



Reto 2 – 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:

Previamente en el *Reto1*, se tenía una aseguradora que precisaba de un sistema que le permitiera liquidar las prestaciones sociales de un empleado, tomando con base su salario y el tiempo laborado durante un periodo estipulado. Dicha aseguradora, presentaba los siguientes atributos y comportamientos:

| Aseguradora |
|---|
| + liquidarPrestaciones(salario:int, diasTrabajados:int): double + liquidarSegSocial(salario:int, diasTrabajados:int): double |

La asegura ahora necesita de un sistema que le permita, calcular la liquidación de la seguridad social de un empleado a partir de su salario y el tiempo durante el cual prestó sus servicios laborales. Lo anterior, se obtiene a partir de la suma de los conceptos que componen la seguridad social, salud (8.5%), pensión (12%) y riesgos laborales (0.522%), que se aplican sobre el valor devengado.

Reto:

Nota: Siga construyendo los métodos de la clase **Aseguradora** implementados en el *Reto1*, y realice los cambios pertinentes(refactorización) según los requerimientos del reto actual.

Siga construyendo las funcionalidades de la clase **Aseguradora**, desarrolle los métodos **liquidarPrestaciones** y **liquidarSegSocial**, para definir el costo a pagar por seguridad social de un empleado. Este método recibirá el salario y los días trabajados, y retornará el valor correspondiente.

Nota: Sí el salario devengado por un empleado es hasta dos veces el salario mínimo legal (correspondiente a \$908.526), se debe tener en cuenta el auxilio de transporte (correspondiente a \$106.454) el cálculo de las prestaciones sociales, pero **no** en el cálculo de la seguridad social.



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

```
public class Aseguradora {  
    public static double liquidarPrestaciones(int salario, int diasTrabajados){  
    }  
    public static double liquidarSegSocial(int salario, 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>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 | Salario: \$1'810.050 | Fecha Inicio: 14/06/2021 | Fecha Finalización: 13/11/2021 | Valor: 2.097.605,69 |
| Empleado | | | | | | |
| Salario: \$1'810.050 | | | | | | |
| Fecha Inicio: 14/06/2021 | | | | | | |
| Fecha Finalización: 13/11/2021 | | | | | | |
| liquidarSegSocial | <table><tr><td>Empleado</td></tr><tr><td>Salario: \$2'000.000</td></tr><tr><td>Fecha Inicio: 14/06/2021</td></tr><tr><td>Fecha Finalización: 13/11/2021</td></tr></table> | Empleado | Salario: \$2'000.000 | Fecha Inicio: 14/06/2021 | Fecha Finalización: 13/11/2021 | Valor: 1.927.910,80 |
| Empleado | | | | | | |
| Salario: \$2'000.000 | | | | | | |
| Fecha Inicio: 14/06/2021 | | | | | | |
| Fecha Finalización: 13/11/2021 | | | | | | |



| | | |
|----------------------|--------------------------------|-------------------------------|
| liquidarPrestaciones | Empleado | Valor: 2.212.444,44 |
| | Salario: \$2'000.000 | |
| | Fecha Inicio: 14/06/2021 | |
| | Fecha Finalización: 13/11/2021 | |

Entrega:

1. Suba a la plataforma un archivo con el nombre de **Aseguradora.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.