

Introducción a la programación con Python + Flask

Disertante:

- **Tec.** Angel Francisco Schulze Bidegorry

Duración:

- **3 MESES**

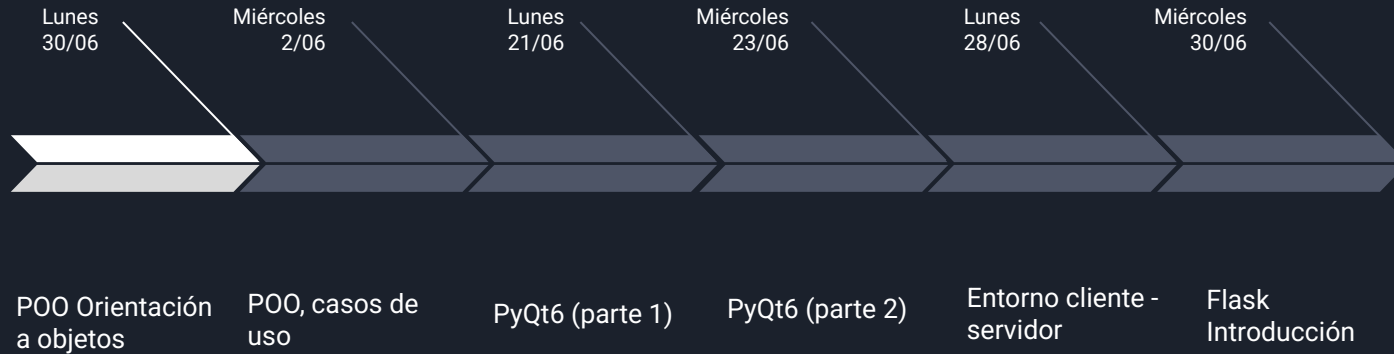
Horarios:


- **Lunes:** 16:00 ~ 18:00
- **Miércoles:** 16:00 ~ 18:00





Cronología del curso



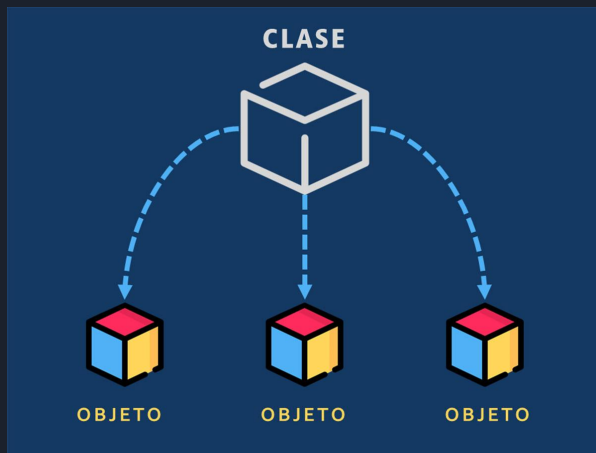
The background is a dark blue-grey gradient. On the left, there are two overlapping geometric shapes: a blue parallelogram and a light green parallelogram. Below these, a circular inset shows a detailed, high-contrast image of a printed circuit board (PCB) with various electronic components. In the top right corner, there is a faint, stylized pattern of interconnected lines and squares, resembling a circuit or a data network.

Programación Orientada a Objetos

¿Qué es la Programación Orientada Objetos?

La Programación Orientada a Objetos (POO) es un paradigma de programación que organiza el software en objetos.

Cada objeto es una instancia de una clase, y la clase define las características y comportamientos que esos objetos tendrán.



Clases y Objetos


Una **clase** es como un plano o plantilla para crear objetos. Define atributos (propiedades) y métodos (funciones) que los objetos de esa clase tendrán.

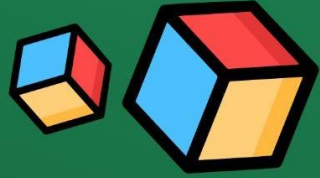
Un **objeto** es una instancia de una clase. Puedes tener múltiples objetos de la misma clase, pero cada uno puede tener diferentes valores para sus atributos.

Clase: *Coche*

Clase Coche	← nombre de la clase
arrancar, ir, parar, girar	← métodos (funciones)
color, velocidad, carburante	← atributos (datos)

♦ Objeto: *Ferrari*

coche.ferrari	← nombre del objeto
	← métodos arrancar, ir, parar, girar
	← datos rojo, 280 km/h, lleno



¡PILARES DE LA PROGRAMACIÓN ORIENTADA A OBJETOS!

ABSTRACCIÓN



Es el proceso de **definir los atributos y los métodos** de una clase.



ENCAPSULAMIENTO



Protege la información de manipulaciones no autorizadas.

POLIMORFISMO



Da la misma orden a varios objetos para que respondan de diferentes maneras.

HERENCIA



Las **clases hijo heredan atributos y métodos** de las clases padre.

Estos pilares definen una forma de estructurar el código, haciendo más fácil de leer, comprender, re-utilizar, y extender.

De esta manera se pueden modularizar mejor las soluciones al problema planteado, verificar posibles errores en secciones más pequeñas de código, y mejorar la eficiencia del desarrollo del software.

POO - Video



[Ir al video >>](#)



Creación de clases en Python

`__init__()` : el uso de esta palabra es propia de Python para referirnos al constructor de la clase.

A diferencia de otros lenguajes solamente se puede crear un único constructor.

self: se utiliza en los métodos de una clase para referirse al objeto actual que está invocando ese método. Es un primer argumento que debe ser incluido en todos los métodos de una clase (excepto en los métodos estáticos)

El uso de **doble guión bajo** antes del nombre de un atributo, lo vuelve 'privado'. Solo puede ser visible dentro de la misma.