



**Universidad Autónoma de Zacatecas**

**Unidad Académica de Ingeniería Eléctrica**

**Programa Académico de Ingeniería de Software.**

**Nombre de la Práctica      Repaso de POO**

**Numero de Práctica      1**

**Nombre de la carrera      Ingeniería de Software**

**Nombre de la materia      Lab. Estructuras de Datos**

**Nombre del alumno      Jesús Manuel Juárez Pasillas**

**Nombre del docente      Aldonso Becerra Sánchez**

**Fecha: 16/08/2021**

# Práctica 1: Repaso de POO

## Introducción:

Repasar la programación orientada a objetos con el fin de tener más presente este paradigma, nos permite recordar como es que se hacia un programa orientado a objetos, además de pulir todos los rubros que esta práctica conlleva, tales como el encapsulamiento, modularidad, etc.

Si todos los principios de la programación orientada a objetos son seguidos al pie de la letra, el programa que se esté haciendo tendrá mucha mejor estructura, además que será mas legible para cualquier persona que entienda estos principios.

## Desarrollo:

La realización de esta práctica será sobre una compañía transportista la cual se dedica a llevar diferentes tipos de materiales de un lugar a otro. La compañía contara con diferentes tipos de vehículos, estos vehículos serán manejados por un chofer designado el cual llevara los materiales solicitados de un origen a un destino, para esto se deberá poder dar de alta vehículos, choferes, orígenes, destinos y materiales, a esta acción se le llamara transportar y se registraran los transportes que se hagan. Además, también se solicita que se pueda solicitar la información de cada parte del transporte y del transporte como tal.

En el proyecto se crearon 4 paquetes en los cuales src es la base ya que dentro de este se encuentra el modelo, vista y controlador.

En el paquete de modelo se agrego la clase Chofer, Locacion, Material, TipoVehiculo (este es un enumerado el cual contiene todos los tipos de vehículos que se tienen), Transporte, Vehiculo, estas clases tienen sus atributos propios, asi como los métodos de getters, setters y el toString.

En el paquete vista solo se agregó la clase Pruebatransportista que contiene el main e inicializa todo el programa.

En el paquete controlador se agregaron las clases de Controlador, EntradaConsola y Menu (Este es un enumerado el cual contiene todas las opciones del menú), la clase

Controlador contiene todos los métodos con los cuales se cubren los requisitos del programa, EntradaConsola contiene 2 métodos con los cuales se pedirán datos al usuario mediante línea de comando.

*Nota: En la sección de código agregado se ven todos los paquetes clases y métodos de cada apartado. Además, para mayor comodidad se creó un método que agrega valores a todas las listas excepto la lista que guarda los registros de transporte.*

