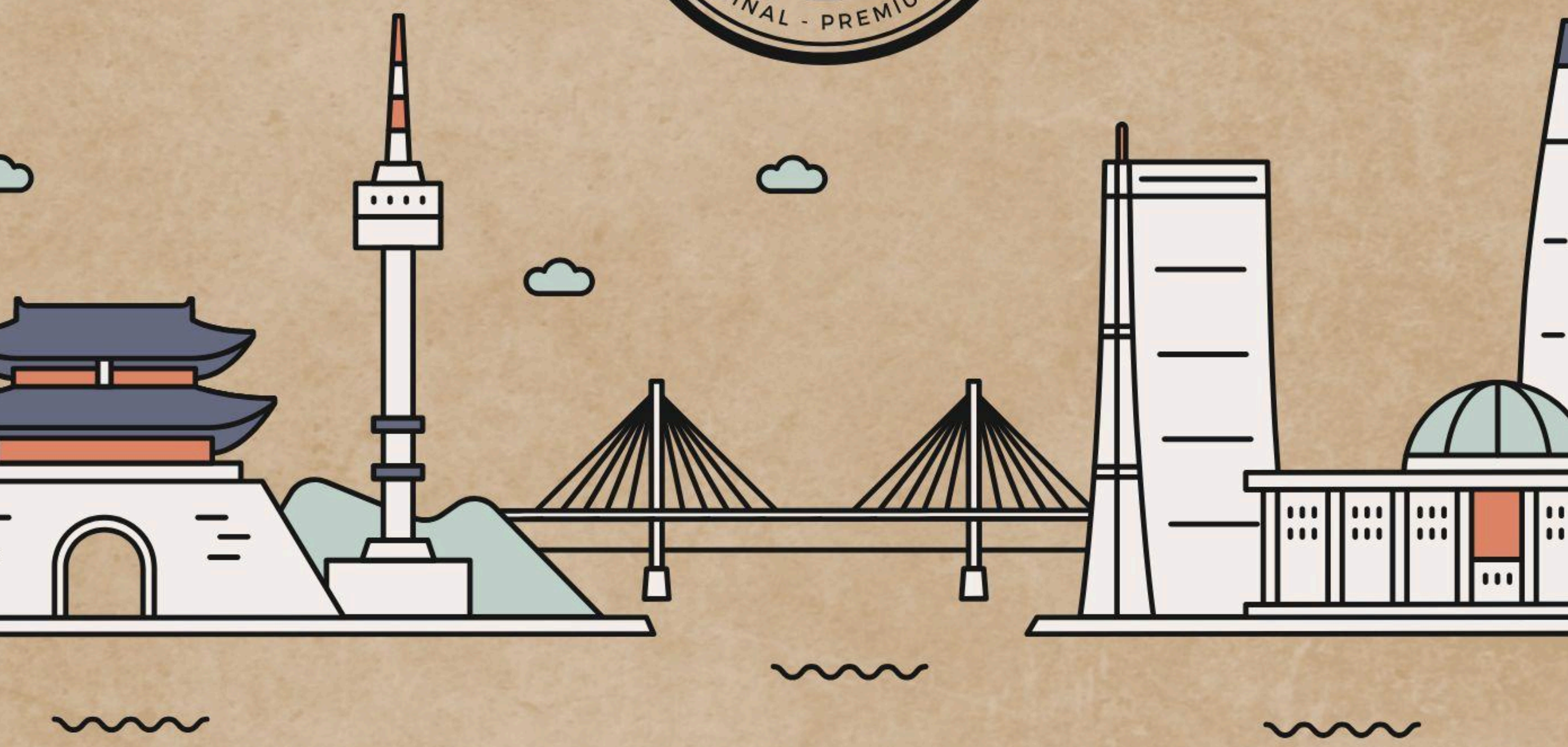
Problemario

# TEMA 5

Métodos Numéricos

# Índice

# Métodos de interpolación

Los métodos de interpolación se utilizan cuando tenemos un conjunto limitado de datos conocidos y queremos estimar valores intermedios dentro de ese rango. La idea fundamental es que si conocemos cómo se comporta una variable en ciertos puntos, podemos suponer razonablemente cómo se comporta entre ellos. Esto es muy útil, por ejemplo, en ingeniería cuando tenemos una tabla de datos experimentales o valores de sensores y queremos hacer estimaciones rápidas.

> “La interpolación se refiere a la determinación de una función que pase exactamente por un conjunto de puntos conocidos.”
> — Chapra & Canale, 2006, p. 304

## Interpolación Lineal

La interpolación lineal es el enfoque más básico y directo para estimar valores entre dos puntos conocidos. Se basa en la suposición de que el cambio entre dos datos es uniforme, es decir, sigue una línea recta.

Se utiliza cuando se necesita una solución rápida y sencilla, por ejemplo, al consultar una tabla de propiedades físicas (densidad, viscosidad, etc.) y el valor deseado no aparece exactamente, pero se encuentra entre dos datos conocidos.

