

## CASO PRÁCTICO 4

- **TÍTULO: Programación orientada a objetos con Java**

- **SITUACIÓN**

Tenemos que resolver los siguientes problemas para la empresa de programación para la que trabajamos.

- **INSTRUCCIONES**

En la empresa multinacional Hasentur se desea implementar un sistema de gestión de empleados que permita calcular el salario total de cada trabajador, teniendo en cuenta su salario base y posibles bonificaciones. Para garantizar un control adecuado de estos valores, se han establecido ciertas restricciones que deben cumplirse para evitar errores en los cálculos.

Cada empleado tiene un **salario base** que debe ser un **valor positivo**, ya que no tiene sentido que un trabajador perciba un salario igual o inferior a cero. Si se intenta asignar un salario no válido al momento de crear un empleado, el sistema debe generar una excepción **SalarioInvalidoException** e impedir que el empleado sea registrado.

Además, la empresa otorga bonificaciones adicionales a sus empleados en función de su desempeño o logros, pero estas **no pueden ser negativas ni superar el 50% del salario base** del trabajador. Si se intenta asignar una bonificación fuera de este rango, el sistema debe lanzar una excepción **BonificacionInvalidaException** y rechazarla y mostrar un mensaje de error.

El programa debe permitir la creación de empleados con un salario válido, asignarles bonificaciones dentro de los límites establecidos y calcular el salario total sumando ambos valores. También debe proporcionar un método para visualizar los datos del empleado de manera clara. Si en algún momento se ingresan valores incorrectos, el sistema debe manejar la situación sin interrumpir su ejecución, mostrando el mensaje de error adecuado para que el usuario comprenda el problema y pueda corregirlo.

En la implementación del sistema, se utilizarán **excepciones personalizadas** para manejar estos errores de manera estructurada. Cuando un usuario intente crear un empleado con un salario inválido, se lanzará una excepción específica para señalar el problema. Del mismo modo, si se asigna una bonificación fuera de los límites permitidos, se generará otra excepción que indicará el motivo del error. Estas excepciones serán **capturadas en el programa principal** para evitar fallos inesperados y asegurar que la ejecución del sistema continúe sin problemas.

La información que tenemos de cada Empleado será:

- nombre: String
- salarioBase: double
- bonificacion: double

Este sistema permitirá a la empresa gestionar de manera eficiente la información de sus empleados, asegurando que los cálculos salariales sean precisos y que no se ingresen datos inconsistentes que puedan afectar la contabilidad o la administración de recursos humanos.

Ejemplo de ejecución:

```
Introduzca el nombre del empleado (o 'FIN' para salir): Pepe
Introduzca el salario base: 1200
Introduzca la bonificación: 100
Empleado: Pepe
Salario Base: 1200,00€
Bonificación: 100,00€
Salario Total:1300,00€

Introduzca el nombre del empleado (o 'FIN' para salir): Juan
Introduzca el salario base: 1350
Introduzca la bonificación: 1000
Error: La bonificación debe estar entre 0 y el 50% del salario base.
Introduzca los datos de nuevo

Introduzca el nombre del empleado (o 'FIN' para salir): Juan
Introduzca el salario base: 1350
Introduzca la bonificación: 500
Empleado: Juan
Salario Base: 1350,00€
Bonificación: 500,00€
Salario Total:1850,00€

Introduzca el nombre del empleado (o 'FIN' para salir): fin
Finalizando la entrada de empleados.
```