EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLÓGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE COMITAN

Alumnos:

- Ruedas Velasco Pedro Eduardo__19700073.
- Hernández Méndez Levi Magdiel__19700039.
- Molina Cifuentes Adriel David__19700061.

Docente: Vera Guillen José Flavio.

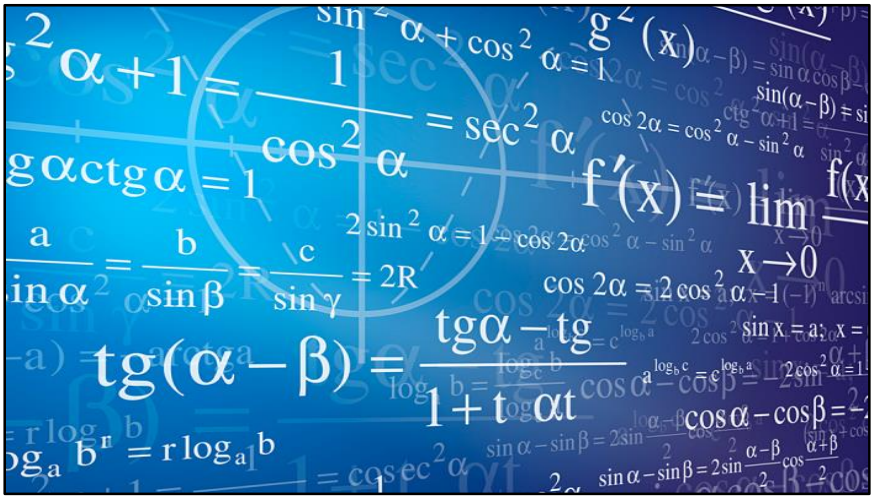
Materia: Métodos Numéricos.

Semestre: Cuarto

Grupo: “A”

Actividad: PA10_Codigos en el lenguaje seleccionado.

Comitán de Domínguez Chiapas, a 10 de Junio del 2021.



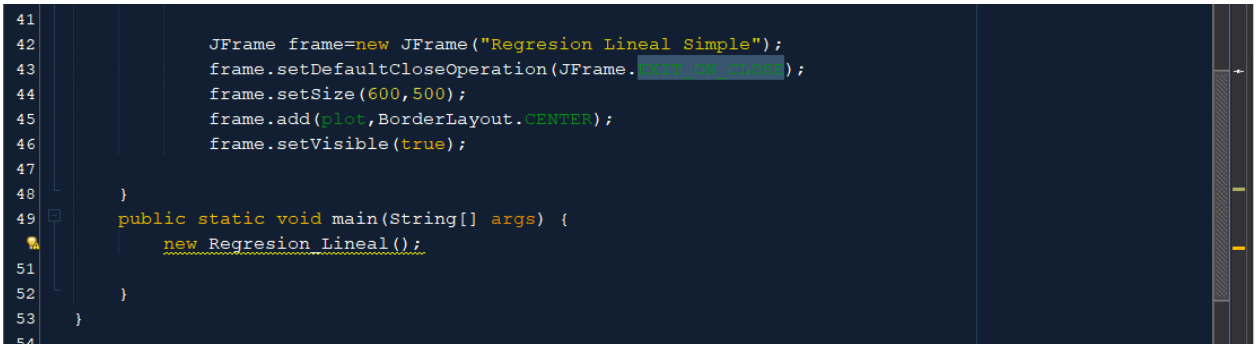
Regresión Lineal Simple.

El método de regresión simple realizado en java tiene como objetivo tratar de explicar la relación que existe entre una variable dependiente Y un conjunto de variables independientes X1,..., Xn.

En un modelo de regresión lineal simple se trata de explicar la relación que existe entre la variable respuesta Y y una única variable explicativa X.

