**Características de la Programación Orientada a Objetos**

Las características principales que definen la programación orientada a objetos son:

* Abstracción
* Encapsulamiento
* Herencia
* Polimorfismo

**ABSTRACCIÓN**

La abstracción es la capacidad de extraer del mundo real las características (atributos y métodos) significativas que intervendrán en el programa.

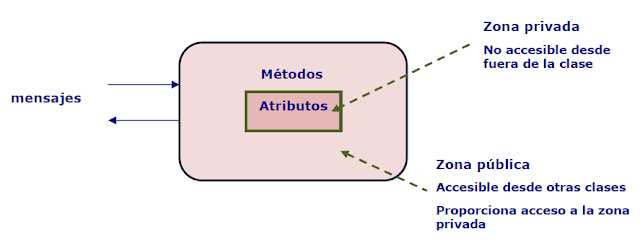
La clave de la POO está en abstraer los métodos y los datos comunes a un conjunto de objetos y almacenarlos en una clase.

Por ejemplo, para hacer una aplicación para el cálculo de nóminas la abstracción consiste en identificar todas las clases que van a intervenir y para cada clase especificar los atributos y métodos necesarios. En un programa de cálculo de nóminas una clase Trabajador podría tener como atributos el nombre o la fecha de nacimiento, pero no es necesario que tenga atributos como color de pelo, raza, religión, etc.

**ENCAPSULAMIENTO**

Esta característica permite ver un objeto como una caja negra que contiene en su interior toda la información relacionada con dicho objeto.

Mediante el encapsulamiento se separan los aspectos externos del objeto (las partes a las que pueden acceder otros objetos o zona pública) de los detalles de implementación internos (ocultos a otros objetos o zona privada).

[](https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEiVgvdFpvyTwERZP5vJTIyQSHCLZ4OvK8EadKsBn0HO9n8C1_6TuaGwl4XudmG3WjsphqhhAj9jfVZtEBWXgjxu8Eugn6T0ZhyV9Bqb_2GwVUwoSkiy28R3aj8pPY1jTIpHhHEsox29Yp5O-z0bLS7laKf7ox-6OzRMMTlouoro5qXmiEUllyUUtjwZ-w/s887/Java-encapsulamiento-POO.png)

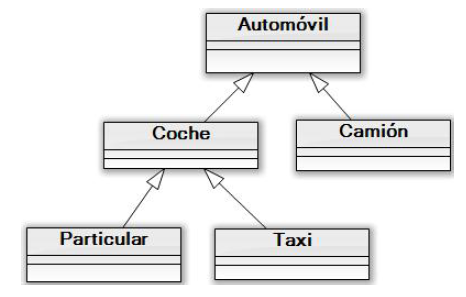
Los objetos se pueden manipular, pero su estructura interna permanecerá oculta a los demás. Sólo son visibles las partes necesarias para utilizar un objeto (interfaz pública). Las demás partes son ocultas (privadas). No es necesario saber cómo está hecho el objeto ni los datos que contiene, para poder utilizarlo.

**HERENCIA**

La herencia permite definir una clase a partir de otra ya existente. Una clase podrá heredar de otra un conjunto de propiedades (métodos y atributos) y a esa nueva clase se le puede añadir atributos o métodos, ocultar métodos heredados o redefinir métodos heredados.

Esta característica está muy relacionada con la reutilización de código en la POO.

Llamaremos clases derivadas a las que heredan de otra clase, a la que denominaremos clase base. A través de la herencia se crea una jerarquía de clases.

[](https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEgnA2uz7tKRu9vmph0rWj95kEMbwNA9bxk4ascWVy8YQ9RzYJh84wGSpP8RTcCVjme_k246-2W4wb0t6X_11SzPezIVK3hwLD_P-DNK7AgQ494SiJ5Ke3d-2g-TV5fTMei3H9-Z49kcPtu-KitVRocFg8R3MNJvaX_f-t9nZZpCzq7iCCqstblmFy7yPg/s466/JAVA-POO-Herencia.png)

**POLIMORFISMO**

Polimorfismo es la cualidad que tienen los objetos para responder de distinto modo a un mismo mensaje.