



**UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA**

**MODALIDAD PRESENCIAL**

**FACULTAD: INGENIERÍA SISTEMAS**

**TITULACIÓN DE COMPUTACIÓN**

**Taller De Programación**

**Docente: ing. María del Carmen Cabrera Loayza**

**Nombre: Bryam Joel Jiménez Hernández**

**Paralelo: Fundamentos Computacionales – Introducción a la Programación**

### Ejercicio 1.

Leer n números enteros, almacenarlos en un vector. Luego leer un entero y determinar cuántos números de los almacenados en el vector terminan en el mismo dígito que el último valor leído.

#### Mini especificación

##### Inicio

```
int n,i,cont, num;

cont = 0;

<<"Ingrese la longitud del vector"

>> n

<< "ingrese un número entero:"

>>num

A [], i[0-n]

Para( i ← 0; i < n; i++) {

    <<"Ingrese datos del vector A["+i+"] = "

    >>A[i]

Fin_Para

Para i ← 0; i < 4; i++ hacer

    << A[i] + " ")

Fin_para

para (i ← 0; i < n; i++) hacer

    SI (A[i]%10) == num entonces

        cont ← cont+1

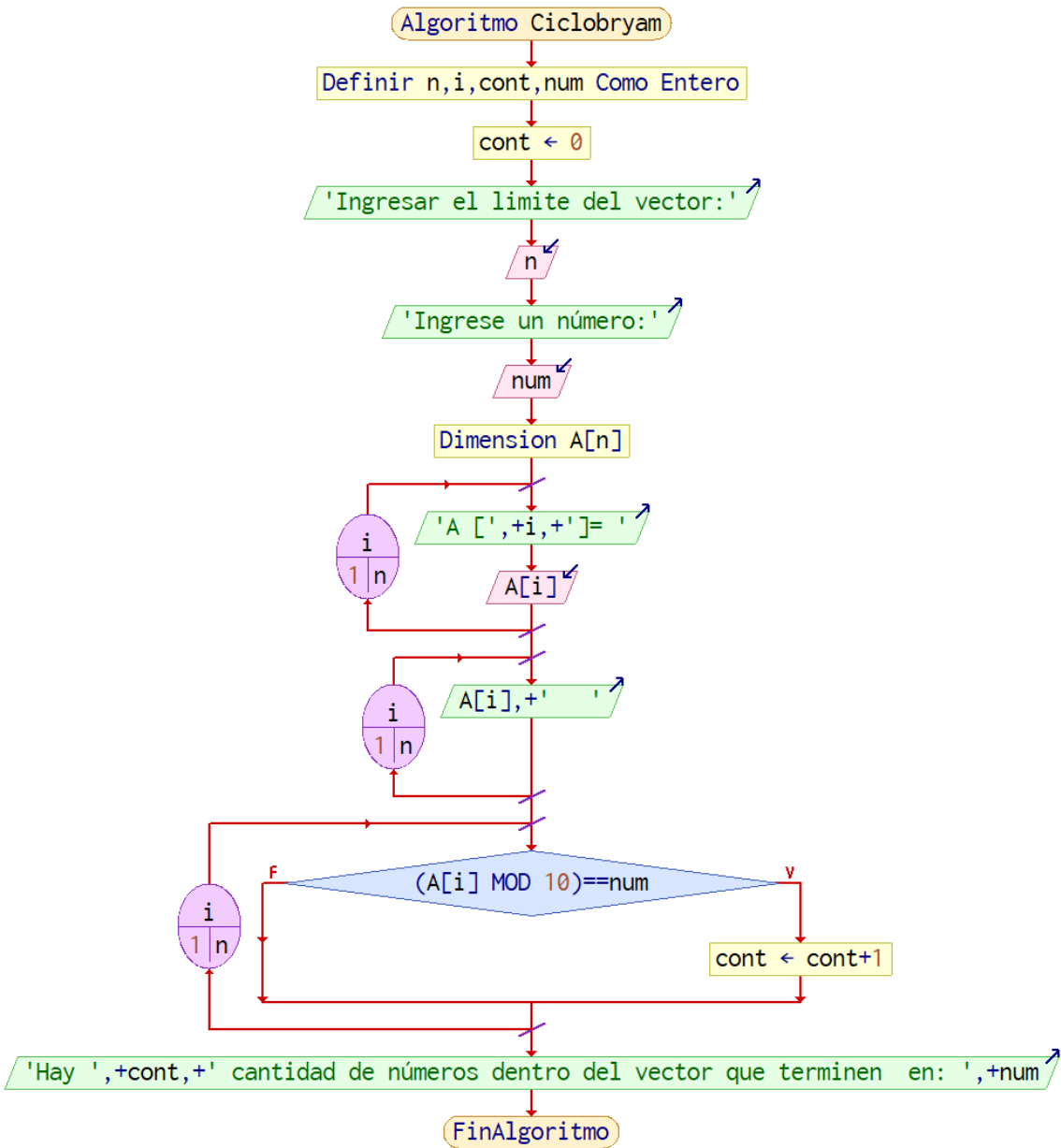
    Fin_Si

Fin_para

<<"Hay "+ cont, + " números dentro del vector que terminen en: ", + num
```

