

**Nombre del estudiante:** Arley Ivan Salgado Mañay

**Asignatura:** Programación estructurada y funcional

**Fecha:** 29/06/2025

**Título:** Evaluación en Contacto con el Docente

## 1 Introducción

Dentro del presente trabajo se muestra una simulación del funcionamiento de un peaje, dentro de esta se va a tener un registro de los vehículos y el monto a recaudado. Se usan clases orientadas a objetos en las cuales van a representar carros, motos y camiones aplicando conocimientos de herencia y polimorfismo. Dentro de este trabajo la estación es la que lleva el control de los valores recaudados y muestra el resumen de los vehículos que han transitado.

## 2 Clases

### 2.1 Vehículo

Se crea una clase publica ya que es la principal para el resto de las clases, se crea el atributo *placa* que es a la que va a acceder todas las clases secundarias, por esta razón es que se la coloca como *protected*.

```
1 public class Vehiculo {
2     protected String placa;
```

Se crea el constructor que es donde inicializa la placa del vehículo.

```
4     public Vehiculo(String placa) {
5         this.placa = placa;
6     }
```

El objetivo de *getPlaca* es devolver la placa y el método *imprimir* es utilizado para que se muestre la información del vehículo.

```
8     public String getPlaca() {
9         return placa;
10    }
11
12    public void imprimir() {
13        System.out.println("Placa: " + placa);
14    }
15 }
```

### 2.2 Carro

Se crea una clase que es hereda información de vehículo y se coloca el valor del peaje para los carros que se le coloca como 2.

```
1 public class Carro extends Vehiculo {
2     private int valorPeaje = 2;
```

Se llama al constructor de la clase vehículo

```
4 public Carro(String placa) {
5     super(placa);
```

Se colocan 2 métodos para el peaje del carro, el primero sirve para poder obtener el peaje y el segundo para poder modificar el valor de este.

```
8 public int getValorPeaje() {
9     return valorPeaje;
10 }
11
12 public void setValorPeaje(int valorPeaje) {
13     this.valorPeaje = valorPeaje;
14 }
```

Se imprime la información que es sobre un carro.

```
16 @Override
17 public void imprimir() {
18     System.out.println("Carro - Placa: " + placa + ", Peaje: $" + valorPeaje);
19 }
20 }
```

### 2.3 Moto

Al igual que en carro se crea la clase que también hereda de la clase principal que es vehículo, a diferencia de la clase carro esta clase tiene el peaje en 1 y no en 2.

```
1 public class Moto extends Vehiculo {
2     private int valorPeaje = 1;
```

Ya que son clases similares en este caso se realiza lo mismo que con carro, pero adaptándolo para motos. Se llama al constructor de vehículo, se crean métodos para modificar y obtener el valor del peaje y al final que se muestre la información sobre la moto.

```
4 public Moto(String placa) {
5     super(placa);
6 }
7
8 public int getValorPeaje() {
9     return valorPeaje;
10 }
11
12 public void setValorPeaje(int valorPeaje) {
13     this.valorPeaje = valorPeaje;
14 }
15
16 @Override
17 public void imprimir() {
18     System.out.println("Moto - Placa: " + placa +
19         ", Peaje: $" + valorPeaje);
20 }
21 }
```

### 2.4 Camión

En esta clase igualmente hereda de la clase vehículo, se diferencia a las clases de carro o en moto de que esta cobra el peaje dependiendo el número de ejes, de ahí tiene el mismo valor de peaje que moto.

```
1 public class Camion extends Vehiculo {
2     private int valorPeajeEje = 1;
3     private final int numeroEjes;
```

Tal como se menciona en la clase moto, el funcionamiento es el mismo, pero en esta ocasión se añade que el valor del paje se es multiplicado por el número de ejes.

```
5 public Camion(String placa, int numeroEjes) {
6     super(placa);
7     this.numeroEjes = numeroEjes;
8 }
9
10 public int getValorPeajeEje() {
11     return valorPeajeEje;
12 }
13
14 public void setValorPeajeEje(int valorPeajeEje) {
15     this.valorPeajeEje = valorPeajeEje;
16 }
17
18 public int getNumeroEjes() {
19     return numeroEjes;
20 }
21
22 @Override
23 public void imprimir() {
24     System.out.println("Camión - Placa: " + placa + ", Ejes: " +
25         numeroEjes + ", Peaje: $" + (valorPeajeEje * numeroEjes));
26 }
27 }
```

## 2.5 Peaje

Se crea una clase peaje que consta con un nombre y una ubicación.

```
J Peaje.java > ...
1 import java.util.ArrayList;
2
3 public class Peaje {
4     private String nombre;
5     private String departamento;
```

Se crean variables que representan los contadores de los vehículos, el total de dinero que ha sido recaudado y un listado en el que se va a almacenar los vehículos que pasaron por el peaje.

```
6 private int totalPeaje = 0;
7 private int totalCamiones = 0;
8 private int totalMotos = 0;
9 private int totalCarros = 0;
10
11 private ArrayList<Vehiculo> vehiculos = new ArrayList<>();
```

