

## Trabajo Práctico P2 (1era parte)

|  
OLIVERA NICOLÁS

P2 COM 1

### ESPECIFICACIÓN DE TADs

#### CLASES Y SUS ATRIBUTOS

EMPRESA	TRANSPORTE	DEPOSITO
<ul style="list-style-type: none"><li>*<b>String</b> Nombre</li><li>*<b>String</b> CUIT</li><li>*Colección de Transportes</li><li>*Colección de Viajes</li><li>*Colección de Depositos</li></ul>	<ul style="list-style-type: none"><li>*<b>String</b> Número ID</li><li>*<b>double</b> Capacidad máx (volumen)</li><li>* <b>double</b> Carga máx (peso)</li><li>* <b>double</b> Costo por KM</li><li>* <b>double</b> Costo final del viaje</li><li>* <b>Viaje</b> Destino</li><li>* <b>boolean</b> En viaje</li><li>*Colección de Paquetes</li></ul>	<ul style="list-style-type: none"><li>*<b>int</b> Número ID</li><li>*<b>boolean</b> frigorífico</li><li>* <b>double</b> Capacidad máx(volumen)</li><li>*<b>boolean</b> propio/tercerizado</li><li>*Colección de Paquetes</li></ul>
<ul style="list-style-type: none"><li>CrearEmpresa()</li><li>AgregarDeposito()</li><li>AgregarTransporte</li><li>AsignarDestino()</li><li>Cargartransporte()</li><li>Iniciarviaje()</li><li>Finalizarviaje()</li><li>ObtenerCosto()</li></ul>	<ul style="list-style-type: none"><li>*cargarTransporte()</li><li>*cargarPaquete()</li><li>*descargarTransporte()</li><li>*controlDeCarga()</li><li>*actualizarCarga()</li><li>*obtenerCosto()</li><li>*setViaje()</li><li>*agregarViaje()</li></ul>	<ul style="list-style-type: none"><li>*cargarDeposito()</li><li>*descargarPaquete()</li><li>*actualizarCapacidad()</li></ul>

PAQUETE	VIAJE	TRANSPORTE CON/SIN FRIO (Subclase de TRANSPORTE)
<ul style="list-style-type: none"><li>*<b>double</b> Peso</li><li>*<b>double</b> Volumen</li><li>*<b>String</b> Destino</li><li>*<b>boolean</b> Confrio</li></ul>	<ul style="list-style-type: none"><li>*<b>String</b> Destino</li><li>*<b>int</b> Distancia</li></ul>	<ul style="list-style-type: none"><li>*<b>boolean</b> frigorífico</li><li>*<b>double</b> Seguro de carga</li></ul>
		<ul style="list-style-type: none"><li>*calcularCostoPor DepTercerizFrio()</li></ul>

## Trabajo Práctico P2 (1era parte)

|

<b>DEPTERCERIZFRIO</b> (Subclase de Deposito)
* <b>double</b> costo por cada 1000 kg

<b>TRAILER</b> (Hereda de TRANSPORTE C/S FRIO)	<b>MEGATRILER</b> (Hereda de TRANSPORTE C/S FRIO)	<b>FLETE</b> (Hereda de TRANSPORTE)
* <b>int</b> Distancia Máxima (500km)	* <b>double</b> Costo fijo de viaje * <b>double</b> Costo extra * <b>int</b> Distancia mínima de 500 km	* <b>int</b> Cantidad de acompañantes * <b>double</b> Costo fijo por pasajero * <b>int</b> distancia máxima(En principio)

