



## Reto 1 – Hospital

### 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 hospital requiere de un sistema que le permita calcular la liquidación de la seguridad social de un empleado de su nómina. La información de dicho empleado se presenta a continuación:

Nombre	Email	Tipo de contrato	Salario	Especialidad
Hugo	<a href="mailto:hugo@email.com">hugo@email.com</a>	Termino Fijo	1'200.000	Dermatología

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

Una manera alternativa de visualizar la información anterior es por medio de la notación de un diagrama UML, utilizando esta notación un empleado del hospital se representaría de la siguiente manera:

Empleado
- id: int - nombre: String - email: String - tipoContrato: String - salario: int - especialidad: String
+ Empleado(nombre:String, email:String, tipoContrato:String, salario:int, especialidad:String) + LiquidarSalud(Empleado empleado): double + getters + setters

El hospital está al tanto de que los porcentajes que componen los conceptos de seguridad social son 8.5% correspondiente a salud, 12% correspondiente a pensión y 0.522% a correspondientes a riesgos laborales para cada uno de sus empleados.



Dado que, el empleado devenga un salario mensual de 1'200.000, se le debe otorgar el auxilio de transporte. Este auxilio equivale a \$106.454 y aunque hace parte de sus ingresos, para efectos de la seguridad social **no se tiene en cuenta**.

### Reto:

Desarrolle una solución que, por medio de los elementos del paradigma de la programación orientada a objetos, realice el cálculo de la liquidación de la seguridad social de un empleado. Para ello, debe de efectuar la sumatoria de cada uno de los conceptos sobre el total devengado del empleado.

En el siguiente snippet de código se ejemplifica un esbozo de cómo debería de implementarse esta solución.

```
public class Empleado {
    private int id;
    private String nombre;
    private String email;
    private String tipoContrato;
    private int salario;
    private String especialidad;

    public Empleado(String nombre, String email, String tipoContrato, int salario, String especialidad){
    }

    public double liquidarSalud(Empleado empleado) {
    }

}
```



## 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.						<b>Valor:</b> 252264
	Nombre	Email	Tipo de contrato	Salario	Especialidad	
	Hugo	<a href="mailto:hugo@email.com">hugo@email.com</a>	Termino Fijo	1'200.000	Dermatología	

## 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.