

PARADIGMAS DE PROGRAMACIÓN:

Un **paradigma de programación** es una manera o estilo de programación. Existen diferentes formas de diseñar un programa y varios modos de trabajar para obtener los resultados que necesitan los programadores. Por lo que un paradigma de programación se trata de un conjunto de métodos sistemáticos aplicables en todos los niveles del diseño de programas para resolver problemas.

PROGRAMACIÓN ORIENTADA A OBJETOS

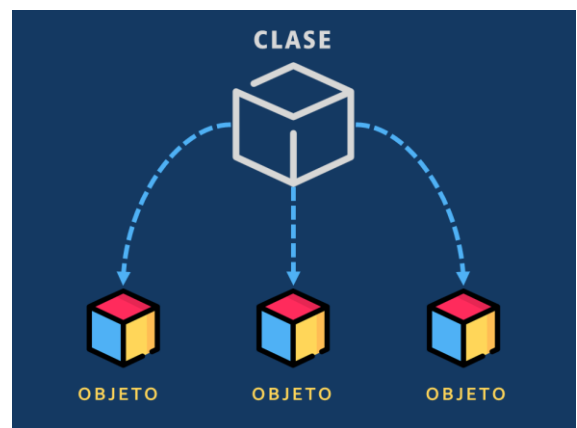
Los Objetos:

Si miramos a nuestro alrededor, vemos que estamos rodeados de objetos. Y cuando debemos desarrollar un nuevo software para resolver un determinado problema, verificamos que los elementos que componen el problema son objetos. Pensemos en un sistema de gestión de negocios. Al analizar las características del problema, nos encontramos con que podríamos tener objetos como: productos, facturas, remitos, depósitos, clientes, entre otros. Y no sólo objetos concretos, sino también abstractos, como ser: vencimientos e impuestos.

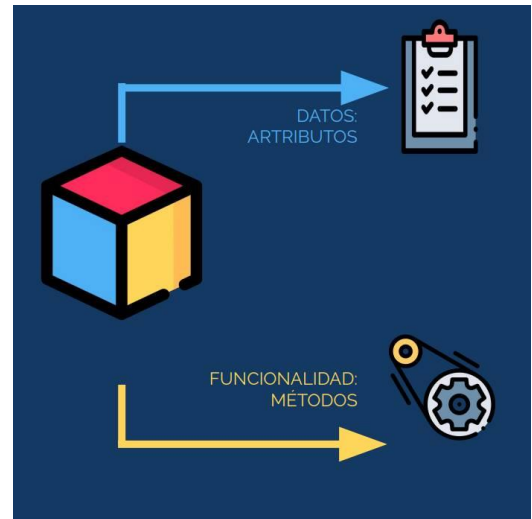
La programación orientada a objetos es un paradigma basado en la identificación de los objetos inherentes al problema y en la escritura del código que modele esos objetos, sus propiedades, sus comportamientos y la forma en que se relacionan para resolver el problema. Visto de esta manera, la programación orientada a objetos resulta opuesta a la programación imperativa clásica, donde los **programas son secuencias de instrucciones**.

Clases y Objetos:

Cuando se analiza un problema para resolverlo mediante las técnicas de programación estructurada, nos enfocamos en identificar los datos que debemos manipular y las transformaciones que sufren como parte de la solución. Con el paradigma orientado a objetos, la tarea de análisis se centrará en la identificación de los objetos, de sus características y de cómo se relacionan entre sí para resolver el problema.

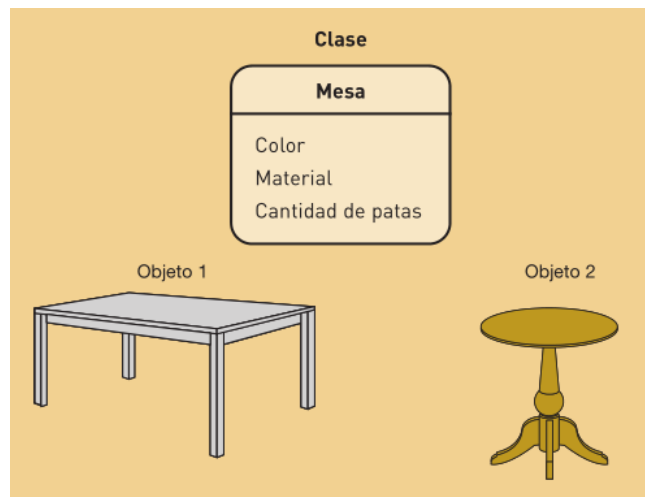


Ya dentro del contexto formal de la teoría de orientación a objetos, podemos definir un objeto como un concepto, abstracción o elemento con significado claro dentro del problema en cuestión (por ejemplo, un Vehículo). Como tal, un objeto se caracteriza por tener un estado, que es el conjunto de valores de sus propiedades en un momento del tiempo (por ejemplo: la marca del Vehículo, la cantidad de combustible que tiene el tanque). Además, todo objeto tiene un comportamiento, es decir, las acciones que puede realizar y que modificarán sus estados (Por ejemplo: un vehículo puede avanzar o retroceder). Por último, un objeto se caracteriza por tener identidad propia, esto es, por más que dos objetos tengan el mismo comportamiento y el mismo estado, resultan totalmente diferentes e identificables.



