



## **INSTITUTO TECNOLÓGICO DE CULIACÁN**

### **Materia:**

**Tópicos de IA**

### **Tarea 1:**

### **Documentación**

**Red neuronal de operaciones aritméticas**

### **Profesor:**

**Zuriel Dathan Mora Felix**

### **Integrantes:**

**Rafael Plata Angulo**

**Anette Leticia Robles Zamora**

### **Fecha:**

**29/05/2025**

## **Introducción:**

Este proyecto implementa una calculadora de operaciones aritméticas básicas (suma, resta, multiplicación y división) usando un modelo de inteligencia artificial entrenado con un dataset sintético. Incluye scripts para generar el dataset, entrenar el modelo y una interfaz gráfica para el usuario final.

## **Estructura de Archivos:**

genera\_dataset.py: Genera un dataset de operaciones aritméticas de un dígito.

generaModelo.py: Entrena y guarda un modelo de red neuronal con el dataset generado.

app.py: Interfaz gráfica (GUI) que utiliza el modelo entrenado para predecir resultados de operaciones.

dataset\_operaciones.csv: Dataset generado con operaciones y resultados.

## **GENERA\_DATASET.PY**

### **Propósito:**

Genera un archivo CSV con 10,000 operaciones aritméticas aleatorias de un dígito y su resultado.

### **Funcionamiento:**

Usa números aleatorios entre 0 y 9 para los operandos a y b.

Selecciona aleatoriamente una operación entre +, -, \*, /.

Evita divisiones por cero cambiando b a 1 si la operación es / y b es 0.

Calcula el resultado usando eval().

Guarda los datos en dataset\_operaciones.csv con las columnas: a, op, b, result.

## GENERAMODELO.PY

### Propósito:

Entrena un modelo de red neuronal para predecir el resultado de operaciones aritméticas básicas.

### Funcionamiento:

Carga el dataset generado por `genera_dataset.py`.

### Preprocesamiento:

Codifica el operador (op) usando `LabelEncoder`.

Normaliza los operandos y el resultado.

Divide los datos en conjuntos de entrenamiento y prueba.

### Modelo:

Red neuronal secuencial con dos capas ocultas de 64 neuronas (activación `ReLU`).

Capa de salida con una neurona (regresión).

Optimiza con `Adam` y usa error cuadrático medio como función de pérdida.

Entrenamiento:

Entrena por 1000 épocas con validación interna.

Guardado:

Guarda el modelo entrenado como `modelo_aritmetico.h5`.

## APP.PY

### Propósito:

Provee una interfaz gráfica para que el usuario ingrese una operación y obtenga el resultado estimado por IA.

### Funcionamiento:

Carga el modelo entrenado (modelo\_aritmetico.h5).

Codifica el operador de la entrada del usuario igual que en el entrenamiento.

Normaliza la entrada antes de predecir y desnormaliza la salida.

Muestra el resultado estimado en la interfaz.

Permite calcular tanto con el botón como presionando Enter.

### Estructura:

Columnas: a (operando 1), op (operador), b (operando 2), result (resultado de la operación).

10,000 filas de operaciones aleatorias.

### Paquetes:

numpy, pandas, scikit-learn, tensorflow, tkinter.

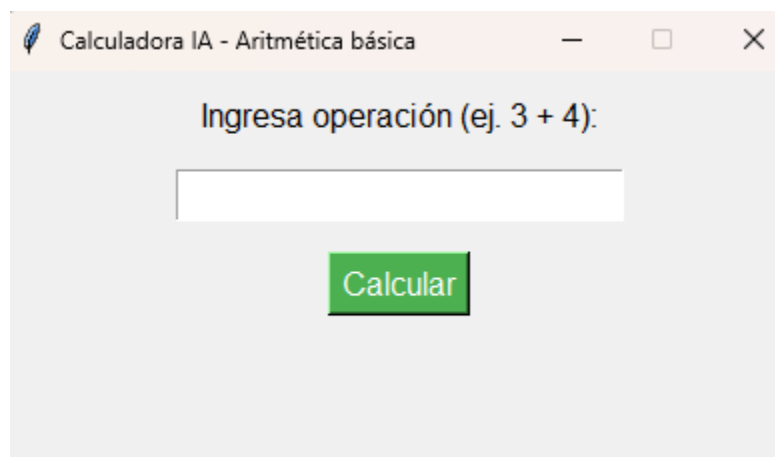
Ejecución

Generar el dataset:

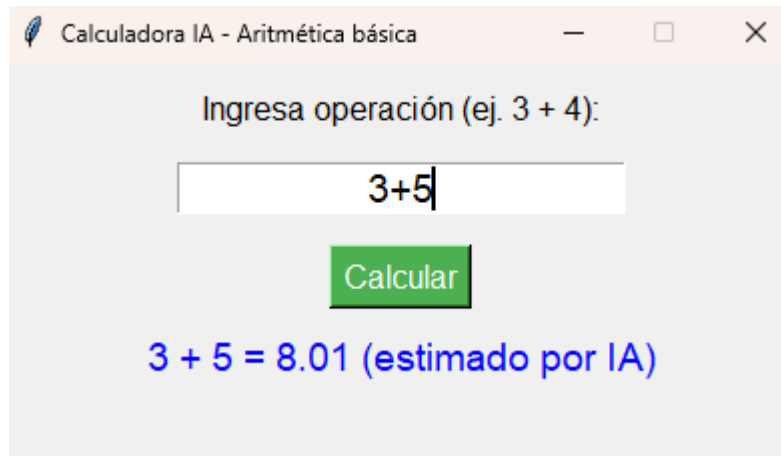
Entrenar el modelo:

Ejecutar la interfaz gráfica:

### Interfaz ejemplos:



### SUMA:



Calculadora IA - Aritmética básica

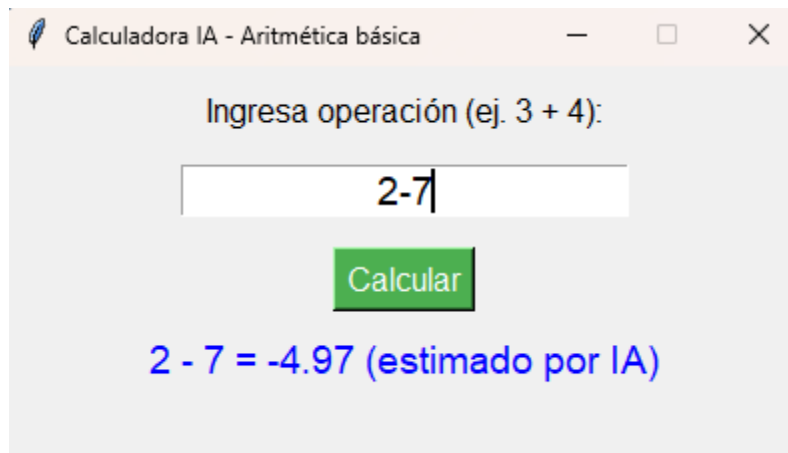
Ingresa operación (ej. 3 + 4):

3+5

Calcular

3 + 5 = 8.01 (estimado por IA)

### RESTA:



Calculadora IA - Aritmética básica

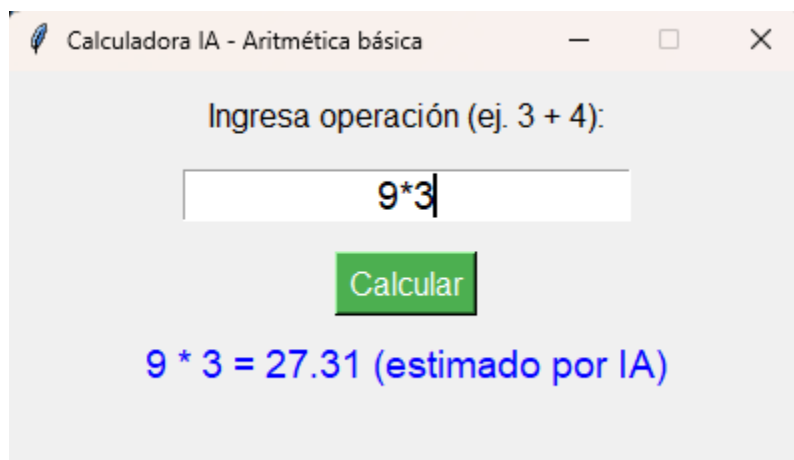
Ingresa operación (ej. 3 + 4):

2-7

Calcular

2 - 7 = -4.97 (estimado por IA)

### MULTIPLICACIÓN:



Calculadora IA - Aritmética básica

Ingresa operación (ej. 3 + 4):

9\*3

Calcular

9 \* 3 = 27.31 (estimado por IA)

## DIVISIÓN:

Calculadora IA - Aritmética básica

Ingresa operación (ej. 3 + 4):

5/2

Calcular

5 / 2 = 2.47 (estimado por IA)

Calculadora IA - Aritmética básica

Ingresa operación (ej. 3 + 4):

6/0

Calcular

Error: división por cero