



# UNIVERSIDAD BANCARIA DE MÉXICO

*"Constancia Unidad y trabajo"*

## INGENIERÍA EN SISTEMAS COMPUTACIONALES

RECONOCIMIENTO DE VALIDEZ OFICIAL DE ESTUDIOS DE LA SECRETARÍA  
DE EDUCACIÓN PÚBLICA No. 2022241 DE FECHA 13 DE SEPTIEMBRE DE 2002.

NOMBRE DE LA MATERIA:

Métodos numéricos

NOMBRE DEL PROFESOR(A) :

Mauricio Gómez Gallegos

CUATRIMESTRE:

6to Cuatrimestre

TÍTULO DEL TRABAJO O INVESTIGACIÓN:

Documentación de programa

Regresión Lineal con coordenadas bidimensionales

NOMBRE DE ALUMNO(S) :

Francisco de Jesus Pincle Puente

FECHA DE ENTREGA:

06 de junio 2025



# PROGRAMA: Regresión lineal con coordenadas bidimensionales

## Introducción

Este programa es una herramienta interactiva que permite a los usuarios **introducir puntos (x, y)** o **expresiones matemáticas** (como  $e^2$ ,  $\sin(\pi/2)$ ) para calcular la **recta de regresión lineal** que mejor se ajusta a esos puntos. La aplicación incluye:

- Interfaz gráfica con estilo oscuro "RGB gamer".
- Evaluación matemática segura de expresiones.
- Tabla con los datos utilizados (x, y, xy,  $x^2$ ).
- Visualización gráfica de los puntos y la recta resultante.
- Ecuación final mostrada de forma clara.

---

## Tecnologías Usadas

- Tkinter: Para la interfaz gráfica.
- NumPy: Para cálculos matemáticos eficientes.
- Matplotlib: Para graficar los resultados.
- math y re: Para evaluar expresiones matemáticas de forma segura.

```
1  # Importamos las librerías necesarias para la interfaz y los cálculos
2  import tkinter as tk # Para crear la interfaz gráfica
3  from tkinter import ttk # Para widgets con estilos (botones, tablas, etc.)
4  from tkinter import messagebox # Para mostrar mensajes emergentes (errores, alertas)
5  import numpy as np # Para manejar arreglos y operaciones matemáticas
6  import matplotlib.pyplot as plt # Para graficar
7  from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg # Para insertar gráficas en Tkinter
8  import math # Para funciones matemáticas como e, sin, cos, etc.
9  import re # Para expresiones regulares
```

Estas librerías permiten crear la interfaz, manejar expresiones matemáticas, graficar y trabajar con matrices.

## Clase Principal: RegresionLinealApp

La clase RegresionLinealApp encapsula toda la lógica de la interfaz, el procesamiento de datos y la regresión lineal.

### `__init__()`

- Inicializa la ventana principal.
- Configura el estilo RGB gamer usando colores oscuros con texto brillante (verde, cian, amarillo).
- Llama a **create\_widgets()** para construir los elementos de la interfaz.

## Evaluación de Expresiones Matemáticas

*evaluar\_expresion(self, expresion, x\_val=None)*

Evalúa expresiones como "e^2", "sin(pi/2)", "log(10)", con seguridad:

- Usa eval() dentro de un contexto\_seguro.
- Soporta uso de x o X como variable en expresiones tipo e^x.
- Evita ejecutar funciones maliciosas.

## Lectura de Puntos desde Texto

*parsear\_punto(self, punto\_str)*

Convierte texto como "2, 4" o "sin(pi/2)" en un punto (x, y).

- Soporta entrada en formato (x,y) o expresiones sueltas para y, asignando x automáticamente.

*extraer\_puntos\_del\_texto(self, texto)*

Extrae todos los puntos del campo de entrada:

- Detecta y separa expresiones sueltas (e^2, sin(x)) o tuplas ((2,4)).
- Admite múltiples formatos mezclados con comas y espacios.

## Interfaz de Usuario

*create\_widgets(self)*

Construye todos los widgets:

- **Etiqueta de instrucción**
- **Campo de texto** para ingresar puntos
- **Botón de cálculo**
- **Tabla tipo Treeview** con columnas para x, y, xy y x<sup>2</sup>
- **Etiqueta para mostrar la ecuación final**
- **Etiqueta para mostrar puntos evaluados**

## Cálculo de Regresión Lineal

*calcular\_regresion(self)* Función

principal:

1. Extrae y evalúa los puntos desde el texto.
2. Muestra errores si los datos no son válidos.
3. Calcula:
  - Productos xy y x<sup>2</sup>. ○ Sumas necesarias:  $\sum x$ ,  $\sum y$ ,  $\sum xy$ ,  $\sum x^2$ .
  - Pendiente m y ordenada al origen b usando las fórmulas:

```
m = (n * sum_xy - sum_x * sum_y) / denominador
b = (sum_y - m * sum_x) / n
```

4. Muestra los resultados en:
  - Tabla con x, y, xy, x<sup>2</sup>.
  - Ecuación  $Y = mX + b$ .
  - Puntos evaluados.

## Visualización Gráfica

*mostrar\_grafica(self, x\_vals, y\_vals, m, b)* Genera

una gráfica con:

- Puntos (x, y) en cian.
- Recta de regresión punteada en verde.
- Fondo oscuro y etiquetas estilo RGB gamer.
- Anotación de cada punto con su índice (P1, P2, ...).

Ingresar los puntos (x, y) o expresiones matemáticas:

Ejemplos: (2,4), (e^2, 5), e^x2, sin(pi/2), (2, 7)

(2,4), e^2, (0, e^2-4), sin(pi/2)

Calcular Recta de Regresión

x	y	xy	x^2
---	---	----	-----

## Posibles Mejoras Futuras

- Exportar la gráfica como imagen.
- Guardar los resultados en archivo CSV.
- Añadir opciones de regresión polinomial.
- Permitir edición visual de puntos en la gráfica.

## Requisitos para Ejecutar

Asegúrate de tener instalados: **pip**

**install matplotlib numpy**

Y luego ejecuta el archivo en Python:

**py REGRESIÓN LINEAL.py**

## Conclusión

Este programa no solo realiza regresión lineal de forma interactiva, sino que también permite a los usuarios aprender visualmente cómo los puntos afectan a la línea de ajuste. Su estilo atractivo y su flexibilidad para aceptar expresiones matemáticas lo convierten en una excelente herramienta educativa y técnica.