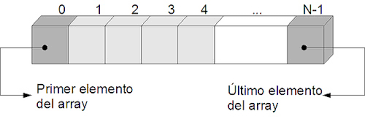
**Evidencia No 4.2**

**Evidencia de conocimiento (Vector)**

****

1. **Fecha:** Viernes, Marzo 7 de 2025
2. **Entregables:**
3. Proyecto netbeans (Código en lenguaje java)
4. Diagrama de clases de cada ejercicio (Usar Visual Paradigm)
5. **Actividad**

**Proyecto No 1.**  **Diagonal principal**

Cree una matriz de tamaño especificado por el usuario (FxC) y permita a través de métodos:

* Llenar la matriz con datos enteros
* Imprimir todos los datos de la matriz
* Imprimir los datos ubicados en la diagonal principal

**Proyecto No 2. Letra N**

Cree una matriz de tamaño especificado por el usuario (FxC) y permita a través de métodos:

* Llenar la matriz con datos enteros
* Imprimir todos los datos de la matriz
* Imprimir los datos que conforman la letra N

**Proyecto No 3.**  **Imprimir datos de una columna especificada por el usuario**

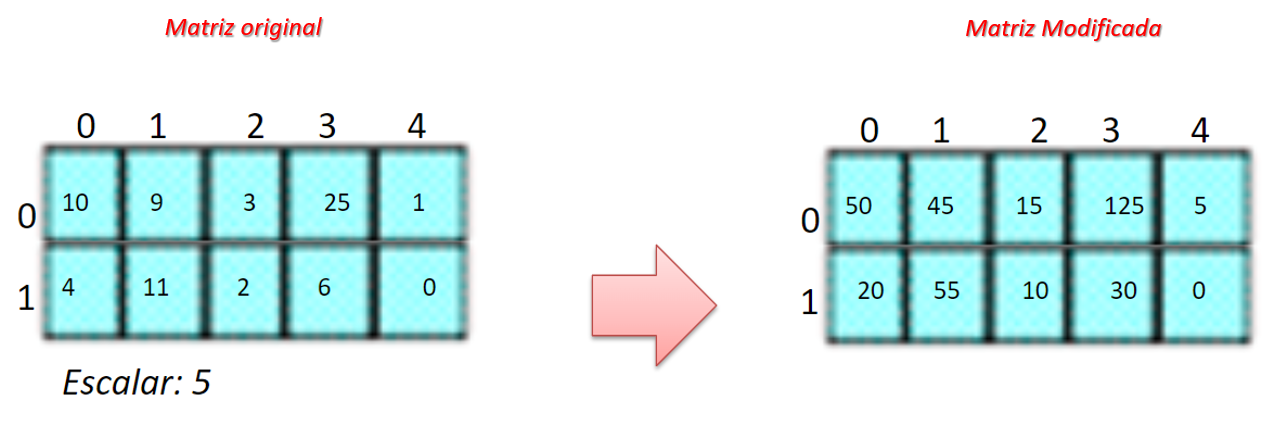
Cree una matriz de tamaño especificado por el usuario (FxC) y permita a través de métodos:

* Llenar la matriz con datos enteros
* Imprimir todos los datos de la matriz
* Recibir por parámetros el número de una columna e imprimir los datos de la misma
* Main: Repetir el proceso usando **el ciclo de su preferencia**

**Proyecto 4. Multiplicación escalar**

En matemáticas, el término multiplicación escalar se refiere al producto de un número real (llamado escalar) por una matriz.

En la multiplicación escalar, cada entrada en la matriz se multiplica por el escalar dado:

****

Calcular la multiplicaciòn de un escalar y una matriz de tamaño M x N



- Usar arreglos unidimensionales

- Usar métodos estáticos siguiendo el modelo de entrada-proceso-salida

- Valide lo que usted considere necesario.

- Repita mientras el usuario así lo desee

**Proyecto No 5.**  **Relaciones de amistad entre personas**

****

La organización llamada "Amigos y Conexiones" ha decidido realizar un estudio único para comprender mejor cómo las relaciones de amistad influyen en nuestras vidas, en el trabajo, en el bienestar emocional y en la comunidad. Para asegurar que el estudio sea lo más representativo posible, la organización está invitando a las personas de diversas edades, profesiones y ubicaciones a participar.

Para participar, las personas deben registrarse y compartir sus relaciones de amistad de forma segura y confidencial.

Se desea representar la lista de amigos de cada persona en una red social. Los datos serán almacenados en una tabla. Por lo anterior, utilizaremos un array bidimensional para almacenar esta información.

* Cada fila del array representa a una persona
* Cada columna dentro de una fila representa un amigo de esa persona.
* Si la persona en la fila i es amiga de la persona en la columna j, se marcará con un **1**, de lo contrario se marcará con un **0**.

