

Taller  
Derivadas Numéricas y Raíces de funciones  
Herramientas Computacionales  
2016661

Ricardo Amézquita  
Departamento de Física  
Universidad Nacional de Colombia  
Sede Bogotá

28 de septiembre de 2018

## Taller de programación

Escriba un modulo llamado taller.py en la cual se definan la siguientes clases:

```
class Derivada:
    def __init__(self, f, metodo ="adelante", dx= 0.001):
        """Clase usada para calcular derivadas numéricas de una función
        Argumentos:
        f: Función a la que se le va a evaluar la derivada numérica
        metodo: Cadena de caracteres que contiene el método a usar para
        calcular la derivada numérica. Los valores validos son los
        siguientes:
        "adelante": Calcular la derivada usando el método de diferencias
        hacia adelante.
        "central": Calcula la derivada usando el método de la diferencia
        central.
        "extrapolada": Calcula la derivada usando el método de la
        diferencia extrapolada.
        "segunda": Calculo de la segunda derivada
        dx: Delta usado para el calculo de la derivada
        """
        .....
    def calc(self,x):
        """ Método que retorna el valor numérico de la derivada de la
        función f evaluada en el punto x"""
        ...
```

```

class Zeros:
    def __init__(self, f, metodo, error=1e-4, max_iter=100):
        """Clase que retorna el cero de una función
Argumentos:
f: Función a la que se le va a evaluar el cero
metodo: Cadena de caracteres que contiene el método a usar para la
búsqueda del cero. Los valores validos son los siguientes:
"newton": Usa el método de newton para el calculo.
"bisectriz": Usa el método de la bisectriz para el calculo.
"interpolacion": Usa el método de la interpolación para el el
calculo
"newton-sp": Usa la función newton definida en scipy para el
calculo.
"fsolve-sp": Usa la función fsolve definida en scipy para el
calculo.
"brentq-sp": Usa la función brentq definida en scipy para el
calculo.
error: Valor máximo que puede tener f(x0) donde X0 es el cero encontrado
numéricamente. Para las funciones de scipy, este es el valor pasado
a xtol.
max_iter: Numero máximo de iteraciones a usar en el calculo.
"""
    ....
    def zero(self,vi):
        """Método que retorna el valor del x0 encontrado
Argumento:
vi: Valor inicial a usar en la búsqueda del cero. Para los algoritmos
que usen un solo punto, debe ser un flotante, para los algoritmos
que usen 2 puntos, debe ser una tupla (a, b) de flotantes.
"""
    ....

```

Adicionalmente el modulo debe tener una sección de ejecución como programa, donde se muestre un ejemplo del uso de las 2 clases. Esto quiere decir, debe tener algo del estilo a:

```

if __name__ == "__main__":
    .... código con el ejemplo

```

## Nota:

1. Los programas deben subirse en en la raíz del repositorio asignado de github-classroom (no en carpetas).