1. ¿Qué es la Programación Orientada a Objetos?

Es un método de desarrollo de software basado en el comportamiento de los objetos en el mundo real. La POO es una manera de plantearse la programación.

1. **¿Cómo se piensa en la POO?**

Pensar en términos de objetos es muy parecido a cómo lo haríamos en la vida real. Por ejemplo, un coche sería el objeto, las propiedades serían las características como el color o el modelo y los métodos serían las funcionalidades asociadas como ponerse en marcha o parar. Con una fracción, esta será el objeto y tendrá dos propiedades, el numerador y el denominador. Luego podría tener varios métodos como simplificarse, sumarse con otra fracción o número, restarse con otra fracción, etc.

1. **Conceptos claves de la POO**

**OBJETO:** Es una porción de código fuente que contiene datos y proporciona servicios.

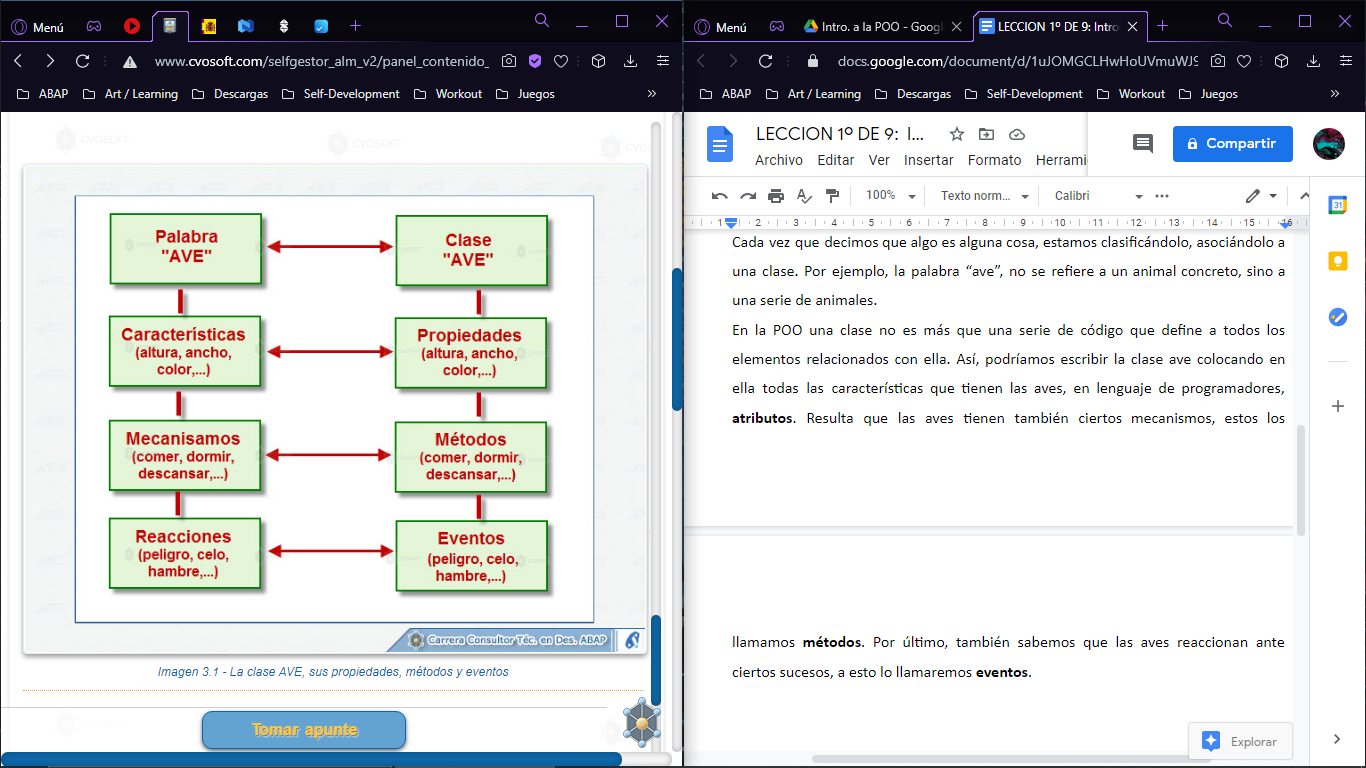
Un objeto es, por tanto, la representación en un programa de un concepto y contiene toda la información necesaria para abstraerlo: datos que describen sus atributos y operaciones que pueden realizarse sobre los mismos. Los servicios que proporciona el objeto se conocen como métodos.

Normalmente los métodos operan con los datos privados del objeto, de esta manera, los atributos de un objeto no pueden ser cambados directamente por el usuario del objeto, solo pueden ser combinados por los métodos de ese objeto. Así se garantiza la consistencia interna del objeto.

**CLASE:** Es un conjunto de objetos que tiene la misma estructura y comportamiento.

Cada vez que decimos que algo es alguna cosa, estamos clasificándolo, asociándolo a una clase. Por ejemplo, la palabra “ave”, no se refiere a un animal concreto, sino a una serie de animales.

En la POO una clase no es más que una serie de código que define a todos los elementos relacionados con ella. Así, podríamos escribir la clase ave colocando en ella todas las características que tienen las aves, en lenguaje de programadores, **atributos**. Resulta que las aves tienen también ciertos mecanismos, estos los llamamos **métodos**. Por último, también sabemos que las aves reaccionan ante ciertos sucesos, a esto lo llamaremos **eventos**.



**INSTANCIA:** Es un objeto de una clase en particular creado en tiempo de ejecución. Cada vez que creamos una nueva instancia, esta adquiere atributos, métodos y eventos de la clase a la que pertenece, sin embargo, cada instancia es independiente de las otras. Ventajas:

* Si hago algún cambio en la clase, todas las instancias de esta clase se actualizaran automáticamente.
* Al ser independientes de las otras instancias, puedo darles valores diferentes sin que afecten a las demás. Aunque comparten la misma estructura, pueden programarse individualmente, dando versatilidad y flexibilidad al código.

1. **Principales ventajas de la orientación a objetos**

* Sistemas de software complejos son más simples de comprender.
* Es posible realizar cambios al nivel de las clases, esto reduce el costo total del mantenimiento necesario.
* A través del **polimorfismo** y la **herencia** es posible la reutilización de componentes individuales. ***Polimorfismo*** *es la capacidad que tienen los objetos de una clase de responder al mismo mensaje o evento en función de los parámetros utilizados durante su invocación. Un objeto polimórfico es una entidad que puede contener valores de diferentes tipos durante la ejecución del programa. La* ***herencia***  *es la capacidad que tiene una clase de heredar las propiedades y métodos de otra clase.*
* La cantidad de trabajo en revisión y mantenimiento del sistema se reduce debido a que muchos problemas pueden ser detectados y corregidos en la fase de diseño.
* Facilita el trabajo en equipo debido a que varios desarrolladores pueden trabajar en distintos componentes de un mismo desarrollo.