##### Fin

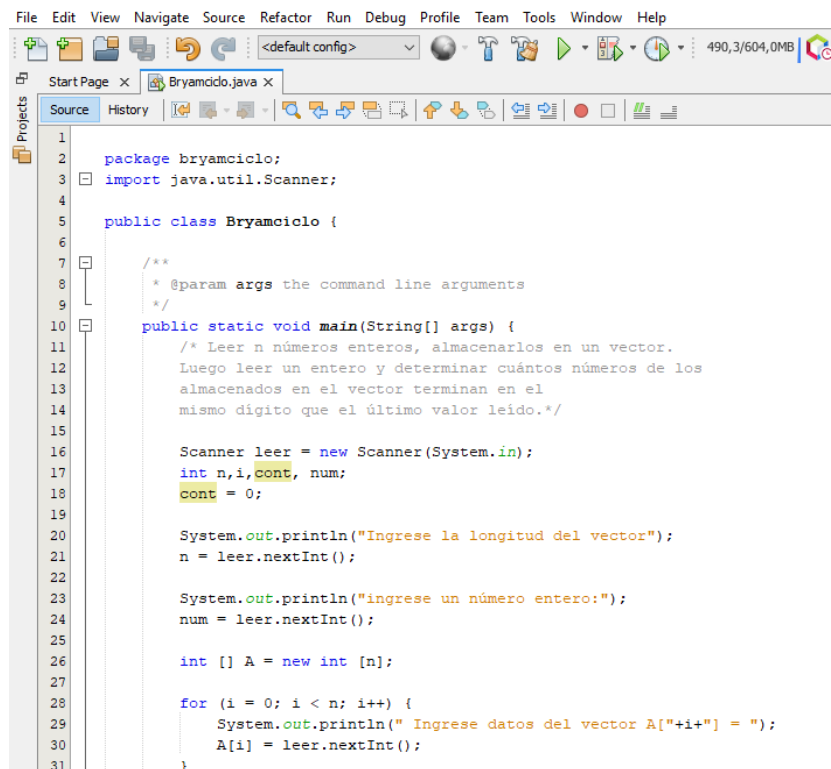
Diagrama de Flujo



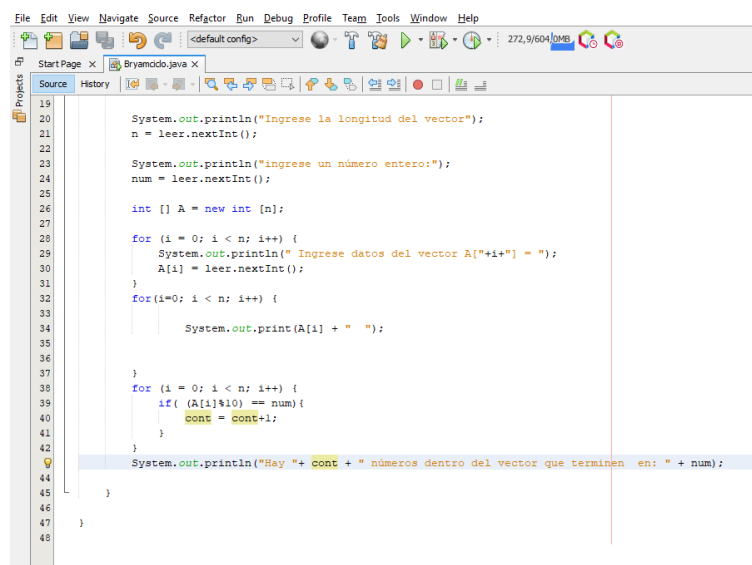
Prueba de escritorio

Prueba de Escritorio						
Proceso/SubProceso	Linea(inst)	n	i	num	A[i]	cont
1.CICLOBRYAM	1(1)	<<Variable no inicializada (N)>>	<<Variable no inicializada (I)>>	<<Variable no inicializada (NUM)>>	ificador A no corresponde a un arreglo o subp	<<Variable no inicializada (CONT)>>
1.CICLOBRYAM	2(1)	<<Variable no inicializada (N)>>	<<Variable no inicializada (I)>>	<<Variable no inicializada (NUM)>>	ificador A no corresponde a un arreglo o subp	<<Variable no inicializada (CONT)>>
1.CICLOBRYAM	3(1)	<<Variable no inicializada (N)>>	<<Variable no inicializada (I)>>	<<Variable no inicializada (NUM)>>	ificador A no corresponde a un arreglo o subp	<<Variable no inicializada (CONT)>>
1.CICLOBRYAM	4(1)	<<Variable no inicializada (N)>>	<<Variable no inicializada (I)>>	<<Variable no inicializada (NUM)>>	ificador A no corresponde a un arreglo o subp	0
1.CICLOBRYAM	5(1)	<<Variable no inicializada (N)>>	<<Variable no inicializada (I)>>	<<Variable no inicializada (NUM)>>	ificador A no corresponde a un arreglo o subp	0
1.CICLOBRYAM	6(1)	4	<<Variable no inicializada (I)>>	<<Variable no inicializada (NUM)>>	ificador A no corresponde a un arreglo o subp	0
1.CICLOBRYAM	7(1)	4	<<Variable no inicializada (I)>>	<<Variable no inicializada (NUM)>>	ificador A no corresponde a un arreglo o subp	0
1.CICLOBRYAM	8(1)	4	<<Variable no inicializada (I)>>	2	ificador A no corresponde a un arreglo o subp	0
1.CICLOBRYAM	9(1)	4	<<Variable no inicializada (I)>>	2	<<Variable no inicializada (I)>>	0
1.CICLOBRYAM	10(1)	4	1	2	0	0
1.CICLOBRYAM	11(1)	4	1	2	0	0
1.CICLOBRYAM	9(1)	4	1	2	1225	0
1.CICLOBRYAM	10(1)	4	2	2	0	0
1.CICLOBRYAM	11(1)	4	2	2	0	0
1.CICLOBRYAM	9(1)	4	2	2	3	0
1.CICLOBRYAM	10(1)	4	3	2	0	0
1.CICLOBRYAM	11(1)	4	3	2	0	0
1.CICLOBRYAM	9(1)	4	3	2	45	0
1.CICLOBRYAM	10(1)	4	4	2	0	0
1.CICLOBRYAM	11(1)	4	4	2	0	0
1.CICLOBRYAM	9(1)	4	4	2	132	0
1.CICLOBRYAM	12(1)	4	5	2	<<Subindice (S) fuera de rango (1..4)>>	0
1.CICLOBRYAM	13(1)	4	5	2	<<Subindice (S) fuera de rango (1..4)>>	0
1.CICLOBRYAM	14(1)	4	1	2	1225	0
1.CICLOBRYAM	13(1)	4	1	2	1225	0
1.CICLOBRYAM	14(1)	4	2	2	3	0
1.CICLOBRYAM	13(1)	4	2	2	3	0
1.CICLOBRYAM	14(1)	4	3	2	45	0
1.CICLOBRYAM	13(1)	4	3	2	45	0
1.CICLOBRYAM	14(1)	4	4	2	132	0
1.CICLOBRYAM	13(1)	4	4	2	132	0
1.CICLOBRYAM	15(1)	4	5	2	<<Subindice (S) fuera de rango (1..4)>>	0
1.CICLOBRYAM	16(1)	4	5	2	<<Subindice (S) fuera de rango (1..4)>>	0
1.CICLOBRYAM	17(1)	4	3	2	45	0
1.CICLOBRYAM	19(1)	4	3	2	45	0
1.CICLOBRYAM	16(1)	4	3	2	45	0
1.CICLOBRYAM	17(1)	4	4	2	132	0
1.CICLOBRYAM	18(1)	4	4	2	132	0
1.CICLOBRYAM	19(1)	4	4	2	132	1
1.CICLOBRYAM	16(1)	4	4	2	132	1
1.CICLOBRYAM	20(1)	4	5	2	<<Subindice (S) fuera de rango (1..4)>>	1
1.CICLOBRYAM	21(1)	4	5	2	<<Subindice (S) fuera de rango (1..4)>>	1
1.CICLOBRYAM	23(1)	4	5	2	<<Subindice (S) fuera de rango (1..4)>>	1

## Programa en Java



```
1
2 package bryamcielo;
3 import java.util.Scanner;
4
5 public class Bryamcielo {
6
7     /**
8      * @param args the command line arguments
9      */
10    public static void main(String[] args) {
11        /* Leer n números enteros, almacenarlos en un vector.
12         Luego leer un entero y determinar cuántos números de los
13         almacenados en el vector terminan en el
14         mismo dígito que el último valor leído.*/
15
16        Scanner leer = new Scanner(System.in);
17        int n,i,cont, num;
18        cont = 0;
19
20        System.out.println("Ingrese la longitud del vector");
21        n = leer.nextInt();
22
23        System.out.println("ingrese un número entero:");
24        num = leer.nextInt();
25
26        int [] A = new int [n];
27
28        for (i = 0; i < n; i++) {
29            System.out.println(" Ingrese datos del vector A["+i+"] = ");
30            A[i] = leer.nextInt();
31        }
```



```
19
20    System.out.println("Ingrese la longitud del vector");
21    n = leer.nextInt();
22
23    System.out.println("ingrese un número entero:");
24    num = leer.nextInt();
25
26    int [] A = new int [n];
27
28    for (i = 0; i < n; i++) {
29        System.out.println(" Ingrese datos del vector A["+i+"] = ");
30        A[i] = leer.nextInt();
31    }
32    for(i=0; i < n; i++) {
33
34        System.out.print(A[i] + " ");
35
36    }
37
38    for (i = 0; i < n; i++) {
39        if ( (A[i]%10) == num) {
40            cont = cont+1;
41        }
42    }
43    System.out.println("Hay "+ cont + " números dentro del vector que terminen en: " + num);
44
45    }
46
47    }
48 }
```

## Programa en Python

```
File Edit Refactoring Source Navigate Search Project Pydev Run Window Help
[Icons]
validarVector
3
4 @author: User
5 ...
6 from sys import stdin, stdout
7
8 def main():
9     vector = []
10
11     stdout.write('Ingresar números enteros para el vector:\n')
12     # Leer número entero desde el teclado, para almacenarlos en un vector
13     for i in range(1,7):
14         nuevoDato = int( input( " Dato numero {}: ".format(i) ))
15         vector.append(nuevoDato)
16
17     print ('El vector tiene los siguientes datos: ')
18     print (vector)
19     # Leer número entero desde el teclado
20     numero = int( input( "\nIngresar un número entero: ".format(i) ))
21
22     # Presentar resultado de la comparación
23     cantidadRepeticion=ContarNumerosRepetidos(vector,numero)
24     stdout.write('\nEl número '+str(numero)+" se repite: "+str(cantidadRepeticion));
25
26     if cantidadRepeticion>1:
27         stdout.write(' veces')
28     else:
29         stdout.write(' vez')
30
31 # Funcion para contar las veces que se repite el número ingresado
32 def ContarNumerosRepetidos(vector,numero):
33     contador=0
34     for i in range(1,7):
35         if vector[i-1]==numero:
36             contador+=1
37     return contador
38
39 if __name__ == '__main__':
40     main()
41
```

## Ejercicio 2.

Leer dos matrices 4x6 enteras y determinar si el mayor número almacenado en una de ellas que pertenezca a la serie de Fibonacci es igual al mayor número almacenado en la otra matriz que pertenezca a la serie de Fibonacci

### Mini especificación

#### Inicio

```
//creación e inicialización de variables
j,i,k,mayorA,n1,n2,n3,mayor i[0-n]
n1←0;n2←1;n3←0;mayorA←0;mayor←0;
A [][] , i [0-n];
B [][] , i [0-n]
fib [] , i [0-100]
para(k←0; k<100;k++) hacer
    n3←n1+n2;
    fib[k] ← n3;

    n1←n2
    n2←n3
Fin_para
para (i = 0; i < 4; i++) hacer
    para ( j = 0; j < 6; j++) hacer
        << " Ingrese datos de la matriz A: [",+i,+"]["",+j,+"]=" "
        >>A[i][j]
    Fin_para
Fin_para
para (i = 0; i < 4; i++) hacer
    para ( j = 0; j < 6; j++) hacer
        <<" Ingrese datos de la matriz B: [",+i,+"]["",+j,+"]=" "
        >>B[i][j]
    Fin_para
Fin_para
//Presentar matrices
```

```

<< " matriz A: "

para (i = 0; i < 4; i++) hacer
    para ( j = 0; j < 6; j++) hacer
        << A[i][j] + " "

    Fin_para
    << " "

Fin_para

<< "matriz B: "

para (i = 0; i < 4; i++) hacer
    para ( j = 0; j < 6; j++) hacer
        << B[i][j] + " "

    Fin_para
    << " "

Fin_para

//Comprobar vector A

para (i = 0; i < 4; i++) hacer
    para ( j = 0; j < 6; j++) hacer
        para (k=0;k<100;k++) hacer
            Si (A[i][j] == fib[k]) &&( mayorA < A[i][j]){
                mayorA ← A[i][j];
            }
        Fin_si
    Fin_para
Fin_para

Fin_para

<< "El mayor de  matriz A es: ",+mayorA


//Comprobar vector B

para (i = 0; i < 4; i++) hacer
    para ( j = 0; j < 6; j++) hacer
        para (k=0;k<100;k++) hacer
            Si (B[i][j] == fib[k]) &&( mayorB < B[i][j]){

