UNIVERSIDAD DEL NORESTE

Universidad del Noreste

Área de ingeniería y Ciencias Químicas

Ingeniería en Sistemas Computacionales y Electrónicos

Introducción a la Programación

Ing. Myriam Janeth Rodríguez Martínez

1º L

Cruz Muñiz Alex Eduardo

"Actividad 11.1. Ejercicios ciclos semana 10"

21 de Noviembre, 2024.

Sube aquí el análisis, diagrama y programa de cada problema.

1. Lee 5 números e imprima cuál es el mayor:

Análisis del problema:

• Datos de entrada:

Dos variables una donde se almacenen los números ingresados, otra donde se vayan almacenando los números mayores.

Variables:

n: (dato tipo numérico con decimales).

auxMayor: (dato tipo numérico con decimales).

• Datos de salida:

auxMayor con el número mayor

auxMayor: (dato tipo numérico con decimales).

• Proceso:

Establecer un ciclo for donde i sea igual a 1 y el limite sea 5 para que este se repita 5 veces y que en cada repetición pida un numero y con un if verificar si este es mayor que el otro, en caso de que lo sea se actualiza el valor de auxMayor para que valga lo mismo que este.

• Diseño de la solución:

Algoritmo (pasos para resolver un problema, tiene 3 características: finitud, precisión,

determinista).

1.-Inicio

2.-Lectura de datos: n, auxMayor.

3.-Inicializar un ciclo for que se repita 5 veces.

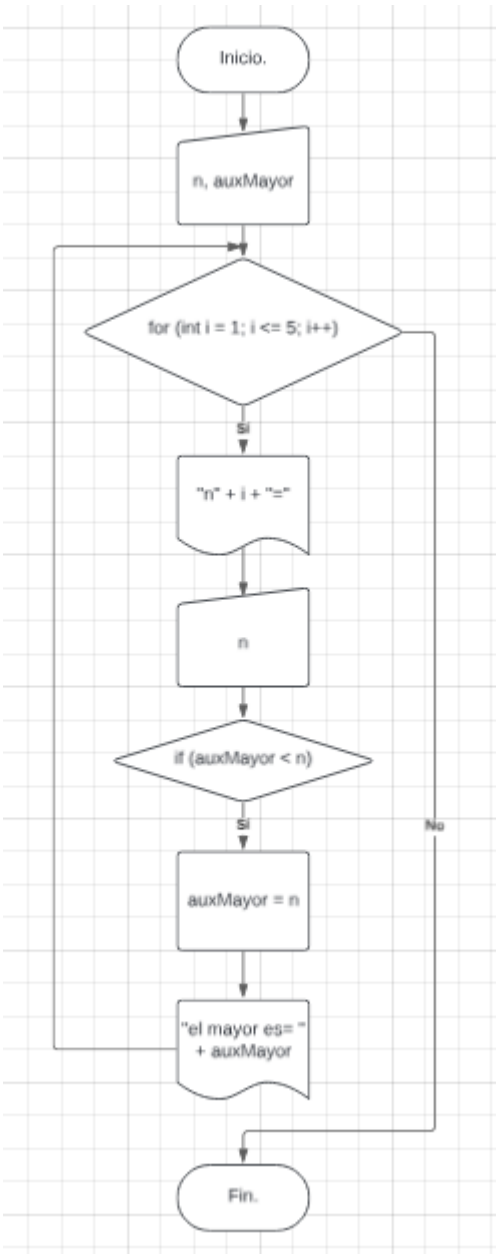
4.-en cada repetición del ciclo pedir un numero y que se almacene en la variable n.

5.-en cada repetición verificar si el nuevo valor almacenado de N es mayor que auxMayor, en caso de que sea TRUE almacenar el nuevo valor de n en auxMayor.

6.-imprimir el valor de auxMayor.

7.-Fin.

Diagrama de flujo:



Programa:

```java
import java.util.Scanner;
public class mayores {
    public static void main(String[] args) {
        int n, auxMayor=0;
        Scanner s = new Scanner(System.in);
        for (int i = 1; i <= 5; i++){
            System.out.print("n"+i+"=");
            n=s.nextInt();
            if (auxMayor < n){
                auxMayor =n;
            }
            System.out.println("El mayor es=" + auxMayor);
        }
    }
}
```

Resultado:

```
n1=5
El mayor es=5
n2=4
El mayor es=5
n3=3
El mayor es=5
n4=6
El mayor es=6
n5=7
El mayor es=7
```

2. Lee 5 números e imprima cuál es el menor:

Análisis del problema:

• Datos de entrada:

Dos variables una donde se almacenen los números ingresados, otra donde se vayan almacenando los números menores.