> “Aunque simple, la interpolación lineal es adecuada cuando los puntos están suficientemente cercanos y el comportamiento de la variable es aproximadamente lineal.”
> — Chapra & Canale, 2006, p. 305

**Pseudocódigo**

Entradas: x0, y0, x1, y1, x

Paso 1: Calcular la pendiente: m = (y1 - y0) / (x1 - x0)

Paso 2: Calcular f(x): y = y0 + m * (x - x0)

Salida: y

**Código en Java**

```
class InterpolacionLineal {
    public static double interpolar(double x0, double y0, double x1,
double y1, double x) {
        double m = (y1 - y0) / (x1 - x0);
        return y0 + m * (x - x0);
    }
}
public class Main {
    public static void main(String[] args) {
        double resultado = InterpolacionLineal.interpolar(2, 4, 6, 10,
3);
        System.out.println("Valor interpolado: " + resultado);
    }
}
```

## Ejemplo 1

**Estimar la temperatura a las 3 horas**

**Tabla de datos**

| Hora (x) | Temperatura (y °C) |
|---|---|
| 1 | 36.5555 |
| 2 | 37.2222 |
| 4 | 38.1111 |

**Planteamiento**:

El médico desea estimar la temperatura del paciente a las **3 horas**.

**Datos introducidos:**

- $x_0 = 2$, $y_0 = 37.2222$
- $x_1 = 4$, $y_1 = 38.1111$
- $x = 3.0$

**Resultado esperado:**

**37.6667 °C**

**Ejecución de código:**

Valor interpolado: 37.6667

## Ejemplo 2

**Nivel del agua en un tanque a los 5 minutos**

**Tabla de datos**

| **Minuto (x)** | **Nivel (y litros)** |
|---|---|
| 2 | 10.5000 |
| 6 | 18.9000 |

**Planteamiento**:

Estimar el nivel del agua al minuto **5**.

**Datos introducidos:**

- $x_0 = 2$, $y_0 = 10.5000$
- $x_1 = 6$, $y_1 = 18.9000$
- $x = 5.0$

**Resultado esperado:**

**16.8000 litros**

**Ejecución de código:**

Valor interpolado: 16.8000

## Ejercicio 3

**Estimar la presión a los 20 segundos**

**Tabla de datos:**

| Tiempo (x en s) | Presión (y en Pa) |
|---|---|
| **10** | **500.0000** |
| **30** | **1500.0000** |

**Planteamiento**:

Estimar la presión a los **20 segundos**.

**Datos introducidos:**

- $x_0$ = 10, $y_0$ = 500.0000
- $x_1$ = 30, $y_1$ = 1500.0000
- x = 20.0

**Resultado esperado:**

**1000.0000 Pa**

**Ejecución de código:**

Valor interpolado: 1000.0000

## Ejercicio 4

**Estimación de población**

**Tabla de datos:**

| Tiempo (x en años) | Habitantes |
|---|---|
| **2000** | **5000.0000** |
| **2010** | **6200.0000°C** |

**Planteamiento**:

Estimar la población en el año **2005.**

**Datos introducidos:**

- $x_0$ = 2000, $y_0$ = 5000.0000
- $x_1$ = 2010, $y_1$ = 6200.0000
- x = 2005.0

**Resultado esperado:**

**5600.0000 min**

**Ejecución de código:**

**Valor interpolado: 5600.0**

## Ejercicio 5 (Caso de error)

**Caso de error de estimación (Coordenadas X Iguales)**

**Tabla de datos:**

| Tiempo (x en s) | Altura en m |
|---|---|
| 5 | 10.0000 m |
| 5 | 12.0000 m |

**Planteamiento**:

Estimar la altura de un objeto en el tiempo de 7 segundos**.**

**Datos introducidos:**

- $x_0 = 5$, $y_0 = 10.0000$
- $x_1 = 5$, $y_1 = 12.0000$
- $x = 7.0$

**Resultado esperado:**

**Al ejecutar el código se provocará una división por leroy por lo tanto una excepción**

**Ejecución de código:**

**Valor interpolado: Infinity**

## Interpolación polinómica: Newton

La interpolación polinómica de Newton es una técnica más avanzada que permite construir una curva (polinomio) que pase exactamente por varios puntos conocidos. Su ventaja principal es que se puede modificar fácilmente cuando se agregan nuevos datos, sin recalcular todo desde cero.

Es útil en situaciones donde se quiere una mayor precisión que la que ofrece la interpolación lineal, o cuando se tienen múltiples puntos que se deben ajustar exactamente. Sin embargo, su uso con muchos datos puede causar oscilaciones no deseadas (efecto Runge).

> "La forma de Newton es eficiente para agregar nuevos datos y para implementar algoritmos computacionales paso a paso."
> — Chapra & Canale, 2006, p. 312

