TRABAJO PRÁCTICO P.2-

Análisis

Según el enunciado, se pide modelar una estructura la cual nos permita automatizar las operaciones y consultas habituales que responden a las actividades de carga y entrega de la mercadería de una empresa que trabaja con una flota de transportes llamada “La Santafesina”.

TADs

En el detalle del problema a resolver se pueden identificar estas entidades:

- **Empresa.**
- **Transportes.**
- **Paquetes de mercadería.**
- **Depósitos.**
- **Viaje.**
- **Centro de distribución.**

Si bien el enunciado nombra a la entidad Centro de distribución, no aporta ningún dato de cómo son estos Centros de distribución no lo vamos a modelar.

TAD Empresa:

Esta empresa cuenta con depósitos que contienen paquetes de mercadería donde se carga cada transporte al momento de hacer unos viajes, en este mismo se cargan los paquetes de mercadería, estos depósitos pueden ser comunes o con refrigeración para almacenar paquetes que necesiten frio. Si son terciarizados con frigorífico tienen un valor de costo por cada 1000kg de cada camión.

Controla los viajes los transportes, los depósitos y los paquetes con su mercadería.

- Datos:
    1. Nombre: Nombre de la empresa.
    2. CUIT: Va a tener un CUIT.
    3. Transportes: Registro de todos los transportes que posee la empresa es decir la flota.
    4. Depósitos: Va a tener un conjunto de todos sus depósitos a disposición (sean tercierizados o no).

- **Acciones:**

  1. Crear la empresa con CUIT y Nombre.

  2. Obtener nombre / CUIT de la empresa.

  2. Agregar un depósito a la empresa: Recibe su capacidad máxima (en volumen), si es frigorífico o no, y si es propio sss tercerizado a si mismo se agregaría al conjunto de todos los depósitos, en caso que ya este el deposito en ese conjunto se emitirá el código de ese depósito.

  3. Incorporar un paquete a un depósito. Recibe los datos del paquete (destino, volumen, peso, si necesita frío o no), y lo pone en un depósito de acuerdo a su tipo (frigorífico o no). Devuelve booleano indicando si se pudo incorporar.

  4. Agregar un transporte a la empresa. Recibe los datos del transporte (identificación, peso y volumen de carga máximos, y si tienen cámara frigorífica o no).

  5. Asignar un destino a un transporte. Recibe la identificación del transporte, el destino y la cantidad de km del viaje. Genera excepción si el transporte ya tiene mercadería cargada.

6. Cargar un transporte con mercadería. Recibe la identificación del transporte. Devuelve el volumen cargado. Genera excepción si el transporte no tiene asignado un destino o si está en viaje.

7. Iniciar viaje. Recibe la identificación del transporte y registra el transporte como que está en viaje. Genera excepción si ya está en viaje, si no tiene destino asignado, o si no tiene ningún paquete cargado.

8. Finalizar viaje. Recibe la identificación del camión. El camión vacía su carga y blanquea su destino. Genera excepción si el camión no está en viaje.

9. Obtener el costo del viaje de un camión. Recibe la identificación del camión. Devuelve el costo. Genera excepción si el camión no está en viaje.

10. Obtener un transporte igual al que tengamos, recibe la identificación y devuelve un transporte igual al que tengamos.

11. Cambiar transporte averiado, un transporte se rompe, entonces la empresa verifica si tiene uno del mismo tipo, con la misma capacidad o mayor y que no esté en viaje, si encuentra se cambian.

TAD transporte: Estos transportes van a tener su carga máxima en peso, su capacidad máxima es decir volumen, si tienen o no refrigeración y un costo por km de viaje

- Datos:
    - Número de identificación.
    - Carga máxima del transporte: cantidad máxima que puede cargar el transporte.
    - Volumen máximo del transporte: cantidad máxima de volumen que puede entrar en el transporte.
    - Peso del transporte: Peso actual del transporte
    - Volumen del transporte: Volumen actual del transporte.
    - Refrigeración: Si cuenta con equipo de refrigeración.
    - Viaje: Que viaje está haciendo.
    - Saber si esta viaje: Ver si el transporte está en viaje o no.
    - Costo de viaje: Ver cuánto sale el costo dependiendo cada transporte.