```



mayorB  $\leftarrow$  A[i][j];

Fin\_si

Fin\_para

Fin\_para

Fin\_para

<< "El mayor de matriz B es: ",+mayorB

Si (mayorA == mayorB) entonces

<< "Las 2 matrices tienen el mismo número mayor perteneciente a la serie Fibonacci"

Sino

<< "Las 2 matrices no tienen el mismo número mayor perteneciente a la serie Fibonacci"

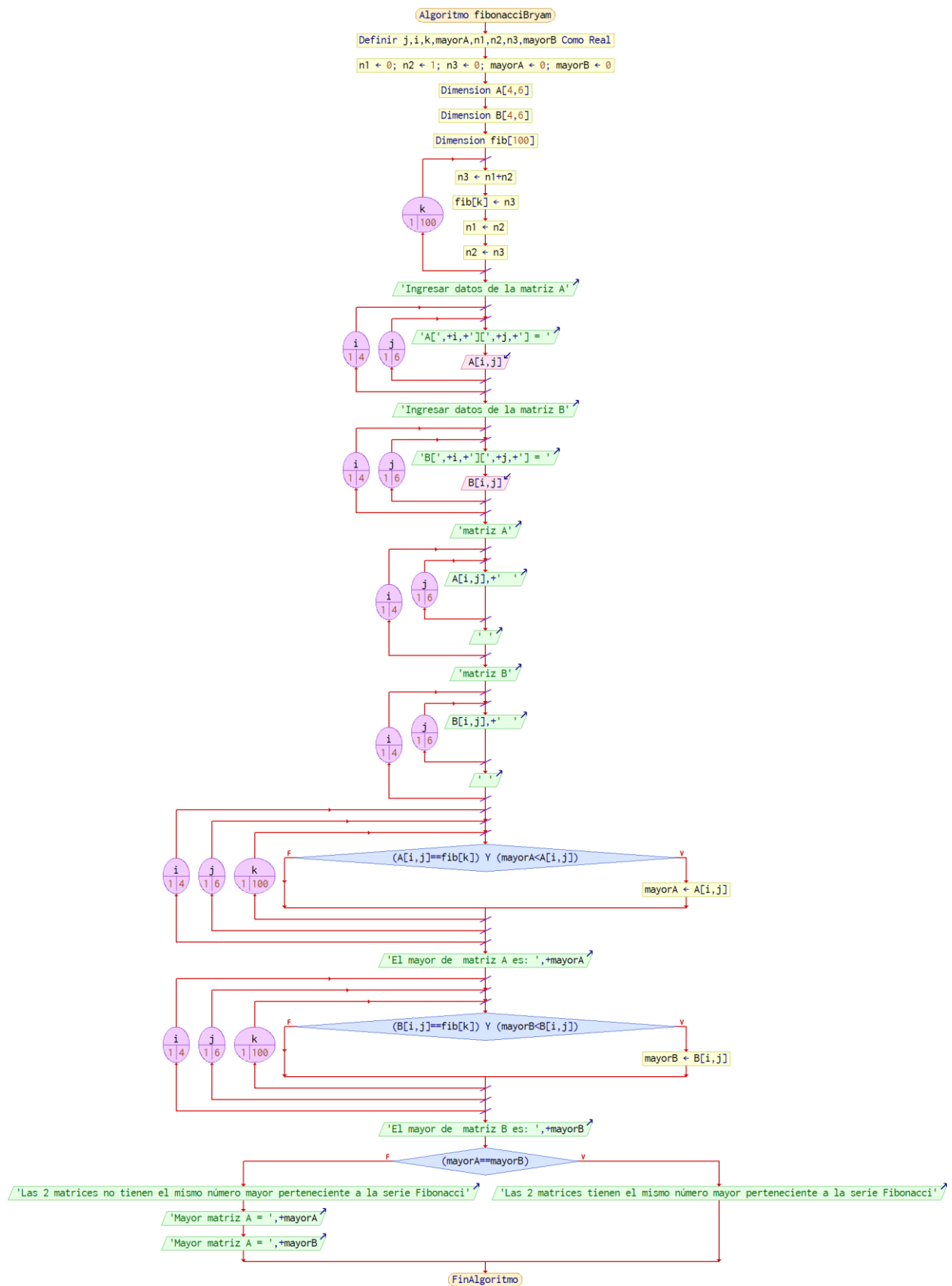
<< "Mayor matriz A = ", +mayorA

<< "Mayor matriz B = ",+ mayorB

Fin\_si

Fin

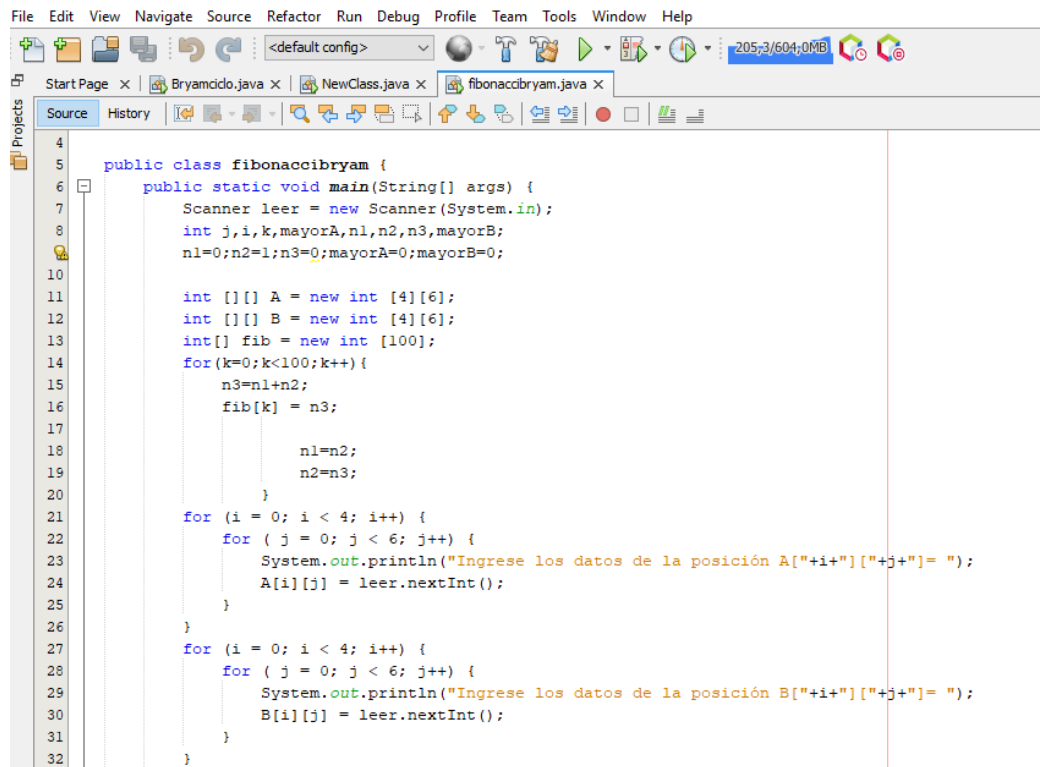
## Diagrama de Flujo



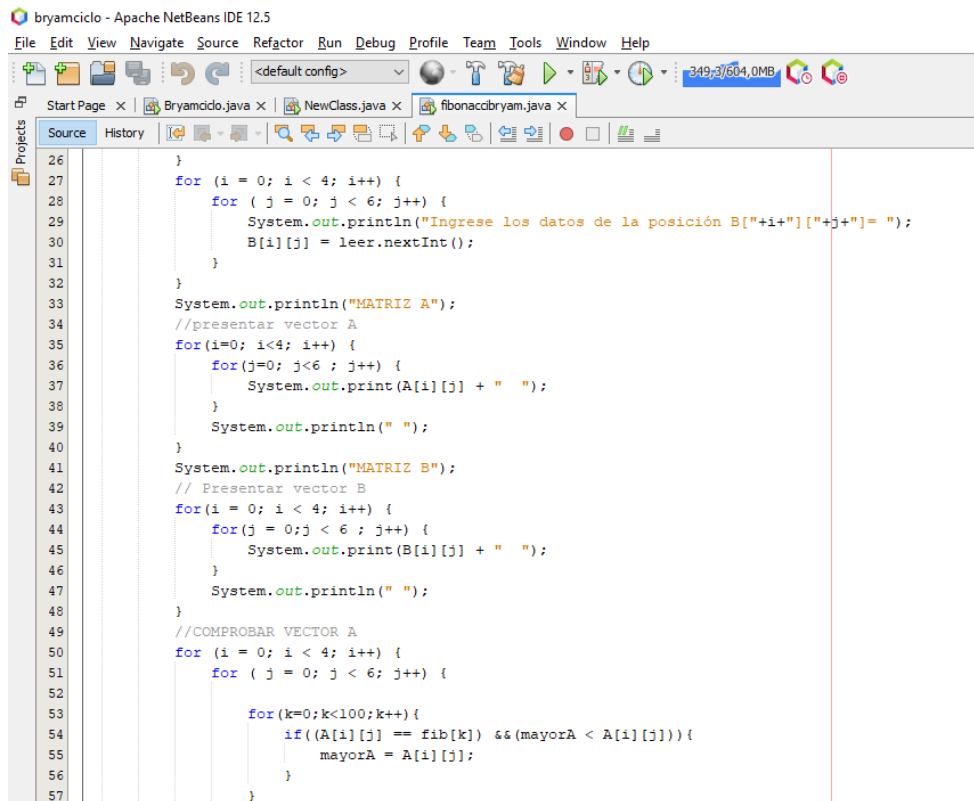




## Programa en Java



```
4
5 public class fibonaccibryam {
6     public static void main(String[] args) {
7         Scanner leer = new Scanner(System.in);
8         int j,i,k,mayorA,n1,n2,n3,mayorB;
9         n1=0;n2=1;n3=0;mayorA=0;mayorB=0;
10
11         int [][] A = new int [4][6];
12         int [][] B = new int [4][6];
13         int[] fib = new int [100];
14         for(k=0;k<100;k++){
15             n3=n1+n2;
16             fib[k] = n3;
17
18             n1=n2;
19             n2=n3;
20         }
21         for (i = 0; i < 4; i++) {
22             for ( j = 0; j < 6; j++) {
23                 System.out.println("Ingrese los datos de la posición A["+i+"]["+j+"]= ");
24                 A[i][j] = leer.nextInt();
25             }
26         }
27         for (i = 0; i < 4; i++) {
28             for ( j = 0; j < 6; j++) {
29                 System.out.println("Ingrese los datos de la posición B["+i+"]["+j+"]= ");
30                 B[i][j] = leer.nextInt();
31             }
32         }
```



```
bryamiciclo - Apache NetBeans IDE 12.5
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Start Page x | Bryamiciclo.java x | NewClass.java x | fibonaccibryam.java x
Source History
26     }
27     for (i = 0; i < 4; i++) {
28         for ( j = 0; j < 6; j++) {
29             System.out.println("Ingrese los datos de la posición B["+i+"]["+j+"]= ");
30             B[i][j] = leer.nextInt();
31         }
32     }
33     System.out.println("MATRIZ A");
34     //presentar vector A
35     for(i=0; i<4; i++) {
36         for(j=0; j<6 ; j++) {
37             System.out.print(A[i][j] + " ");
38         }
39         System.out.println(" ");
40     }
41     System.out.println("MATRIZ B");
42     // Presentar vector B
43     for(i = 0; i < 4; i++) {
44         for(j = 0;j < 6 ; j++) {
45             System.out.print(B[i][j] + " ");
46         }
47         System.out.println(" ");
48     }
49     //COMPROBAR VECTOR A
50     for (i = 0; i < 4; i++) {
51         for ( j = 0; j < 6; j++) {
52
53             for(k=0;k<100;k++){
54                 if((A[i][j] == fib[k]) &&(mayorA < A[i][j])){
55                     mayorA = A[i][j];
56                 }
57             }
```

Start Page x | Bryamciclo.java x | NewClass.java x | fibonaccibryam.java x

Source History

```
54         if((A[i][j] == fib[k]) && (mayorA < A[i][j])){
55             mayorA = A[i][j];
56         }
57     }
58 }
59 }
60 System.out.println("El mayor de matriz A es: "+mayorA);
61
62 //VECTOR B a COMPROBAR
63 for (i = 0; i < 4; i++) {
64     for (j = 0; j < 6; j++) {
65
66         for(k=0;k<100;k++){
67
68             if((B[i][j] == fib[k]) && (mayorB < B[i][j])){
69                 mayorB = B[i][j];
70             }
71         }
72     }
73 }
74 System.out.println("El mayor de matriz B es: "+mayorB);
75 if(mayorA == mayorB){
76     System.out.println("Las 2 matrices tienen el mismo número mayor perteneciente a la serie Fibonacci");
77 }
78 else{
79     System.out.println("Las 2 matrices no tienen el mismo número mayor perteneciente a la serie Fibonacci");
80     System.out.println("Mayor matriz A = "+mayorA);
81     System.out.println("Mayor matriz B = "+mayorB);
82 }
83
84 }
```

## Programa en Python

```
wsk_001 - ContarNumerosVector/vector/fibonacci.py - Eclipse IDE
File Edit Refactoring Source Navigate Search Project Pydev Run Window Help

PyDev Package Explorer
vector
  fibonacci.py
  validarVector.py
python10 (C:\Use...hon310\python
Ejemplo
EJERCICIOS
  ContarNumerosVector
    vector
      fibonacci.py
      validarVector.py
  MatrizInversa
    Inversa
      matriz_inversa.py
  MaximoValorMatrices
    Matrices
      _init_.py
      evaluar_matrices.py
  NumRepeticionMaximoValorMatriz
    ContarRepetNumMaxMatriz
    python10 (C:\Use...hon310\python
  MatrizInversa
  MaximoValorMatrices
  NumRepeticionMaximoValorMatriz
  proyecto
  ProyectoCiclo

10 '''
11 Created on 23 ene. 2022
12
13 @author: User
14
15 def main():
16     #Declaracion de variable
17
18     #Leer datos de la matriz A
19     print('Ingresar datos de la matriz A:')
20     A = leeMatriz(4,6)
21     print('Matriz A')
22     print(A)
23
24     #Leer datos de la matriz B
25     print('Ingresar datos de la matriz B:')
26     B = leeMatriz(4,6)
27     print('Matriz B')
28     print(B)
29
30     mayorA = determinarMaximoValorMatriz(A)
31     mayorB = determinarMaximoValorMatriz(B)
