



---

## UNIDAD 3. CARACTERÍSTICAS DE POO Y EXCEPCIONES

---

**MATERIA: Programación orientada a objetos I**



15 DE MAYO DE 2020

Guadalupe García Vázquez  
MATRICULA: ES1921008556  
Desarrollo en Software

**Docente: Claudia Erika González Gómez**

**Actividad 2: Herencia y Polimorfismo. Fase 2****Contenido**

Introducción.....	3
Desarrollo.....	4
Conclusiones .....	10
Bibliografía .....	11

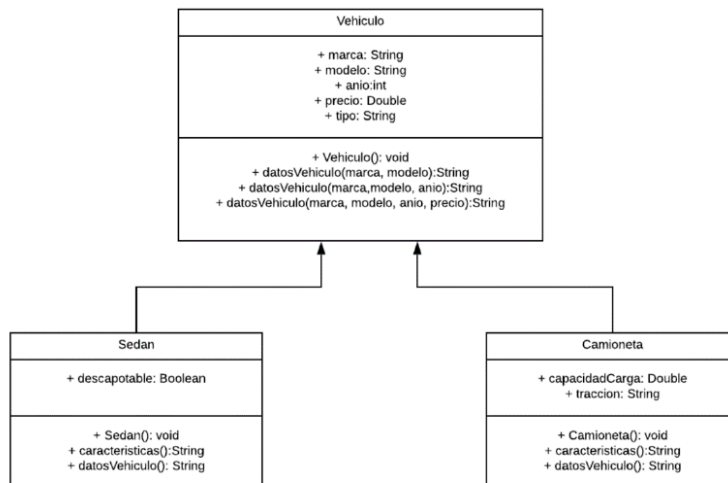
## Introducción

El siguiente tema veremos los conceptos de Herencia, Superclase, subclase, y polimorfismo.

Aplicándolo a un ejercicio y programándolo en Netbeans, se detectará la superclase(clase padre), la subclase(clase hija), jerarquía de clases y polimorfismo.

## Desarrollo

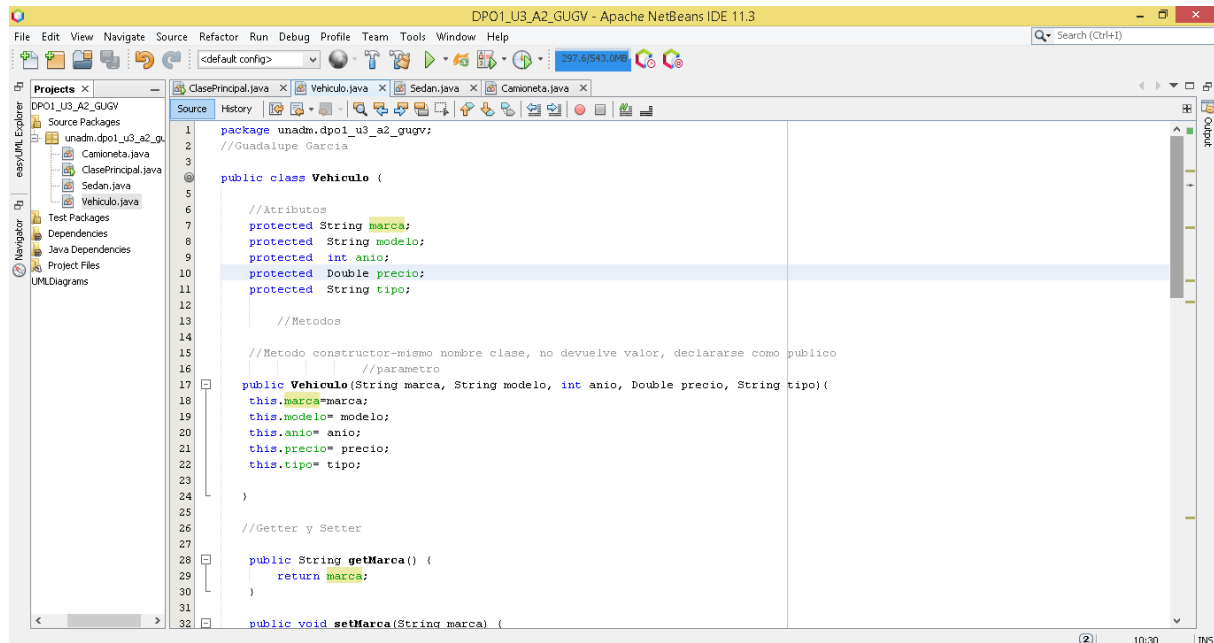
### 1. Analiza el siguiente diagrama de clases



