



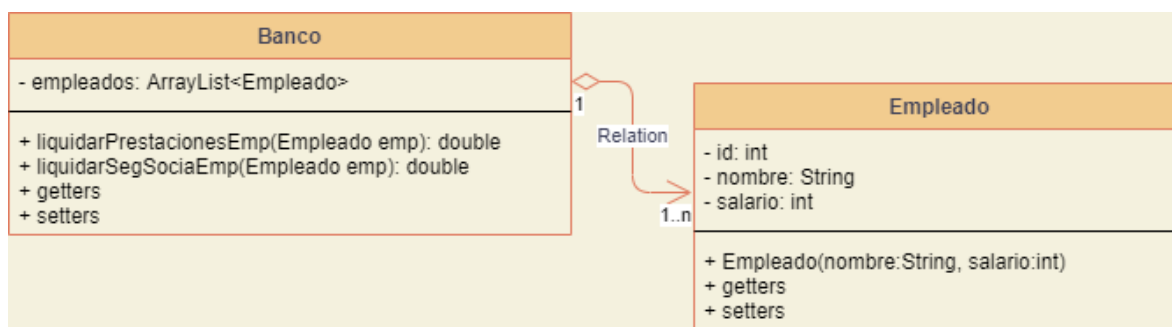
Reto 2 – Banco

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:

Partiendo de la clase **Banco** del *Reto1*, refactorice sus atributos y métodos, para lograr desarrollar una nueva solución que presente la estructura del siguiente diagrama UML



Como se logra observar en el diagrama, la clase **Banco** se compone ahora por una lista de empleados y presenta dos métodos que efectúan la liquidación de las prestaciones sociales y la seguridad social de un **Empleado**.

De la liquidación de las prestaciones sociales se toman los porcentajes de prima de servicios (8.33% del salario de un empleado), cesantías (8.33% del salario de un empleado), intereses sobre las cesantías (12% del valor de las cesantías) y vacaciones (4,16% del salario de un empleado), y se suman para obtener el valor a liquidar.

Por otra parte, la liquidación de la seguridad social se calcula sumando los conceptos de salud (8.5% del salario de un empleado), pensión (12% del salario de un empleado) y ARL (0.522% del salario de un empleado).



Reto:

Nota: Reestructure las clases y métodos del proyecto **Banco** implementadas en el *Reto1*, y realice los cambios pertinentes (refactorización) según los requerimientos del reto actual.

Debe de implementar las clases **Banco** y **Empleado** como se muestran en el diagrama de clases con sus respectivos métodos **get** y **set**. Para el desarrollo de los métodos **liquidarPrestacionesEmp** y **liquidarSegSocialEmp**, tome la información proporcionada previamente en el contexto del reto.

Un ejemplo de la estructura el código de las clases a implementar se muestra a continuación:

```
public class Banco {  
    ArrayList<Empleado> empleados = new ArrayList<>();  
  
    public static double liquidarPrestacionesEmp(Empleado empleado){  
    }  
  
    public static double liquidarSegSocialEmp(Empleado empleado){  
    }  
}
```

```
public class Empleado {  
    private int id;  
    private String nombre;  
    private int salario;  
  
    public Empleado(String nombre, int salario) {  
    }  
}
```



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			
<i>liquidarPrestacionesEmp</i>	<table><tr><td>Empleado</td></tr><tr><td>Nombre: Manuel</td></tr><tr><td>Salario: \$2'500.000</td></tr></table>	Empleado	Nombre: Manuel	Salario: \$2'500.000	Valor: 545.490,00
Empleado					
Nombre: Manuel					
Salario: \$2'500.000					
<i>liquidarSegSocialEmp</i>	<table><tr><td>Empleado</td></tr><tr><td>Nombre: Manuel</td></tr><tr><td>Salario: \$2'500.000</td></tr></table>	Empleado	Nombre: Manuel	Salario: \$2'500.000	Valor: 525.500,00
Empleado					
Nombre: Manuel					
Salario: \$2'500.000					

Entrega:

1. Suba a la plataforma los archivos **Banco.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.