**Pseudocódigo**

```
Proceso InterpolacionNewton
    Definir x[100], y[100], dd[100,100] Como Real
    Definir n, i, j Como Entero
    Definir xEval, resultado, producto Como Real

    // Leer cantidad de puntos
    Escribir "Ingrese la cantidad de puntos:"
    Leer n

    // Leer los valores de x
    Para i <- 0 Hasta n - 1 Con Paso 1
        Escribir "x[", i, "] = "
        Leer x[i]
    FinPara

    // Leer los valores de y
    Para i <- 0 Hasta n - 1 Con Paso 1
        Escribir "y[", i, "] = "
        Leer y[i]
    FinPara

    // Inicializar la primera columna de diferencias divididas con y
    Para i <- 0 Hasta n - 1 Con Paso 1
        dd[i, 0] <- y[i]
```

```
    FinPara

    // Calcular diferencias divididas
    Para j <- 1 Hasta n - 1 Con Paso 1
        Para i <- 0 Hasta n - j - 1 Con Paso 1
            dd[i, j] <- (dd[i + 1, j - 1] - dd[i, j - 1]) / (x[i + j] - x[i])
        FinPara
    FinPara

    // Evaluar el polinomio
    Escribir "Ingrese el valor de x para evaluar el polinomio:"
    Leer xEval

    resultado <- dd[0, 0]
    producto <- 1

    Para i <- 1 Hasta n - 1 Con Paso 1
        producto <- producto * (xEval - x[i - 1])
        resultado <- resultado + dd[0, i] * producto
    FinPara

    Escribir "El valor interpolado en x = ", xEval, " es: ", resultado
FinProceso
```

**Código en Java**

```java
public class NewtonInterpolation {

    // Método para calcular las diferencias divididas
    public static double[][] calcularDiferenciasDivididas(double[] x, double[] y) {
        int n = x.length;
        double[][] dd = new double[n][n];

        // Inicializar primera columna con y
        for (int i = 0; i < n; i++) {
            dd[i][0] = y[i];
        }

        // Calcular las diferencias divididas
        for (int j = 1; j < n; j++) {
            for (int i = 0; i < n - j; i++) {
                dd[i][j] = (dd[i + 1][j - 1] - dd[i][j - 1]) / (x[i + j] - x[i]);
            }
        }

        return dd;
    }

    // Evaluar el polinomio interpolado en un punto xEval
    public static double evaluarPolinomio(double[][] dd, double[] x, double xEval) {
        int n = x.length;
        double resultado = dd[0][0];
        double producto = 1.0;

        for (int i = 1; i < n; i++) {
            producto *= (xEval - x[i - 1]);
            resultado += dd[0][i] * producto;
        }

        return resultado;
    }

    // Método principal de prueba
    public static void main(String[] args) {
        double[] x = {9, 12, 15}; // Ejemplo de valores de x
        double[] y = {15, 21, 18}; // Ejemplo de valores de y

        double[][] dd = calcularDiferenciasDivididas(x, y);
        double xEval = 13.5;
        double resultado = evaluarPolinomio(dd, x, xEval);

        System.out.printf("Valor interpolado en x = %.2f es: %.2f\n", xEval,
resultado);
    }
}
```

## Ejemplo 1

Un paciente es monitoreado cada hora después de una operación. La temperatura corporal (en °C) registrada fue:

| Hora(x) | Temperatura(y) |
|---|---|
| 1 | 36.5555 |
| 2 | 37.2222 |
| 4 | 38.1111 |

El médico desea estimar la temperatura a las **3 horas**.

Datos introducidos:

En los valores de x ingresamos los siguientes datos: 1 , 2, 4.

En los valores de y se ingresan: 36.5555, 37.2222, 38.1111.

En el codigo seria algo asi:

```
double[] x = {1, 2, 4};
double[] y = {36.5555, 37.2222, 38.1111};
double xEval = 3.000;
```

El resultado lanzado seria de : 37.6777 °C

En codigo seria algo asi:
Valor interpolado en x = 3.0000 es: 37.7407

Por lo que podemos concluir que a las 3 horas, la temperatura estimada del paciente es **37.7407 °C**.

**Ejemplo 2**

Un agricultor mide la altura de una planta (en cm) cada semana:

| Semana (x) | Altura (y) |
|---|---|
| 1 | 12.0000 |
| 3 | 15.3333 |
| 6 | 19.2222 |

Quiere estimar la altura en la **semana 4**.

Datos introducidos:

En los valores de x ingresamos los siguientes datos: 1 , 3, 6.

En los valores de y se ingresan: 12.0000, 15.3333, 19.2222.

En el codigo seria algo asi:

```
double[] x = {1, 3, 6};
double[] y = {12.0000, 15.3333, 19.2222};
double xEval = 4;
```

El resultado lanzado seria de : 16.7777

En codigo seria algo asi:
Valor interpolado en x = 4.0000 es: 16,7777

La altura estimada en la semana 4 es **16.7777 cm**.

## Ejemplo 3

Un analista financiero registra el precio de una acción (en USD) al final de ciertos días:

| Dia (x) | Precio (y) |
|---|---|
| 1 | 120.0000 |
| 2 | 123.5555 |
| 5 | 130.2222 |

Desea estimar el precio para el **día 3**.

Datos introducidos:

En los valores de x ingresamos los siguientes datos: 1 , 2, 5.

En los valores de y se ingresan:120.0000, 123.5555, 130.2222.

En el codigo seria algo asi:

```
double[] x = {1, 2, 5};
double[] y = {120.0000, 123.5555, 130.2222};
double xEval = 3;
```

El resultado lanzado seria de :

En codigo seria algo asi:

Valor interpolado en x = 3.0000 es: 126.4444

El precio estimado de la acción el día 3 es **126.4444 USD**

## Ejemplo 4

Una empresa registra la producción de una máquina (en unidades) cada hora.

| Hora (x) | Producción (y) |
|---|---|
| 1 | 100 |
| 3 | 170 |
| 6 | 290 |

Se desea estimar la producción a las **4 horas**.