- Acciones:

- Iniciar el viaje del transporte.
- Cargar el transporte.
- Saber si está en viaje o no dicho transporte.
- Saber si el transporte está lleno o todavía no.
- Terminar el viaje.
- Darle un destino al transporte.
- Dependiendo el camión que sea obtener un costo.
- Cambiar el cargamento a otro transporte si es que se rompe.
- Comprobar si las condiciones del transporte son correctas para un reemplazo.

Estos transportes se dividen en 2 tipos:

1. Con tráiler: Estos pueden o no tener equipo refrigerante y tienen un seguro de carga.
    - Camiones de tráiler común: Pueden utilizar en viajes con un destino no mayor de 500 km.
    - Camiones mega tráiler: Estos van a tener un costo por fijo por viaje, lo cuales se agregan al costo por km, además de la comida del conductor, estos se utilizan en destinos mayores a 500 km.

2. Fletes: Tienen un costo fijo por cada pasajero acompañante y no poseen equipo de frio.

TAD Viaje:

- Datos:
  - Destino del viaje.
  - Distancia

- Acciones:
  - Consultar todo su registro.

TAD Depósito:

- Datos:
  - Paquetes de mercadería.
  - Código del depósito.
  - Si son propios o no.
  - Cámara frigorífica.
  - Cantidad de capacidad máxima.
- Acciones:
  - Consultar estado del depósito.
  - Cargar a depósito paquetes de mercadería.
  - Vaciar depósito a micros: Cargar micros con paquetes de mercadería.
  - Depósito lleno: Saber cuándo o no está lleno del depósito.
  - Sumarle plus, le suma el plus de costo a viaje si es un deposito con cámaras frigoríficas.
  - Cargar un trasporte que aborda en el depósito.
  - Contar las cantidad de paquetes que tiene el depósito.

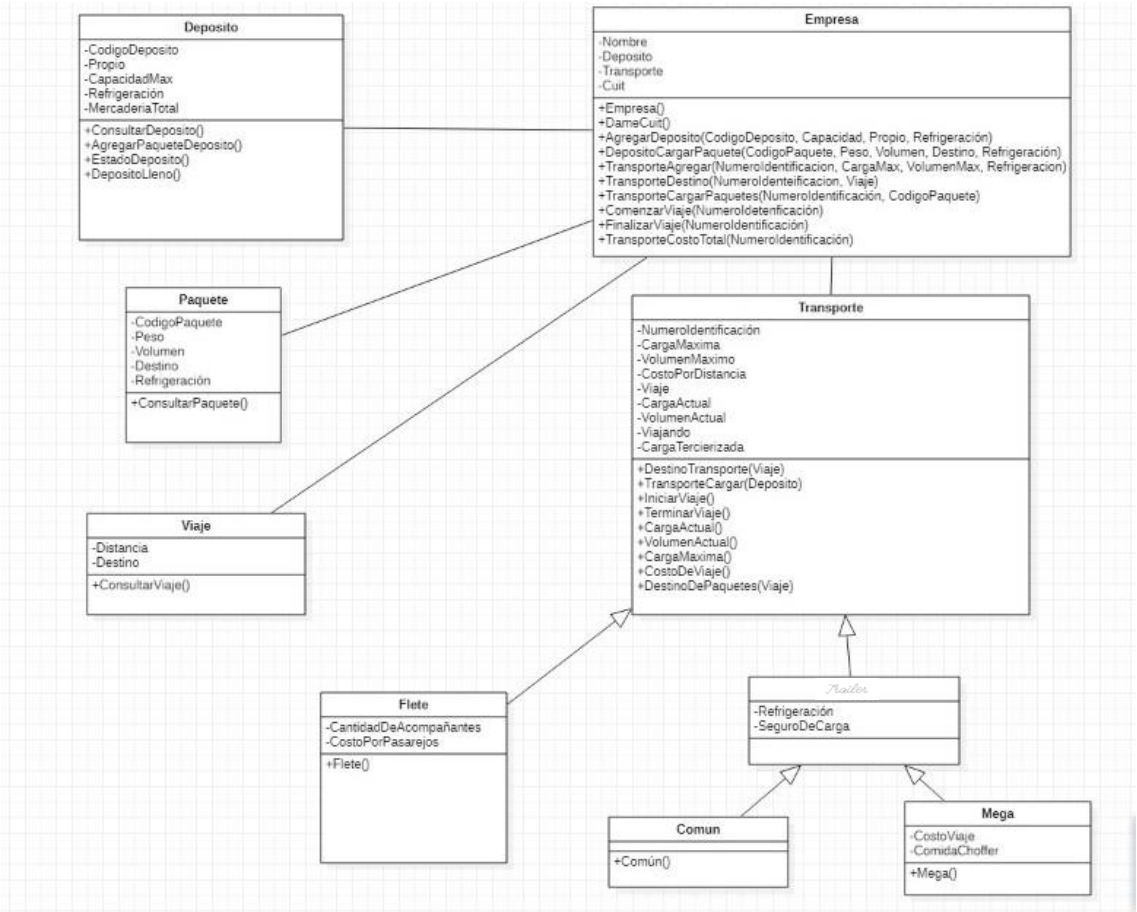
TAD Paquetes de Mercadería:

- Datos:
  - Peso: Define el peso del paquete.

- Volumen: Define el volumen del paquete.
- Destino: Define el destino del depósito a cual va dicho paquete.
- Frio: Avisa si el paquete de mercadería necesita o no frio.

- Acciones:
  - Consultar el paquete de Mercadería.

Diagrama de clases.



Interfaz:

➢ EMPRESA

- **Empresa()**

  Crear la empresa con CUIT y Nombre.

- **DameCuit()**

  Obtener nombre / CUIT de la empresa.

- **AgregarDeposito(Capacidad, Propio, Refrigeración)**

  Agrega un depósito a la empresa, requiere recibir su capacidad máxima, si es frigorífico o no, y si es propio devolviendo un numero único.

- **AgregarDepositoTercerizFrio(Capacidad,CostoXTonelada)**
  Agrega un deposito tercerizado con refrigeración a la empresa devolviendo su número único.

- **IncorporarPaquete(Peso, Volumen, Destino, Refrigeración)**
  Incorpora un paquete a un depósito requiere recibir volumen y peso del paquete si necesita frío o no y un destino, intenta acomodarlo según la necesidad del paquete e indica si lo pudo incorporar o no.

- **AgregarFlete(NumeroIdentificacion, CargaMax, Capacidad, CostoKm, Acomp, CostoPorAcomp)**

  Agrega un flete a la empresa. requiere recibir identificación, peso y volumen de carga máximas del transporte, el costo por Km, la cantidad de acompañantes y el costo por cada uno de ellos.

- **AgregarTrailer(IdTransp, CargaMax, Capacidad, Frigorifico, CostoKm, SegCarga)**
  Agrega un tráiler común a la empresa, se necesita la identificación la carga máxima, el volumen máximo, si tiene o no refrigeración, el costo por KM y el seguro de carga.
- **AgregarMegaTrailer(IdTransp, CargaMax, Capacidad, Frigorifico, CostoKm, SegCarga, CostoFijo, Comida)**
  Agrega un tráiler mega a la empresa, para este necesitamos identificación, la carga y el volumen máximo soportados, si tiene refrigeración, el costo por km, su seguro de carga, el costo fijo y la comida del chófer.
- **AgregarDestino(Destino, Distancia)**
  Se agrega un nuevo viaje.

