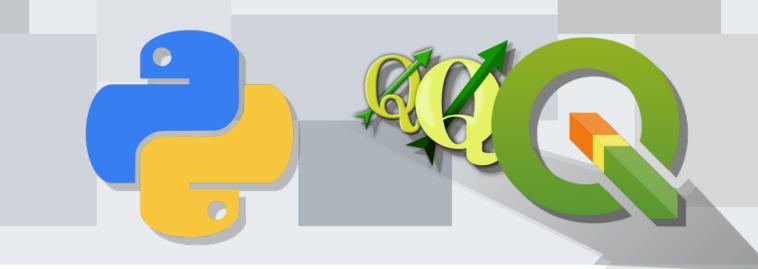


#### Universidad Autónoma del Estado de México Facultad de Geografía





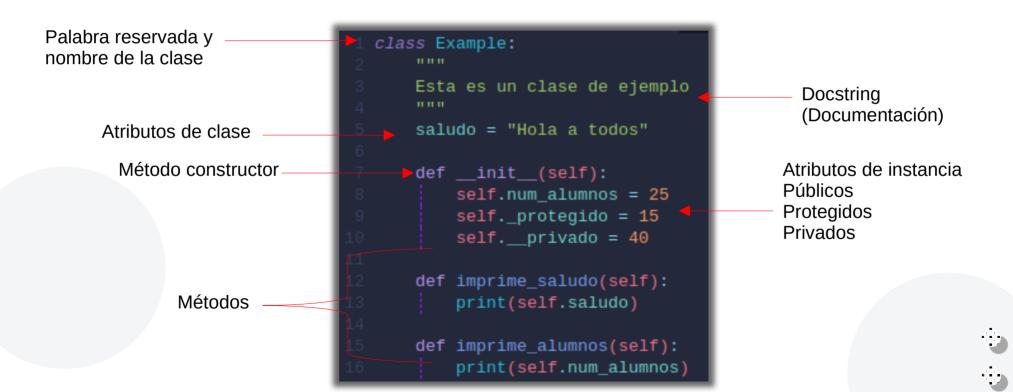
#### Programación con Python

Orientado a la automatización de procesos en QGIS

Ing. Adonai Emmanuel Nicanor Bautista

- La Programación Orientada a Objetos es un paradigma de programación y Python es un lenguaje multiparadigma.
- **Clase:** Es un tipo de "plantilla" que permite crear diferentes objetos con ciertas características en común. La creación de un objeto se llama **Instancia.**
- Objetos: Son un tipo de dato el cual contiene atributos y métodos.
- Atributos (Clase o instancia): Pueden ser cualquier otro tipo de datos como números, strings, etc. Ademas hay tres tipos: Públicos, protegidos (\_) y privados(\_\_).

- **Métodos:** Son funciones interactúan con los objetos y los atributos que contiene, permitiendo generar nueva información.
- Las clases se crean con la palabra reservada **class** seguido del nombre de la clase, la cual debe iniciar con **Mayúscula**.
- El Método **Constructor** sirve para inicializar valores cuando la clase ha sido instanciada, así como realizar los procesos y operaciones necesarias para generar objetos, este método puede recibir o no valores. Este método siempre sera definido como **\_\_init\_\_**.
- La palabra reservada **self** permite tener acceso a los atributos y métodos dentro de una clase.
- Con **del** podemos eliminar una instancia.



- La **Herencia (Extend)** sirve para aprovechar el máximo código posible en objetos con estructuras similares. Como su nombre lo indica, hereda propiedades de una clase a otra.
- La clase que brinda estas propiedades se denomina Clase Padre.
- La nueva clase (**Clase Hija**) puede incluir atributos y métodos adicionales a los que le han sido heredados.
- Para acceder a los atributos de la clase padre, es necesario usar la palabra reservada **super**.
- También existe la herencia múltiple, es decir, una clase hija puede incluir distintas clases padre, dando prioridad de izquierda a derecha según sean declaradas.

```
class Curso(Example):
class Example:
    saludo = "Hola a todos"
                                      def __init__(self):
                                          super(). init ()
    def init (self):
        self.num alumnos = 25
                                      def nom_curso(self):
                                          self.n curso = "Programación con Python"
    def imprime saludo(self):
        print(self.saludo)
                                      def imprime_curso(self):
                                          print(self.n_curso)
    def imprime alumnos(self):
        print(self.num alumnos)
```