

# ESCUELA DE INGENIERÍA

## Programa Ingeniería de Sistemas

Modalidad - Presencial

**Curso: Programación Orientada a Objetos** 

<u>Proyecto Series</u> <u>Funciones / Métodos en Python</u>

### PROYECTO SEGUNDO CORTE

El ejercicio plantea la solución a un problema utilizando el lenguaje Python, implementando el patrón de arquitectura Modelo-Vista-Controlador (MVC). El objetivo es crear un programa que satisfaga las necesidades expuestas. Como parte del aprendizaje, se solicita gestionar el proyecto en un versionador de código, como GitHub u otra plataforma similar. Esto permitirá desarrollar buenas prácticas en el control de versiones y colaboración en futuros proyectos. Este ejercicio servirá como base para futuras implementaciones, facilitando el desarrollo de aplicaciones organizadas, mantenibles y escalables.

### ENERGÍA ELÉCTRICA

#### SERIE DE TAYLOR

En matemática, una **serie de Taylor** es una aproximación de funciones mediante una serie de potencias o suma de potencias enteras de polinomios como  $(x-a)^n$  llamados términos de la serie, dicha suma se calcula a partir de las derivadas de la función para un determinado valor o punto a suficientemente derivable sobre la función y un entorno sobre el cual converja la serie. A la serie centrada sobre el punto cero, es decir, cuando a=0, se le denomina también **serie de Maclaurin**.

### PYTHON PROJECT SERIES DE TAYLOR - ESCENARIO

Lista de Series de Maclaurin de algunas funciones comunes:

La función exponencial  $e^x$  tiene como serie de Maclaurin  $e^x = \sum_{n=0}^\infty \frac{x^n}{n!} =$ 

$$e^x = \sum_{n=0}^{\infty} rac{x^n}{n!} = 1 + x + rac{x^2}{2!} + rac{x^3}{3!} + \cdots$$

Las funciones trigonométricas usuales y sus inversas tienen como series de Maclaurin:

UNIMINUTO BOGOTÁ

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022-2000

2022

$$\sin x = \sum_{n=0}^{\infty} rac{(-1)^n}{(2n+1)!} \; x^{2n+1} \quad ext{para toda } x$$

$$\cos x = \sum_{n=0}^{\infty} rac{(-1)^n}{(2n)!} \; x^{2n} \quad ext{para toda } x$$

$$rcsen \ x = \sum_{n=0}^{\infty} rac{(2n)!}{4^n (n!)^2 (2n+1)} \ x^{2n+1} \quad ext{para} \ |x| < 1$$

$$\arccos x = \frac{\pi}{2} - \arcsin x$$

$$ext{senh } x = \sum_{n=0}^{\infty} rac{x^{2n+1}}{(2n+1)!} \quad ext{para toda } x$$

$$\cosh x = \sum_{n=0}^{\infty} \frac{x^{2n}}{(2n)!}$$
 para toda  $x$ 

### **FUNCIONALIDAD**

Es importante considerar que no se permite el uso de funciones nativas o integradas del lenguaje Python. Todas las funcionalidades requeridas deben ser implementadas manualmente utilizando exclusivamente estructuras de control básicas, operadores y tipos de datos primitivos del lenguaje.

La aplicación o software desarrollado debe calcular los siguientes procesos desde un ambiente de terminal y web:

- 1.  $e^x$
- 2. sen(x)
- 3.  $\cos(x)$
- 4. arcsen(x)
- 5. arccos(x)
- 6. senh(x)
- 7.  $\cosh(x)$

### **ENTREGABLES**

- 1. Cargar el proyecto en un software versionador, donde se evidencie su uso.
- 2. Hacer la entrega del código de la aplicación o software al aula virtual de curso.
- 3. Generar un paquete en el proyecto Python llamado docs donde se entregue un informe de la realización del proyecto. Análisis, Diseño y codificación.

Se aplica el siguiente instrumento de calificación únicamente a los trabajos entregados en aula virtual en el apartado y en los tiempos estipulados.

1. La calificación resulta de dos momentos:

2. El entregable en grupo que se califica de 1.0 a 5.0.

3. La defensa la podrá realizar aquellos que hayan realizado la entrega, y se calificará de 0,0 a 5,0.

Ejemplo: Entrega perfecta calificación 5,0.

Defensa perfecta calificación 1,0

Calificación final del trabajo =  $E*D \Rightarrow 5.0 * 1.0 = 5.0$ 

### RUBRICA DE LA REVISIÓN DE ALGORITMOS

La rúbrica de calificación es una herramienta de evaluación que define de manera clara y objetiva los criterios y niveles de desempeño para calificar la entrega de los estudiantes. Incluye criterios específicos que serán evaluados, como la claridad de la argumentación, el uso de fuentes y la profundidad del análisis, junto con distintos niveles de desempeño que describen la calidad del trabajo, desde excelente hasta deficiente. Además, asigna puntajes o escalas para cada nivel, permitiendo una evaluación más equitativa y transparente. Su uso facilita la comprensión de las expectativas académicas y proporciona retroalimentación estructurada para que los estudiantes mejoren su desempeño.

RUBRICAS DE ENTREGA								
Entrega	Incumplimiento en lo solicitado en un 90% falta de correspondencia	Cumple con 30% de lo solicitado en la entrega, de forma como de diseño.	Cumple con un 50% de lo solicitado en la entrega, de forma y diseño	Cumple con 70% de Diseño y la forma en la entrega tiene o no falencias.	Cumple con todo lo solicitado, puede tener faltas de forma.			
Calificación	0,0 a 1,0	1,1 a 2,0	2,1 a 3,0	3,1 a 4,0	4,1 a 5,0			

RUBRICAS DE DEFENSA								
Entrega	Se realizan hasta	Responde dejando	La respuesta es	La respuesta es				
	tres preguntas y las	dudas de fondo se	objetiva, pero	contundente y no				
	respuestas no	realiza hasta tres	deja espacio para	deja dudas, asiste				
	tienen contexto.	preguntas.	realizar otra.	a tutorías.				
Calificación	0,0 a 0,2	0,3 a 0,5	0,6 a 0,8	0,9 a 1,0				