32
33     if mayorA == mayorB :
34         print(f'Las 2 matrices tienen el mismo número mayor perteneciente a la serie Fibonacci: {mayorA}')
35     else:
36         print('Las 2 matrices no tienen el mismo número mayor perteneciente a la serie Fibonacci')
37         print('Maximo valor de la matriz A: '+str(mayorA))
38         print('Maximo valor de la matriz B: '+str(mayorB))
39
40 def creaMatriz(n,m):
41     matriz = []
42     for i in range(n):
43         a = [0]*m
44         matriz.append(a)
45     return matriz
46
47 def leeMatriz(n,m):
48     A = creaMatriz(n,m)
49     for i in range(n):
50         for j in range(m):
51             A[i][j] = int(input('Introduce dato (%d,%d):'%(i,j)))
52     return A
53
54 def determinarMaximoValorMatriz(matriz):
55     max_valor=[0][0]
56     for x in range(len(matriz)):
57         for z in range(len(matriz[x])):
58             if z>max_valor and validarNumeroFibonacci(z):
59                 max_valor=z
60     return max_valor
61
62 def validarNumeroFibonacci(max_valor):
63     n1=0
64     n2=1
65     n3=0
66     f=[]
67     flag=False
68     for i in range(100):
69         n3=n1+n2
70         f.append(n3)
71         n1=n2
72         n2=n3
73     for i in range(100):
74         if f[i-1] == max_valor:
75             flag=True
76     return flag
77
78 if __name__ == '__main__':
79     main()
80
81 '''
```

```
wsk_001 - ContarNumerosVector/vector/fibonacci.py - Eclipse IDE
File Edit Refactoring Source Navigate Search Project Pydev Run Window Help

PyDev Package Explorer
vector
  fibonacci.py
  validarVector.py
python10 (C:\Use...hon310\python
Ejemplo
EJERCICIOS
  ContarNumerosVector
    vector
      fibonacci.py
      validarVector.py
  MatrizInversa
    Inversa
      matriz_inversa.py
  MaximoValorMatrices
    Matrices
      _init_.py
      evaluar_matrices.py
  NumRepeticionMaximoValorMatriz
    ContarRepetNumMaxMatriz
    python10 (C:\Use...hon310\python
  MatrizInversa
  MaximoValorMatrices
  NumRepeticionMaximoValorMatriz
  proyecto
  ProyectoCiclo

49 mayorB = determinarMaximoValorMatriz(B)
50
51 if mayorA == mayorB :
52     print(f'Las 2 matrices tienen el mismo número mayor perteneciente a la serie Fibonacci: {mayorA}')
53 else:
54     print('Las 2 matrices no tienen el mismo número mayor perteneciente a la serie Fibonacci')
55     print('Maximo valor de la matriz A: '+str(mayorA))
56     print('Maximo valor de la matriz B: '+str(mayorB))
57
58 def creaMatriz(n,m):
59     matriz = []
60     for i in range(n):
61         a = [0]*m
62         matriz.append(a)
63     return matriz
64
65 def leeMatriz(n,m):
66     A = creaMatriz(n,m)
67     for i in range(n):
68         for j in range(m):
69             A[i][j] = int(input('Introduce dato (%d,%d):'%(i,j)))
70     return A
71
72 def determinarMaximoValorMatriz(matriz):
73     max_valor=[0][0]
74     for x in range(len(matriz)):
75         for z in range(len(matriz[x])):
76             if z>max_valor and validarNumeroFibonacci(z):
77                 max_valor=z
78     return max_valor
79
80 def validarNumeroFibonacci(max_valor):
81     n1=0
82     n2=1
83     n3=0
84     f=[]
85     flag=False
86     for i in range(100):
87         n3=n1+n2
88         f.append(n3)
89         n1=n2
90         n2=n3
91     for i in range(100):
92         if f[i-1] == max_valor:
93             flag=True
94     return flag
95
96 if __name__ == '__main__':
97     main()
98
99 '''
```

```
wsk_001 - ContarNumerosVector/vector/fibonacci.py - Eclipse IDE
File Edit Refactoring Source Navigate Search Project Pydev Run Window Help

PyDev Package Explorer
vector
  fibonacci.py
  validarVector.py
python10 (C:\Use...hon310\python
Ejemplo
EJERCICIOS
  ContarNumerosVector
    vector
      fibonacci.py
      validarVector.py
  MatrizInversa
    Inversa
      matriz_inversa.py
  MaximoValorMatrices
    Matrices
      _init_.py
      evaluar_matrices.py
  NumRepeticionMaximoValorMatriz
    ContarRepetNumMaxMatriz
    python10 (C:\Use...hon310\python
  MatrizInversa
  MaximoValorMatrices
  NumRepeticionMaximoValorMatriz
  proyecto
  ProyectoCiclo

45 def leeMatriz(n,m):
46     A = creaMatriz(n,m)
47     for i in range(n):
48         for j in range(m):
49             A[i][j] = int(input('Introduce dato (%d,%d):'%(i,j)))
50     return A
51
52 def determinarMaximoValorMatriz(matriz):
53     max_valor=[0][0]
54     for x in range(len(matriz)):
55         for z in range(len(matriz[x])):
56             if z>max_valor and validarNumeroFibonacci(z):
57                 max_valor=z
58     return max_valor
59
60 def validarNumeroFibonacci(max_valor):
61     n1=0
62     n2=1
63     n3=0
64     f=[]
65     flag=False
66     for i in range(100):
67         n3=n1+n2
68         f.append(n3)
69         n1=n2
70         n2=n3
71     for i in range(100):
72         if f[i-1] == max_valor:
73             flag=True
74     return flag
75
76 if __name__ == '__main__':
77     main()
78
79 '''
```