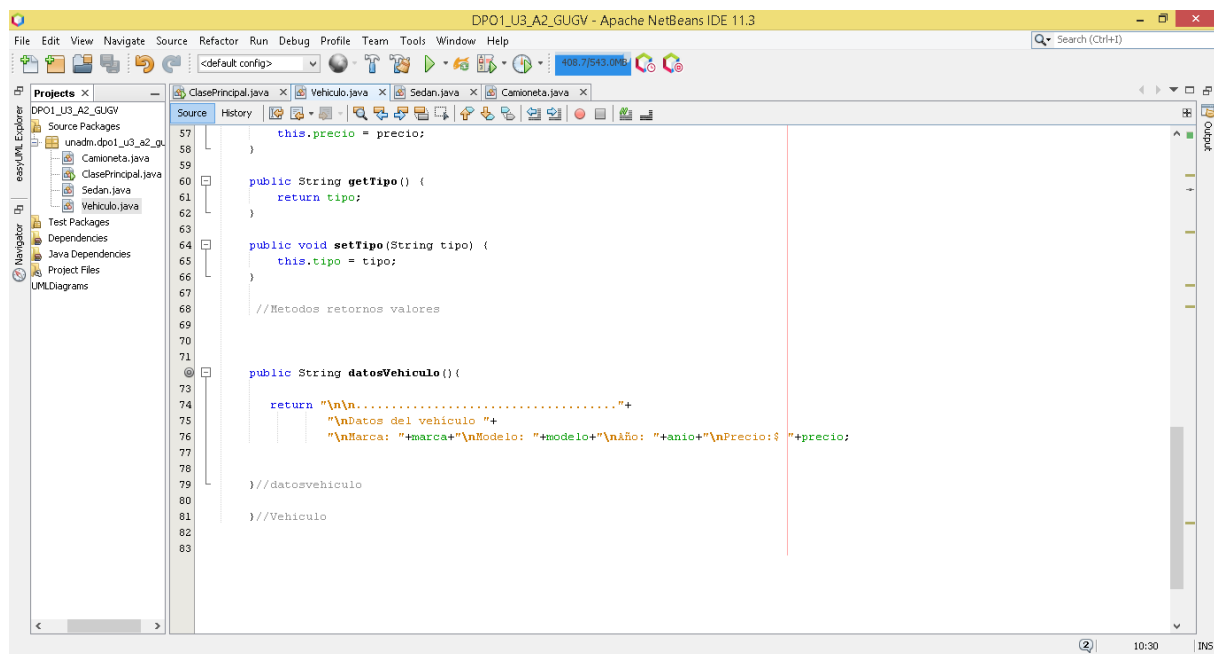
### 2. Aplicando el concepto de herencia y polimorfismo, realice la codificación del diagrama de clases indicado en el paso 1.

Para lograr aplicar de manera adecuada los conceptos de Herencia y polimorfismo en Netbeans.

