**CÓDIGO DIAGRAMA DE CLASES *plantUML***

Las clases se definen con **class** seguido del nombre de la clase entre llaves. Los atributos se definen dentro de la clase utilizando el signo + para indicar atributos públicos, # para atributos protegidos y - para atributos privados. Los métodos se definen de manera similar a los atributos, con el prefijo +, # o - seguido del nombre del método y los parámetros si los tiene.

Las relaciones entre clases se establecen con diferentes flechas:

* **Asociación**: Una relación básica entre dos clases se indica simplemente con una línea que conecta ambas clases.
* **Herencia**: Se representa con una flecha con un triángulo en el extremo. Se usa para mostrar una relación de herencia entre una clase "hija" y una clase "padre".
* **Agregación**: Se representa con un diamante en el extremo de la línea. Indica que una clase "contiene" a la otra, pero no depende exclusivamente de ella.
* **Composición**: Similar a la agregación, pero con un diamante relleno. Indica una relación más fuerte, donde una clase "posee" a la otra y su existencia está vinculada.

Ejemplo de relación básica entre clases:

@startuml

class Cliente {

- id: int

- nombre: string

+ obtenerInformacion(): string

}

class Factura {

- id: int

- total: float

+ calcularTotal(): float

}

Cliente -- Factura : Realiza

@enduml

En este ejemplo:

* Cliente y Factura son clases.
* Los atributos id y nombre son privados en Cliente, y id y total son privados en Factura.
* Los métodos obtenerInformacion() y calcularTotal() son públicos en Cliente y Factura respectivamente.
* La relación entre Cliente y Factura se establece con la asociación Realiza

**CÓDIGO DIAGRAMA DE CLASES *plantUML***

@startuml

class Ruta {

- tipoRuta: String

- puertoOrigen: String

- puertoDestino: String

- distancia: int

- duracion: int

- coste: int

- id: int

+ Ruta(tipoRuta: String, puertoOrigen: String, puertoDestino: String, distancia: int, duracion: int, coste: int, id: int)

+ getTipoRuta(): String

+ getDistancia(): int

+ getDuracion(): int

+ getCoste(): int

+ getId(): int

+ setTipoRuta(tipoRuta: String): void

+ calcularTiempoEstimadoLlegada(velocidadPromedio: double): double

+ calcularCostoTotalEstimado(costoPorKilometro: int): int

}

class Empaquetado {

- numeroContenedores: int

- peso: double

- tamano: double

+ Empaquetado(numeroContenedores: int, peso: double, tamano: double)

+ getNumeroContenedores(): int

+ setNumeroContenedores(numeroContenedores: int): void

+ getPeso(): double

+ setPeso(peso: double): void

+ getTamano(): double

+ setTamano(tamano: double): void

+ calcularDensidad(): double

}

class Facturacion {

- codigo: String

- fecha: String

- hora: String

- tipo: String

- matricula: String

- nombre: String

- dni: String

- concepto: String

- importe: double

- estado: String

+ Facturacion(codigo: String, fecha: String, hora: String, tipo: String, matricula: String, nombre: String, dni: String, concepto: String, importe: double, estado: String)

+ getCodigo(): String

+ setCodigo(codigo: String): void

+ getFecha(): String

+ setFecha(fecha: String): void

+ getHora(): String

+ setHora(hora: String): void

+ getTipo(): String

+ setTipo(tipo: String): void

+ getMatricula(): String

+ setMatricula(matricula: String): void

+ getNombre(): String

+ setNombre(nombre: String): void

+ getDni(): String

+ setDni(dni: String): void

+ getConcepto(): String

+ setConcepto(concepto: String): void

+ getImporte(): double

+ setImporte(importe: double): void

+ getEstado(): String

+ setEstado(estado: String): void

}

class Operacion {

- tipo: String

- cantidad: int

- id: int

- idActual: int

- ruta: Ruta

- empaquetado: Empaquetado

- facturacion: Facturacion

+ Operacion(tipo: String, cantidad: int, ruta: Ruta, empaquetado: Empaquetado, facturacion: Facturacion)

+ getTipo(): String

+ getCantidad(): int

+ getId(): int

+ setTipo(tipo: String): void

+ setCantidad(cantidad: int): void

+ crearOperacion(): void

+ eliminarOperacion(): void

+ modificarOperacion(): void

+ toString(): String

}

Operacion --|> Ruta

Operacion --|> Empaquetado

Operacion --|> Facturacion

@enduml