Datos introducidos:

En los valores de x ingresamos los siguientes datos: 1, 3, 6.

En los valores de y se ingresan: 100, 170, 290.

En el código sería algo así:

```
double[] x = {1, 3, 6};
double[] y = {100, 170, 290};
double xEval = 4.0;
```

El resultado lanzado sería:

Valor interpolado en x = 4.0000 es: 210.0000

Por lo que podemos concluir que a las 4 horas, la producción estimada de la máquina es **210.0000 unidades**.

**Ejemplo 5 (Caso de error)**

Un meteorólogo registra la temperatura en grados Celsius en ciertas horas del día.

| Hora (x) | Temperatura (y) |
|---|---|
| 6 | 15.0 |
| 12 | 22.5 |
| 6 | 19.0 |

El meteorólogo desea estimar la temperatura a las **9 horas**.

Datos introducidos:

En los valores de x ingresamos los siguientes datos: 6, 12, 6.

Aquí se comete el error: los valores de x no son únicos (6 se repite).

En los valores de y se ingresan: 15.0, 22.5, 19.0.

En el código sería algo así:

```
double[] x = {6, 12, 6}; // Error: valores duplicados
double[] y = {15.0, 22.5, 19.0};
double xEval = 9.0;
```

Resultado:

El código arrojará un error o resultado incorrecto debido a una división por cero en el cálculo de diferencias divididas, ya que dos valores de x son iguales.

# Métodos de regresión

La regresión, a diferencia de la interpolación, parte del reconocimiento de que los datos pueden tener errores o variaciones naturales. En lugar de pasar exactamente por todos los puntos, busca una curva que represente la tendencia general de los datos, incluso si no pasa por todos ellos.

## Correlación

La correlación evalúa qué tan fuerte y en qué dirección están relacionadas dos variables. Es un paso preliminar importante antes de aplicar cualquier modelo de regresión, ya que permite verificar si tiene sentido suponer que una variable depende de otra.

Por ejemplo, si estás estudiando la relación entre la velocidad del viento y la producción de una turbina eólica, primero se analiza la correlación: si es alta, aplicar una regresión tiene sentido; si no, puede que los datos no estén relacionados directamente.

> "La correlación cuantifica la fuerza de una relación lineal entre dos variables, y sirve como base para el análisis de regresión."
> — Chapra & Canale, 2006, p. 333

**Pseudocódigo**

```
Algoritmo Correlacion
    Definir n, i Como Entero
    Definir xi, yi Como Real
    Definir sumaX, sumaY, sumaXY, sumaX2, sumaY2 Como Real
    Definir coefPendiente, coefIntercepto, r Como Real

    Escribir "Ingrese el número de datos:"
    Leer n

    Dimension datosX[n], datosY[n]

    sumaX <- 0
    sumaY <- 0
    sumaXY <- 0
    sumaX2 <- 0
    sumaY2 <- 0
```

```
Para i <- 1 Hasta n Con Paso 1
	Escribir "Dato ", i
	Escribir "Ingrese X[", i "]: "
	Leer xi
	Escribir "Ingrese Y[", i, "]: "
	Leer yi

	datosX[i] <- xi
	datosY[i] <- yi

	sumaX <- sumaX + xi
	sumaY <- sumaY + yi
	sumaXY <- sumaXY + xi * yi
	sumaX2 <- sumaX2 + xi * xi
	sumaY2 <- sumaY2 + yi * yi
Fin Para

coefPendiente <- (n * sumaXY - sumaX * sumaY) / (n * sumaX2 - sumaX * sumaX)
coefIntercepto <- (sumaY - coefPendiente * sumaX) / n

r <- (n * sumaXY - sumaX * sumaY) / raiz( (n * sumaX2 - sumaX^2) * (n * sumaY2 - sumaY^2) )
Escribir "----------------------------------------"
Escribir "Ecuación de regresión: Y = ", coefIntercepto, " + ", coefPendiente, " * X"
Escribir "Coeficiente de correlación r = ", r
Escribir "Interpretación:"
Si r > 0 Entonces
```

**Código en Java**

```java
import java.util.Scanner;


public class correlacion {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        int n;
        double xi, yi;
        double sumaX = 0, sumaY = 0, sumaXY = 0, sumaX2 = 0, sumaY2 = 0;
        double coefPendiente, coefIntercepto, r;

        System.out.print("Ingrese el número de datos: ");
        n = scanner.nextInt();


        double[] datosX = new double[n];
        double[] datosY = new double[n];


        for (int i = 0; i < n; i++) {
            System.out.println("Dato " + (i + 1));
            System.out.print("Ingrese X[" + (i + 1) + "]: ");
            xi = scanner.nextDouble();
            System.out.print("Ingrese Y[" + (i + 1) + "]: ");
            yi = scanner.nextDouble();


            datosX[i] = xi;
            datosY[i] = yi;

            sumaX += xi;
            sumaY += yi;
            sumaXY += xi * yi;
            sumaX2 += xi * xi;
            sumaY2 += yi * yi;
        }


        coefPendiente = (n * sumaXY - sumaX * sumaY) / (n * sumaX2 - sumaX * sumaX);

        coefIntercepto = (sumaY - coefPendiente * sumaX) / n;
        r = (n * sumaXY - sumaX * sumaY) /
            Math.sqrt((n * sumaX2 - sumaX * sumaX) * (n * sumaY2 - sumaY * sumaY));


        System.out.println("-----------------------------------------");
        System.out.printf("Ecuación de regresión: Y = %.4f + %.4f * X%n",
```

```java
coefIntercepto, coefPendiente);
        System.out.printf("Coeficiente de correlación r = %.4f%n", r);


        System.out.println("Interpretación:");
        if (r > 0) {
            System.out.println("Relación positiva.");
        } else if (r < 0) {
            System.out.println("Relación negativa.");
        } else {
            System.out.println("La correlación es nula.");
        }
        scanner.close();
    }
}
```