### Ejercicio 3

Construir una función que reciba como parámetro una matriz 4x4 entera y retorne el número de la fila en donde se encuentre por primera vez el número mayor de la matriz.

#### Mini especificación

Inicio

```
//creación e inicialización de variables
j,i,mayor,cont i[0-n]
mayor←0;cont←0;
A [3][4] , i [0-n];
para (i = 0; i < 3; i++) hacer
    para ( j = 0; j < 4; j++) hacer
        << " Ingrese datos de la matriz A: [",+i,+"][",+j,+"]= "
        >>A[i][j]
    Fin_para
Fin_para
Fin_para
Mayor←A[0][0];
cont ← mayorRep(A,mayor)
<< "El número mayor se repitió ",+ cont ,+ " cantidad de veces."
```

mayorRep( B, mayor1)

```
cont i[0-n]
cont←0
para (i = 0; i < 4; i++) hacer
    para ( j = 0; j < 6; j++) hacer
        Si mayor1<B[i][j] entonces
            mayor1 = B[i][j]
        Fin_si
    Fin_para
    << " "
Fin_para
para (i = 0; i < 4; i++) hacer
    para ( j = 0; j < 6; j++) hacer
```



Si (mayor1 == B[i][j]) entonces

cont = cont + 1

Fin\_si

Fin\_para

<< " "

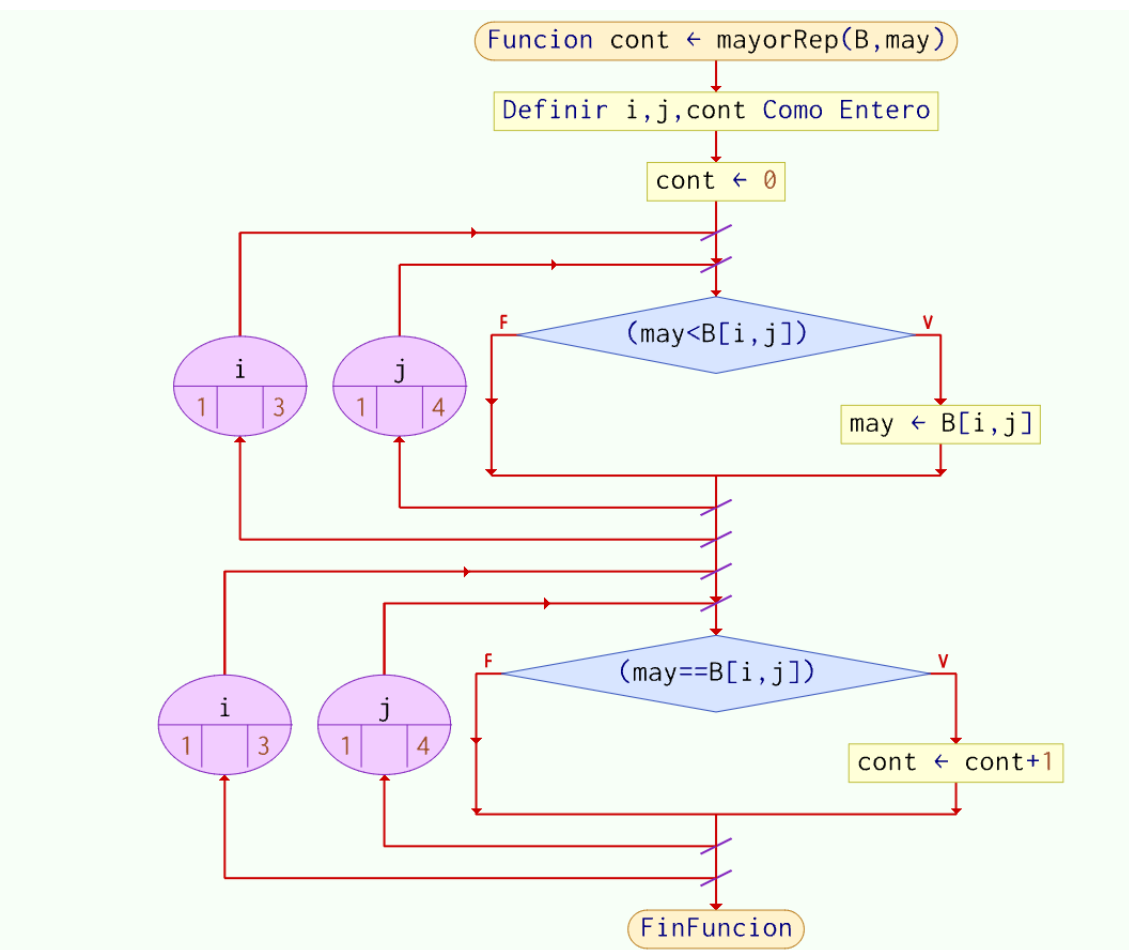
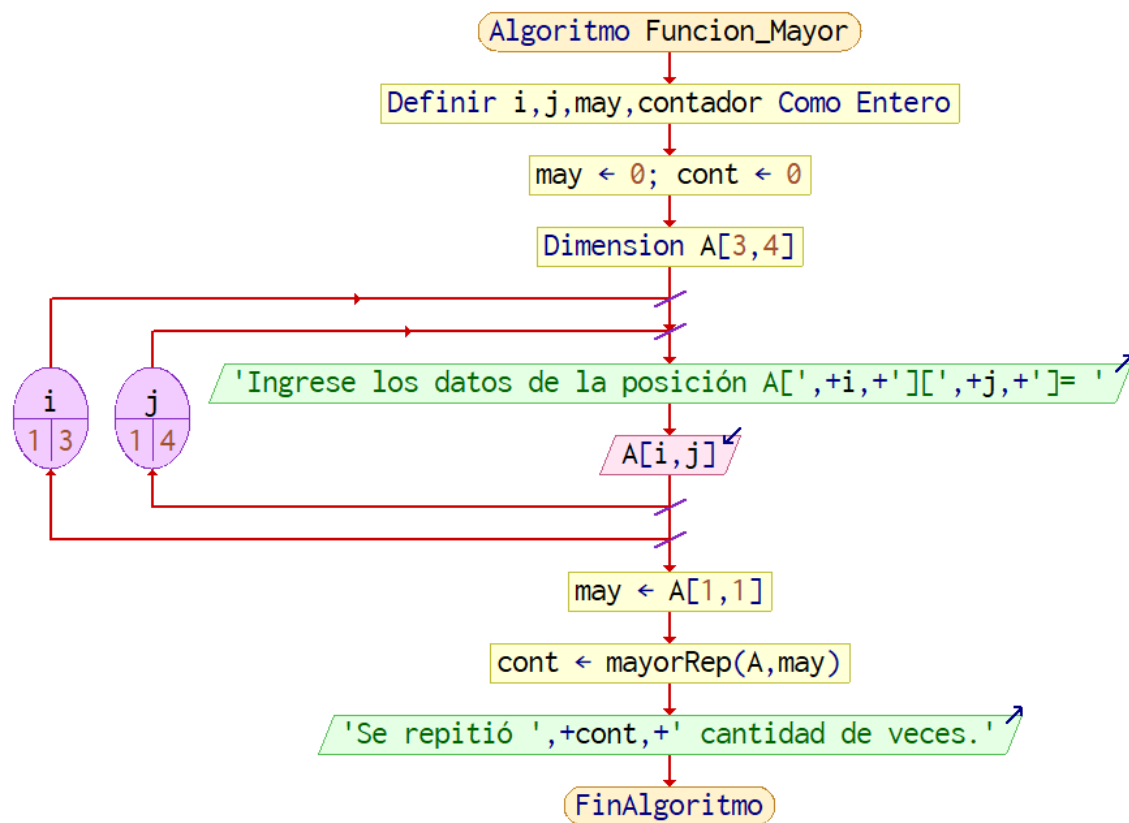
Fin\_para