Código en Java.

```
 * @author Eduardo Velasco
 */
public class Regresion_Lineal {
    double x[]={1,2,3,5,7,8,12,13,16,18};
    double y[]={1.3,3.4,5.4,7.2,10.3,9.13,12,13,16,18};

    SimpleRegression sr=new SimpleRegression();
    Plot2DPanel plot=new Plot2DPanel();
    JTextArea resul=new JTextArea();

    public Regresion_Lineal() {
        for (int i = 0; i < x.length; i++) {
            sr.addData(x[i],y[i]);
        }
            double [] yc=new double [y.length];
            for (int i = 0; i < x.length; i++) {
            yc[i]=sr.predict(x[i]);
            }
            plot.addLegend("South");
            plot.addScatterPlot("Datos",x, y);
            plot.addLinePlot("Regresion",x, yc);
            BaseLabel bl=new BaseLabel("Ejemplo de regresion lineal",Color.MAGENTA,0.5,1.1);
            plot.addPlotable(bl);
```



```
            JFrame frame=new JFrame("Regresion Lineal Simple");
            frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
            frame.setSize(600,500);
            frame.add(plot,BorderLayout.CENTER);
            frame.setVisible(true);

    }
    public static void main(String[] args) {
        new Regresion_Lineal();

    }
}
```

Mínimos Cuadrados

El método consiste en acercar una línea o una curva, según se escoja, lo más posible a los puntos determinados por la coordenadas [x, f(x)], que normalmente corresponden a muestras de algún experimento, sirve para interpolar valores, dicho en otras palabras, se usa para buscar valores desconocidos usando como referencia otras muestras del mismo evento.

Existen numerosas leyes físicas en las que se sabe de antemano que dos magnitudes x e y se relacionan a través de una ecuación lineal

$y = ax + b$

Donde las constantes b (ordenada en el origen) y a (pendiente) dependen del tipo de sistema que se estudia y, a menudo, son los parámetros que se pretende encontrar. Cabe aclarar que este método, aunque es sencillo de implantar no es del todo preciso, pero si proporciona una interpolación aceptable. Como se comentó previamente se puede usar una recta o una curva como base para calcular nuevos valores.

