

NIVEL 3

Herencia

Al igual que sucede en el mundo real, la herencia en esencia es el traspaso de elementos de una generación a otra. Aplicado de manera técnica a la programación “Es el mecanismo por el cual una clase permite heredar las características (atributos y métodos) de otra clase”.[1]

Es uno de los pilares fundamentales de la Programación Orientada a Objetos ya que permite que se reutilice código al no tener que definir nuevamente los atributos y métodos que pueden estar presentes en otra clase.

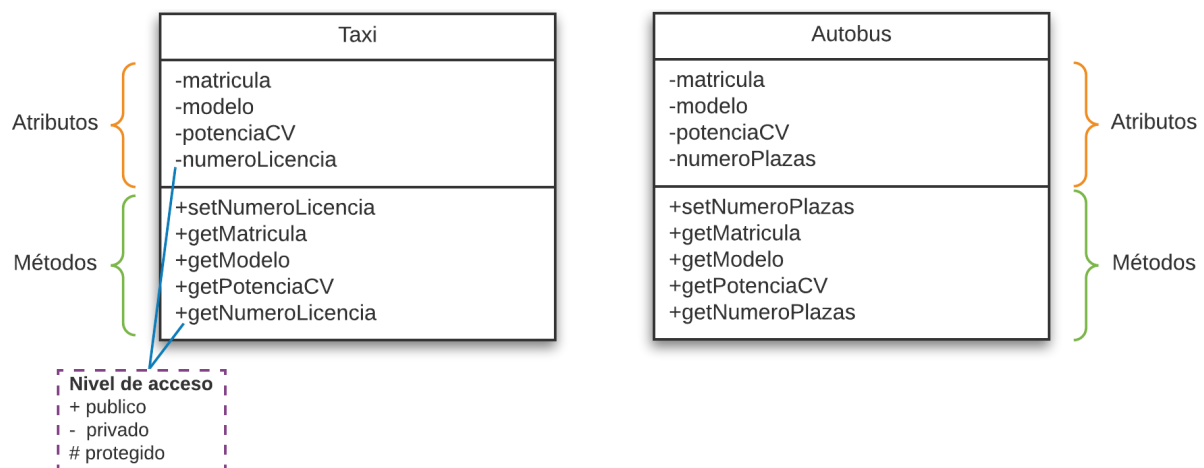
Para entender mejor la herencia es necesario conocer dos términos:

- **Superclase o clase madre:** Portador de los métodos y atributos que se traspasaran a las subclases.
- **Subclase o clase hija:** Clase o clases que heredan las características de la clase madre.

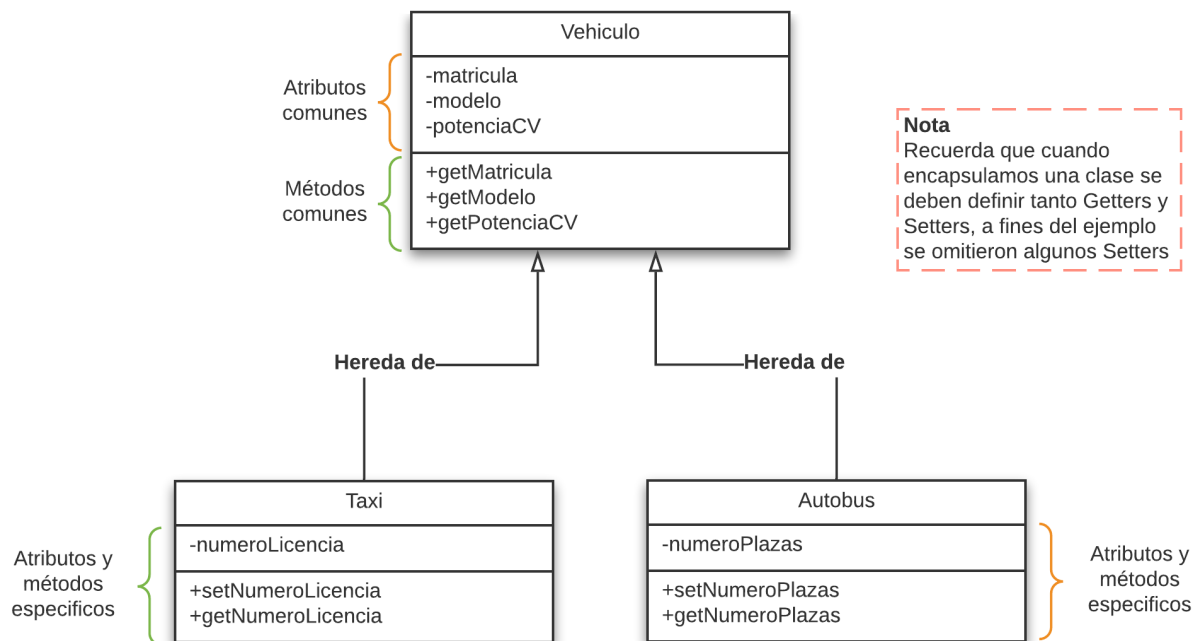
“Hay dos tipos de herencia: Herencia Simple y Herencia Múltiple. La primera indica que se pueden definir nuevas clases solamente a partir de una clase inicial mientras que la segunda indica que se pueden definir nuevas clases a partir de dos o más clases iniciales. Java sólo permite herencia simple.”[3]