Según el diccionario, una clase es un grupo de elementos de un conjunto que tiene características comunes. Las técnicas de orientación a objetos se centran en la identificación de esos elementos comunes entre los objetos para poder agruparlos en clases y, así manipularlos fácilmente. Podemos decir que una clase es una abstracción de un grupo de objetos, porque no existe por sí sola.

Por ejemplo, pensemos en una mesa. Hay mesas rectangulares, mesas redondas, de madera, de metal, incluso de diferentes colores y tamaños. Todas son mesas distintas e identificables, es decir: objetos. Sin embargo, podemos verificar que todas tienen los mismos atributos, aunque con distintos valores. Todas tienen un color, un material, una forma, una dimensión; por lo tanto, podemos agruparlas bajo un mismo concepto: la mesa. Desde el punto de vista de la orientación a objetos, podemos decir que mesa es una clase.



Propiedades y Métodos:

Como decíamos, los objetos poseen atributos y comportamientos. Ya desde el punto de vista de la programación, los atributos de los objetos se traducen en propiedades de las clases que modelan.

El comportamiento en cambio, está representado por procedimientos y funciones que creamos dentro de la clase. En la jerga de la programación orientada a objetos, los procedimientos y funciones que creamos dentro de una clase se denominan métodos, y representan un conjunto de actividades que un objeto puede realizar, es decir, su comportamiento.

Volviendo a la definición de método, justamente podemos decir, que los métodos son los encargados de interceptar y enviar mensajes desde y hacia otros objetos y, al mismo tiempo, alterar el estado actual del objeto.

Identificación de los objetos que modelan un problema

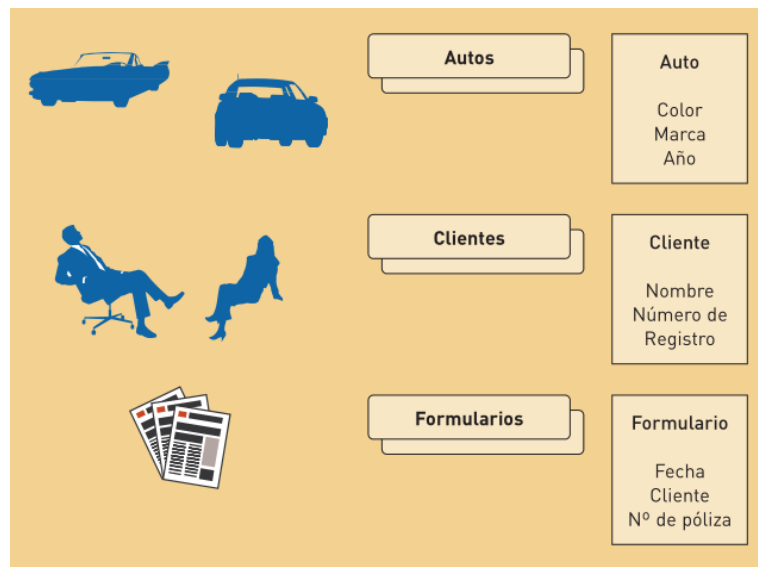
El primer paso para modelar un problema usando la POO es la identificación de los objetos que son relevantes para representar el problema. No todos los posibles objetos que podemos hallar son necesarios en nuestro modelo. Los primeros candidatos a ser objetos son los sustantivos (nombres comunes) en el enunciado de un problema. Por ejemplo, considere el siguiente problema:

Imaginemos que necesitamos implementar una aplicación de alquiler de autos, y nos presentan el siguiente requerimiento (resumido): La empresa dispone de autos para alquilar a sus clientes. Cuando se alquila un auto, el cliente firma una póliza de seguro. En un formulario de alquiler se registra la fecha, el nombre del cliente, el número de su registro de conductor y el número de su tarjeta de crédito.

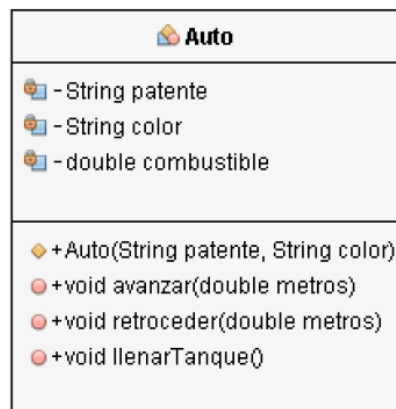
Si aplicamos la técnica de los sustantivos, podremos identificar rápidamente tres objetos fundamentales para la aplicación: *auto*, *cliente* y *formulario de alquiler*. Dependiendo del tipo de análisis y diseño que queramos hacer, podríamos considerar como objetos también a la *tarjeta de crédito* e incluso, el *alquiler*.

Para identificar a las propiedades aplicamos una técnica similar a los sustantivos, pero buscando cualidades o adjetivos que ya hemos identificado. Siguiendo con el ejemplo de alquiler de autos, podemos ver que el formulario de alquiler tiene propiedades como: la fecha, el nombre del cliente, etc.

Una vez que hemos identificado los objetos y sus propiedades estamos en condiciones de definir clases para agruparlos y abstraer todas las posibles instancias de ellos.



Si representamos con diagrama de clases UML a nuestro Auto, tendríamos lo siguiente:



Como podemos observar, las características de Auto (sus atributos), no son ni más ni menos que variables de un tipo determinado que almacenaran un estado; y su comportamiento (métodos) no son otra cosa que en otros lenguajes conocemos como procedimientos o funciones (si devuelven algo); y es allí en donde utilizaremos las estructuras de programación aprendidas como iteraciones, condicionales o simples estructuras secuenciales.