Devolver cont

Fin\_mayorRep

**Prueba de escritorio**

Diagrama de Flujo



Prueba de escritorio

Prueba de Escritorio						
Proceso/SubProceso	Línea(inst)	i	j	B(i,j)	may	cont
2MAYORREP	21(1)	1	1	1	1	0
2MAYORREP	23(1)	1	1	1	1	0
2MAYORREP	20(1)	1	1	1	1	0
2MAYORREP	21(1)	1	2	2	1	0
2MAYORREP	22(1)	1	2	2	1	0
2MAYORREP	23(1)	1	2	2	2	0
2MAYORREP	20(1)	1	2	2	2	0
2MAYORREP	21(1)	1	3	3	2	0
2MAYORREP	22(1)	1	3	3	2	0
2MAYORREP	23(1)	1	3	3	3	0
2MAYORREP	20(1)	1	3	3	3	0
2MAYORREP	21(1)	1	4	4	3	0
2MAYORREP	22(1)	1	4	4	3	0
2MAYORREP	23(1)	1	4	4	4	0
2MAYORREP	20(1)	1	4	4	4	0
2MAYORREP	24(1)	1	5	<<Subíndice (5) fuera de rango (1..4)>>	4	0
2MAYORREP	19(1)	1	5	<<Subíndice (5) fuera de rango (1..4)>>	4	0
2MAYORREP	20(1)	2	5	<<Subíndice (5) fuera de rango (1..4)>>	4	0
2MAYORREP	21(1)	2	1	5	4	0
2MAYORREP	22(1)	2	1	5	4	0
2MAYORREP	23(1)	2	1	5	5	0
2MAYORREP	20(1)	2	1	5	5	0
2MAYORREP	21(1)	2	2	6	5	0
2MAYORREP	22(1)	2	2	6	5	0
2MAYORREP	23(1)	2	2	6	6	0
2MAYORREP	20(1)	2	2	6	6	0
2MAYORREP	21(1)	2	3	7	6	0
2MAYORREP	22(1)	2	3	7	6	0
2MAYORREP	23(1)	2	3	7	7	0
2MAYORREP	20(1)	2	3	7	7	0
2MAYORREP	21(1)	2	4	8	7	0
2MAYORREP	22(1)	2	4	8	7	0
2MAYORREP	23(1)	2	4	8	8	0
2MAYORREP	20(1)	2	4	8	8	0
2MAYORREP	24(1)	2	5	<<Subíndice (5) fuera de rango (1..4)>>	8	0

Prueba de Escritorio						
Proceso/SubProceso	Línea(inst)	i	j	B(i,j)	may	cont
2MAYORREP	22(1)	3	1	9	8	0
2MAYORREP	23(1)	3	1	9	9	0
2MAYORREP	20(1)	3	1	9	9	0
2MAYORREP	21(1)	3	2	7	9	0
2MAYORREP	23(1)	3	2	7	9	0
2MAYORREP	20(1)	3	2	7	9	0
2MAYORREP	21(1)	3	3	8	9	0
2MAYORREP	23(1)	3	3	8	9	0
2MAYORREP	20(1)	3	3	8	9	0
2MAYORREP	21(1)	3	4	9	9	0
2MAYORREP	23(1)	3	4	9	9	0
2MAYORREP	20(1)	3	4	9	9	0
2MAYORREP	24(1)	3	5	<<Subíndice (5) fuera de rango (1..4)>>	9	0
2MAYORREP	19(1)	3	5	<<Subíndice (5) fuera de rango (1..4)>>	9	0
2MAYORREP	25(1)	4	5	<<Subíndice (4) fuera de rango (1..3)>>	9	0
2MAYORREP	26(1)	4	5	<<Subíndice (4) fuera de rango (1..3)>>	9	0
2MAYORREP	27(1)	1	5	<<Subíndice (5) fuera de rango (1..4)>>	9	0
2MAYORREP	28(1)	1	1	1	9	0
2MAYORREP	30(1)	1	1	1	9	0
2MAYORREP	27(1)	1	1	1	9	0
2MAYORREP	28(1)	1	2	2	9	0
2MAYORREP	30(1)	1	2	2	9	0
2MAYORREP	27(1)	1	2	2	9	0
2MAYORREP	28(1)	1	3	3	9	0
2MAYORREP	30(1)	1	3	3	9	0
2MAYORREP	27(1)	1	3	3	9	0
2MAYORREP	28(1)	1	4	4	9	0
2MAYORREP	30(1)	1	4	4	9	0
2MAYORREP	27(1)	1	4	4	9	0
2MAYORREP	31(1)	1	5	<<Subíndice (5) fuera de rango (1..4)>>	9	0
2MAYORREP	26(1)	1	5	<<Subíndice (5) fuera de rango (1..4)>>	9	0
2MAYORREP	27(1)	2	5	<<Subíndice (5) fuera de rango (1..4)>>	9	0
2MAYORREP	28(1)	2	1	5	9	0
2MAYORREP	30(1)	2	1	5	9	0
2MAYORREP	27(1)	2	1	5	9	0

2MAYORREP	28(1)	2	2	6	9	0
2MAYORREP	30(1)	2	2	6	9	0
2MAYORREP	27(1)	2	2	6	9	0
2MAYORREP	28(1)	2	3	7	9	0
2MAYORREP	30(1)	2	3	7	9	0
2MAYORREP	27(1)	2	3	7	9	0
2MAYORREP	28(1)	2	4	8	9	0
2MAYORREP	30(1)	2	4	8	9	0
2MAYORREP	27(1)	2	4	8	9	0
2MAYORREP	31(1)	2	5	<<Subíndice (5) fuera de rango (1..4)>>	9	0
2MAYORREP	26(1)	2	5	<<Subíndice (5) fuera de rango (1..4)>>	9	0
2MAYORREP	27(1)	3	5	<<Subíndice (5) fuera de rango (1..4)>>	9	0
2MAYORREP	28(1)	3	1	9	9	0
2MAYORREP	29(1)	3	1	9	9	0
2MAYORREP	30(1)	3	1	9	9	1
2MAYORREP	27(1)	3	1	9	9	1
2MAYORREP	28(1)	3	2	7	9	1
2MAYORREP	30(1)	3	2	7	9	1
2MAYORREP	27(1)	3	2	7	9	1
2MAYORREP	28(1)	3	3	8	9	1
2MAYORREP	30(1)	3	3	8	9	1
2MAYORREP	27(1)	3	3	8	9	1
2MAYORREP	28(1)	3	4	9	9	1
2MAYORREP	29(1)	3	4	9	9	1
2MAYORREP	30(1)	3	4	9	9	2
2MAYORREP	27(1)	3	4	9	9	2
2MAYORREP	31(1)	3	5	<<Subíndice (5) fuera de rango (1..4)>>	9	2
2MAYORREP	26(1)	3	5	<<Subíndice (5) fuera de rango (1..4)>>	9	2
2MAYORREP	32(1)	4	5	<<Subíndice (4) fuera de rango (1..3)>>	9	2
2MAYORREP	33(1)	4	5	<<Subíndice (4) fuera de rango (1..3)>>	9	2
INICION_MAYOR	13(1)	4	5	itificador B no corresponde a un arreglo o subp	1	2
INICION_MAYOR	14(1)	4	5	itificador B no corresponde a un arreglo o subp	1	2