Variables:

n: (dato tipo numérico con decimales).

auxMenor: (dato tipo numérico con decimales).

• Datos de salida:

auxMenor con el número mayor

auxMenor: (dato tipo numérico con decimales).

• Proceso:
Pedir al usuario que ingrese un numero y se almacene en n y que el valor de auxMenor sea el mismo que n después establecer un ciclo for donde i sea igual a 2 y el límite sea 5 para que este se repita 4 veces y que en cada repetición pida un número y con un if verificar si este es menor que el otro, en caso de que lo sea se actualiza el valor de auxMenor para que valga lo mismo que este.

• Diseño de la solución:

Algoritmo (pasos para resolver un problema, tiene 3 características: finitud, precisión,

determinista).

1.-Inicio

2.-Lectura de datos: n, auxMayor.

3.-Pedirle al usuario que ingrese un numero entero y almacenarlo en n y almacenar el valor de n en auxMayor.

4.-Inicializar un ciclo for que se repita 4 veces.

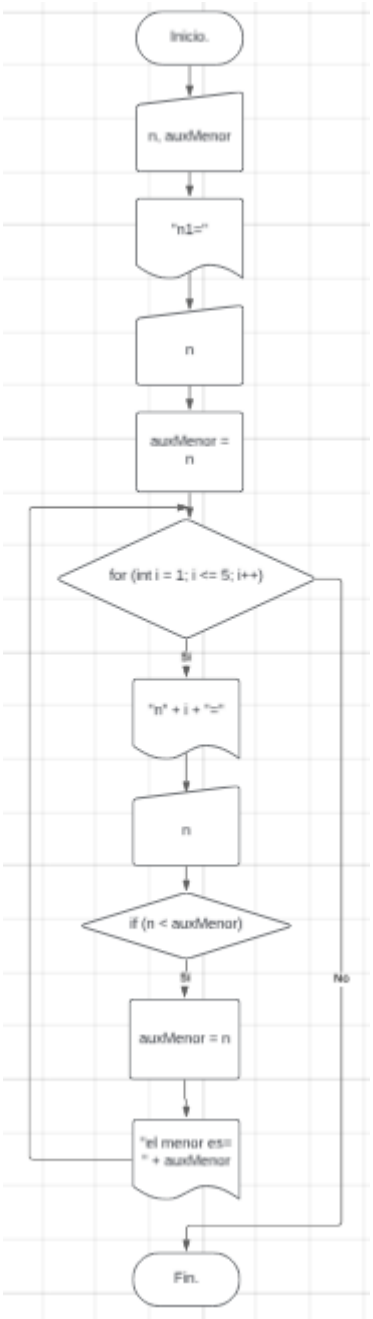
5.-en cada repetición del ciclo pedir un numero y que se almacene en la variable n.

6.-en cada repetición verificar si el nuevo valor almacenado de N es menor que auxMenor, en caso de que sea TRUE almacenar el nuevo valor de n en auxMenor.

7.- imprimir el valor de auxMenor

8.-Fin.

Diagrama de flujo:



Programa:

```java
import java.util.Scanner;
public class Menor {
    public static void main(String[] args) {
        int n, auxMenor;
        Scanner s = new Scanner(System.in);
        System.out.print("n1=");
        n = s.nextInt();
        auxMenor = n;
        for (int i = 2; i <= 5; i++){
            System.out.print("n"+i+"=");
            n=s.nextInt();
            if (n < auxMenor){
                auxMenor =n;
            }
            System.out.println("El menor es=" + auxMenor);
        }
    }
}
```

Resultado:

```
n1=4
n2=3
El menor es=3
n3=6
El menor es=3
n4=7
El menor es=3
n5=8
El menor es=3
```

3.Realizar un programa que imprima 25 términos de la serie 11 - 22 - 33 - 44, etc. (No se ingresan valores por teclado):

Análisis del problema:

• Datos de entrada:

La variable i que se utilizara para hacer el ciclo for

i: (dato tipo numérico con decimales).

• Datos de salida:

La i del ciclo for pero esta aumentara su valor y se imprimirá dos veces para formar números iguales.

• Proceso:
Establecer un ciclo for en donde el valor inicial de i sea 1 y que este vaya aumentando uno por uno hasta llegar a 25 y que cada que este se repita imprima dos veces seguidas el valor de i.

• Diseño de la solución:

Algoritmo (pasos para resolver un problema, tiene 3 características: finitud, precisión,

determinista).