- **AsignarDestino(NumeroIdenteificacion, Destino)**

Asigna un destino a un transporte. Requiere recibir la identificación del transporte, y el destino. Permite la asignación si el camión se encuentra vacío.

- **CargarTransporte(NumeroIdentificación)**
  Carga un transporte con mercadería. Requiere recibir la identificación del transporte.

- **ComenzarViaje(NumeroIdetenficación)**
  Inicia un viaje. Requiere el número de identificación.

- **FinalizarViaje(NumeroIdentificación)**
  Indica que el transporte llegó a destino de forma correcta y vacía la carga. Requiere el número de identificación del transporte.

- **TransporteCostoTotal(NumeroIdentificación)**
  Obtiene el costo del viaje de un camión. Requiere la identificación del camión. Devuelve el costo si el camión se encuentra viajando hacia destino.

➢ PAQUETE

- ConsultarPaquete()
  Devuelve todos los datos que caracterizan al paquete como volumen, código de identificación.

➢ TRANSPORTE

- **TransporteCargar(Paquete)**

  Carga el transporte, requiere que le pasen un paquete

- **IniciarViaje()**
  Da inicio a un viaje hacia su despacho.

- **TerminarViaje()**
  Finaliza el viaje cuando el transporte llega a destino y deja sus paquetes.

- **CargaActual()**

  Controla el estado de la carga del camión al momento de la consulta.

- **VolumenActual()**
  Controla el nivel de carga que tiene el transporte y cuanto le queda disponible según el tipo.

- **CargaMaxima()**

  Controla que el transporte no se sobre cargue.

- **CostoDeViaje()**

  Retorna el costo del viaje en relación al tipo de transporte utilizado en el mismo.
- **costoTercierizado(doublé Costo)**

➢ VIAJE

- **ConsultarViaje()**
  Retorna el viaje.

➢ FLETE

- **Flete()**
  Retorna la cantidad de acompañantes del Flete según transporte y costo por pasajero

➢ TRAILER

- **Comun()**
  - **Comun()**
    Toma viajes de hasta 500km con o sin refrigeración en relación a la necesidad establecida y la disponibilidad dada.

- **Mega()**
  - **Mega()**
    Toma viajes mayores a 500km y ajusta el monto a un adicional por kg de carga reportado.

➢ DEPOSITO

- **ConsultarDeposito()**

Retorna el estado del depósito en relación a su capacidad**.**

- **AgregarPaqueteDeposito(PaqueteNuevo)**

  Agrega un paquete nuevo si cumple con las condiciones.

- **DamePaquete(IndicePaquete)**

  Retorna el paquete ubicado en ese índice.

- **EliminarPaquete(indicePaquete)**

  Elimina paquete con el índice enviado por parámetro.

Cambios al implementar lo antes planificado:

A medida que se fue implementando lo antes planificado optamos por hacer cambios que surgieron para una mejoría en la implementación, en la interfaz ya están los métodos nuevos que decidimos agregar.

En el TAD Transporte la clase elegimos hacerla Abstracta ya que había métodos que se tenían que definir dependiendo el transporte que terminaba siendo (Flete, Tráiler Común, Tráiler Mega). Los métodos que luego se terminan en dichas clases son:

CostoDeViaje(): Dependiendo que transporte es hace una suma para el costo total diferente, es decir que si es un flete no tendría su seguro de carga, tendría que fijarse la cantidad de acompañantes y el costo por cada uno

AsignarDestino(Destino, Distancia): Lo mismo en este caso, con este método dependiendo que transporte es va a definir su distancia y destino con este método lo que hago es controlar si la distancia de viaje para el transporte pedido es el adecuado, es decir que si la distancia es mayor a 500 por ejemplo no podría ser asignado a un tráiler común.

En el TAD Empresa, se agregaron los métodos para construir cada transporte (Flete, Tráiler Común, Tráiler Mega).

