
	<b>Curso</b>	DAM on-line			
	<b>Módulo</b>	PROGRAMACIÓN			
	<b>Fecha</b>	11 - Junio - 2020	<b>Duración</b>	120 minutos	
	<b>Parcial</b>	1 (JUNIO)	<b>Temas</b>	1 al 5	
<b>Criterios de calificación</b>					<b>Puntuación</b>
Preguntas (realizar solo 2 de las 3 propuestas)				1 punto cada una	2 puntos
<u>Ejercicio Práctico</u>					8 puntos
<b>Total</b>					<b>10 puntos</b>
<b>Instrucciones generales</b>					
La entrega del examen se realizara en un único fichero zip utilizando la plataforma avanza. El nombre del fichero será <b>apellido1_apellido2_nombre_PROG_1Parcial</b> y contendrá un documento de texto con las respuestas a las preguntas elegidas y el proyecto practico realizado en netbeans.					

<b>Nombre</b>	<b>PC</b>
---------------	-----------

PREGUNTAS (de las 3 propuestas solo debes realizar 2 de ellas) 2 puntos

**IMPORTANTE:**

- Vuelvo a insistir **de las 3 preguntas** propuestas **solo debes hacer 2 preguntas**.
- Para responder **no es necesario implementar nada en netbeans**, solo explicarlo. Se permiten errores en código o instrucciones que se utilicen como ejemplo en la explicación. (Esto es como si lo hiciéramos escrito en papel)
- Las **respuestas** deben ir en un **documento de texto**, con el formato de costumbre `apellido1_apellido2_nombre_Preguntas.doc`

**PREGUNTA 1: Estructuras de control (1 punto)**

Si deseamos hacer un menú de opciones (en línea de comandos). ¿Qué estructuras de control debemos utilizar? ¿Cómo sería?

**Ejemplo**

```
1.- Crear vendedor
2.- Añadir venta
3.- Mostrar datos vendedor
4.- Salir
```

**PREGUNTA 2: Validación (1 punto)**

Tenemos un atributo `codigo`, de empleado, con el formato `xx-x` siendo `x` un dígito. Ejemplo `10-1`. ¿Cuál es la forma más correcta de validar? ¿Cómo lo harías? ¿Sería necesario el control de excepciones?

**PREGUNTA3: Estructura de datos (1 punto)**

Si tenemos un objeto vendedor (como el del ejercicio práctico) que estructura de datos sería la más correcta para almacenar varios vendedores sabiendo que cada objeto vendedor es único, no vamos a almacenar más de 4 ó 5 vendedores y las búsquedas (que se realizarán pocas) se podrán hacer por distintos atributos, es decir a veces buscamos por código, nombre o nif. Justifica la respuesta.

## EJERCICIO PRÁCTICO: **Vendedor.** 8 puntos

### IMPORTANTE:

- **No** será necesario **validar** los datos. La especificación de los atributos se realiza para la elección del tipo de dato.
- **No** es necesario el control de **excepciones**.
- Para este ejercicio **no** es necesario **pedir datos** al usuario. Pondremos directamente **nuestros datos personales de ejemplo**.

Implementaremos la clase **vendedor** con una serie de atributos y métodos. En el programa principal crearemos un objeto de dicha clase y realizaremos una serie de operaciones.

### Clase Vendedor:

Con los siguientes atributos:

- **NIF**, ocho dígitos mas una letra.
- **Nombre**, un texto de 20 caracteres.
- **Apellidos** una texto de 40 caracteres.
- **codigo**, de empleado, con el formato **XX-X** siendo **X** un dígito. Ejemplo 10-1.
- **sueldoBase**, sueldo base del empleado una cantidad entera. Ejemplo: 900
- **turno**, que puede ser Mañana o Tarde. Ejemplo: Tarde
- **comision**, será un numero entero que indica un porcentaje fijo (1-100), que se aplicara sobre las ventas realizadas. Ejemplo: 2 que indica un 2%.
- **ventas**, un número con decimales que va contabilizando el total de ventas de un vendedor. Al principio vale 0.

Los **métodos de la clase Vendedor** serán los métodos **set** y **get** para cada atributo y el método **toString**. Además:

- Un método **realizarVenta** que **recibe** como parámetro el importe de la venta realizada, este es un número con decimal, el método debe sumar dicha cantidad a la cantidad que tenga el atributo **ventas**. Para ello debemos utilizar el **get** y **set** de **ventas**.
- El método **calcularSueldo** **devuelve** un número con decimales que se calcula según el porcentaje que tenga de comisión sobre las ventas que ha realizado  $(comisión * ventas) / 100$  y le sumamos el **sueldoBase**. Por lo tanto sería aplicar la siguiente fórmula:

$$((comisión * ventas) / 100) + sueldoBase$$

### En el programa principal (no hay menú)

- Crea un vendedor **con tus datos personales**.
- Añadir varias ventas a ese vendedor.
- Calcular y mostrar el sueldo, de ese vendedor.
- Mostrar todos los datos del vendedor.

#### Criterios de puntuación: 8 puntos

1 punto – Atributos de la Clase

0,5 puntos – Constructores

1 punto – Getters, Setters y toString

1,5 puntos – método realizarVenta

2 puntos – método calcularSueldo

Programa principal

0,5 punto - Crear vendedor

0,5 puntos – añadir varias ventas a ese vendedor

0,5 puntos – calcular y mostrar el sueldo de ese vendedor

0,5puntos – mostrar los datos del vendedor