## Programa en java

```
bryamciclo - Apache NetBeans IDE 12.5
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config> 271,5/604,0MB
Start Page x | Bryamciclo.java x | NewClass.java x | fibonaccibryam.java x
Source History
6
7 public class NewClass {
8     public static void main(String[] args) {
9         Scanner leer = new Scanner(System.in);
10        int j,i,mayor,cont;
11        mayor=0;cont=0;
12
13
14        int [][] A = new int [3][4];
15        for (i = 0; i < 3; i++) {
16            for ( j = 0; j < 4; j++) {
17                System.out.println("Ingrese los datos de la posición A["+i+"]["+j+"]= ");
18                A[i][j] = leer.nextInt();
19            }
20        }
21        mayor=A[0][0];
22        cont = mayorRep(A,mayor);
23
24        System.out.println("Se repitió " + cont + " cantidad de veces.");
25    }
26    public static int mayorRep(int B[][], int mayor1){
27        int cont=0;
28        // Encontrar el mayor
29        for (int i = 0; i < 3; i++) {
30            for (int j = 0; j < 4; j++) {
31                if(mayor1<B[i][j]){
```

```
bryamciclo - Apache NetBeans IDE 12.5
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config> 366,4/604,0MB
Start Page x | Bryamciclo.java x | NewClass.java x | fibonaccibryam.java x
Source History
19    }
20    }
21    mayor=A[0][0];
22    cont = mayorRep(A,mayor);
23
24    System.out.println("Se repitió " + cont + " cantidad de veces.");
25    }
26    public static int mayorRep(int B[][], int mayor1){
27        int cont=0;
28        // Encontrar el mayor
29        for (int i = 0; i < 3; i++) {
30            for (int j = 0; j < 4; j++) {
31                if(mayor1<B[i][j]){
32                    mayor1 = B[i][j];
33                }
34            }
35        }
36
37        // contar las veces que se repite el numero mayor
38        for (int i = 0; i < 3; i++) {
39            for ( int j = 0; j < 4; j++) {
40                if(mayor1 == B[i][j]){
41                    cont = cont + 1;
42                }
43            }
44        }
45        return cont;
46    }
47 }
```

#### Ejercicio4

Ejercicio 4.4 - 3. Libro de Mancilla. Pág. 269

##### Mini especificación

Inicio

j,i,n,m i[0-n]

<< "Ingrese el número de filas"

n←0;m←1;

A [][] , i [0-n]

para (i = 0; i < n; i++) hacer

    para ( j = 0; j < m; j++) hacer

        << " Ingrese datos de la matriz A[" + i + "][" + j + "]= "

        >>A[i][j]

    Fin\_para

Fin\_para

<< " matriz A: "

para (i = 0; i < n; i++) hacer

    para ( j = 0; j < m; j++) hacer

        << A[i][j] + " "

    Fin\_para

<< " "

Fin\_para

<< " matriz A invertida "

para (i = n; i >= 0; i--) hacer

    para ( j = 0; j < 6; j++) hacer

        << A[i][j] + " "

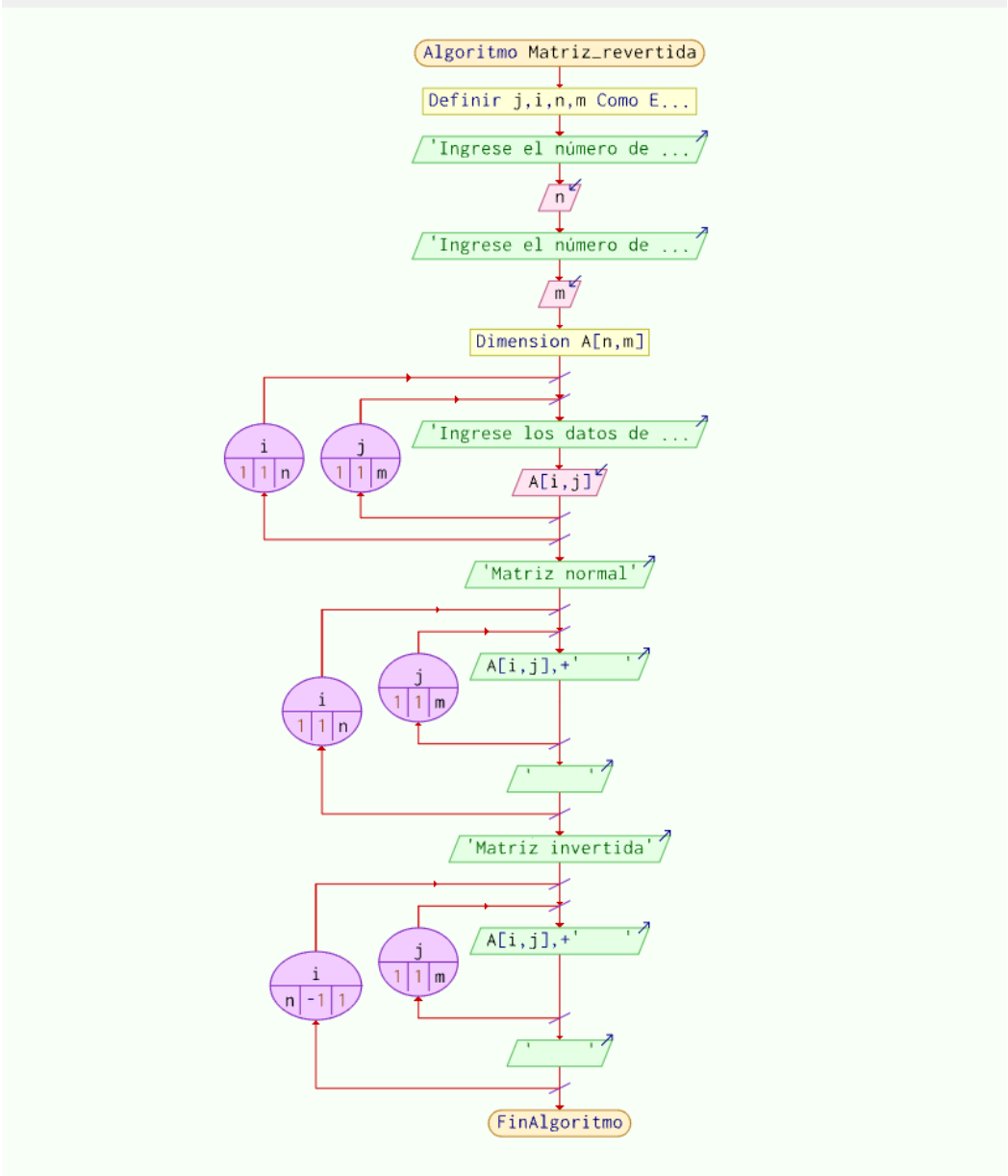
    Fin\_para

<< " "