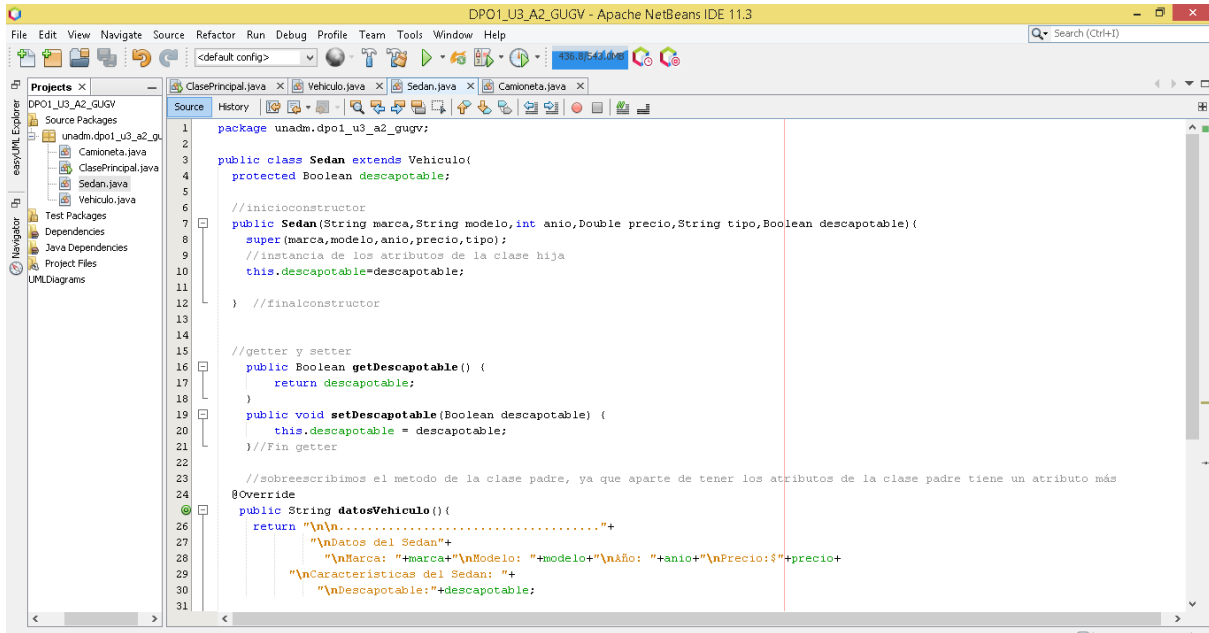
Primero creamos la Superclase "Vehiculos" con los atributos y métodos correspondientes.



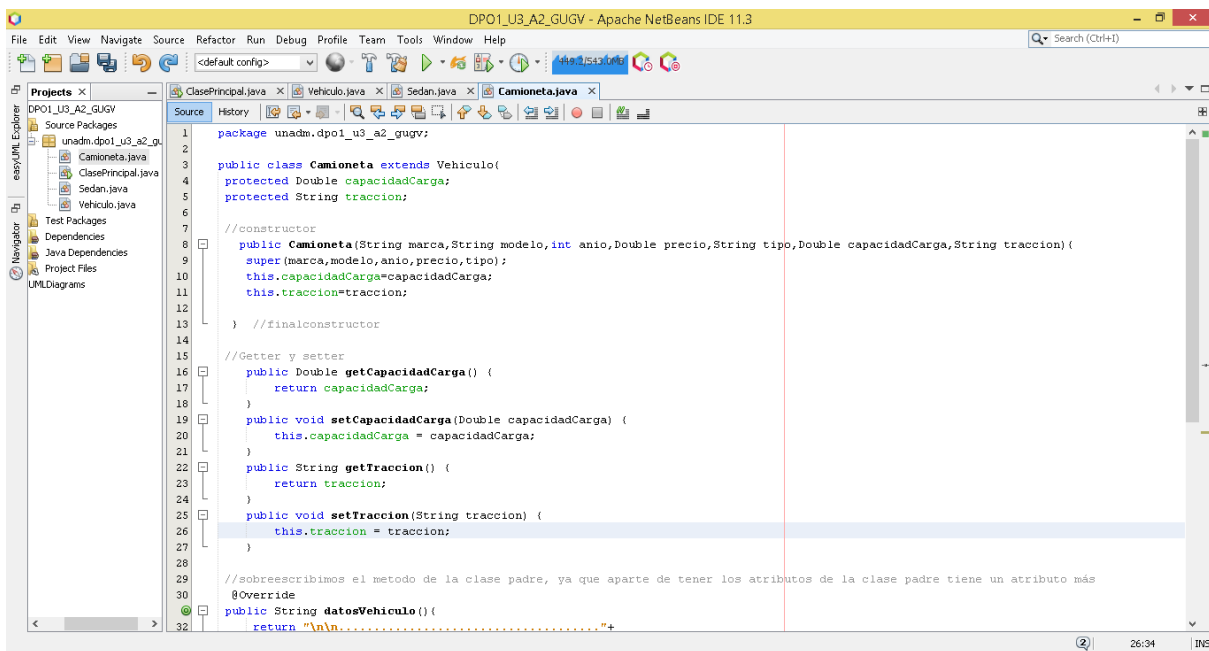
Así mismo se utilizaron los métodos Get y Set. Los métodos get y set, son simples métodos que usamos en las clases para mostrar (get) o modificar (set) el valor de un atributo, esto con la finalidad de que los valores de las clases estuvieran protegidos.



