

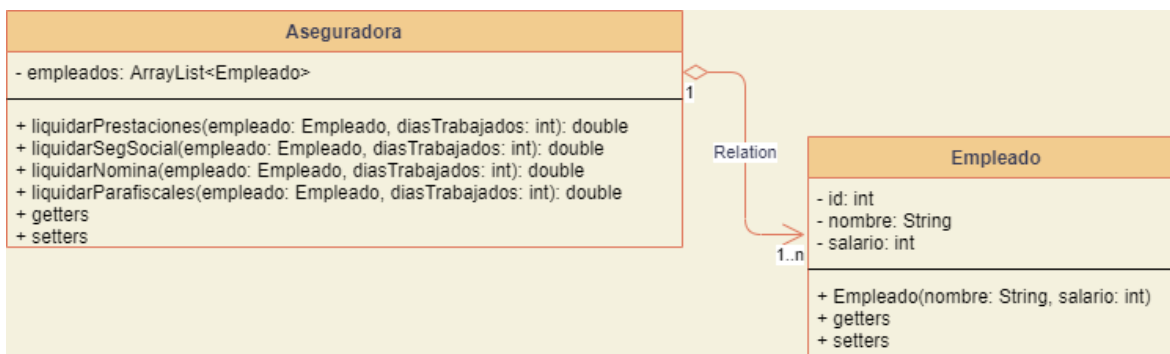
Reto 3 – Aseguradora

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:

Siguiendo la línea de trabajo de *Reto1* y *Reto2*, se desea construir un sistema de liquidación para una empresa **Aseguradora**. Por tal motivo, es necesario que el sistema adquiera una abstracción más cercana a la realidad e incorpore nuevas entidades al sistema. Realizando estas adaptaciones, el sistema en representación UML se ilustraría de la siguiente manera:



Como se logra apreciar en la imagen, el sistema se compone de las clases **Empleado** y **Aseguradora**, en esta última se conservaron los métodos implementados en los retos pasados y ahora reciben como parámetros un objeto de tipo **Empleado** y los días trabajados del mismo. De la misma manera, se han agregado dos nuevas funcionalidades que realizarán los cálculos de nómina y parafiscales.

La nómina de un empleado se obtiene a partir de su total devengado, el cual está compuesto por el salario y el auxilio de transporte (de presentarlo), y a dicho valor se le descuenta los porcentajes de salud y pensión, que corresponde al 4% del devengado (excluyendo el auxilio de transporte) y corren por parte del empleado.

Análogamente, para el cálculo de los parafiscales se deben tomar todos los conceptos que componen el total devengado (excluyendo el auxilio de transporte) y deducirle 4% de caja de compensación familiar, 3% de ICBF y 2% del SENA. La sumatoria de estos valores indicará el monto a liquidar por conceptos de parafiscales.

Nota: el auxilio de transporte tiene un valor de \$106.454 y se les otorga a aquellos empleados que devenga un salario de hasta dos veces el salario mínimo legal vigente (correspondiente a \$908.526).



Reto:

Nota: Reestructure los métodos de la clase **Aseguradora** implementados en el *Reto1* y *Reto2*, y realice los cambios pertinentes según los requerimientos del reto actual.

Se requiere que reinterprete el sistema de liquidación, agregue la clase **Empleado** dentro de su proyecto e implemente los nuevos métodos de la clase **Aseguradora** teniendo en cuenta la información brinda en el contexto del reto. Para aquellos métodos realizados en los retos pasados, reescriba su código con base a los nuevos parámetros que deben de recibir y realice el cálculo siguiendo los mismos datos usados con anterioridad.

Un ejemplo de la estructura el código de las clases **Aseguradora** y **Empleado** se muestra a continuación:

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

```
public class Aseguradora {  
  
    private ArrayList<Empleado> empleados = new ArrayList<>();  
  
    public static double liquidarPrestaciones(Empleado empleado, int diasTrabajados){  
  
    }  
    public static double liquidarSegSocial(Empleado empleado, int diasTrabajados){  
  
    }  
    public static double liquidarNomina(Empleado empleado, int diasTrabajados){  
  
    }  
    public static double liquidarParafiscales(Empleado empleado, int diasTrabajados){  
  
    }  
  
}
```



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					
liquidarPrestaciones	<table><tr><td>Empleado</td></tr><tr><td>Nombre: Sara</td></tr><tr><td>Salario: \$1'810.050</td></tr><tr><td>Fecha Inicio: 14/06/2021</td></tr><tr><td>Fecha Finalización: 13/11/2021</td></tr></table>	Empleado	Nombre: Sara	Salario: \$1'810.050	Fecha Inicio: 14/06/2021	Fecha Finalización: 13/11/2021	Valor: 2.097.605,69
Empleado							
Nombre: Sara							
Salario: \$1'810.050							
Fecha Inicio: 14/06/2021							
Fecha Finalización: 13/11/2021							
liquidarSegSocial	<table><tr><td>Empleado</td></tr><tr><td>Nombre: Sara</td></tr><tr><td>Salario: \$1'810.050</td></tr><tr><td>Fecha Inicio: 14/06/2021</td></tr><tr><td>Fecha Finalización: 13/11/2021</td></tr></table>	Empleado	Nombre: Sara	Salario: \$1'810.050	Fecha Inicio: 14/06/2021	Fecha Finalización: 13/11/2021	Valor: 2.212.444,44
Empleado							
Nombre: Sara							
Salario: \$1'810.050							
Fecha Inicio: 14/06/2021							
Fecha Finalización: 13/11/2021							
liquidarNomina	<table><tr><td>Empleado</td></tr><tr><td>Nombre: Sara</td></tr><tr><td>Salario: \$1'810.050</td></tr><tr><td>Fecha Inicio: 14/06/2021</td></tr><tr><td>Fecha Finalización: 13/11/2021</td></tr></table>	Empleado	Nombre: Sara	Salario: \$1'810.050	Fecha Inicio: 14/06/2021	Fecha Finalización: 13/11/2021	Valor: 8.976.613,33
Empleado							
Nombre: Sara							
Salario: \$1'810.050							
Fecha Inicio: 14/06/2021							
Fecha Finalización: 13/11/2021							
liquidarParafiscales	<table><tr><td>Empleado</td></tr><tr><td>Nombre: Sara</td></tr><tr><td>Salario: \$1'810.050</td></tr><tr><td>Fecha Inicio: 14/06/2021</td></tr><tr><td>Fecha Finalización: 13/11/2021</td></tr></table>	Empleado	Nombre: Sara	Salario: \$1'810.050	Fecha Inicio: 14/06/2021	Fecha Finalización: 13/11/2021	Valor: 825.382,80
Empleado							
Nombre: Sara							
Salario: \$1'810.050							
Fecha Inicio: 14/06/2021							
Fecha Finalización: 13/11/2021							



Entrega:

1. Suba a la plataforma los archivos de **Aseguradora.java** y **Empleado.java**, estos nombres deben 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.