

Ejercicio 1. Define los siguientes atributos indicando su tipo.

estaAbierta ; indica si una puerta está abierta.	
nombreJugador , indica el nombre de un jugador de fútbol.	
altura , indica la altura, en centímetros.	
altura , indica la altura, en metros.	
Sexo , indica el sexo de una persona con dos opciones ('M', 'F').	
tienePlaya , indica si una población tiene playa.	
resultadoDado , indica el resultado de un dado de 6 caras.	

Ejercicio 2: Evalúa como True o False las siguientes expresiones.

$(x > 0) \ \&\& \ (x < 15)$, con $x = 8$	
$(x > 0) \ \&\& \ (x < 15)$, con $x = 20$	
$!(x \geq 10) \ \ (x > 15)$, con $x = 5$	
$(x > 0) \ \ (x < 15)$, con $x = 20$	
$(x \neq 2) \ \&\& \ (x == 5)$, con $x = 3$	
$(x < 3) \ \&\& \ !(x > 10)$, con $x = 1$	
$(x < 8) \ \ (y < 0)$, con $x = 10$ e $y = 0$	
$(x < 8) \ \&\& \ (y < 0)$, con $x = 3$ e $y = -2$	
$(x \neq 6) \ \ !(y < 0)$, con $x = 6$ e $y = 4$	
$(x == 2) \ \&\& \ (y < 10)$, con $x = 5$ e $y = 6$	

Ejercicio 3

Genere una clase llamada **Pistola** que tenga los siguientes atributos:

- **Modelo** (de tipo texto)
- **numBalas** (entero) → Número de balas que tiene la pistola.
- **numMaximoBalas** (entero) → Número máximo de balas que tiene.

Y deberá tener los siguientes métodos:

- **Disparar()**: El número de balas que tenga deberá reducirse en uno. No es necesaria la lógica de implementar si $\text{numBalas} = 0$.
- **Recargar()**: El número de balas será el número de balas máximo.
- **obtenerNumBalas()**: Devuelve el número de balas que tiene la pistola.

NOTA: No es necesario realizar el constructor.

Ejercicio 4

Genere una clase llamada ***Ordenador*** que tenga los siguientes atributos:

- **Marca** (de tipo texto)
- **Modelo** (de tipo texto)
- **Memoria**(entero) → Memoria del disco duro en GB.
- **MemLibre** (entero) → Memoria del disco duro disponible en GB.
- **numSWInstalado** (entero) → Número de programas o softwares instalados en el ordenador.

Y deberá tener los siguientes métodos:

- **InstalarSW():** Tendrá un parámetro de entrada que indicará (en GB) la memoria del software a instalar, por lo tanto, en este método la memoria disponible se reducirá según la memoria del software a instalar y el numero de programas instalados aumentan en 1.
- **Formatear():** El número de programas se reduce a 0 y la memoria disponible será la memoria total.
- **obtenerMemLibre():** Devuelve la memoria disponible en GB.

NOTA: No es necesario realizar el constructor.

Las entregas de los ejercicios 3 y 4 se pueden realizar en ficheros txt, java, docx o pdf.