Se crea el constructor de la clase para poder iniciar con los datos del peaje.

```
13 public Peaje(String nombre, String departamento) {
14     this.nombre = nombre;
15     this.departamento = departamento;
16 }
```

Al igual que en las anteriores clases que crearon 2 métodos para poder acceder y modificar los valores del peaje, en esta clase se crean los mismos métodos, pero para acceder o modificar el nombre y el departamento.

```

18  ✓ public String getNombre() {
19      return nombre;
20  }
21
22  ✓ public void setNombre(String nombre) {
23      this.nombre = nombre;
24  }
25
26  ✓ public String getDepartamento() {
27      return departamento;
28  }
29
30  ✓ public void setDepartamento(String departamento) {
31      this.departamento = departamento;
32  }

```

Se crea un método que permite mantener un registro de los vehículos que pasaron por el peaje y se utiliza un bucle if para poder condicionar el tipo de vehículo que pasó y de esta manera actualizar los contadores.

```

34      public void añadirVehiculo(Vehiculo vehiculo) {
35          int valor = calcularPeaje(vehiculo);
36          totalPeaje += valor;
37
38          if (vehiculo instanceof Carro) {
39              totalCarros++;
40          } else if (vehiculo instanceof Moto) {
41              totalMotos++;
42          } else if (vehiculo instanceof Camion) {
43              totalCamiones++;
44          }
45
46          vehiculos.add(vehiculo);
47      }

```

Usa el mismo mecanismo que el punto anterior, se crea un bucle if por el cual se va a identificar el tipo de vehículo y según eso se va a proceder con el pago del peaje.

```

49  ✓ public int calcularPeaje(Vehiculo vehiculo) {
50  ✓     if (vehiculo instanceof Carro) {
51      return ((Carro) vehiculo).getValorPeaje();
52  ✓     } else if (vehiculo instanceof Moto) {
53      return ((Moto) vehiculo).getValorPeaje();
54  ✓     } else if (vehiculo instanceof Camion) {
55      Camion c = (Camion) vehiculo;
56      return c.getValorPeajeEje() * c.getNumeroEjes();
57  ✓     } else {
58      return 0;
59     }
60 }

```

Primero se va a mostrar los vehículos que están registrados y al final muestra el resumen de peaje como el total de los carros, motos, camiones y el valor total recaudado.

```

62  public void imprimir() {
63      System.out.println("=== Peaje: " + nombre + " - " + departamento + " ===");
64      System.out.println("Vehículos registrados:");
65      for (Vehiculo v : vehiculos) {
66          v.imprimir();
67      }
68      System.out.println("\n--- Estadísticas ---");
69      System.out.println("Total Carros: " + totalCarros);
70      System.out.println("Total Motos: " + totalMotos);
71      System.out.println("Total Camiones: " + totalCamiones);
72      System.out.println("Total Recaudado: $" + totalPeaje);
73  }
74  }

```

## 2.6 Sistema de peaje

Se crea la clase en la que va a funcionar el main para poder ejecutar el programa. Dentro de esta clase se va a dar nombre al peaje y al departamento que pertenece.

```

1  public class SistemaPeaje {
    Run | Debug | Run main | Debug main
2  public static void main(String[] args) {
3      Peaje peaje = new Peaje(nombre:"Peaje Sur", departamento:"Guayas");

```

Se crean 3 objetos los cuales están guardados en variables de tipo vehículo ya que es la clase principal (padre).

```

5      Vehiculo v1 = new Carro(placa:"ABC123");
6      Vehiculo v2 = new Moto(placa:"XYZ789");
7      Vehiculo v3 = new Camion(placa:"DEF456", numeroEjes:3);
8

```

Por medio de AñadirVehiculo se van añadiendo los vehículos que pasaron por el peaje y finalmente se llamada al método imprimir() que pertenece a la clase peaje en la que se va a mostrar las estadísticas del número de vehículos que paso por cada tipo de vehículo y el monto recaudado en total.

```

9      peaje.añadirVehiculo(v1);
10     peaje.añadirVehiculo(v2);
11     peaje.añadirVehiculo(v3);
12
13     peaje.imprimir();
14 }
15 }

```

## 3 Códigos

Se adjunta enlace de GitHub en donde se encuentran las 6 clases creadas

<https://github.com/arsalgadoma/Programaci-nEstructuradaYFuncional.git>

## 4 Resultados

Se muestra los vehículos que transitaron, el valor que tienen que cancelar cada uno, en la estadística se tiene el valor el número de cada vehículo que transitó y el total recaudado entre todos los vehículos.

```
=== Peaje: Peaje Sur - Guayas ===  
Vehículos registrados:  
Carro - Placa: ABC123, Peaje: $2  
Moto - Placa: XYZ789, Peaje: $1  
Camión - Placa: DEF456, Ejes: 3, Peaje: $3  
  
--- Estadísticas ---  
Total Carros: 1  
Total Motos: 1  
Total Camiones: 1  
Total Recaudado: $6
```