**Abstracción**

**¿Que se solicita finalmente?**

Realizar un programa de un sistema de peaje llevando el control del costo de los vehículos que se transportan y el costo de dichos vehículos de acuerdo al tipo de vehículo

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

EstacionPeaje Vehiculo <abstract> VehiculoCarro

-nombre \*placa -tipoCarro

-departamento \*numeroPeajesPagados

-totalPeajeRecolectado

TipoCarro <<Enum>> VehiculoMoto VehiculoCamión

Eléctrico -cilindraje -capacidadCarga

Serviciopublico -numeroEjes

Persona <<abstract>> Recaudador

-nombre -sueldoMensual

-apellido

-id

-fechaNacimiento

Conductor <<record>>

IDescripcion <<interface>>

-informacionGeneralVehiculo

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

|  |
| --- |
| **EstacionPeaje** |
| -String nombre  -String departamento  -double totalPeajeRecolectado |

|  |
| --- |
| **Vehiculo**  **<<abstract>>** |
| \*String placa  \*int numeroPeajesPagados |

|  |
| --- |
| **VehiculoCarro** |
| -TipoCarro tipoCarro |

|  |
| --- |
| **TipoCarro**  **<<Enum>>** |
| ELECTRICO  SERVICIOPUBLICO |

|  |
| --- |
| **VehiculoMoto** |
| -int cilindraje |

|  |
| --- |
| **VehiculoCamion** |
| -double capacidadCarga  -int numeroEjes |

|  |
| --- |
| **Persona**  **<<abstract>>** |
| -String nombre  -String apellido  -String id  -LocalDate fechaNacimiento |

|  |
| --- |
| **Recaudador** |
| -double sueldoMensual |

|  |
| --- |
| **Conductor**  **<<record>>** |
|  |

|  |
| --- |
| **IDescripcion**  **<<interface>>** |
| +String informacionGeneralVehiculo |

**¿Qué funcionalidades se solicitan finalmente?**

Realizar un sistema de control de peajes con métodos que ayuden a controlar la información requerida

**Descomposición**

**¿Cómo se distribuyen las funcionalidades?**

En la carpeta model del repositorio se encuentra el diagrama de clases

**Codificación**

**¿Como escribo la solución en java?**

En la carpeta code está el código generado con todas las clases y métodos hechos

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

|  |  |  |
| --- | --- | --- |
| **Prueba** | **Entrada de datos** | **Salida** |
| asssertEquals(7000, tarifaCalculada) | Debe validarse de que el método para calcular la tarifa en una moto sea igual a la calculada | Lanzar true; |
| assertnotNull(ListEstacion  peajes()) | Debe validarse de que la lista de estaciones  De peajes no deba ser nula | Pasar la prueba y lanzar true |

**Reconocimiento de patrones**

**¿Qué puedo reutilizar?**

No hay nada por reutilizar

**Codificación**

**¿Cómo pruebo la solución en java?**

En la carpeta code en la parte de test están las pruebas del código