



Reto 1 – Concesionario

Objetivo:

El objetivo de este reto es que el estudiante reconozca y aplique los elementos básicos del paradigma de la programación orientada a objetos en un escenario abstraído de la cotidianidad.

Contexto:

Un empleado del concesionario **CARMAX S.A.S** se puede representar con la notación UML de la siguiente manera:

| Empleado |
|--|
| - id: int - nombre: String - apellido: String - comision: int - horasExtra: int - salario: int |
| + Empleado(nombre:String, apellido:String, comision:int, horasExtra:int, salario:int) + calcularMiNomina(Empleado empleado): double + getters + setters |

Los empleados o asesores de ventas del concesionario tienen un sueldo mensual base de \$1'875.000 COP. Además de este sueldo base, pueden obtener más ingresos si trabajan horas extra, o, si tienen un muy buen mes de venta, el concesionario los premia con un bono de comisión adicional.

Diego Moreno fue un asesor de ventas premiado con un bono de comisión de \$500.000 COP, y aunque no trabajó horas extras, este reconocimiento lo hizo muy feliz por que se tendrá en cuenta para la liquidación de su nómina.

Nota: Tome la siguiente información para verificar el correcto funcionamiento de su implementación.

Reto:

Se requiere que construya una solución que le permita a los empleados del concesionario **CARMAX S.A.S** liquidar su nómina. Para ello, los empleados deben de tener presente que se deben agrupar los conceptos que conforman el total devengado (salario base, comisiones, horas extra, auxilio de transporte), y luego restarle a ese valor las deducciones de nómina que corresponden a los aportes a salud y pensión.



Nota: Los porcentajes de salud y pensión que le corresponde al empleado son del 4% respectivamente. Estos porcentajes se efectúan sobre la suma de los conceptos que conforman el total devengado, pero sin tener en cuenta el auxilio de transporte.

Un ejemplo de la estructura del código de este programa se muestra a continuación:

```
public class Empleado {  
    private int id;  
    private final String nombre;  
    private final String apellido;  
    private final int comision;  
    private final int horasExtra;  
    private final int salario;  
  
    public Empleado(String nombre, String apellido, int comision, int horasExtra, int salario) {  
    }  
  
    public static double calcularMiNomina(Empleado empleado){  
    }  
}
```

Para corroborar el funcionamiento de su implementación, tome la información antes brindada del empleado Diego Moreno y realice el cálculo de su nómina.

Casos de prueba:

Para verificar el funcionamiento de su programa se sugiere que tenga en consideración los siguientes casos de prueba

| # CASO DE PRUEBA | DATOS DE ENTRADA | | | SALIDA ESPERADA |
|------------------|------------------|-------------|-------------|--------------------------|
| 1. | Comisión | Horas extra | salario | Valor: 2185000 |
| | \$500.000 | 0 | \$1'875.000 | |
| | | | | |



Entrega:

1. Suba a la plataforma un archivo con el nombre de **Empleado.java**, este nombre debe de respetarse, dado que, si no se nombre de dicha manera no se tendrá en cuenta para la calificación del reto.
2. **Importante:** Los métodos deben de llamarse **exactamente igual** a como se muestra en el ejemplo de la estructura del código.