El Diagrama de Clases fue modificado para que se vean los cambios que se hicieron sobre lo planificado.

Nuevo métodos agregados.

Se agregaron nuevos métodos:

Transporte:

public boolean condicionCargarTransporte(Paquete p), verifica que se cumplan todas las verificaciones para que ese paquete se pueda cargar en el transporte.

public void cambiarTransporte(Transporte reemplazo), método nuevo en el cual se manda un transporte de reemplazo para reemplazar al transporte original, en caso de cumplir con sus pautas lo cambia.

public double costoDeViajeTotal() Calcula el costo de viaje total, así retornando dicho costo.

public boolean comprobarCondiciones(Transporte reemplazo) Comprueba si se puede cambiar dicho transporte (si son iguales y el reemplazo tiene la misma o mayor capacidad del otro)

public doublé cargarTransporte(Paquete p): Se le pasa un paquete y verifica si se puede cargar, en caso de cargarlo retorna el peso cargado

Trailer:

public void cancelarSeguro(): Cancela el seguro del transporte común o del transporte mega

Fletes:

public void vaciarAcomp(): Vacia la cantidad de acompañantes del flete.

Mega:

p**ublic void vaciarCostoYComida()** Deja en 0 el costo de comida y costo por el choffer.

Empresa

public boolean reemplazoTransporte(String identificación) método nuevo en el cual se manda un transporte de reemplazo para reemplazar al transporte original, en caso de cumplir con sus pautas lo cambia.

public Transporte dameTransporte(String identificación): Retorna el transporte pasado por su identificacion.

Deposito:

public int contarPaquetes() : Cuenta la cantidad de paquetes en el depósito.

Public void cargarTransporte(Transporte trans): Carga el transporte recibido si cumple las condiciones.

COMPLEJIDAD METODO:

```
public String obtenerTransporteIgual(String idTransp) {
    if (transporte.containsKey(idTransp)) {
        Transporte t = transporte.get(idTransp);
        for (Transporte t2 : transporte.values()) {
            if (t.equals(t2) && !t.getNumeroIdentificacion().equals(t2.getNumeroIdentificacion())) {
                return t2.getNumeroIdentificacion();
            }
        }
    }
    else {
        throw new RuntimeException("No existe ese transporte");

    }
    return null;
}
```

```
public boolean equals(Object obj) {
    if (obj == null) {
        return false;
    }
    if (getClass() != obj.getClass())
        return false;
    if (obj instanceof Transporte) {
        Transporte other = (Transporte) obj;
        if (this.destino == null && other.getDestino() != null) {
            return false;

        } else if (!this.destino.equals(other.getDestino()))
            return false;

        if (this.paquete.size() != other.paquete.size()) {
            return false;
        }
        ArrayList<Paquete> listaPrimaria = new ArrayList<>();
        listaPrimaria.addAll(other.paquete);
        for (Paquete paquete : this.paquete) {
            listaPrimaria.remove(paquete);
        }
        if (listaPrimaria.size() != 0) {
            return false;
        }
    }
    return true;
}
```

Para obtener la complejidad del método obtenerTransporteIgual debemos primero analizar la función equals, ya que el método equals se llama dentro de obtenerTransporteIgual.

Complejidad del método equals:

La complejidad de este método es O (j^2). Esta complejidad se encuentra en el peor caso en la parte que se recorre la cantidad de paquetes de transportes, ya que usamos un for que en su peor caso se repite J veces siendo J la cantidad que tenga de paquetes ese transporte, y luego dentro de ese mismo usamos el remove object, se repite J veces, y en el peor de los casos seria que encuentre a lo último para remover ese paquete, teniendo en cuenta eso diríamos que los va removiendo y depende de cómo los encuentra, siempre supongamos que los encuentre en el peor de los casos es decir que a todos los paquetes los va encontrando siempre a lo último entonces también sería J, tomando eso tendríamos que O (J*J) por ende la complejidad del método equals es la antes mencionada.