Subclase Sedan,

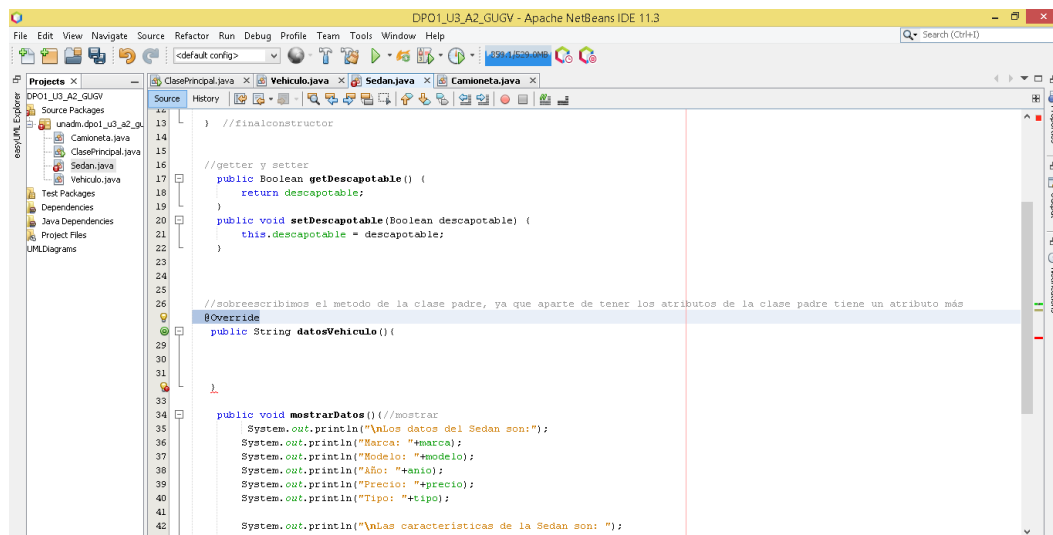
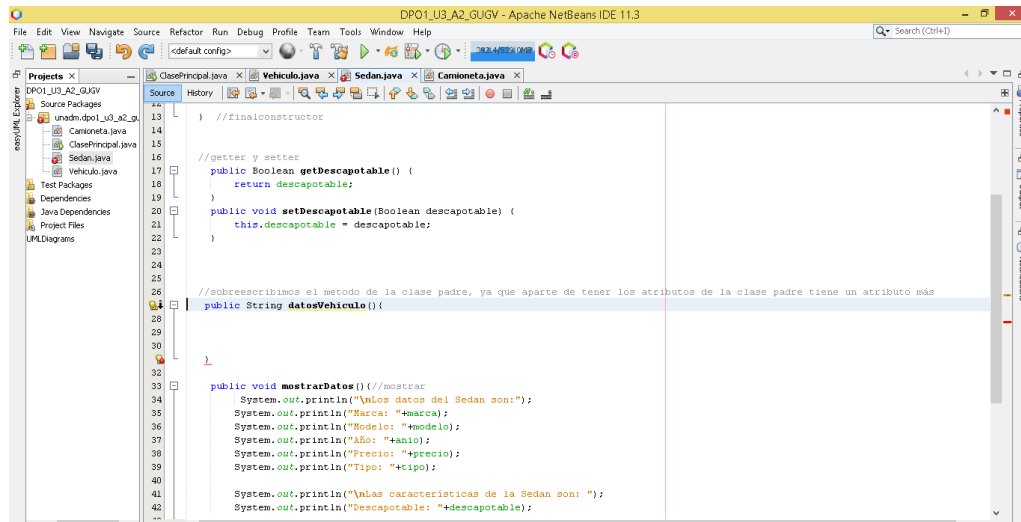