Fin\_para

Fin

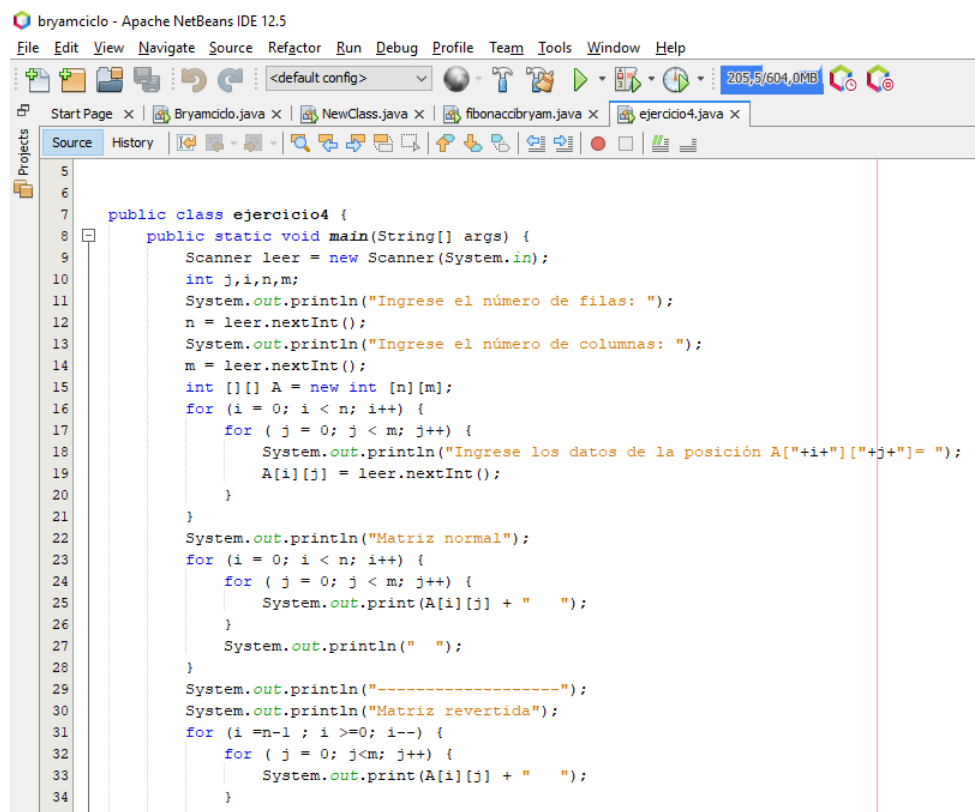
Diagrama de flujo



Prueba de escritorio

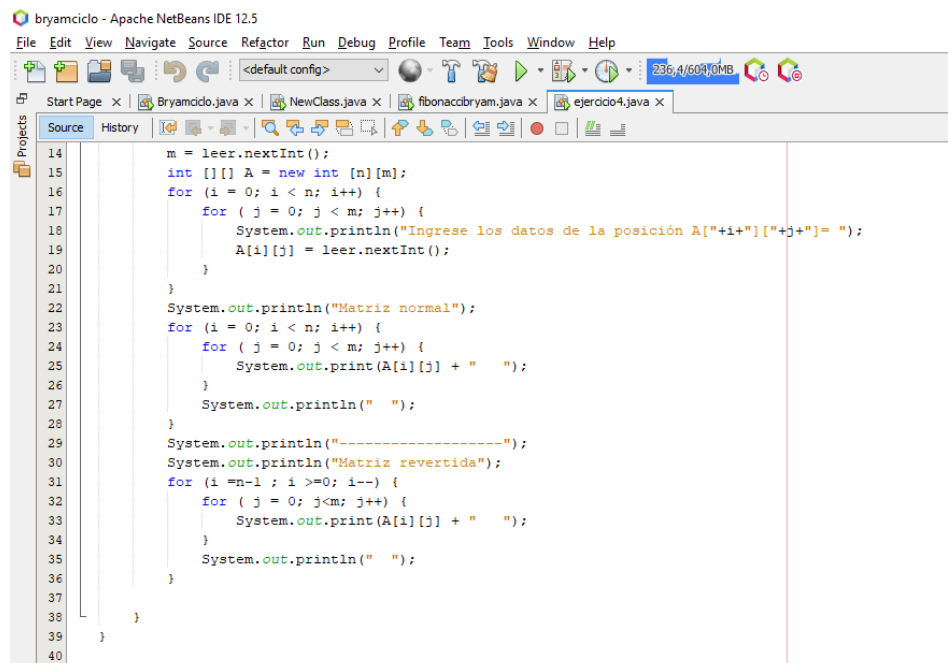
Prueba de Escritorio				
Proceso/SubProceso	Linea(inst)	i	j	A[i,j]
1.MATRIZ_REVERTIDA	1(1)	<<Variable no inicializada (I)>>	<<Variable no inicializada (J)>>	<<El identificador A no corresponde a un arreglo o subproceso>>
1.MATRIZ_REVERTIDA	2(1)	<<Variable no inicializada (I)>>	<<Variable no inicializada (J)>>	<<El identificador A no corresponde a un arreglo o subproceso>>
1.MATRIZ_REVERTIDA	3(1)	<<Variable no inicializada (I)>>	<<Variable no inicializada (J)>>	<<El identificador A no corresponde a un arreglo o subproceso>>
1.MATRIZ_REVERTIDA	4(1)	<<Variable no inicializada (I)>>	<<Variable no inicializada (J)>>	<<El identificador A no corresponde a un arreglo o subproceso>>
1.MATRIZ_REVERTIDA	5(1)	<<Variable no inicializada (I)>>	<<Variable no inicializada (J)>>	<<El identificador A no corresponde a un arreglo o subproceso>>
1.MATRIZ_REVERTIDA	6(1)	<<Variable no inicializada (I)>>	<<Variable no inicializada (J)>>	<<El identificador A no corresponde a un arreglo o subproceso>>
1.MATRIZ_REVERTIDA	7(1)	<<Variable no inicializada (I)>>	<<Variable no inicializada (J)>>	<<El identificador A no corresponde a un arreglo o subproceso>>
1.MATRIZ_REVERTIDA	8(1)	<<Variable no inicializada (I)>>	<<Variable no inicializada (J)>>	<<Variable no inicializada (I)>>
1.MATRIZ_REVERTIDA	9(1)	1	<<Variable no inicializada (J)>>	<<Variable no inicializada (J)>>
1.MATRIZ_REVERTIDA	10(1)	1	1	
1.MATRIZ_REVERTIDA	11(1)	1	1	
1.MATRIZ_REVERTIDA	9(1)	1	1	1
1.MATRIZ_REVERTIDA	10(1)	1	2	
1.MATRIZ_REVERTIDA	11(1)	1	2	
1.MATRIZ_REVERTIDA	9(1)	1	2	2
1.MATRIZ_REVERTIDA	12(1)	1	3	<<Subindice (3) fuera de rango (1..2)>>
1.MATRIZ_REVERTIDA	8(1)	1	3	<<Subindice (3) fuera de rango (1..2)>>
1.MATRIZ_REVERTIDA	9(1)	2	3	<<Subindice (3) fuera de rango (1..2)>>
1.MATRIZ_REVERTIDA	10(1)	2	1	
1.MATRIZ_REVERTIDA	11(1)	2	1	
1.MATRIZ_REVERTIDA	9(1)	2	1	3
1.MATRIZ_REVERTIDA	10(1)	2	2	
1.MATRIZ_REVERTIDA	11(1)	2	2	
1.MATRIZ_REVERTIDA	9(1)	2	2	4
1.MATRIZ_REVERTIDA	12(1)	2	3	<<Subindice (3) fuera de rango (1..2)>>
1.MATRIZ_REVERTIDA	8(1)	2	3	<<Subindice (3) fuera de rango (1..2)>>
1.MATRIZ_REVERTIDA	13(1)	3	3	<<Subindice (3) fuera de rango (1..2)>>
1.MATRIZ_REVERTIDA	29(1)	3	3	<<Subindice (3) fuera de rango (1..2)>>

## Programa en java



The screenshot shows the Apache NetBeans IDE 12.5 interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, and Help. The toolbar contains various icons for file operations, running, and debugging. The project explorer on the left shows a project named 'bryamciclo' with files 'Bryamciclo.java', 'NewClass.java', 'fibonaccibryam.java', and 'ejercicio4.java'. The main editor window displays the source code of 'ejercicio4.java'.

```
5
6
7 public class ejercicio4 {
8     public static void main(String[] args) {
9         Scanner leer = new Scanner(System.in);
10        int j,i,n,m;
11        System.out.println("Ingrese el número de filas: ");
12        n = leer.nextInt();
13        System.out.println("Ingrese el número de columnas: ");
14        m = leer.nextInt();
15        int [][] A = new int [n][m];
16        for (i = 0; i < n; i++) {
17            for ( j = 0; j < m; j++) {
18                System.out.println("Ingrese los datos de la posición A["+i+"]["+j+"] = ");
19                A[i][j] = leer.nextInt();
20            }
21        }
22        System.out.println("Matriz normal");
23        for (i = 0; i < n; i++) {
24            for ( j = 0; j < m; j++) {
25                System.out.print(A[i][j] + " ");
26            }
27            System.out.println(" ");
28        }
29        System.out.println("-----");
30        System.out.println("Matriz revertida");
31        for (i =n-1 ; i >=0; i--) {
32            for ( j = 0; j<m; j++) {
33                System.out.print(A[i][j] + " ");
34            }
35        }
36    }
37 }
```



This block shows the continuation of the source code for 'ejercicio4.java' in the NetBeans IDE. The code continues from line 35, showing the completion of the matrix reversal loop and the closing of the main method and class.

```
35        System.out.println(" ");
36    }
37 }
38
39
40 }
```



Video en YouTube: <https://www.youtube.com/watch?v=JBDHPxncrzI>

Git: Bryamjim