

## Solución Desafío - Automotora

- Reconocer las clases que están en el enunciado con sus atributos.
- Crear las clases.
  - Accesadores y Seteadores para cada campo.
  - Constructores
  - Operaciones según clase
- Crear relaciones asociadas.

### Requerimiento 1

Crear la clase vehículo que contiene atributos, “getters and setters” para cada campo y un Constructor.

```
//Clase Vehiculo
public class Vehiculo {
    private String color;
    private String patente;

    public Vehiculo(String color, String patente) {
        super();
        this.color = color;
        this.patente = patente;
    }

    public String getColor() {
        return color;
    }

    public void setColor(String color) {
        this.color = color;
    }

    public String getPatente() {
        return patente;
    }

    public void setPatente(String patente) {
        this.patente = patente;
    }
}
```

## Requerimiento 2

Crear la clase Taxi que contiene un atributo llamado "valorPasaje", generamos su "getter and setter" para el campo y un Constructor.

```
//Clase Taxi

public class Taxi extends Vehiculo {

    private int valorPasaje;

    public Taxi(String color, String patente, int valorPasaje) {
        super(color, patente);
        this.valorPasaje = valorPasaje;
    }

    public int getValorPasaje() {
        return valorPasaje;
    }

    public void setValorPasaje(int valorPasaje) {
        this.valorPasaje = valorPasaje;
    }

    /**
     * Método pagarPasaje :
     * Se ingresa el monto por parámetro, antes de pagar el pasaje se
     debe validar
     * que este monto sea mayor al valor del pasaje inicial.
     * Si se cumple devolver el vuelto
     * Si no se cumple devolver el pasaje original
     * @param pasaje
     * @return
     */
    public int pagarPasaje(int pasaje) {
        int vuelto = 0 ;

        if(pasaje >= getValorPasaje()) {
            vuelto = getValorPasaje() - pasaje;
        }
        return vuelto ;
    }
}
```

## Requerimiento 3

Crear la clase Bus que extiende de vehículo, que contiene el atributo “cantidadDeAsientos”, generamos su “getter and setter” para el campo y un Constructor.

```
//Clase Bus

public class Bus extends Vehiculo {

    private int cantidadDeAsientos;

    public Bus(String color, String patente, int cantidadDeAsientos) {
        super(color, patente);
        this.cantidadDeAsientos = cantidadDeAsientos;
    }

    public int getCantidadDeAsientos() {
        return cantidadDeAsientos;
    }

    public void setCantidadDeAsientos(int cantidadDeAsientos) {
        this.cantidadDeAsientos = cantidadDeAsientos;
    }

    /**
     * @return
     */
    public int asientosDisponibles() {
        // TODO implement here
        return getCantidadDeAsientos();
    }

}
```

## Requerimiento 4

Crear la clase Minibus que extiende de Bus, que contiene un atributo llamado "tipoViaje", el "getter and setter" correspondiente y su Constructor.

```
//Clase Minibus

public class MinuBus extends Bus {

    private String tipoViaje;

    public MinuBus(String color, String patente, int cantidadDeAsientos,
String tipoViaje) {
        super(color, patente, cantidadDeAsientos);
        this.tipoViaje = tipoViaje;
    }

    public String getTipoViaje() {
        return tipoViaje;
    }

    public void setTipoViaje(String tipoViaje) {
        this.tipoViaje = tipoViaje;
    }

    public void imprimeBus() {
        System.out.println( " color " + super.getColor());
        System.out.println( " patente " + super.getColor());
        System.out.println( " cantidadDeAsientos " +
super.getCantidadDeAsientos() );
        System.out.println( " tipoViaje " + getTipoViaje());
    }

}
```

## Requerimiento 5

Crear la clase Persona que contiene atributos, “getters and setters” para cada campo y un Constructor.

```
//Clase Persona

public class Persona {

    private String rut;
    private String nombre;
    private int edad;

    public Persona(String rut, String nombre, int edad) {
        super();
        this.rut = rut;
        this.nombre = nombre;
        this.edad = edad;
    }
    public String getRut() {
        return rut;
    }
    public void setRut(String rut) {
        this.rut = rut;
    }
    public String getNombre() {
        return nombre;
    }
    public void setNombre(String nombre) {
        this.nombre = nombre;
    }
    public int getEdad() {
        return edad;
    }
    public void setEdad(int edad) {
        this.edad = edad;
    }
}
```

## Requerimiento 6

Crear la clase Cliente que extiende de Persona, contiene un atributo llamado "edad", su "getter and setter" para el campo y un Constructor.

```
//Clase Cliente

public class Cliente extends Persona {

    private int edad;

    public Cliente(String rut, String nombre, int edad, int edad2) {
        super(rut, nombre, edad);
        edad = edad2;
    }

    public int getEdad() {
        return edad;
    }

    public void setEdad(int edad) {
        this.edad = edad;
    }

}
```

## Requerimiento 7

Crear la clase Vendedor que extiende de Persona, contiene un atributo llamado "dirección", su "getter and setter" para el campo y un Constructor.

```
//Clase Vendedor

public class Vendedor extends Persona {

    private String direccion;

    public Vendedor(String rut, String nombre, int edad, String
direccion) {
        super(rut, nombre, edad);
        this.direccion = direccion;
    }

    public String getDireccion() {
        return direccion;
    }

    public void setDireccion(String direccion) {
        this.direccion = direccion;
    }

}
```

## Requerimiento 8

Crear la clase Tienda que contiene atributos, “getters and setters” para cada campo y un Constructor. Además, incorpora el método existeStock() para obtener el stock.

```
//Clase Tienda
public class Tienda {

    private Vendedor vendedor;
    private Vehiculo vehiculo;
    private int stock;

    public Tienda(Vendedor vendedor, Vehiculo vehiculo, int stock) {
        this.vendedor = vendedor;
        this.vehiculo = vehiculo;
        this.stock = stock;
    }

    public Vendedor getVendedor() {
        return vendedor;
    }

    public void setVendedor(Vendedor vendedor) {
        this.vendedor = vendedor;
    }

    public Vehiculo getVehiculo() {
        return vehiculo;
    }

    public void setVehiculo(Vehiculo vehiculo) {
        this.vehiculo = vehiculo;
    }

    public int getStock() {
        return stock;
    }

    public void setStock(int stock) {
        this.stock = stock;
    }

    public String existeStock() {
        // TODO implement here
        return "cantidad de stock es " + getStock();
    }

}
```



## Diagrama del caso