Subclase Camioneta

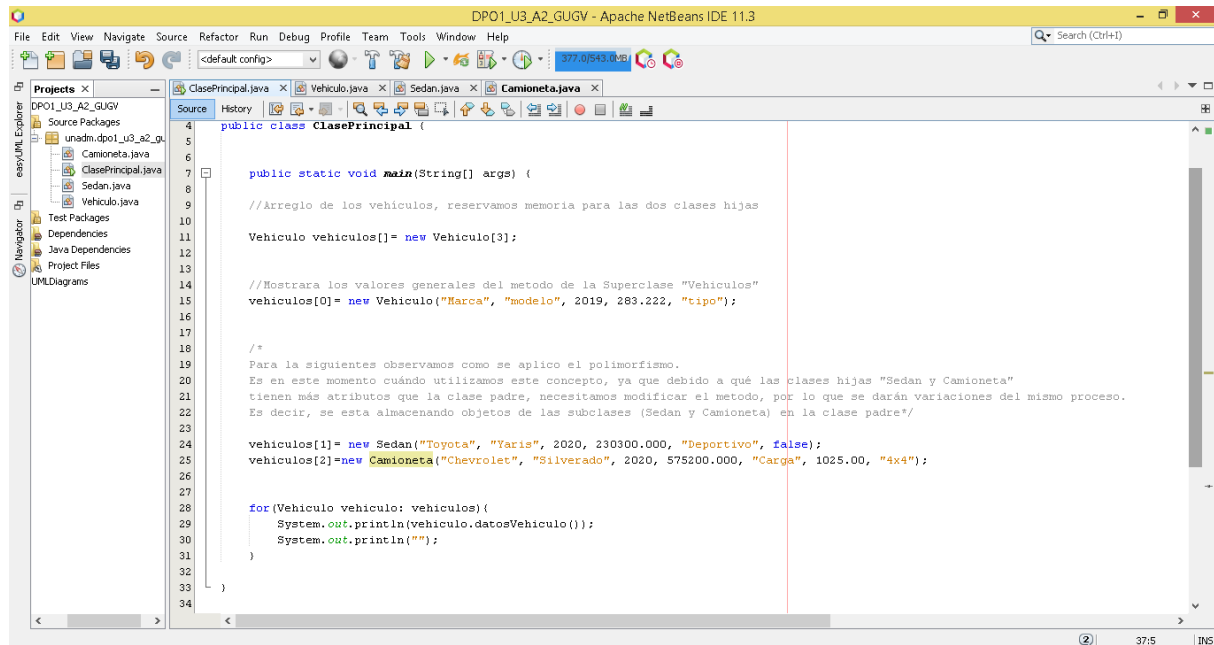


En este puntos nos encontramos con un problema.

Cómo podemos observar en la siguiente imagen Netbeans nos marcaba un error, ya que estamos utilizando el método de la clase Padre, se escribe `@Override` para poder sobrescribir el método.



Y por último en la Clase principal, instanciamos las subclases.