Con respecto a la complejidad del método obtenerTransporteIgual(IdTransporte), el peor caso lo podemos encontrar en el for each cuando buscamos el transporte ya que al no saber la cantidad de transportes que tenemos se repite una cantidad de N veces donde N representa la cantidad de transportes que tenemos en la empresa, y dentro de ese ciclo llamamos N método equals, así mismo la complejidad de este método se extiende a O(N*J^2).

INVARIANTE DE REPRESENTACIÓN:

EMPRESA:

La longitud del número del cuit debe ser mayor a 7, y el nombre de la empresa debe ser distinto a vacío, además debemos contar con por lo menos un depósito. El transporte está representado con un diccionario, el cual debe tener su número de id un string distinto a vacío y debe ser un transporte no ningún otro tipo de objeto.

El viaje también está representado de la misma forma, en el cual debe ir un destino no vacío y su distancia mayor a 0.

- ### DEPOSITO

Siempre debemos, tener que un deposito puede ser propio o tercerizado, es decir, la variable propio, debe ser o true o falso, nunca podría llegar en null.

La refrigeración debe ser o true o false, nunca puede llegar null.

La capacidad total del depósito tiene que ser mayor a 0.

La mercadería total actual del depósito nunca puede superar la capacidad total del depósito.

Además si es un deposito tercierizado debemos tener en cuenta que el costo adicional debe ser >= 0.

También debemos mencionar que al cargar el deposito con paquetes debemos tener en cuenta lo siguiente, solamente se van a poder cargar Paquetes, es decir no puede ser otro tipo de objeto ni mucho menos null, y se verificara si su refrigeración es la misma al del depósito y que el volumen de ese paquete no sobrepase la capacidad de dicho depósito.

➢ VIAJE

El destino debe ser distinto a vacío, no puede ser un destino vacío o nulo.

La distancia debe ser mayor estricto a 0.

➢ PAQUETE

Tanto como el peso del paquete como su volumen deben ser mayor estrictos a 0. Con respecto a su destino es un string el cual debe ser distinto de vacío, y su refrigeración debe ser o true o falso, impidiendo ser un null.

➢ TRANSPORTE

La longitud del número de identificación del transporte debe ser mayor estricto

que 4 y menor estricto que 16 y será único para cada transporte.

La carga máxima y el volumen máximo del transporte debe ser mayor estricto a 0

El costo por km de los transportes debe ser mayor o igual que 0.

Con respecto al peso y volumen de este transporte, siempre el peso actual y volumen actual tiene que ser menor que al peso máximo y al volumen máximo soportado por el transporte.

Por último, también se verifica al cargar un paquete en el transporte que se cumpla lo siguiente: que sea un Paquete, y que no sea otro tipo de objeto, también que no sea null, además se verifica que lleven el mismo destino, que lleven la misma refrigeración, y que el volumen de ese paquete y el peso del paquete no rebalsen el peso máximo y el volumen máximo del transporte.

- ## TRAILER

  Tendrán un indicador de Refrigeración, este tendrá siempre el valor de "True" o "False" este nunca puede recibirlo como null. También el seguro de carga no puede ser negativo, es decir debe ser mayor o igual a 0.

Se dividen en dos tipos, (Por definición ambos respetan el IRREP heredado)

- Si es lo que denominaremos "**Comun"** sólo se podrá utilizar en viajes con un destino no mayor de 500 km.
- Si es lo que denominaremos "**Mega"** sólo se podrá utilizar en viajes con un destino cuya con un recorrido cuya distancia sea mayor a 500 km, tendrán dos indicadores compuestos cada uno por un número decimal o entero positivo que indicarán el costo adicional fijo por viaje y el costo de la comida del chofer respectivamente .

- ## FLETE

  Cada flete contará con una cantidad neutra o positiva de acompañantes representado con un entero no negativo y en relación a esta cantidad de forma directamente proporcional se agrega un costo representado por un número decimal o entero no negativo.