## INTERFACES

### Empresa

**CrearEmpresa**(String nombre, String CUIT)

**double AgregarDeposito**(double capmax, boolean frio, boolean propio) {return numDeposito}

**Boolean IncorporarPaquete**(Viaje destino, double volumen, double peso, boolean frio){return true/false}

**Void AgregarTrasporte**(int ID, double peso, double volmax, boolean frio)

**Void AsignarDestino**(int IDtransporte, String destino, int kmviaje)

**double CargarTransporte**(int IDtransporte){return EvolumenCargado}

**Void IniciarViaje**(int IDtransporte)

**Void FinalizarViaje**(int IDtransporte)

**double ObtenerCosto**(int IDtransporte){return costo}

## Trabajo Práctico P2 (1era parte)

|

### Transporte

**Void cargarTransporte**(Deposito d)

**Boolean cargarPaquete**(Paquete p)

**Void descargarTransporte**()

**Boolean controlDeCarga**(Paquete p)

**Void actualizarCarga**(Paquete p)

**double calcularCosto**()

**void agregarViaje**(Viaje v)

**void setViaje**()

### TranspCSF

**Double ObtenerCostoPorDTF**(Deposito d)

### Deposito

**Boolean agregarPaquete**(Paquete p)

**Void descargarPaquete**(Paquete p)

En base al problema planteado por la empresa “La santafesina”, y a sus requerimientos especificados con anterioridad, presentamos este diseño de una posible solución a dicha cuestión. Para ello se realizó una especificación de los TADs, junto a sus atributos y operaciones correspondientes, los cuales son los siguientes:

### EMPRESA

## Trabajo Práctico P2 (1era parte)

|  
Por medio de este TAD se podrá crear el objeto de la clase empresa, la cual contará con un nombre, un número de CUIT. Para esto, vamos a precisar dos datos de tipo String tanto para el nombre de la empresa como para el número de CUIT. Las operaciones que podremos manejar desde este TAD son, por ejemplo la de agregar un deposito a la empresa, y la de agregar un transporte. Aquí vamos a contar con las operaciones `AgregarDeposito()` y `AgregarTransporte()`. Estas lo que harán es crear objetos de la clase `DEPOSITO` Y `TRANSPORTE` respectivamente. También podremos agregar instancias de la clase `Viaje`, por medio de la operación `asignarDestino()`, las cuales serán almacenadas en colecciones de tipo lista. A partir de aquí también podremos ejecutar las operaciones, `incorporarPaquete()`, `cargarTransporte()`, `iniciarViaje()`, `finalizarViaje()` y `obtenerCosto()`.

### TRANSPORTE

Por su parte, el tad `TRANSPORTE` va a funcionar a su vez como superclase de las clases **TranspCSF** y **Flete** (el cual no cuenta con equipo de frío). Aquí, utilizaremos los conceptos de “herencia”, “polimorfismo” y “sobreescritura”. Los tipos de transporte con los que se trabajaran aquí son con los que no puedan contar con equipo de refrigeración. Los atributos que posee un transporte básico son por ejemplo, el de tener una capacidad y una carga máxima de paquetes, un número de identificación, un estado de viaje (True o false) y un costo por kilómetro de viaje. Respecto a los datos relacionados a la capacidad y carga de paquetes, y al costo por km, estos serán de tipo double. Los métodos principales que se emplearan en cada clase son `agregarViaje()` que agrega un `Viaje` con destino y distancia al mismo, y `CargarTransporte()` que recibe un objeto de la clase `Deposito` y a partir de ahí se accede su lista de paquetes. Luego se llama a la operación `cargarPaquete()` la cual va a verificar que el paquete cumpla con las condiciones necesarias (peso, volumen y destino) y de ser así se va a incorporar el paquete a la coleccion de tipo `Paquete` de la clase `Transporte`. Además de cargar el transporte necesitaremos descargarlo mediante la operación `descargarTransporte()`, y por ende tendremos que actualizar la capacidad y carga en peso de paquetes por medio de `actualizarCarga()`. Debido a que este tad se encargará de obtener el costo final del viaje, vamos a incorporar el uso de una variable de tipo double

## Trabajo Práctico P2 (1era parte)

|  
llamada costoFinal y así mismo, se empleará una operación para calcular dicho costo.