## Ejemplo 1

Análisis de ventas vs inversión en publicidad

Una empresa desea saber si existe una relación entre la cantidad invertida en publicidad (en miles de pesos) y el total de ventas mensuales (en miles de pesos). Se recogen los siguientes datos:

| **Publicidad (X)** | **Ventas (Y)** |
|---|---|
| 2 | 4 |
| 4 | 7 |
| 6 | 9 |
| 8 | 13 |
| 10 | 15 |

**Resultado esperado**:

- Ecuación: Y = 1.2000 + 1.4000 * X
- r ≈ 0.9949
- Interpretación: Correlación positiva muy fuerte.

**Resultado en el código:**

Ingrese el número de datos: 5
Dato 1
Ingrese X[1]: 2
Ingrese Y[1]: 4
Dato 2
Ingrese X[2]: 4
Ingrese Y[2]: 7
Dato 3
Ingrese X[3]: 6
Ingrese Y[3]: 9
Dato 4
Ingrese X[4]: 8
Ingrese Y[4]: 13
Dato 5
Ingrese X[5]: 10
Ingrese Y[5]: 15

Ecuación de regresión: Y = 1.2000 + 1.4000 * X
Coeficiente de correlación r = 0.9949
Interpretación:
Relación positiva.

## Ejemplo 2

**Edad vs número de libros leídos este mes**

Se busca saber si hay relación entre la edad de un grupo de 5 personas y la cantidad de libros que leyeron este mes.

| n | Edad (X) | Libros leídos (Y) |
|---|---|---|
| 1 | 20 | 3 |
| 2 | 25 | 2 |
| 3 | 30 | 4 |
| 4 | 35 | 3 |
| 5 | 40 | 2 |

**Resultado esperado:**

- Ecuación: Y = 3.4000 - 0.0200 * X
- r ≈ -0.1890
- Interpretación: Hay una relación que tiende a ser negativa entre las variables.

**Resultado en el código:**

Ingrese el número de datos: 5
Dato 1
Ingrese X[1]: 20
Ingrese Y[1]: 3
Dato 2
Ingrese X[2]: 25
Ingrese Y[2]: 2
Dato 3
Ingrese X[3]: 30
Ingrese Y[3]: 4
Dato 4
Ingrese X[4]: 35
Ingrese Y[4]: 3
Dato 5
Ingrese X[5]: 40
Ingrese Y[5]: 2

Ecuación de regresión: Y = 3.4000 + -0.0200 * X
Coeficiente de correlación r = -0.1890
Interpretación:
Relación negativa.

## Ejemplo 3

**Temperatura vs ventas de sopa caliente**

Un restaurante analiza cómo afecta la temperatura exterior a las ventas de sopa caliente.

| Temperatura (°C) (X) | Ventas (Y) |
|---|---|
| 30 | 50 |
| 28 | 60 |
| 26 | 70 |
| 24 | 85 |
| 22 | 95 |

**Resultado esperado:**

- Ecuación: Y = 221.5000 - 5.7500 * X
- r ≈ -0.9972
- Interpretación: Correlación negativa muy fuerte.

**Resultado en el código:**

Ingrese el número de datos: 5
Dato 1
Ingrese X[1]: 30
Ingrese Y[1]: 50
Dato 2
Ingrese X[2]: 28
Ingrese Y[2]: 60
Dato 3
Ingrese X[3]: 26
Ingrese Y[3]: 70
Dato 4
Ingrese X[4]: 24
Ingrese Y[4]: 85
Dato 5
Ingrese X[5]: 22
Ingrese Y[5]: 95

Ecuación de regresión: Y = 221.5000 + -5.7500 * X
Coeficiente de correlación r = -0.9972
Interpretación:
Relación negativa.

## Ejemplo 4

**Edad vs número de libros leídos este mes**

Se busca saber si hay relación entre la edad de un grupo de personas y la cantidad de libros que leyeron este mes.

| Edad (X) | Libros leídos (Y) |
|---|---|
| **20** | **3** |
| **25** | **2** |
| **30** | **4** |
| **35** | **3** |
| **40** | **2** |

**Resultado esperado:**

- Ecuación: Y = 3.0000 + 0.0000 * X
- r ≈ 0.0000
- Interpretación: No hay correlación entre las variables.