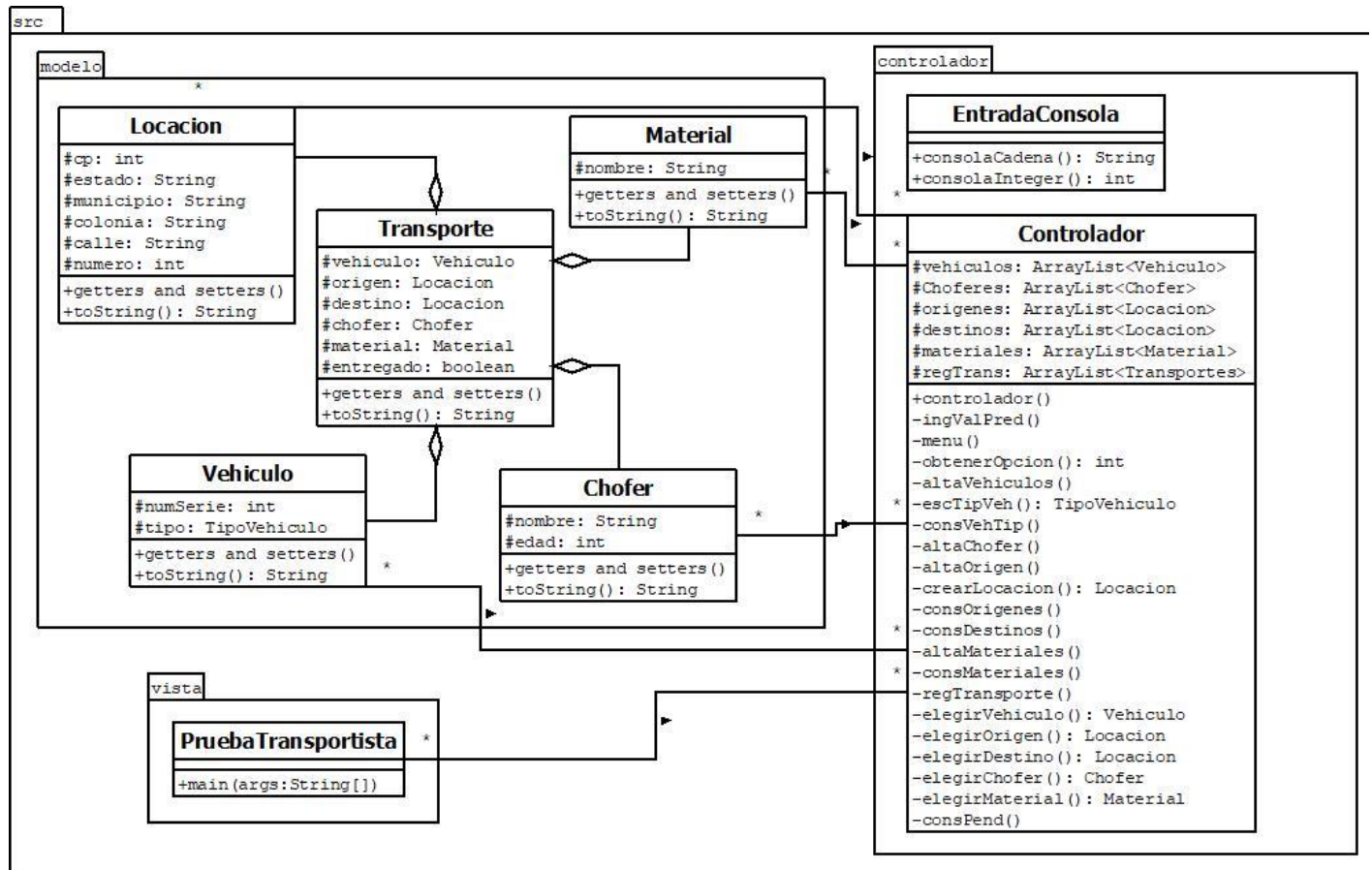
### Capturas del programa funcionando:

```
ELIGE EL NUMERO DE LA OPCION:
1.- Dar de alta vehiculos
2.- Consultar vehiculos por tipo
3.- Registrar transporte de origen a destino con chofer
4.- Dar de alta choferes
5.- Dar de alta origen
6.- Dar de alta destino
7.- Consultar origen
8.- Consultar destino
9.- Dar de alta materiales
10.- Consultar materiales
11.- Consultar el listado completo de materiales pendientes de entregar
12.- Salir
3
Elige el numero del vehiculo que deseas:
1. Vehiculo: 525372554, Tipo: Camione frigorifico
2. Vehiculo: 525372555, Tipo: Camion parachoque
3. Vehiculo: 525372556, Tipo: Camion con plataforma abierta
```

```
11.- Consultar el listado completo de materiales pendientes de entregar
12.- Salir
11
Transporte{vehiculo=Vehiculo: 525372556, Tipo: Camion con plataforma abierta, origen:
```

## Código agregado:

Todas las clases y métodos son nuevos.



## Pre-evaluación:

Pre-Evaluación para prácticas de Laboratorio de Estructuras de Datos	PRE-EVALUACIÓN DEL ALUMNO
CUMPLE CON LA FUNCIONALIDAD SOLICITADA.	Sí
DISPONE DE CÓDIGO AUTO-DOCUMENTADO.	-
DISPONE DE CÓDIGO DOCUMENTADO A NIVEL DE CLASE Y MÉTODO.	-
DISPONE DE INDENTACIÓN CORRECTA.	Sí
CUMPLE LA POO.	Sí
DISPONE DE UNA FORMA FÁCIL DE UTILIZAR EL PROGRAMA PARA EL USUARIO.	Sí
DISPONE DE UN REPORTE CON FORMATO IDC.	Sí
LA INFORMACIÓN DEL REPORTE ESTÁ LIBRE DE ERRORES DE ORTOGRAFÍA.	Sí
SE ENTREGÓ EN TIEMPO Y FORMA LA PRÁCTICA.	Sí
INCLUYE LA DOCUMENTACIÓN GENERADA CON JAVADOC.	-
INCLUYE EL CÓDIGO AGREGADO EN FORMATO UML.	Sí
INCLUYE LAS CAPTURAS DE PANTALLA DEL PROGRAMA FUNCIONANDO.	Sí
LA PRÁCTICA ESTÁ TOTALMENTE REALIZADA (ESPECIFIQUE EL PORCENTAJE COMPLETADO).	100%
Observaciones:	

## Conclusión:

El uso de la POO para crear programas amplios nos permite separarlo en partes para poder hacerlo mucho más fácil, también nos permite tener un código mas limpio y mas entendible, con el cual se pueden localizar cada parte del programa sin dificultades y con esto obtenemos que el mantenimiento o mejoras al programa sea mucho más fácil de realizar.