

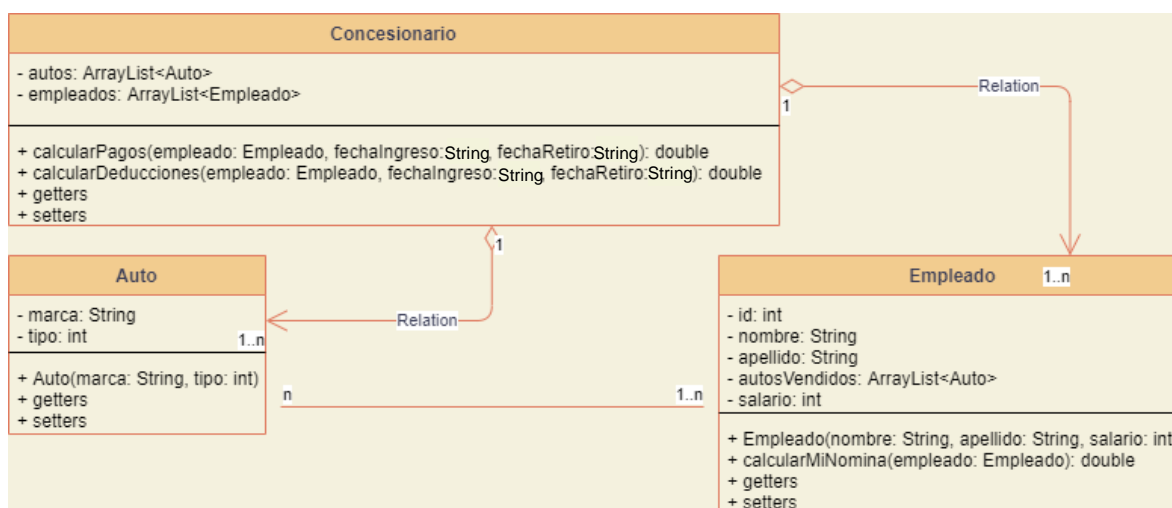
## Reto 3 – Concesionario

### 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 concesionario **CARMAX S.A.S** ha tenido un crecimiento empresarial significativo, por tal motivo, su sistema de liquidación de nómina ha adquirido una reestructuración, que le permitirá al concesionario no realizar únicamente liquidaciones, sino que también, podrá calcular otros pagos que guardan gran relación y complementan la nómina. La representación del nuevo sistema en notación de diagrama de clase UML se verá de la siguiente manera:



Cómo se observa en la imagen, las entidades **Empleado** y **Auto**, presenta el mismo comportamiento que el *Reto2*, sus atributos y métodos siguen la misma lógica de negocio del concesionario. En donde, los autos están representados de forma numérica, (1) para deportivos y camionetas, (2) para autos eléctricos y (3) para autos tradicionales, y la nómina de los empleados depende de salario y el número de autos vendidos.

En adición a lo anterior, ha surgido la entidad **Concesionario**, que estará encargada de llevar el listado de autos y empleados de la empresa, así como también, realizar el cálculo de todos los pagos de un empleado en un tiempo determinado e igualmente el cálculo del total de deducciones del mismo.



## Reto:

**Nota:** Por medio la información brindada en el contexto del reto actual, sigue desarrollando la estructura de tu proyecto **Concesionario** e integra las nuevas funcionalidades a las desarrolladas en los **retos 1 y 2**.

Se requiere que implemente los métodos **calcularPagos** y **calcularDeducciones** de la clase **Concesionario**, para el primero de ellos, realice la sumatoria de todos los pagos durante un tiempo de trabajo determinado, teniendo en cuenta, que un pago está compuesto por el salario del empleado y los bonos de comisión recibidos por los autos vendidos. Por otra parte, para realizar el **“calcularDeducciones”** realice la sumatoria de todas las deducciones por concepto de seguridad social que tiene el empleado durante el periodo que trabajó.

## Casos de prueba:

Para verificar el correcto funcionamiento de su programa se sugiere que tenga en consideración los siguientes casos de prueba

# CASO DE PRUEBA	DATOS DE ENTRADA				SALIDA ESPERADA																
calcularPagos	<table><tr><th>Empleado</th><th>Autos Venidos periodo</th><th>Fecha Ingreso</th><th>Fecha Retiro</th></tr><tr><td>Nombre: Diego</td><td>Auto Eléctrico</td><td>2020-01-01</td><td>2020-12-30</td></tr><tr><td>Apellido: Moreno</td><td>Camioneta</td><td>-</td><td>-</td></tr><tr><td>Salario: \$1'875.000</td><td>Deportivo</td><td>-</td><td>-</td></tr></table>				Empleado	Autos Venidos periodo	Fecha Ingreso	Fecha Retiro	Nombre: Diego	Auto Eléctrico	2020-01-01	2020-12-30	Apellido: Moreno	Camioneta	-	-	Salario: \$1'875.000	Deportivo	-	-	Valor: 24.437.500,00
	Empleado	Autos Venidos periodo	Fecha Ingreso	Fecha Retiro																	
	Nombre: Diego	Auto Eléctrico	2020-01-01	2020-12-30																	
	Apellido: Moreno	Camioneta	-	-																	
	Salario: \$1'875.000	Deportivo	-	-																	
NOTA: El formato de aceptación de las fechas es AAAA-MM-dd.																					



calcularDeducciones

Empleado	Autos Venidos	Fecha Ingreso	Fecha Retiro
Nombre: Diego	Auto Eléctrico	2020-01-01	2020-12-30
Apellido: Moreno	Camioneta	-	-
Salario: \$1'875.000	Deportivo	-	-

Valor:  
1.955.000,00

**NOTA:** El formato de aceptación de fecha es AAAA-MM-dd.

### Entrega:

1. Suba a la plataforma los archivos **Auto.java**, **Empleado.java** y **Concesionario.java**, recuerde que estos nombres deben ser respetarse, de lo contrario no se calificará el reto.
2. **Importante:** Los métodos deben de llamarse **exactamente igual** a como se muestran en la firma de los siguientes snippet de código:

```
public class Auto {  
    private String marca;  
    private int tipo;  
  
    public Auto(String marca, int tipo) {  
  
    }  
  
}
```



```
public class Empleado {  
    private int id;  
    private String nombre;  
    private String apellido;  
    private ArrayList<Auto> autos = new ArrayList<>();  
    private int salario;  
  
    public Empleado(String nombre, String apellido, int salario){  
    }  
  
    public static double calcularMiNomina(Empleado empleado){  
    }  
}
```

```
public class Concesionario {  
    private ArrayList<Auto> autos = new ArrayList<>();  
    private ArrayList<Empleado> empleados = new ArrayList<>();  
  
    public static double calcularPagos(Empleado empleado, String ingreso, String retiro){  
    }  
  
    public static double calcularDeducciones(Empleado empleado, String ingreso, String retiro){  
    }  
}
```