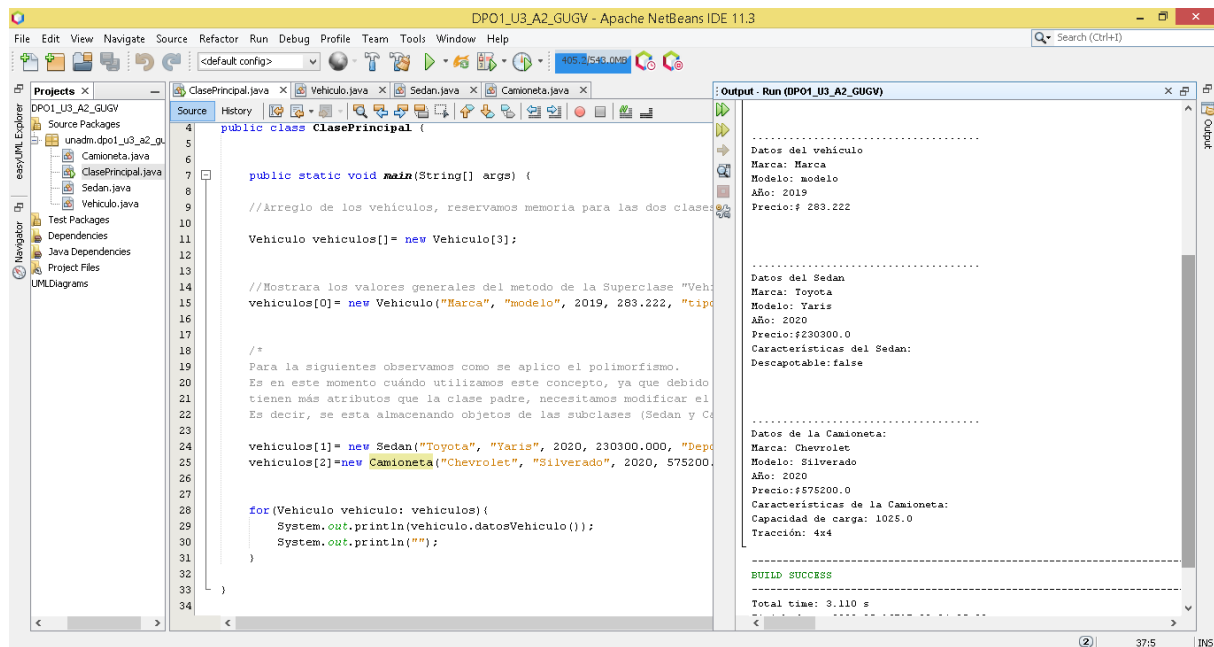


```

4 public class ClasePrincipal {
5
6     public static void main(String[] args) {
7
8         //Arreglo de los vehiculos, reservamos memoria para las dos clases hijas
9
10        Vehiculo vehiculos[] = new Vehiculo[3];
11
12
13        //Mostrara los valores generales del metodo de la Superclase "Vehiculos"
14        vehiculos[0] = new Vehiculo("Marca", "modelo", 2019, 283.222, "tipo");
15
16
17        /*
18        Para la siguientes observamos como se aplica el polimorfismo.
19        Es en este momento cuando utilizamos este concepto, ya que debido a que las clases hijas "Sedan y Camioneta"
20        tienen más atributos que la clase padre, necesitamos modificar el metodo, por lo que se darán variaciones del mismo proceso.
21        Es decir, se esta almacenando objetos de las subclases (Sedan y Camioneta) en la clase padre*/
22
23        vehiculos[1] = new Sedan("Toyota", "Yaris", 2020, 230300.000, "Deportivo", false);
24        vehiculos[2] = new Camioneta("Chevrolet", "Silverado", 2020, 575200.000, "Carga", 1025.00, "4x4");
25
26        for(Vehiculo vehiculo: vehiculos){
27            System.out.println(vehiculo.datosVehiculo());
28            System.out.println("");
29        }
30    }
31
32
33
34

```

Resultando lo siguiente. La primera representa el método del padre, sin ninguna modificación. La segunda hacemos el llamado a las Subclases de Sedan y Camioneta, con la modificación del método de la superclase, utilizando el polimorfismo.



```

4 public class ClasePrincipal {
5
6     public static void main(String[] args) {
7
8         //Arreglo de los vehiculos, reservamos memoria para las dos clases hijas
9
10        Vehiculo vehiculos[] = new Vehiculo[3];
11
12
13        //Mostrara los valores generales del metodo de la Superclase "Vehiculos"
14        vehiculos[0] = new Vehiculo("Marca", "modelo", 2019, 283.222, "tipo");
15
16
17        /*
18        Para la siguientes observamos como se aplica el polimorfismo.
19        Es en este momento cuando utilizamos este concepto, ya que debido a que las clases hijas "Sedan y Camioneta"
20        tienen más atributos que la clase padre, necesitamos modificar el metodo, por lo que se darán variaciones del mismo proceso.
21        Es decir, se esta almacenando objetos de las subclases (Sedan y Camioneta) en la clase padre*/
22
23        vehiculos[1] = new Sedan("Toyota", "Yaris", 2020, 230300.000, "Deportivo", false);
24        vehiculos[2] = new Camioneta("Chevrolet", "Silverado", 2020, 575200.000, "Carga", 1025.00, "4x4");
25
26        for(Vehiculo vehiculo: vehiculos){
27            System.out.println(vehiculo.datosVehiculo());
28            System.out.println("");
29        }
30    }
31
32
33
34

