

# Clases en Python

José David Ruiz Álvarez

josed.ruiz@udea.edu.co

Instituto de Física, Facultad de Ciencias Exactas y Naturales

**Universidad de Antioquia**

10 de marzo de 2020

## 1. Clases de Python: Programación Orientada a Objetos

Las clases son objetos de alto nivel de Python que permiten una mayor flexibilidad en términos de los elementos del objeto y que se adaptan especialmente a algunos problemas de computación. En general permiten un código más portable, menos susceptibles a errores del usuario y más compacto en términos de líneas de código y conceptualmente.

Declaración de una clase y sus elementos:

```
class NombreDeLaClase:

    #Atributos

    atributo1=0.1

    atributo2=1.34


    #Instancias

    def __init__(self, variable1, variable2, variable3):

        self.Instancia1=variable1*variable2

        self.Instancia2=variable1/variable2


    #Metodos

    def UnaFuncion(self, VariableX,VariableY):

        self.Instancia1=self.Instancia2*VariableX/VariableY

        return VariableX-VariableY
```

Una clase puede tener atributos, simples variables asociadas a la clase, instancias, elementos de la clase de tipo variable que son inicializados cuando se instancia una clase, y también puede tener métodos, que son elementos de la clase de tipo función y que ejecutan acciones bien sea sobre los mismos miembros de la clase

u otros. Así pues los elementos/miembros de una clase pueden ser: atributos, instancias o métodos.

Instanciar una clase es crear un objeto del tipo la clase en cuestión. La acción de instanciar una clase asigna a una variable la estructura de ella con todos sus elementos que pueden ser accedidos a través del operado “.”

```
A=NombreDeLaClase(valor1,valor2,valor3) #El valor1 es asignado a la variable1, valor2 a la variable2, e

print A.atributo1
print A.Instancial
print A.UnaFuncion(ValorX,ValorY) #El ValorX es asignado a la VariableX, ValorY a la VariableY
```

**Ejemplo 1:** Definir una clase para describir una partícula

```
class Particle:
    #Atributos
    cargada = True

    #Instancias (metodos)
    def __init__(self, x, y, z, vx, vy, vz, m, carga): #Funcion que se aplica sobre la classe misma (se
        self.X = x
        self.Y = y
        self.Z = z
        self.VX = vx
        self.VY = vy
        self.VZ = vz
        self.M = m
        self.Carga = carga
```

**Ejercicio 2:** Defina métodos para describir la cinemática de la clase partícula y que responda a la cinemática de una partícula puntual sujeta a una fuerza.

**Ejercicio 3:** En otro código utilice la clase partícula escrita anteriormente como librería y describa el movimiento de una partícula cargada en un campo magnético. Realice un gráfico en uno de los planos del sistema coordenado de la posición de la partícula.

**Tarea (Fecha de entrega: 10 de marzo a más tardar a las 23:59 hora colombiana):** Cree un sistema de dos partículas de carga opuesta y con la misma masa inmersas en un campo magnético constante en la dirección z. Asuma que las partículas tienen velocidad inicial 0. Adicionalmente asuma que

la posición inicial de la partícula 1 es  $(0, 0, 0)$  y la partícula 2 en  $(1, 0, 0)$ . Dicho sistema está regido por la fuerza de Lorentz. Los entregables son el código desarrollado y un gráfico de la posición de ambas partículas después de 10000 iteraciones con un paso de 0.01 , masa de las partículas de 10.0, y campo magnético de 10.0. Debe subirse a la carpeta “Seguimiento1/SuNumeroDeCedula” del repositorio central con el nombre “SegClasesPython.py” y “SegClasesPython.png”. Resuelva el mismo problema en C++ y utilice herramientas de graficación de Python. Entregables “SegClasesCpp.cpp” y “SegClasesCpp.png”.