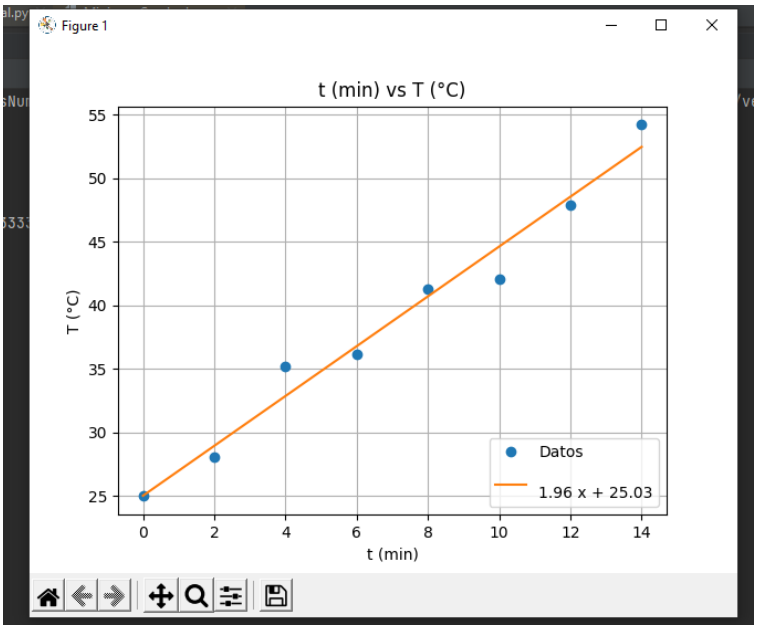
- **Código en Pyton**

```
-   #Se realizó un experimento con un fluido al que se le transfirió
    calor de forma constante. La temperatura se tomó cada
    # dos minutos durante los primeros 14 minutos. La siguiente tabla
    resume los resultados obtenidos, y se observa que la
    # temperatura aumenta de forma lineal.

    ##https://colab.research.google.com/drive/1k3B49BDvmWOLNSyr4ptF8oiJ
    5T7G7FOD?usp=sharing#scrollTo=SV8mDhmClOUi
    #tiempo (minutos)  altura (metros)
    #    0                    25
    #   2                     28.1
    #   4                     35.2
    #   6                     36.2
    #   8                     41.3
    #   10                    42.1
    #   12                    47.9
    #   14                    54.2


    #Importacion de Librerias
    import numpy as np
    import matplotlib.pyplot as plt

    #Funcion para ejecutar la Regresion de un conjunto de datos (x, y)

    def regress (x, y):

      sumx = 0           #Suma de los valores conocidos de x
      sumy = 0           #Suma de los valores conocidos de y
      sumxy = 0          #Suma de los productos x*y
      St = 0             #Suma de los cuadrados de la diferencia entre
    "y" y "promy"
```

```
  sumx2 = 0          #Suma de los cuadrados de x
  Sr = 0             #Sumas residuales
  n = len(x)         #Cantidad de datos conocidos de la variable x

  for i in range (len(x)):  #Se ejecutan cada una de las sumas
    sumx = sumx + x[i]
    sumy = sumy + y[i]
    sumxy = sumxy + x[i]*y[i]
    sumx2 = sumx2 + x[i]**2

  promx = sumx/n             #Promedio de x
  promy = sumy/n             #Promedio de y

  a1 = (n * sumxy - sumx*sumy)/(n * sumx2 - sumx**2)     #Calcula la
pendiente (a1) - Tomado de Chapra
  a0 = promy - a1*promx                                  #Calcula el
intercepto (a0) - Tomado de Chapra

  for i in range (n):
    St = St + (y[i] - promy)**2                          #Calcula la
suma de los cuadrados de la diferencia entre "y" y "promy"
    Sr = Sr + (y[i] - a1*x[i] - a0)**2                   #Calcula la
suma de los residuales - Tomado de Chapra
    Syx = (Sr/(n-2))**0.5                                #Calcula el
error estandar - Tomado de Chapra
    r2 = (St - Sr)/St                                    #Calcula el
coeficiente de determinacion - Tomado de Chapra

  return a1, a0, Syx, r2


# Registramos los datos de la variable independiente "x" y la
variable independiente "y"

x = [0, 2, 4, 6, 8, 10, 12, 14]
y = [25, 28.1, 35.2, 36.2, 41.3, 42.1, 47.9, 54.2]

#Llamar funcion "regress"

a1, a0, syx, r2 = regress (x, y)

#Imprimir resultados de los parametros de la regresion
print ("a1: ", a1)
print ("a0: ", a0)
print ("Syx: ", syx)
print ("r2: ", r2)




#Ecuacion de la Recta
t = np.poly1d ([a1, a0])
#print(t)
print("y = " + str(a1) + "x + " + str(a0))  # y = a1x + a0


plt.plot(x, y, "o", label = 'Datos')                #Grafica de
```

```
dispersion de los datos de entrada (x, y)
plt.plot(x, a1*np.array(x) + a0, label = t)  #Grafica de la recta
de la regresion
plt.xlabel("t (min)")
plt.ylabel("T (°C)")
plt.title("t (min) vs T (°C)")
plt.grid()
plt.legend (loc = 4)
plt.show()
```

- **Ejecución**



INTERPOLACION DE NEWTON:

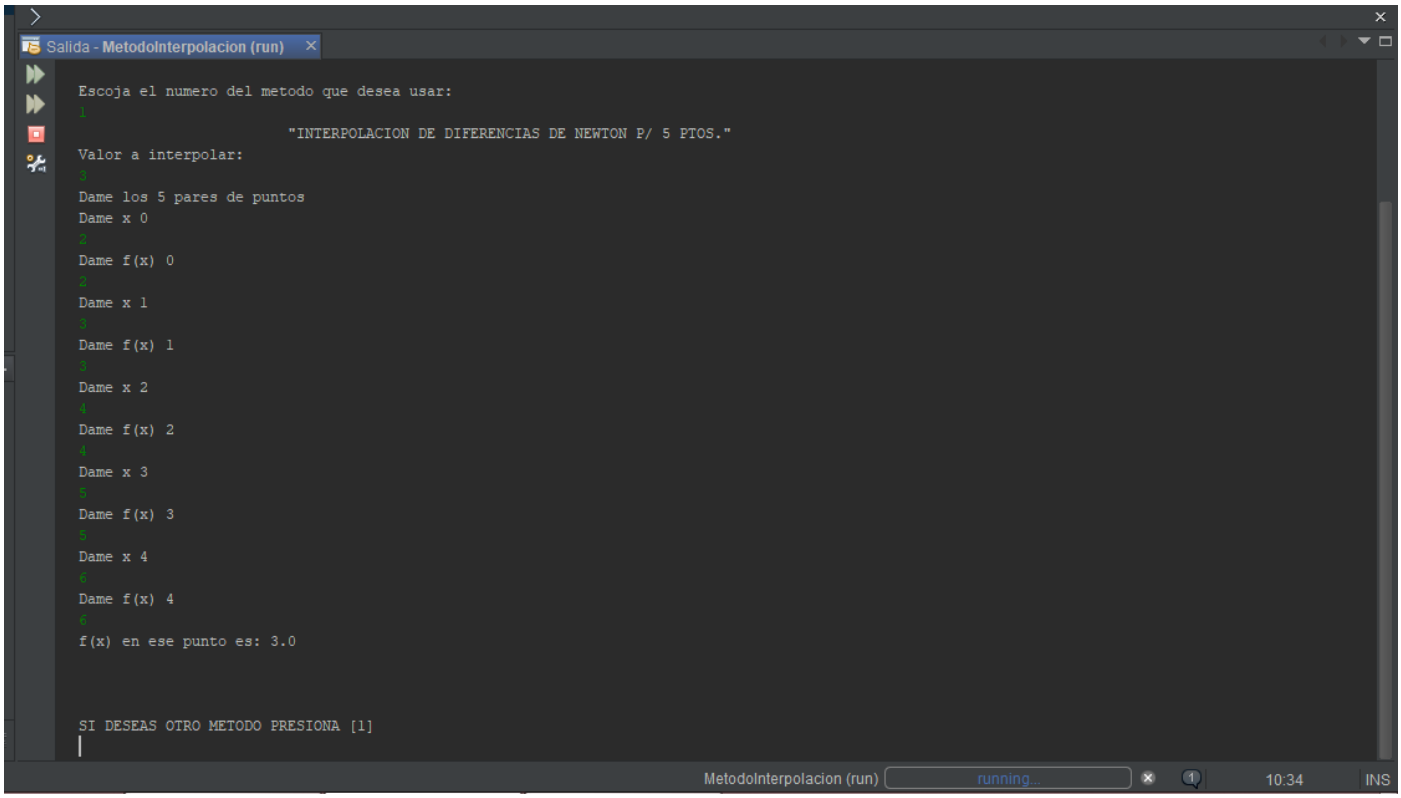
Código en java Es un método de interpolación polinómica. Aunque solo existe un único polinomio que interpola una serie de puntos, existen diferentes formas de calcularlo. Este método es útil para situaciones que requieran un número bajo de puntos para interpolar, ya que a medida que crece el número de puntos, también lo hace el grado del polinomio, en este trabajo se entrega el código fuente de la implementación del método de interpolación de newton, trabajando con varios métodos en la clase main para lograr una mejor eficacia al ejecutar el código

```
package metodointerpolacion;


import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
public class MetodoInterpolacion {


   public static void main(String[] args) {
      MetodoInterpolacion Proyecto = new MetodoInterpolacion();
      Proyecto.menu();  //solo se manda a llamar a menu desde aqui, ya que menu llama a los demas metodos posteriromente
   }


   //(7) Interpolacion Newton
   public void MetodoInterNewton() {
      double a[][] = new double[5][2];
      double x, y, fx1x0, fx2x1, fx3x2, fx4x3, fx2x1x0, fx3x2x1, fx4x3x2, fx3x2x1x0, fx4x3x2x1, fx4x3x2x1x0;
      int i;
      System.out.println("\t\t\t\"INTERPOLACION DE DIFERENCIAS DE NEWTON P/ 5 PTOS.\"");
      System.out.println("Valor a interpolar: ");
      x = lee();
```

```
        System.out.println("Dame los 5 pares de puntos");
        for (i = 0; i < 5; i++) {
           System.out.println("Dame x " + i);
           a[i][0] = lee();
           System.out.println("Dame f(x) " + i);
           a[i][1] = lee();
        }
        fx1x0 = (a[1][1] - a[0][1]) / (a[1][0] - a[0][0]);
        fx2x1 = (a[2][1] - a[1][1]) / (a[2][0] - a[1][0]);
        fx3x2 = (a[3][1] - a[2][1]) / (a[3][0] - a[2][0]);
        fx4x3 = (a[4][1] - a[3][1]) / (a[4][0] - a[3][0]);
        fx2x1x0 = (fx2x1 - fx1x0) / (a[2][0] - a[0][0]);
        fx3x2x1 = (fx3x2 - fx2x1) / (a[3][0] - a[1][0]);
        fx4x3x2 = (fx4x3 - fx3x2) / (a[4][0] - a[2][0]);
        fx3x2x1x0 = (fx3x2x1 - fx2x1x0) / (a[3][0] - a[0][0]);
        fx4x3x2x1 = (fx4x3x2 - fx3x2x1) / (a[3][0] - a[0][0]);
        fx4x3x2x1x0 = (fx4x3x2x1 - fx3x2x1x0) / (a[4][0] - a[0][0]);
        y = a[0][1] + fx1x0 * (x - a[0][0]) + fx2x1x0 * (x - a[0][0]) * (x - a[1][0]) + fx3x2x1x0 * (x -