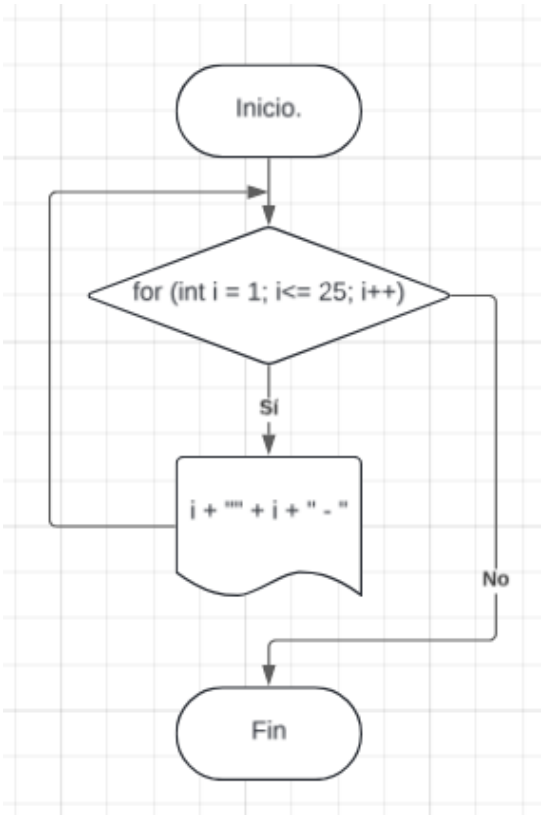
1.-Inicio

2.- Iniciar un ciclo for donde el valor inicial de i sea 1 y que este vaya aumentando de uno en uno cada que este se repita y el límite sea 25

3.-imprimir el valor de i dos veces seguidas cada que el ciclo se repita para que este forme números iguales.

8.-Fin.

Diagrama de flujo:



Programa:

```java
public class Iguales {
    public static void main(String[] args) {
        for (int i = 1; i<= 25; i++){
            System.out.print(i+ "" + i + " - ");
        }
    }
}
```

Resultados:

```
11 - 22 - 33 - 44 - 55 - 66 - 77 - 88 - 99 - 1010 - 1111 - 1212 - 1313 - 1414 - 1515 - 1616 - 1717 - 1818 - 1919 - 2020 - 2121 - 2222 - 2323 - 2424 - 2525 -
Process finished with exit code 0
```

4. Mostrar los múltiplos de 8 hasta el valor 500. Debe aparecer en pantalla 8 - 16 - 24, etc:

Análisis del problema:

• Datos de entrada:

La variable i que se utilizara para hacer el ciclo for

i: (dato tipo numérico con decimales).

• Datos de salida:

La i del ciclo for pero esta aumentara su valor en múltiplos de 8

• Proceso:
Establecer un ciclo for en donde el valor inicial de i sea 8 y que este vaya aumentando de 8 en 8 hasta llegar a 500 y que cada que este se repita imprima el valor de i.

• Diseño de la solución:

Algoritmo (pasos para resolver un problema, tiene 3 características: finitud, precisión,

determinista).

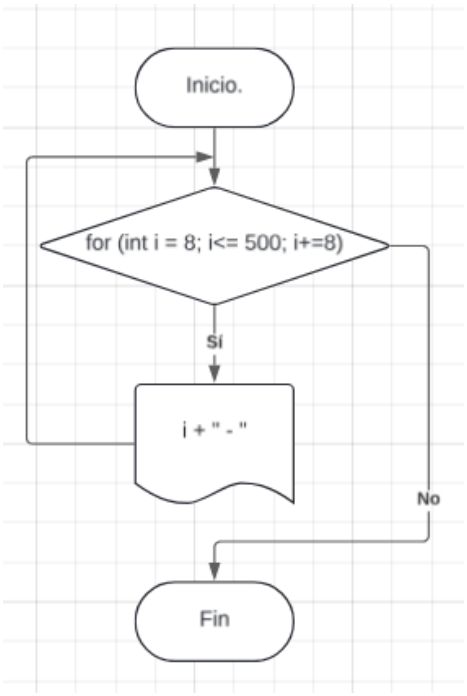
1.-Inicio

2.- Iniciar un ciclo for donde el valor inicial de i sea 8 y que este vaya aumentando de 8 en 8 cada que este se repita y el límite sea 500

3.-imprimir el valor de i.

4.-Fin.