```

```

.....
Datos del vehículo
Marca: Marca
Modelo: modelo
Año: 2019
Precio: 283.222
.....

Datos del Sedan
Marca: Toyota
Modelo: Yaris
Año: 2020
Precio: 230300.0
Características del Sedan:
Descapotable: false
.....

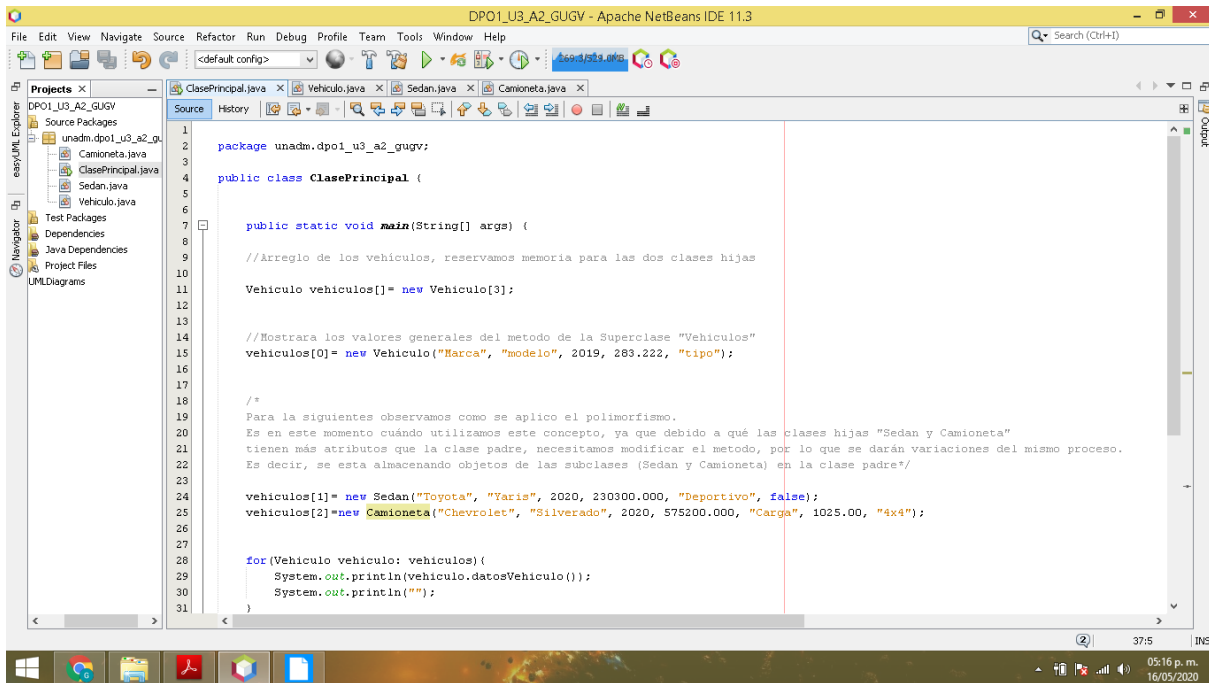
Datos de la Camioneta:
Marca: Chevrolet
Modelo: Silverado
Año: 2020
Precio: 575200.0
Características de la Camioneta:
Capacidad de carga: 1025.0
Tracción: 4x4
.....

BUILD SUCCESS
.....
Total time: 3.110 s
.....