Ejemplo: [2]

Tenemos un programa que tiene objetos Taxi y Autobús.



Como se puede ver en el diagrama, ambas clases tienen muchos atributos y métodos en común, por lo que volveríamos más eficiente el programa si aplicamos la herencia de la siguiente forma:



De esta manera, con la creación de la clase 'vehículo' eliminamos la repetición de código y hacemos que el programa sea más eficaz.

Pero ahora te preguntarás cómo aplico esto a código y más aún como puedo instanciar un objeto de tipo Taxi o Autobús, primero veamos como se definen las clases:

Clase Vehiculo

La clase madre se define de la manera usual

```

public class Vehiculo {
    private String matricula;
    private String modelo;
    private String potenciaCV;

    //-----Constructor-----
    public Vehiculo(String matricula, String modelo, String potenciaCV) {
        this.matricula = matricula;
        this.modelo = modelo;
        this.potenciaCV = potenciaCV;
    }

    //-----Getters y Setters-----
    public String getMatricula() {
        return matricula;
    }

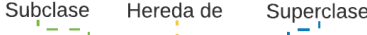
    public String getModelo() {
        return modelo;
    }

    public String getPotenciaCV() {
        return potenciaCV;
    }
}

```

Mientras que las clases hijas deben ir acompañadas de la palabra 'extends' y el nombre de la clase madre de la que están heredando, al igual que de la palabra 'super()' en el constructor.

Clase Taxi



```

public class Taxi extends Vehiculo{

    private long numeroLicencia;

    //-----Constructor-----
    public Taxi(String matricula, String modelo, String potenciaCV, long numeroLicencia) {
        super(matricula, modelo, potenciaCV);
        this.numeroLicencia = numeroLicencia;
    }

    //-----Getters y Setters-----
    public long getNumeroLicencia() {
        return numeroLicencia;
    }

    public void setNumeroLicencia(long numeroLicencia) {
        this.numeroLicencia = numeroLicencia;
    }
}

```

Llamado del constructor Vehiculo

Clase Autobús

```

Subclase      Hereda de      Superclase
public class Autobus extends Vehiculo {
    private int numeroPlazas;

    //-----Constructor-----
    public Autobus(String matricula, String modelo, String potenciaCV, int numeroPlazas) {
        super(matricula, modelo, potenciaCV);
        this.numeroPlazas = numeroPlazas;
    }

    //-----Getters y Setters-----
    public int getNumeroPlazas() {
        return numeroPlazas;
    }

    public void setNumeroPlazas(int numeroPlazas) {
        this.numeroPlazas = numeroPlazas;
    }
}

```

Llamado del constructor Vehiculo

Por último para instanciar una clase que hereda de otra se debe llamar de manera usual el constructor de la clase y pasarle los parámetros que se refieren a los atributos de la super y subclase.

```

public class Main {

    public static void main(String[] args) {

        //Objeto Taxi
        Taxi taxi1 = new Taxi("HSB112", "Hyundai Grand I10", "86 a 5500 rpm", 1425436);
        //Objeto Autobus
        Autobus autobus1 = new Autobus("SDF334", "Paradiso 1200", "428 hp @ 2,000 rpm", 46);
    }
}

```

Referencias

1. ¿Qué es la Herencia en programación orientada a objetos? (2019, June 5). IfgeekthenEveris. <https://ifgeekthen.everis.com/es/herencia-en-programacion-orientada-objetos>
2. Rodríguez, A. (2021). Concepto o definición de herencia en Java (programación orientada a objetos). ¿Qué es? Ejemplos (CU00684B). Aprenderaprogramar.com. https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=651:concepto-o-definicion-de-herencia-en-java-programacion-orientada-a-objetos-ique-es-ejemplos-cu00684b&catid=68&Itemid=188
3. III - Herencia. (2021). Unam.mx. http://profesores.fi-b.unam.mx/carlos/java/java_basico3_4.html