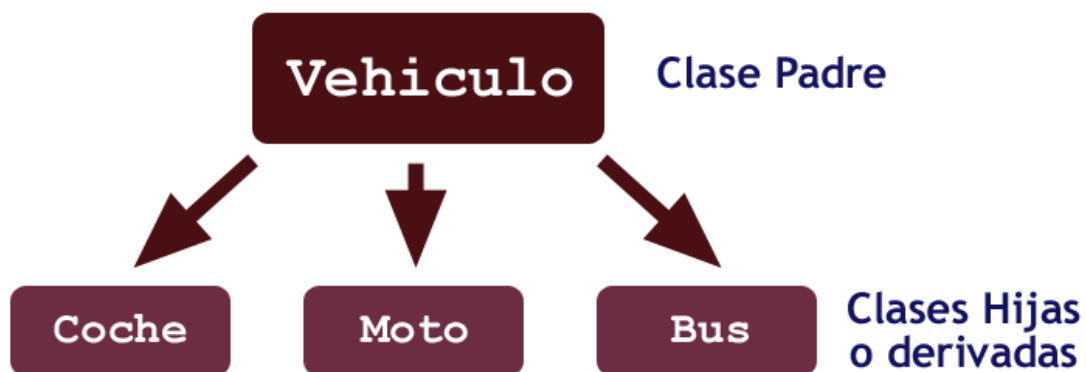


Polimorfismo:

Definición: El polimorfismo es la capacidad que tiene un objeto de tomar distintas formas, de tal manera que una referencia a una clase (atributo, parámetro o declaración local o elemento de un vector) acepta direcciones de objetos de dicha clase y de sus clases derivadas (hijas, nietas, ...).

Veremos que el polimorfismo y la herencia son dos conceptos estrechamente ligados. Conseguimos implementar polimorfismo en jerarquías de clasificación que se dan a través de la herencia. Por ejemplo, tenemos una clase vehículo y de ella dependen varias clases hijas como coche, moto, autobús, etc.



Ejemplo;

Tenemos la clase Parking. Dentro de ésta tenemos un método estacionar(). Puede que en un parking tenga que estacionar coches, motos o autobuses. Sin polimorfismo tendría que crear un método que permitiese estacionar objetos de la clase "Coche", otro método que acepte objetos de la clase "Moto" para estacionarlos, etc. Pero todos estaremos de acuerdo que estacionar un coche, una moto o un bus es bastante similar: "entrar en el parking, recoger el ticket de entrada, buscar un lugar, situar el vehículo dentro de ese lugar...".

Lo ideal sería que nuestro método nos permita recibir todo tipo de vehículos para estacionarlos, primero por reutilización del código, ya que es muy parecido estacionar uno u otro vehículo, pero además porque así si mañana el mercado trae otro tipo de vehículos, como una van, todoterreno hibrido, o una nave espacial, el software sea capaz de aceptarlos sin tener que modificar la clase Parking.

Gracias al polimorfismo, cuando declaramos la función estacionar() podemos decir que recibe como parámetro un objeto de la clase "Vehículo" y el compilador aceptará no solamente vehículos genéricos, sino todos aquellos objetos que hayamos creado que hereden de la clase Vehículo, es decir , coches, motos, buses, etc. Esa relajación del sistema de tipos para aceptar una gama de objetos diferente es lo que llamamos polimorfismo.

Declaro la función:

```
function estacionar ( Vehiculo ) { }
```

Invoco la función: (soporto polimorfismo)

```
estacionar ( Coche ) ;  
estacionar ( Moto ) ;  
estacionar ( Bus ) ;
```

No puedo invocar la función: (no lo permitiría, porque no ser clasificación de herencia de vehículos)

```
estacionar ( Mono ) ;  
estacionar ( INT ) ;
```

En el futuro si podría: (Si creo las clases "Van" o "Nave espacial" y heredan de Vehículo)

```
estacionar ( Van ) ;  
estacionar ( Nave espacial ) ;
```

En fin, esto es lo que significa polimorfismo.

El polimorfismo incorpora conceptos de **métodos Constructor , sobre escritura (override) , Abstract , Extends**, que incluimos en otro material complementario de la entrega anterior pero los volvemos a repasar cómo conceptos previamente para entender más claro estos videos tutoriales. Mirando el Power Point **Repasamos Conceptos**.

Link de video tipo tutorial sobre polimorfismos

<https://www.youtube.com/watch?v=RV2itxdmodM>

<https://www.youtube.com/watch?v=XITwgK9Lgbl>