**Resultado en el código:**

Ingrese el número de datos: 5
Dato 1
Ingrese X[1]: 20
Ingrese Y[1]: 3
Dato 2
Ingrese X[2]: 25
Ingrese Y[2]: 2
Dato 3
Ingrese X[3]: 30
Ingrese Y[3]: 4
Dato 4
Ingrese X[4]: 35
Ingrese Y[4]: 3
Dato 5
Ingrese X[5]: 40
Ingrese Y[5]: 2

Ecuación de regresión: Y 3.0000 + 0.0000 * X
Coeficiente de correlación r = 0.0000
Interpretación:
Relación nula.

## Ejemplo 5  (Caso de error)

**Análisis del salario vs edad de empleados en un departamento** *(Caso con error: varianza en X = 0)*

Una empresa desea analizar si hay relación entre la edad de los empleados y el salario que reciben. Se recolectan los siguientes datos:

| Edad (X) | Salario (Y) |
|---|---|
| 30 | 8000 |
| 30 | 8500 |
| 30 | 8200 |
| 30 | 8300 |
| 30 | 8100 |

Ingrese los valores de X y Y en el programa.
Obtenga la ecuación de regresión y el coeficiente de correlación.
Interprete el valor de r.

**Resultado esperado:**
**Error matemático:**
No se puede calcular la pendiente ni el coeficiente de correlación porque todos los valores de X son iguales, por lo tanto la varianza de X es 0, y eso provoca una división entre cero en las fórmulas.

Interpretación:
No hay variabilidad en los valores de X, por lo tanto no se puede determinar ninguna relación lineal entre X y Y.

## Mínimos Cuadrados

El método de mínimos cuadrados es el más usado para ajustar una curva a datos experimentales. A diferencia de la interpolación, no se obliga a que la curva pase por todos los puntos. En lugar de eso, se busca que en promedio se equivoque lo menos posible.

Este método es especialmente útil en ingeniería, donde los datos reales suelen tener ruido, errores de medición o variabilidad natural. La regresión lineal por mínimos cuadrados es el caso más común, pero también puede aplicarse a modelos cuadráticos, exponenciales, etc.

> “Los mínimos cuadrados permiten encontrar el mejor ajuste posible entre un modelo matemático y los datos reales disponibles.”
> — Chapra & Canale, 2006, p. 335

**Pseudocódigo**

```
Inicio
  Mostrar "Método de mínimos cuadrados - Ajuste lineal"

  Leer n

  Crear lista x vacía
  Crear lista y vacía

  Para i desde 1 hasta n hacer:
    Leer xi
    Leer yi
    Agregar xi a la lista x
    Agregar yi a la lista y
  Fin Para

  Inicializar sum_x ← 0
  Inicializar sum_y ← 0
  Inicializar sum_x2 ← 0
  Inicializar sum_xy ← 0

  Para i desde 0 hasta n-1 hacer:
    sum_x  ← sum_x  + x[i]
    sum_y  ← sum_y  + y[i]
    sum_x2 ← sum_x2 + x[i]²
    sum_xy ← sum_xy + x[i] * y[i]
  Fin Para
```

b ← (n * sum_xy - sum_x * sum_y) / (n * sum_x2 - sum_x$^2$)
a ← (sum_y - b * sum_x) / n

Redondear a y b a 4 cifras decimales

Mostrar "Resultado:"
Mostrar "La recta de regresión es: y = a + bx"

Fin

### Código en Java

```
import java.util.Scanner;
import java.util.ArrayList;
import java.math.RoundingMode;
import java.text.DecimalFormat;

public class MinimosCuadrados {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        DecimalFormat df = new DecimalFormat("#.####"); // 4 cifras decimales
        df.setRoundingMode(RoundingMode.HALF_UP);

        System.out.println("Metodo de minimos cuadrados - Ajuste lineal");

        // Leer la cantidad de pares
        System.out.print("Cuantos pares de datos vas a ingresar? ");
        int n = scanner.nextInt();

        ArrayList<Double> x = new ArrayList<>();
        ArrayList<Double> y = new ArrayList<>();

        // Ingreso de datos
        for (int i = 0; i < n; i++) {
            System.out.print("Ingrese x[" + (i + 1) + "]: ");
            double xi = scanner.nextDouble();

            System.out.print("Ingrese y[" + (i + 1) + "]: ");
            double yi = scanner.nextDouble();
```

```java
            x.add(xi);
            y.add(yi);
        }

        // Cálculo de sumatorias
        double sum_x = 0, sum_y = 0, sum_x2 = 0, sum_xy = 0;

        for (int i = 0; i < n; i++) {
            double xi = x.get(i);
            double yi = y.get(i);
            sum_x += xi;
            sum_y += yi;
            sum_x2 += xi * xi;
            sum_xy += xi * yi;
        }

        // Cálculo de b y a
        double b = (n * sum_xy - sum_x * sum_y) / (n * sum_x2 - sum_x * sum_x);
        double a = (sum_y - b * sum_x) / n;

        // Redondear a 4 cifras decimales
        String a_str = df.format(a);
        String b_str = df.format(b);

        // Mostrar resultado
        System.out.println("\nResultado:");
        System.out.println("La recta de regresion es: y = " + a_str + " + " +
b_str + "x");

        scanner.close();
    }
}
```

## Ejemplo 1

Estimar la temperatura a partir de la hora del día

| Hora (x) | Temperatura (y °C) |
|---|---|
| 6 | 15.0 |
| 9 | 18.4 |
| 12 | 21.5 |
| 15 | 24.0 |
| 18 | 22.1 |

Se desea estimar la temperatura a las 10:00 horas (usando el modelo lineal ajustado).

