

## **PROGRAMACIÓN ORIENTADA A OBJETOS**

**COORPORACION UNIVERSITARIA MINUTE DE DIOS**

**INGENIERÍA DE SISTEMAS**

**CÁLCULO DE FUNCIONES MEDIANTE SERIES DE TAYLOR EN PYTHON**

**Esteban Crespo Gonzalez**

**NRC:**

**73392**

**12/06/2025**

## 1. INTRODUCCIÓN

El presente proyecto tiene como finalidad implementar una aplicación en el lenguaje Python que permita calcular aproximaciones de funciones matemáticas mediante el uso de series de Taylor (específicamente Maclaurin). Para ello, se ha utilizado el patrón de arquitectura Modelo-Vista-Controlador (MVC), facilitando la organización, escalabilidad y mantenibilidad del código.

Las funciones fueron implementadas sin el uso de librerías o funciones nativas de Python como `math.sin` o `math.exp`, cumpliendo con las restricciones del ejercicio. Asimismo, el control de versiones se gestionó mediante un repositorio Github, promoviendo buenas prácticas en el desarrollo colaborativo.

## 2. ANÁLISIS DEL PROBLEMA

Se requiere una aplicación que calcule, desde un entorno terminal, los valores aproximados de las siguientes funciones usando sus respectivas series de Taylor:

- $e^x$
- $\text{sen}(x)$
- $\text{cos}(x)$
- $\text{arcsen}(x)$
- $\text{arccos}(x)$
- $\text{senh}(x)$
- $\text{cosh}(x)$

El número de términos utilizados para la aproximación se estableció por defecto en 100, con excepción de las funciones  $\text{arcsen}$  y  $\text{arccos}$ , que requieren fórmulas más complejas y específicas para su cálculo. Restricciones:

- No utilizar funciones matemáticas integradas.
- Uso exclusivo de estructuras de control básicas y tipos primitivos.
- Separación lógica mediante el patrón MVC.

### 3. DISEÑO DEL SISTEMA

#### PATRÓN ARQUITECTÓNICO: MVC

- Modelo (modelo/Modelo.py)

Encargado del cálculo matemático de cada función. Implementa operaciones básicas como potencia y factorial, así como las series de Maclaurin para cada función.

- Vista (vista/Vista.py)

Se encarga de la interacción con el usuario en el terminal. Presenta el menú, solicita datos y muestra los resultados.

- Controlador (controlador/Controlador.py)

Conecta la vista con el modelo. Toma las entradas del usuario, invoca las funciones correspondientes del modelo y muestra el resultado por pantalla.

- Principal (main.py)

Archivo de entrada del programa que inicializa la ejecución.

#### **4. DESCRIPCIÓN DE LA CODIFICACIÓN**

Dado que existía la restricción de no utilizar funciones propias del lenguaje Python, como el operador `**` para elevar un número a una potencia o la función integrada para calcular factoriales, se optó por implementar manualmente estas funciones. Esto permitió cumplir con los requisitos del proyecto y, además, facilitó la reutilización del código en los distintos cálculos requeridos mediante las series de Taylor.

#### **5. CONCLUSIONES**

- La implementación del patrón de arquitectura MVC permitió estructurar el proyecto de forma clara y modular, facilitando su comprensión y mantenimiento.
- Todas las funciones fueron desarrolladas a partir de las series de Maclaurin, lo que permitió obtener aproximaciones precisas sin recurrir a funciones nativas del lenguaje.
- La restricción de no utilizar funciones integradas fomentó una mayor comprensión del funcionamiento interno de los algoritmos matemáticos, ya que fue necesario construir manualmente funciones como potencia y factorial.

#### **6. REPOSITORIO**

Repositorio del proyecto GitHub:

<https://github.com/VoicedMist2/Taylor>