Diagrama de flujo:



Programa:

```java
public class multiplosdeocho {
    public static void main(String[] args) {
        for (int i = 8; i<= 500; i+=8){
            System.out.print(i+" - ");
        }
    }
}
```

Resultados:

```
8 - 16 - 24 - 32 - 40 - 48 - 56 - 64 - 72 - 80 - 88 - 96 - 104 - 112 - 120 - 128 - 136 - 144 - 152 - 160 - 168 - 176 - 184 - 192 - 200
Process finished with exit code 0
```

```
- 200 - 208 - 216 - 224 - 232 - 240 - 248 - 256 - 264 - 272 - 280 - 288 - 296 - 304 - 312 - 320 - 328 - 336 - 344 - 352 - 360 - 368 -
```

```
360 - 368 - 376 - 384 - 392 - 400 - 408 - 416 - 424 - 432 - 440 - 448 - 456 - 464 - 472 - 480 - 488 - 496 -
```

5. Lee un número e imprime si es primo o no:

Análisis del problema:

• Datos de entrada:

La variable n en donde se ingresará el valor de los números ingresados y la variable i que se utilizará para hacer el ciclo for

n:(dato tipo numérico con decimales).

i: (dato tipo numérico con decimales).

• Datos de salida:

Un texto donde diga si es primo o no es primo

• Proceso:

Pedir al usuario el valor de n (el número que verificaremos si es primo) después establecer un ciclo for en donde el valor inicial de i sea 1 y que este vaya aumentando de 1 en 1 hasta llegar a el valor de n y que cada que este se repita verifique si la variable i se pueda dividir entre n sin dejar residuo asi verificara que cualquier número antes de la variable n sea divisor de esta para saber si es número primo y cada que esta se cumpla se le sume 1 a la variable acumulador.

Después verificar con un if si acumulador es igual a 1 entonces es primo porque solo se puede dividir entre el mismo y si no este no es primo.

• Diseño de la solución:

Algoritmo (pasos para resolver un problema, tiene 3 características: finitud, precisión,

determinista).

1.-Inicio

2.- Pedir el valor de la variable n (el número que verificaremos si es primo)

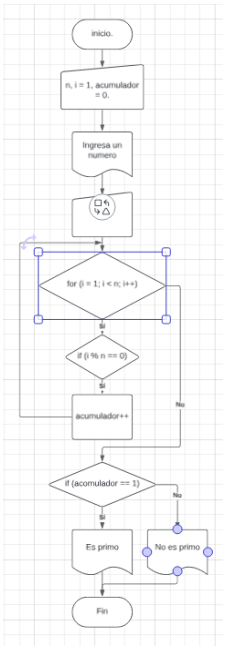
3.-establecer un ciclo for donde el valor inicial de i sea 1 y que vaya aumentando de 1 en 1 hasta que este llegue al valor de n.

4.- dentro del for verificar si este se puede divir entre 1 sin dejar residuo, en caso de que si se cumpla aumentar el valor de la variable acumulador más uno.

5.- fuera del for cuando este acabe verificar con un if si el valor del acumulador es igual a 1, si si se cumple entonces es primo, y si no con un else imprimir que este no es primo

6.-Fin.

Diagrama de flujo:



Programa:

```java
import java.util.Scanner;
public class primos {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        int n, i = 1;
        int acomulador = 0;
        System.out.print("Ingresa un numero: ");
        n = s.nextInt();
        for (i = 1; i <= n; i++) {
            if (i % n == 0) {
                acomulador++;
            }
        }
        if (acomulador == 1){
            System.out.println("es primo");
        } else if (acomulador >= 2) {
            System.out.println("No es primo");
        }
    }
}
```

Resultado:

```
Ingresa un numero: 6
No es primo
```

```
Ingresa un numero: 7
es primo
```

6. Ve el video de ciclos anidados, sube los programas realizados

• Datos de entrada:

La variable i y j que se utilizaran para hacer los ciclos for

i: (dato tipo numérico con decimales).

j: (dato tipo numérico con decimales).

• Datos de salida:

La i y la j de los ciclos for pero esta aumentara su valor haciendo tablas de multiplicar y el resultado de estas será la variable mult.

• Proceso:
Establecer un ciclo for en donde el valor inicial de i sea 1 y que este vaya aumentando uno por uno hasta llegar a 10 y agregar un ciclo for anidado en este con los mismos valores pero cambiando la variable por j cada que se repita el ciclo anidado se le añadira el valor de i * j a la variable mult y se imprimirá de la siguiente forma (i + "*" + j + "=" + mult) .

• Diseño de la solución:

Algoritmo (pasos para resolver un problema, tiene 3 características: finitud, precisión,

determinista).