**Datos introducidos:**

En los valores de x ingresamos los siguientes datos:

```
6, 9, 12, 15, 18
```

En los valores de y se ingresan:

```
15.0, 18.4, 21.5, 24.0, 22.1
```

**En el código sería algo así:**

ArrayList<Double> x = [6.0, 9.0, 12.0, 15.0, 18.0];
ArrayList<Double> y = [15.0, 18.4, 21.5, 24.0, 22.1];

**Resultado lanzado por el código:**
La recta de regresion es: y = 14.6111 + 0.5714x

**Temperatura estimada a las 10:00 h: y = 14.6111 + 0.5714(10) = 20.3251 °C**
**El valor estimado es: 20.3251 °C**

## Ejemplo 2

Relación entre años de experiencia y salario mensual

| **Años de experiencia (x)** | **Salario (y, USD)** |
|---|---|
| 1 | 1500 |
| 3 | 2100 |
| 5 | 2600 |
| 7 | 3000 |
| 9 | 3400 |

Desea estimar el salario para 6 años de experiencia.

**Datos introducidos:**
ArrayList<Double> x = [1.0, 3.0, 5.0, 7.0, 9.0];
ArrayList<Double> y = [1500.0, 2100.0, 2600.0, 3000.0, 3400.0];

**Resultado lanzado por el código:**
La recta de regresion es: y = 1394.2857 + 238.5714x

**Salario estimado para 6 años: y = 1394.2857 + 238.5714(6) = 2821.7141 USD**
**El salario estimado es: 2821.7141 USD**

## Ejemplo 3

Publicidad vs ventas

| Inversión en publicidad (x, miles USD) | Ventas (y, miles USD) |
|---|---|
| 1 | 5.2 |
| 2 | 6.8 |
| 3 | 7.9 |
| 4 | 9.5 |
| 5 | 11.3 |

Desea estimar las ventas si se invierten 3.5 miles de USD.

**Datos introducidos:**
ArrayList<Double> x = [1.0, 2.0, 3.0, 4.0, 5.0];
ArrayList<Double> y = [5.2, 6.8, 7.9, 9.5, 11.3];

**El resultado lanzado sería:**
La recta de regresion es: y = 4.62 + 1.34x

**Ventas estimadas para 3.5:**
y = 4.62 + 1.34(3.5) = 9.31 miles de USD

**Las ventas estimadas con 3.5 de inversión son 9.31 miles de USD**

## Ejemplo 4

Relación entre tiempo y distancia recorrida

| Tiempo (horas, x) | Distancia (km, y) |
|---|---|
| 1 | 50 |
| 2 | 100 |
| 3 | 150 |
| 4 | 200 |
| 5 | 250 |

Desea estimar la distancia para 6 horas.

**Datos introducidos:**
ArrayList<Double> x = [1.0, 2.0, 3.0, 4.0, 5.0];
ArrayList<Double> y = [50.0, 100.0, 150.0, 200.0, 250.0];

**Resultado lanzado por el código:**

**La recta de regresión es:** $y = 0 + 50x$
**Distancia estimada para 6 horas:** $y = 0 + 50(6) = 300$ km
**La distancia estimada es: 300 km**

## Ejemplo 5 (Caso de error)

Relación entre cantidad de fertilizante y rendimiento del cultivo

| **Cantidad de fertilizante (kg, x)** | **Rendimiento (toneladas, y)** |
|---|---|
| 1 | 2.0 |
| 2 | 4.0 |
| 3 | 6.0 |
| 4 | 4.0 |
| 5 | 2.0 |

Desea estimar el rendimiento para 3.5 kg.

**Datos introducidos:**
ArrayList<Double> x = [1.0, 2.0, 3.0, 4.0, 5.0];
ArrayList<Double> y = [2.0, 4.0, 6.0, 4.0, 2.0];

**Resultado lanzado por el código:**

**La recta de regresión es:** y = 3.6 + 0x
**Rendimiento estimado para** 3.5 kg: y = 3.6 + 0(3.5) = 3.6 toneladas
**El rendimiento estimado es: 3.6 toneladas**

**Este método no es adecuado para este problema porque los datos muestran un patrón no lineal (parabólico), mientras que el método de mínimos cuadrados asume una relación lineal. Un modelo cuadrático sería más apropiado para capturar la tendencia de los datos.**

