



## Reto 3 – Call Center

### 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:

El Call Center continua con la elaboración de su sistema de liquidación, previamente había incorporado funcionalidades que le permitían calcular la nómina y las prestaciones sociales de un empleado. Actualmente, el Call Center en su afán de tener un sistema más completo ha decidido añadir el cálculo de las liquidaciones concernientes a la seguridad social y los aportes a parafiscales. Por tal motivo, su representación en el sistema tendrá ahora los siguientes comportamientos:

Call Center
- empleados: ArrayList<Empleado>
+ liquidarNominaEmp(empleados: ArrayList<Empleado>): ArrayList<Double> + liquidarPrestacionesEmp(empleados: ArrayList<Empleado>): ArrayList<Double> + liquidarSegSocialEmp(empleados: ArrayList<Empleado>): ArrayList<Double> + liquidarParafiscalesEmp(empleados: ArrayList<Empleado>): ArrayList<Double> + getters + setters

La liquidación de la seguridad social de un empleado se obtiene cuando se suman los conceptos de salud (8,5%), pensión (12%) y ARL (0,522%), que se encuentran aplicados al total devengado de cada empleado, pero se **excluye** el auxilio de transporte, ya que no es un factor salarial.

Por otro lado, para realizar el cálculo de la liquidación de los parafiscales se suman los conceptos de caja de compensación familiar (4%), ICBF (3%) y SENA (2%), que se efectúan sobre el devengado total de un empleado, pero se exime el concepto de auxilio de transporte dentro de este cálculo, tal y como se realiza en la liquidación de la seguridad social.

### Reto:

**Nota:** Dé continuidad a la construcción del sistema de liquidación del **Call Center** e incluya las nuevas funcionales del reto actual a su solución.



Realice la implementación de los métodos **liquidarSegSocialEmp** y **liquidarParafiscalesEmp** con el fin de que se obtenga el valor de cada concepto, por cada uno de los empleados que trabajan en el **Call Center**. De manera similar a los métodos desarrollados en los retos 1 y 2, los métodos actuales toman una colección de empleados y devuelven una lista con los valores a liquidar para cada empleado que figura en la colección. Si por alguna razón, no se encuentran empleados en la colección, los métodos deberán de retornar una lista vacía.

Una vez integre las nuevas funcionalidades en la clase **Call Center**, está deberá de verse de la siguiente manera:

```
public class CallCenter {  
    private ArrayList<Empleado> empleados = new ArrayList<>();  
    public static ArrayList<Double> liquidarNominaEmp(ArrayList<Empleado> empleados){  
    }  
    public static ArrayList<Double> liquidarPrestacionesEmp(ArrayList<Empleado> empleados){  
    }  
    public static ArrayList<Double> liquidarSegSocialEmp(ArrayList<Empleado> empleados){  
    }  
    public static ArrayList<Double> liquidarParafiscalesEmp(ArrayList<Empleado> empleados){  
    }  
}
```



## 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														
liquidarSegSocialEmp	Vector Vacío	Valor: Vector Vacío														
liquidarSegSocialEmp	<table><tr><th>Empleado 1</th><th>Empleado 2</th></tr><tr><td>Nombre: Carmen</td><td>Nombre: Laura</td></tr><tr><td>Horas Extra: \$40.000</td><td>Horas Extra: \$50.000</td></tr><tr><td>Auxilio: Si</td><td>Auxilio: No</td></tr><tr><td>Salario: \$1'500.000</td><td>Salario: \$2'000.000</td></tr></table>	Empleado 1	Empleado 2	Nombre: Carmen	Nombre: Laura	Horas Extra: \$40.000	Horas Extra: \$50.000	Auxilio: Si	Auxilio: No	Salario: \$1'500.000	Salario: \$2'000.000	Valor: <table><tr><th>Valor 1</th><th>Valor 2</th></tr><tr><td>323.738,80</td><td>430.951,00</td></tr></table>	Valor 1	Valor 2	323.738,80	430.951,00
Empleado 1	Empleado 2															
Nombre: Carmen	Nombre: Laura															
Horas Extra: \$40.000	Horas Extra: \$50.000															
Auxilio: Si	Auxilio: No															
Salario: \$1'500.000	Salario: \$2'000.000															
Valor 1	Valor 2															
323.738,80	430.951,00															
liquidarParafiscalesEmp	Vector Vacío	Valor: Vector Vacío														
liquidarParafiscalesEmp	<table><tr><th>Empleado 1</th><th>Empleado 2</th></tr><tr><td>Nombre: Carmen</td><td>Nombre: Laura</td></tr><tr><td>Horas Extra: \$40.000</td><td>Horas Extra: \$50.000</td></tr><tr><td>Auxilio: Si</td><td>Auxilio: No</td></tr><tr><td>Salario: \$1'500.000</td><td>Salario: \$2'000.000</td></tr></table>	Empleado 1	Empleado 2	Nombre: Carmen	Nombre: Laura	Horas Extra: \$40.000	Horas Extra: \$50.000	Auxilio: Si	Auxilio: No	Salario: \$1'500.000	Salario: \$2'000.000	Valor: <table><tr><th>Valor 1</th><th>Valor 2</th></tr><tr><td>138.600,00</td><td>184.500,00</td></tr></table>	Valor 1	Valor 2	138.600,00	184.500,00
Empleado 1	Empleado 2															
Nombre: Carmen	Nombre: Laura															
Horas Extra: \$40.000	Horas Extra: \$50.000															
Auxilio: Si	Auxilio: No															
Salario: \$1'500.000	Salario: \$2'000.000															
Valor 1	Valor 2															
138.600,00	184.500,00															

## Entrega:

1. Suba a la plataforma los archivos **CallCenter.java** y **Empleado.java**, estos nombres deben de respetarse, dado que, si no se nombran 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.