

## Programación II

### Asociaciones

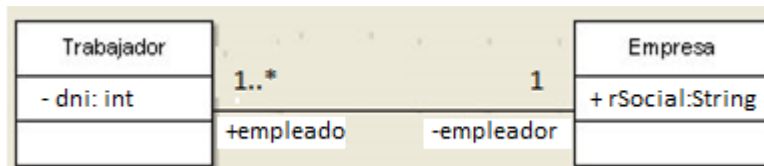
Alumno:

Entrega: 21/4/2025

Ejercicio 1: Suponiendo que se han creado todos los objetos necesarios, con sus atributos para las asociaciones, escriba la línea de código necesaria para mostrar desde un objeto de la relación, una característica del objeto del atributo de relación.

Imagine la relación implementada usando arreglos.

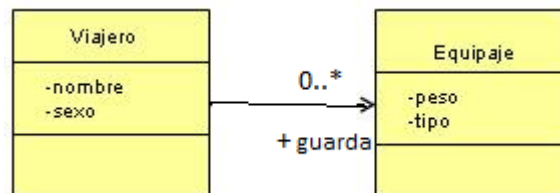
a)



```

public static void main(String[] args) {
    Empleado e = new Empleado(12356);
    Empresa em = new Empresa("Osep");
    em.empleado[0] = e;
    System.out.println(em.empleado[0].getDni());
}
    
```

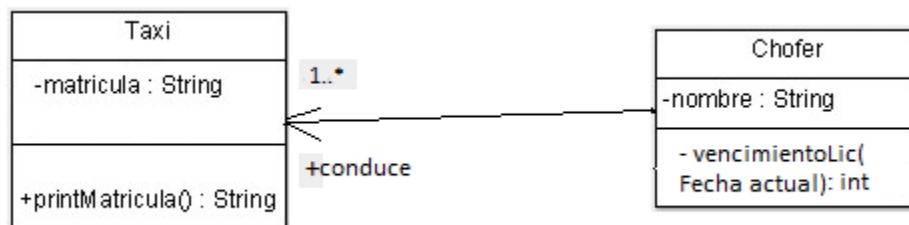
b)



```

public static void main(String[] args) {
    Viajero viajero = new Viajero("Nombre");
    Equipaje equipaje = new Equipaje(150, "Ropa");
    viajero.equipaje[0] = equipaje;
    viajero.equipaje[0].getPeso();
}
    
```

c)



```

public static void main(String[] args) {
    Taxi taxi = new Taxi("ABC999");
    Chofer chofer = new Chofer("Cristian", taxi);
    System.out.println(chofer.taxis[0].printMatricula());
}
    
```

## Programación II

### Asociaciones

Ejercicio 2: Observando los ejemplos del ejercicio anterior, indique cuáles de las relaciones son opcionales y cuáles son obligatorias.

Relacion Trabajador-Empresa: Obligatoria

Relacion Viajero-Equipaje: Opcional

Relacion Chofer-Taxi: Obligatoria

Ejercicio 3: Codifique los ejemplos y complete cada **atributo de relación**, creando objetos de las clases relacionadas:

a) Trabajador, relacionado con 1 objeto Empresa.

```
Empleado e = new Empleado(12356);
Empresa em = new Empresa("Osep");
em.empleado[0] = e;
```

b) Viajero, relacionado con 2 objetos Equipaje.

```
Viajero viajero = new Viajero("Nombre");
Equipaje equipaje1 = new Equipaje(150, "Ropa");
Equipaje equipaje2 = new Equipaje(200, "Electronico");
viajero.equipoaje[0] = equipaje1;
viajero.equipoaje[1] = equipaje2;
```

c) Chofer, relacionado con 3 objetos Taxi.

```
class Chofer{
    private String nombre;
    public ArrayList<Taxi> taxis = new ArrayList();
    public Chofer(String nombre) {
        this.nombre = nombre;
    }
    public void agregarTaxi(Taxi taxi){
        this.taxis.add(taxi);
    }
}
```

```
public static void main(String[] args) {
    Taxi taxi1 = new Taxi("ABC999");
    Taxi taxi2 = new Taxi("CCC888");
    Taxi taxi3 = new Taxi("BB777");
    Chofer chofer = new Chofer("Cristian");
    chofer.agregarTaxi(taxi1);
    chofer.agregarTaxi(taxi2);
    chofer.agregarTaxi(taxi3);
}
```

## Programación II

### Asociaciones

Ejercicio 4: Utilizando lo creado en el punto anterior, visualice el atributo indicado:

- a) Desde un objeto de Trabajador, la razón social del objeto empleador.

```
public static void main(String[] args) {  
    Empleado e = new Empleado(12356);  
    Empresa em = new Empresa("Osep");  
    em.empleado[0] = e;  
    System.out.println(em.rSocial);  
}
```

- b) Desde el objeto Viajero, el peso de todos los equipajes.

```
Viajero viajero = new Viajero("Nombre");  
Equipaje equipaje1 = new Equipaje(150, "Ropa");  
Equipaje equipaje2 = new Equipaje(200, "Electronico");  
viajero.equipaje[0] = equipaje1;  
viajero.equipaje[1] = equipaje2;  
System.out.println(viajero.equipaje[0].getPeso());  
System.out.println(viajero.equipaje[1].getPeso());
```

- c) Desde el objeto Chofer, la matrícula de los tres Taxis.

```
public static void main(String[] args) {  
    Taxi taxi1 = new Taxi("ABC999");  
    Taxi taxi2 = new Taxi("CCC888");  
    Taxi taxi3 = new Taxi("BB777");  
    Chofer chofer = new Chofer("Cristian");  
    chofer.agregarTaxi(taxi1);  
    chofer.agregarTaxi(taxi2);  
    chofer.agregarTaxi(taxi3);  
    for(Taxi taxi: chofer.taxis){  
        taxi.printMatricula();  
    }  
}
```

- d) Desde un objeto Empresa, conocer los dni de los empleados no nulos (Null).

No me deja verificar si el empleado.getDni es null. Así que lo puse como cero.

```
public static void main(String[] args) {  
    Empleado empleado1 = new Empleado(12356);  
    Empleado empleado2 = new Empleado(56789);  
    Empleado empleado3 = new Empleado(0);  
    Empresa empresa = new Empresa("Osep");  
    empresa.agregarEmpleado(empleado1);  
    empresa.agregarEmpleado(empleado2);  
    for (Empleado empleado: empresa.empleados){  
        if(empleado.getDni() != 0)  
            System.out.println(empleado.getDni());  
    }  
}
```