍNDICE

EJERCICIO 1	3
EJERCICIO 2	3
EJERCICIO 3	3
EJERCICIO 4	3
EJERCICIO 5	4
EJERCICIO 6	4
EJERCICIO 7	4
EJERCICIO 8	4

### EJERCICIO 1

Crea una clase *Empleado* con 2 propiedades: *nombre* y *sueldo*. Implementa los getters y setters de las dos propiedades (ten en cuenta la visibilidad adecuada de las propiedades y métodos).

Crea dos empleados rellenando su nombre y sueldo, y haz que salga por pantalla la frase “*nombre\_empleado* tiene un sueldo de *sueldo\_empleado*”.

### EJERCICIO 2

Modifica el ejercicio anterior para que el nombre y el sueldo del empleado se rellenen en el momento en que se crea el empleado y no se pueda modificar después.

### EJERCICIO 3

Añade un método a la clase *Empleado* que devolverá un booleano indicando si el empleado tiene que pagar impuestos (si su sueldo es mayor que 1200) o no (si es menor).

Muestra la frase “*nombre\_empleado* tiene que pagar impuestos” o “*nombre\_empleado* no tiene que pagar impuestos” después de la frase “*nombre\_empleado* tiene un sueldo de *sueldo\_empleado*”.

### EJERCICIO 4

Modifica el ejercicio anterior para que el mostrar la frase de si tiene que pagar impuestos o no sea un método de la clase *Empleado* (diferente del método que calcula si tiene que pagar impuestos o no).

Comprueba que las visibilidades de los métodos son las correctas.

### EJERCICIO 5

Plantea una clase Calculadora que contenga cuatro métodos estáticos (*sumar()*, *restar()*, *multiplicar()* y *dividir()*) estos métodos reciben dos valores y calcularán la operación asociada.

### EJERCICIO 6

Modifica el ejercicio anterior para que el primero de los números sea una propiedad de la clase Calculadora y valga siempre 8.

### EJERCICIO 7

Crea una clase Mamifero con las propiedades *especie*, *sonido* y *familia*, y un constructor donde se le pasarán la especie y el sonido (pero no la familia) para rellenarlos. Además, tendrá un método *sonido()* donde se mostrará la frase “Sonido de *especie*, de la familia *familia*: *sonido*”.

Crea otras dos clases que hereden de Mamifero llamadas Perro y Gato. Ambas clases tendrán sólo un constructor donde se rellenará la familia a la que pertenece cada una (cánidos o felinos).

Comprueba que la aplicación funciona de forma correcta creando un perro y un gato y ejecutando su métodos *sonido()*.

### EJERCICIO 8

Crea una clase Trabajador. Define como atributos su nombre y sueldo. La clase tendrá el método *calcularSueldo()*.

Implementa la clase Empleado, que heredará de la anterior. Para calcular el sueldo tener en cuenta que se le paga 9.50 la hora.

Plantea otra clase Gerente que herede de la clase Trabajador. Para calcular el sueldo tener en cuenta que se le abona el sueldo base (2500) más un porcentaje del beneficio de la empresa (parámetro que se le pasará al constructor de la clase Gerente).

Crear los métodos necesarios en las clases y comprobar su funcionamiento.