**Contexto**  
  
Se requiere desarrollar un programa que modele un sistema de peaje. Las estaciones de peaje tienen un nombre, departamento en el que están ubicadas y un valor numérico que representa el valor total de peaje recolectado. Los vehículos que llegan a un peaje tienen una placa, el número de peajes que ha pagado, y una capacidad de carga en toneladas (aplicable solo para camiones).

El sistema cobra diferentes tarifas de peaje dependiendo del tipo de vehículo y de condiciones adicionales:

* **Carro**: el valor base del peaje es de $10,000, pero si es un carro eléctrico, se aplica un descuento del 20%. Además, si es un vehículo de servicio público, el valor del peaje se incrementa en un 15%.
* **Moto**: el valor base del peaje es de $5,000, pero si la moto tiene un cilindraje mayor a 200 cc, se aplica un recargo adicional de $2,000.
* **Camión**: el valor del peaje depende de los ejes y la carga. Se cobra un valor base de $7,000 por eje. Si la capacidad de carga del camión supera las 10 toneladas, se aplica un recargo del 10% sobre el total calculado por eje.

Las estaciones de peaje deben calcular el valor del peaje de cada vehículo que llegue y registrar el total de peajes recolectados. Al finalizar, el sistema debe imprimir en pantalla un listado de los vehículos que llegaron al peaje, el tipo de vehículo, el cálculo detallado de cada peaje y el total acumulado.

El conductor tiene los siguientes atributos: nombre, apellidos, número de documento de identidad y fecha de nacimiento y la lista de vehículos asignados.

Para optimizar el proceso de cobro, la empresa cuenta con recaudadores que ofrecen el servicio de cobro en la estación de peaje. Los recaudadores tienen nombre, apellidos, número de documento de identidad, fecha de nacimiento, y sueldo mensual.

**Métodos requeridos:**

* Asignar un vehículo a una persona, permitiendo que una persona tenga múltiples vehículos de diferentes tipos.
* Imprimir en pantalla un listado de los vehículos que llegaron al peaje, el tipo de vehículo, el cálculo detallado de cada peaje y el total acumulado.
* Se requiere implementar un método en Vehiculo que retorne una cadena descriptiva con la información del vehículo, incluyendo tipo, placa y detalles particulares como cilindraje o número de ejes y capacidad de carga. Esta descripción se usará para generar reportes de control.
* Consultar el total de dinero pagado en peajes por cada vehículo que una persona tiene asignado, detallando el cálculo realizado.
* Devolver una lista de vehículos de un conductor que coincidan con un tipo específico ("Carro", "Moto", "Camion").
* Calcular el valor del peaje de un vehículo X, actualiza el valor total recaudado y guarda el registro del paso del vehículo.
* Busca un recaudador por nombre completo (nombre + apellidos) ignorando mayúsculas y espacios adicionales.
* La gerencia requiere identificar los conductores que tienen al menos un camión con capacidad de carga mayor a 10 toneladas. Se debe implementar un método que devuelva una lista con estos conductores, permitiendo tomar decisiones sobre tarifas o restricciones de circulación para vehículos de alto tonelaje.

**Abstracción**

**¿Qué se solicita finalmente? (problema)**

Se requiere desarrollar un sistema de gestión de peajes en Java que modele estaciones de peaje, vehículos (carros, motos, camiones), conductores y recaudadores, con las respectivas reglas de cobro y funciones administrativas.

**¿Qué información es relevante dado el problema anterior?**

* **Estación de peaje**: nombre, departamento, valor total recaudado.
* **Vehículos**: placa, peajes pagados, tipo específico (carro, moto, camión) con atributos particulares.
* **Conductores**: nombre, apellidos, documento, fecha de nacimiento, vehículos asignados.
* **Recaudadores**: nombre, apellidos, documento, fecha de nacimiento, sueldo mensual.
* Tarifas y condiciones de cobro específicas según tipo de vehículo.

**¿Cómo se agrupa la información relevante?**

* Clases principales:
  + EstacionPeaje
  + Vehiculo (abstracta o con herencia: Carro, Moto, Camion)
  + Conductor
  + Recaudador
* Enums para tipo de vehículo, etc.
* Interfaces para comportamientos comunes, si aplica.
* Registro del cobro y paso por peaje como clases auxiliares.

**¿Qué funcionalidades se solicitan?**

1. Asignar vehículos a conductores.
2. Imprimir listado de vehículos con detalles de cobro y total acumulado.
3. Método descriptivo en vehículo.
4. Consultar total pagado por cada vehículo de un conductor.
5. Listar vehículos por tipo.
6. Calcular y registrar peaje de un vehículo.
7. Buscar recaudador por nombre completo (ignorando mayúsculas y espacios).
8. Listar conductores con camiones > 10 toneladas.

**Descomposición**

**¿Cómo se distribuyen las funcionalidades?**

* En clases:
  + Vehiculo: método calcularPeaje(), descripcion().
  + Conductor: métodos para consultar pagos, filtrar vehículos por tipo.
  + EstacionPeaje: registrar cobros, listado total.
  + Recaudador: búsqueda por nombre.
* Cada funcionalidad corresponde a un método específico de una o varias clases.

**¿Qué debo hacer para probar las funcionalidades?**

* Crear objetos de prueba (conductores, vehículos, peajes).
* Ejecutar métodos clave:
  + Asignación de vehículos.
  + Calcular peajes con condiciones (eléctrico, servicio público, cilindraje, ejes, tonelaje).
  + Buscar recaudador con nombres variados.
  + Generar salidas esperadas (reportes y listados).
* Usar assert o pruebas manuales/imprimir resultados esperados.

**Reconocimiento de patrones**

**¿Qué puedo reutilizar de la solución de otros problemas?**

* Patrón **herencia y polimorfismo** para manejar tipos de vehículos con cálculo diferente.
* Patrón **composición** para asociar vehículos a conductores.
* Uso de **colecciones** (ArrayList, HashMap) para almacenar registros.
* Comparación de cadenas ignorando mayúsculas y espacios: normalizar nombres con .toLowerCase().replace(" ", "").
* Interfaz Descriptible para métodos descripcion() comunes.

**Codificación**

**¿Cómo pruebo la solución en Java?**

**¿Cómo escribo la solución en Java?**