



# Universidad Autónoma de Baja California



**Asignatura:** Programación con Python

**Alumno:** Rodríguez Gómez Alberto

**Docente:** Parra Galaviz Ray Brunett

**Grupo:** 372

**Semestre:** 2018-2

# Programación Orientada a Objetos en Python

La Programación Orientada a Objetos Cómo tal, nos enseña un método -probado y estudiado- el cual se basa en las interacciones de objetos para resolver las necesidades de un sistema informático.

Los elementos de la POO, pueden entenderse como los materiales que necesitamos para diseñar y programar un sistema, mientras que las características, podrían asumirse como las herramientas de las cuáles disponemos para construir el sistema con esos materiales.

## Clases

Las clases son los modelos sobre los cuáles se construirán nuestros objetos.

En Python, una clase se define con la instrucción `class` seguida de un nombre genérico para el objeto.

## Propiedades

Las propiedades, como hemos visto antes, son las características intrínsecas del objeto. Éstas, se representan a modo de variables, solo que técnicamente, pasan a denominarse propiedades

## Métodos

Los métodos son funciones, solo que técnicamente se denominan métodos, y representan acciones propias que puede realizar el objeto. El primer parámetro de un método, siempre debe ser **SELF**.

## Objeto

Las clases por sí mismas, no son más que modelos que nos servirán para crear objetos en concreto. Podemos decir que una clase, es el razonamiento abstracto de un objeto, mientras que el objeto, es su materialización. A la acción de crear objetos, se la denomina instanciar una clase y dicha instancia, consiste en asignar la clase, como valor a una variable.

## **Herencia**

Algunos objetos comparten las mismas propiedades y métodos que otro objeto, y además agregan nuevas propiedades y métodos. A esto se lo denomina herencia: una clase que hereda de otra. Vale aclarar, que en Python, cuando una clase no hereda de ninguna otra, debe hacerse heredar de object, que es la clase principal de Python, que define un objeto.

## **Polimorfismo**

La palabra viene del griego y significa varias formas. El polimorfismo es la habilidad de responder a una misma función de manera distinta.

## **Encapsulación.**

Lo que hace el encapsulamiento es impedir la visualización o acceso de las variables de manera directa convirtiendo la variable en privada. Para lograr esto en Python el nombre de la función o variable al ser declarado debe comenzar con doble guion bajo (\_\_). Esto bastará para que lo declarado sea reconocido como privado y no se pueda acceder.

## **Bibliografía**

<https://frontendlabs.io/1305--tutorial-basico-de-python-parte-iv-programacion-orientada-a-objetos>

<http://www.iac.es/sieinvens/python-course/source/poo.html>

[https://librosweb.es/libro/python/capitulo\\_5/programacion\\_orientada\\_a\\_objetos.htm](https://librosweb.es/libro/python/capitulo_5/programacion_orientada_a_objetos.htm)  
!