a[0][0]) * (x - a[1][0]) * (x - a[2][0]) + fx4x3x2x1x0 * (x - a[0][0]) * (x - a[1][0]) * (x - a[2][0]) * (x -
a[3][0]);
        System.out.println("f(x) en ese punto es: " + y);
    }

    //para leer desde teclado
    public double lee() {
        double num;
        try {
            InputStreamReader isr = new InputStreamReader(System.in);
            BufferedReader br = new BufferedReader(isr);
            String sdato;
```

```
            sdato = br.readLine();
            num = Double.parseDouble(sdato);
        } catch (IOException ioe) {
            num = 0.0;
        }
        return num;
    }

    //para  leer un entero
    public int leeint() {
        int num;
        try {
            InputStreamReader isr = new InputStreamReader(System.in);
            BufferedReader br = new BufferedReader(isr);
            String sdato;
            sdato = br.readLine();
            num = Integer.parseInt(sdato);
        } catch (IOException ioe) {
            num = 0;
        }
        return num;
    }

    //para salir del programa
    public int Fuera() {
        int sal;
        System.out.println("\n\n\nSI DESEAS OTRO METODO PRESIONA [1]");
        sal = leeint();
        return sal;
```

```
    }

    //despliega menu
    public void menu() {
      int a;
      int p;
      do {
        do {
          System.out.println("\n\n\t\t\tMETODOS NUMERICOS\n\n");
          System.out.println("1.-Interpolacion Newton");
          System.out.println("\n\nEscoja el numero del metodo que desea usar:");
          a = leeint();
        } while (a < 1 || a > 8);
        switch (a) {
          case 1:
            MetodoInterNewton();
            p = Fuera();
            break;
          default:
            System.out.println("Opcion incorrecta");
            p = 1;
            break;
        }
      } while (p == 1);
    }

}
```


Salida - MetodoInterpolacion (run)
Escoja el numero del metodo que desea usar:
1
"INTERPOLACION DE DIFERENCIAS DE NEWTON P/ 5 PTOS."
Valor a interpolar:
3
Dame los 5 pares de puntos
Dame x 0
2
Dame f(x) 0
2
Dame x 1
3
Dame f(x) 1
3
Dame x 2
4
Dame f(x) 2
4
Dame x 3
5
Dame f(x) 3
5
Dame x 4
6
Dame f(x) 4
6
f(x) en ese punto es: 3.0
SI DESEAS OTRO METODO PRESIONA [1]
MetodoInterpolacion (run)
running...
10:34
INS