Usted ha sido contratado para:

1. Método ***registrar participantes***:

En la matriz se debe almacenar únicamente la columna cero, la cual representa

1. Método ***registrar amistad***:

Registrar las relaciones de amistad existentes preguntando para cada participante si es amigo de otro participante. Ejemplo: José, ¿Eres amigo de Claudia?

Para ello, es necesario

1. Método ***imprimir relaciones***:

Imprimir los datos que representan las relaciones de amistad:

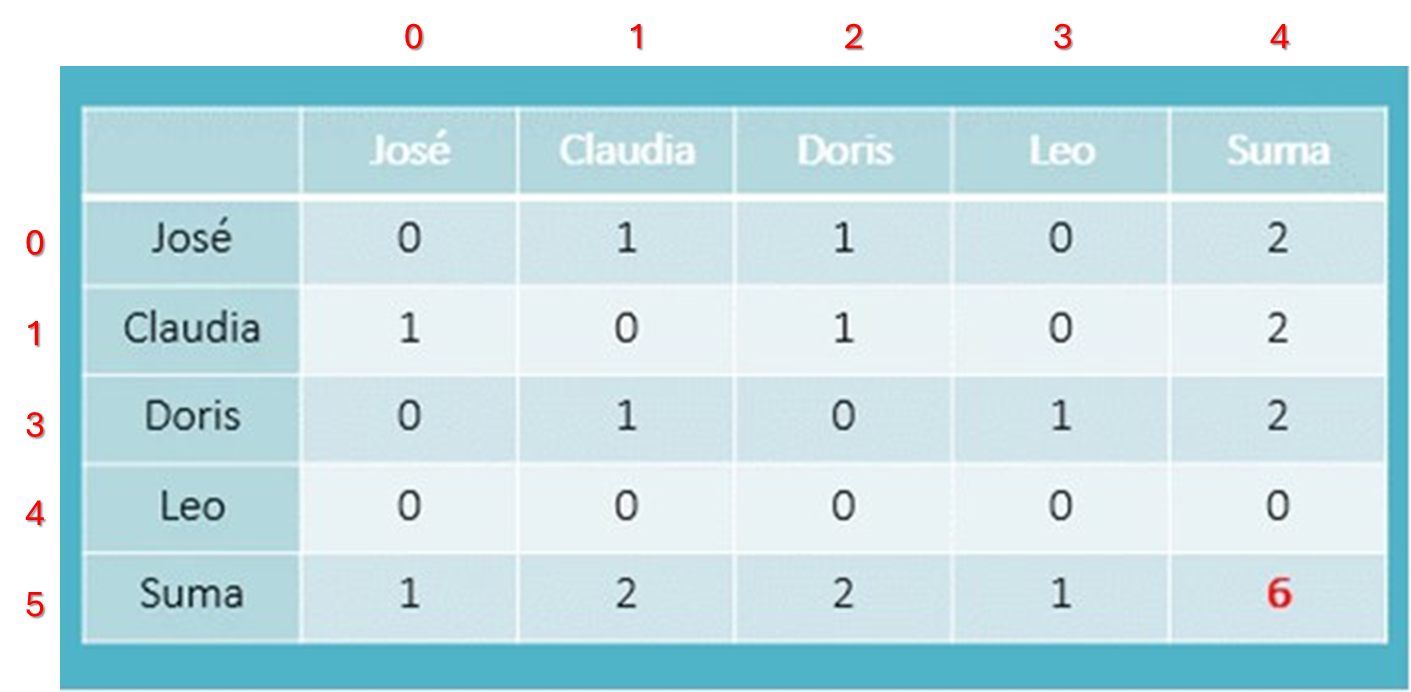
* Imprimir la tabla con valores 1 y 0
* Incluya los nombres de las personas

1. Método ***ver amigos:***

Recibir por parámetros el número de una fila (el cual representa a una persona o usuario de la red social) e imprimir los nombres de sus amigos.

1. Método ***Main***: Repetir el proceso usando **el ciclo de su preferencia**

Ejemplo:

****

Como posible solución, usaremos un array bidimensional (matriz) con el fin de representar las relaciones de amistad:

* Filas: personas
* Columnas: Amigos

**Punto de partida:**

| public class Main  {  public static void main(String[] args) {    int fila;  int columnas;    //Tamaño de la matriz: 5X3  fila= 5;  columnas=3;    //Declarar matriz  int mat[][]= new int[fila][columnas];    //Obtener tamaño:  System.out.println("Filas: "+ mat.length + "\nColumnas: "+mat[1].length);    for (int i = 0; i < mat.length ; i++)  {    for (int j = 0; j < mat[i].length; j++)  {  System.out.print( "["+ i + "." + j + "] ");  }    System.out.println("\n");  }    }  } |
| --- |



- Usar arreglos bidimensionales

- Usar métodos estáticos siguiendo el modelo de entrada-proceso-salida

- Valide lo que usted considere necesario.

- Repita mientras el usuario así lo desee

*-*