```



3. Describa dentro del código de su proyecto (como comentario) donde aplica herencia y polimorfismo, entregue sus clases en Java.



```
1 package unadm.dpo1_u3_a2_gugv;
2
3 public class ClasePrincipal {
4
5
6
7     public static void main(String[] args) {
8
9         //Arreglo de los vehiculos, reservamos memoria para las dos clases hijas
10
11         Vehiculo vehiculos[] = new Vehiculo[3];
12
13
14         //Mostrara los valores generales del metodo de la Superclase "Vehiculos"
15         vehiculos[0] = new Vehiculo("Marca", "modelo", 2019, 283.222, "tipo");
16
17
18         /*
19          Para la siguientes observamos como se aplico el polimorfismo.
20          Es en este momento cuando utilizamos este concepto, ya que debido a que las clases hijas "Sedan y Camioneta"
21          tienen más atributos que la clase padre, necesitamos modificar el metodo, por lo que se darán variaciones del mismo proceso.
22          Es decir, se esta almacenando objetos de las subclases (Sedan y Camioneta) en la clase padre*/
23
24         vehiculos[1] = new Sedan("Toyota", "Yaris", 2020, 230300.000, "Deportivo", false);
25         vehiculos[2] = new Camioneta("Chevrolet", "Silverado", 2020, 575200.000, "Carga", 1025.00, "4x4");
26
27
28         for(Vehiculo vehiculo: vehiculos){
29             System.out.println(vehiculo.datosVehiculo());
30             System.out.println("");
31         }
32     }
33 }
```

## Conclusiones

Como vimos de la actividad anterior la herencia y el polimorfismo son dos conceptos que van unidos. Ambos nos facilitan la programación, siendo dos temas fundamentales dentro de la programación orientada a objetos (POO).

Ahora lo pusimos en práctica y me ayudo a comprender como se utilizan estos conceptos.

En el ejercicio primero detecte cuál era la superclase, en este caso es Vehículos y las subclases Sedan y Camioneta.

La superclase vehículo heredará los métodos y atributos a sus hijas (Sedan y Camioneta) Pero éstas tienen atributos independientes de la clase padre, más específicos de éstas, por lo que la aplicación del método de la clase padre deber modificarse. Es cuando entra el polimorfismo.

## Bibliografía

Programación ATS. (Consultado 14/05/20). *Curso de programación de Java*. Obtenido de Obtenido de: <https://www.youtube.com/watch?v=2ZXiuh0rg3M&list=PLWtYZ2ejMVJkOuTCzIk61j7XKfpIR74K>

Soria, P. R. (Consultado 14/05/20). *Herencia*. Obtenido de Programación orientada a objetos-Java: Obtenido de: <https://catedu.gitbooks.io/programacion-orientada-a-objetos-java/content/herencia.html>

Universidad Abierta y a Distancia de México. (Consultado 14/05/20). *Unidad 3. Características de POO y excepciones*. Obtenido de Obtenido de: file: /DPO1\_U3\_Contenido.pdf

Universidad Nacional Autónoma de México. (Consultado 14/05/20). *Programando con Java para aplicaciones Android*. Obtenido de Obtenido de: <https://www.coursera.org/lecture/programandoconjava/herencia-y-polimorfismo-D8J6y>

Universidad Nacional Autónoma de México. (Consultado 14/05/2020). *Herencia y Polimorfismo*. Obtenido de Programando con Java para aplicaciones Android: Obtenido de: <https://www.coursera.org/lecture/programandoconjava/herencia-y-polimorfismo-D8J6y>  
[https://www.youtube.com/watch?v=LZPafq9Ve\\_Q&list=PLWtYZ2ejMVJkOuTCzIk61j7XKfpIR74K&index=75](https://www.youtube.com/watch?v=LZPafq9Ve_Q&list=PLWtYZ2ejMVJkOuTCzIk61j7XKfpIR74K&index=75)  
[https://www.youtube.com/watch?v=vQZ4RY\\_4l-A&list=PLWtYZ2ejMVJkOuTCzIk61j7XKfpIR74K&index=76](https://www.youtube.com/watch?v=vQZ4RY_4l-A&list=PLWtYZ2ejMVJkOuTCzIk61j7XKfpIR74K&index=76)  
<https://www.youtube.com/watch?v=RXcUTWczbag&list=PLWtYZ2ejMVJkOuTCzIk61j7XKfpIR74K&index=78>