### FLETE

Como vimos anteriormente, Flete será subclase de Transporte. Un transporte de este tipo también tendrá una capacidad y una carga de peso máxima, además de un estado de viaje ( true o false) y un costo por kilómetro. Un flete por su parte, va a contar con un determinado número de pasajeros, y un costo extra por cada pasajero. Usaremos datos de tipo double para el costo e int para la cantidad de pasajeros. Los fletes tendrán una distancia máxima de viajes, en este caso, solo podrán recorrer distancias de hasta 200 km.

### TRANSPCSF

A partir de la clase Transporte, desprendemos esta subclase que nos permitirá trabajar con transportes que puedan tener la característica de tener equipo de frío y la de tener un seguro de carga. Estamos hablando de los transportes de tipo **Trailers** y **MegaTrailers**. Estos van a heredar atributos y operaciones de TranspCSF y, a su vez como TranspCSF es subclase de Transporte, heredarán todos los atributos y operaciones propios de un objeto de la clase Transporte. Lo que no está asegurado es que compartan el mismo comportamiento ya que un MegaTrailer por ejemplo, que cuenta con equipo de frio no realizará la operación cargarTransporte de la misma forma que un flete que no cuenta con equipo de frio. Por otro lado, este tad será el encargada de calcular el costo por tonelada cargada correspondiente a los depósitos tercerizados con frigorífico.

### TRAILER

Los transportes de tipo Trailer, como se dijo anteriormente podrán contar con equipo de refrigeración, además de otras características y operaciones un propias de un objeto de clase TranspCSF. El atributo particular que tienen los trailers es que no podrán realizar viajes con una distancia mayor a 500 km. Para almacenar ese dato se utilizará una variable de tipo int.

## Trabajo Práctico P2 (1era parte)

|

### **MEGATRILER**

Los transportes de tipo MegaTrailer poseerán como agregado un costo extra por provisiones, debido a que se encargaran de viajar largas distancias. Además de un costo fijo por cada viaje. Ambos datos serán almacenados en variables de tipo double.

### **DEPOSITO**

Un objeto de la clase Deposito tendrá atributos tales como una capacidad máxima, si cuenta con equipo de refrigeración o no y si es de carácter o propio o tercerizado. Para la capacidad máxima se utilizará una variable de tipo double y para los demás atributos, variables de tipo boolean.

En el caso del TAD DEPOSITO, este va poder incorporarse instancias de la clase PAQUETE. Acción que se podrá realizar por medio de la operación cargarDeposito(). Estas serán almacenadas en colecciones de tipo lista. Otra de las operaciones que se tendrá que implementar es la de descargarPaquete() y actualizarCarga().

### **DEPTERCERIZFRIO**

Esta clase será subclase de Deposito, aquí solo se podrán almacenar productos que necesiten frío. Un depósito de este tipo tendrá por defecto el valor de variable frigorífico en true y false para la variable propio, indicando que es justamente de carácter tercerizado. Por cada tonelada cargada, estos depósitos tendrán un determinado costo extra por lo que se utilizará una variable de tipo double para guardar dicho valor de costo.

### **PAQUETE**

En este TAD nos encontraremos con los objetos de tipo Paquete, los cuales cuentan con atributos como un peso, volumen y destino. Para los primeros dos utilizaremos datos de tipo double y, para el último, vamos a usar un dato de tipo String. Un paquete puede o no necesitar frío, ese dato va a ser guardado en una variable de tipo boolean. Como se mencionó anteriormente, los objetos de tipo PAQUETE van a ser almacenados primero en colecciones de tipo lista ubicadas en la clase

## Trabajo Práctico P2 (1era parte)

|  
Deposito y posteriormente en colecciones de la clase Transporte. Todo a través de las operaciones ya nombradas con anterioridad.

### **VIAJES**

Respecto a este TAD, podemos decir que cuenta con dos atributos, destino y distancia al destino (String destino, int distancia).

Las instancias de esta clase, serán almacenadas por un lado en la clase Empresa, y por otro en la clase Transporte. A partir de las instancias de la clase Viaje que vayan a ser almacenadas en la clase Empresa se determinará el viaje que realizará cada transporte.