## Conclusión

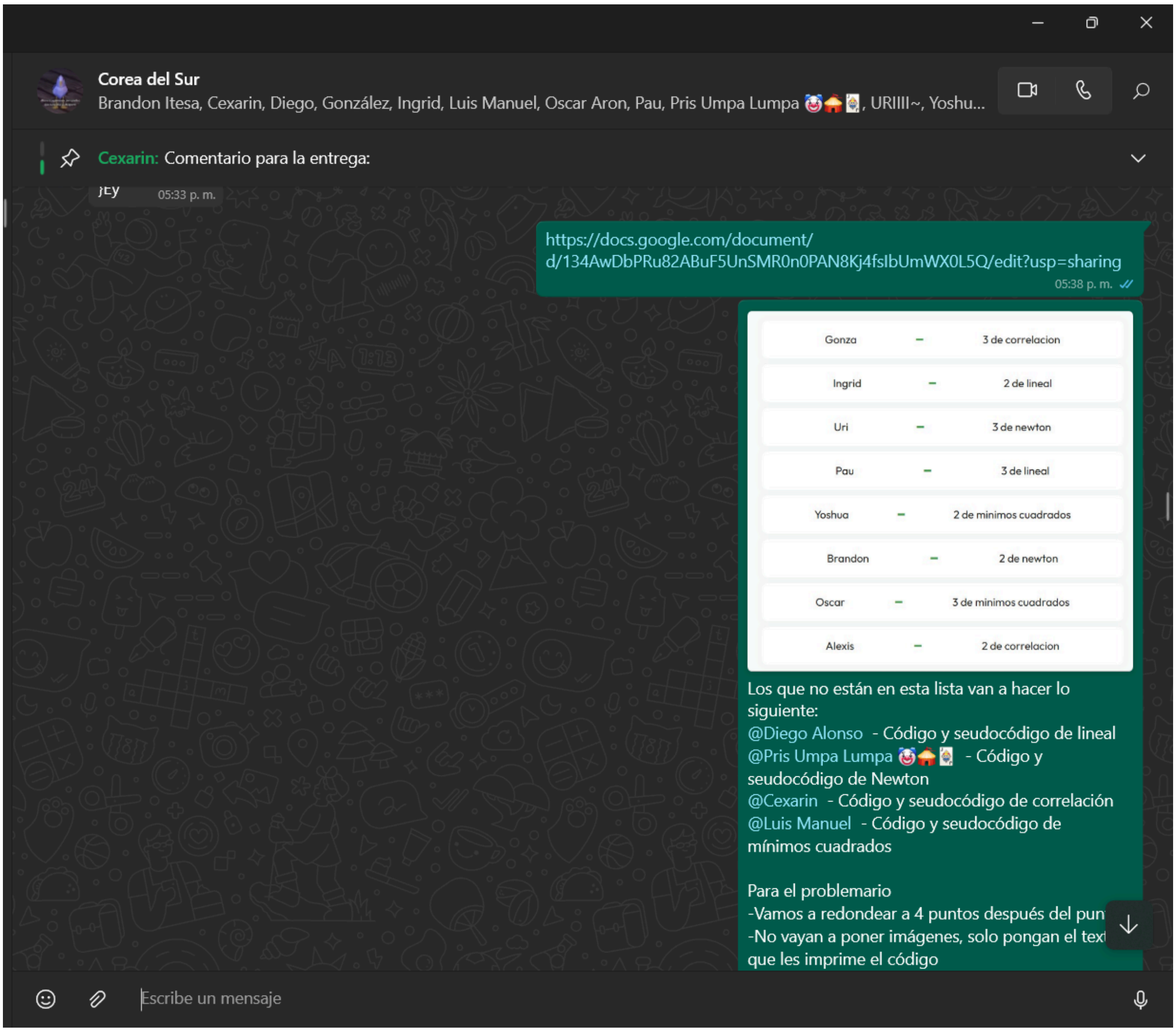
A lo largo de este problemario se han aplicado diversos métodos numéricos fundamentales para el análisis y modelado de datos. En primer lugar, los métodos de interpolación, tanto lineal como polinómica, permitieron estimar valores intermedios dentro del rango de datos conocidos, demostrando su utilidad para aproximaciones en contextos donde la exactitud puntual es clave. La interpolación lineal, por su simplicidad, resultó eficiente en situaciones con pocos datos o comportamiento casi lineal, mientras que la interpolación polinómica ofreció mayor precisión al considerar más puntos, aunque con riesgos de oscilaciones indeseadas si no se usa adecuadamente.

Por otro lado, los métodos de regresión y correlación fueron esenciales para identificar tendencias y relaciones entre variables. La regresión permitió ajustar modelos matemáticos que describen el comportamiento de los datos, mientras que la correlación cuantificó la fuerza y dirección de dicha relación. El método de mínimos cuadrados, aplicado especialmente en regresión lineal, mostró cómo minimizar errores en la predicción, reforzando la importancia de los métodos estadísticos en la toma de decisiones informadas a partir de datos experimentales o recolectados.

En conjunto, estos métodos ofrecen herramientas poderosas para modelar fenómenos reales, realizar predicciones y tomar decisiones con base en evidencia cuantitativa. Su dominio resulta fundamental en el campo de la ingeniería, las ciencias aplicadas y cualquier disciplina que implique análisis de datos.

# Anexos

## Trabajo colaborativo

## Integrantes

Cortes De Lucio Alexis Martin

Cortes Ramirez Priscila

Coronel Vargas Diego Alonso

Delgadillo Fernandez Oscar Aaron

Garcia Ordaz Brandon

Garcia Torres Ernesto Uriel

Gonzalez Ramirez Angel

Hernandez Cruz Luis Manuel

Hernandez Garrido Paulina

Labastida Jimenez Brandon Joshua

Ortega Ramos Ingrid

Rojas Moreno César