1.-Inicio

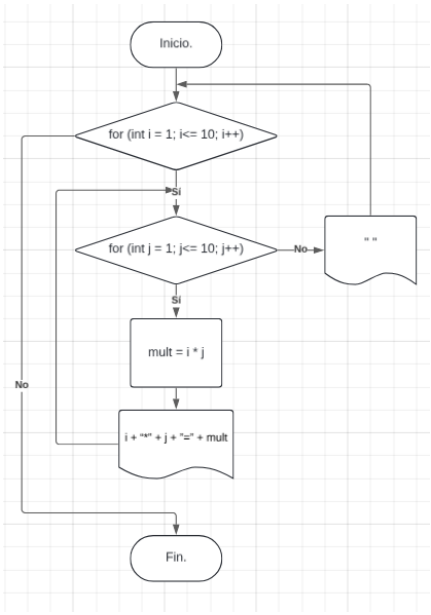
2.- Iniciar un ciclo for donde el valor inicial de i sea 1 y que este vaya aumentando de uno en uno cada que este se repita y el límite sea 10

3.- Iniciar un ciclo for dentro del otro donde el valor inicial de j sea 1 y que este vaya aumentando de uno en uno cada que este se repita y el límite sea 10

4.-cada que se cumpla el ciclo anidado este multiplicara la i * j y se le agregara al valor de mult y se imprimirá de la siguiente forma: (i + "*" + j + "=" + mult) .

5.-Fin.

Diagrama de Flujo:




```java
public class video {
    public static void main(String[] args) {
        int mult;

        for(int i = 1; i <=10; i++){
            for(int j = 1; j<=10; j++){
                mult = i * j;
                System.out.println(i+ " * " +j+" = " + mult);
            }
            System.out.println();
        }
    }
}
```

```
C:\Program Files\Java\jdk-23\bi
1 * 1 = 1
1 * 2 = 2
1 * 3 = 3
1 * 4 = 4
1 * 5 = 5
1 * 6 = 6
1 * 7 = 7
1 * 8 = 8
1 * 9 = 9
1 * 10 = 10

2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
2 * 4 = 8
2 * 5 = 10
2 * 6 = 12
2 * 7 = 14
2 * 8 = 16
2 * 9 = 18
2 * 10 = 20

3 * 1 = 3
3 * 2 = 6
3 * 3 = 9
3 * 4 = 12
3 * 5 = 15
3 * 6 = 18
3 * 7 = 21
3 * 8 = 24
```

```
4 * 1 = 4
4 * 2 = 8
4 * 3 = 12
4 * 4 = 16
4 * 5 = 20
4 * 6 = 24
4 * 7 = 28
4 * 8 = 32
4 * 9 = 36
4 * 10 = 40

5 * 1 = 5
5 * 2 = 10
5 * 3 = 15
5 * 4 = 20
5 * 5 = 25
5 * 6 = 30
5 * 7 = 35
5 * 8 = 40
5 * 9 = 45
5 * 10 = 50

6 * 1 = 6
6 * 2 = 12
6 * 3 = 18
6 * 4 = 24
6 * 5 = 30
6 * 6 = 36
6 * 7 = 42
6 * 8 = 48
```

```
7 * 6 = 42
7 * 7 = 49
7 * 8 = 56
7 * 9 = 63
7 * 10 = 70

8 * 1 = 8
8 * 2 = 16
8 * 3 = 24
8 * 4 = 32
8 * 5 = 40
8 * 6 = 48
8 * 7 = 56
8 * 8 = 64
8 * 9 = 72
8 * 10 = 80

9 * 1 = 9
9 * 2 = 18
9 * 3 = 27
9 * 4 = 36
9 * 5 = 45
9 * 6 = 54
9 * 7 = 63
9 * 8 = 72
9 * 9 = 81
9 * 10 = 90

10 * 1 = 10
10 * 2 = 20
```

7. Escribe un programa que dado un número entero, imprima los números primos menores a éste.

Análisis del problema:

• Datos de entrada:

La variable n y la variable prima

n: (dato tipo numérico con decimales).

primo: (dato tipo verdadero o falso).

• Datos de salida:

Los números primos que sean menores al valor ingresado con la variable i

• Proceso:
Establecer un ciclo for en donde el valor inicial de i sea n y que este vaya disminuyendo de 1 en 1 hasta que i sea menor o igual a 2 y que cada que este se repita el valor de primo sea true y agregar un ciclo for anidado donde el valor de j sea 2 y este vaya aumentando de 1 en 1 hasta que sea mayor que i y cada que este se repita verificar con un if si i módulo de j es igual a 0 imprimir que no es primo y convertir la variable primo a false y cerrar el ciclo y agregar un if fuera de este ciclo pero Adentro del otro donde verifique si primo es true entonces imprimir que este es primo.

• Diseño de la solución:

Algoritmo (pasos para resolver un problema, tiene 3 características: finitud, precisión,

determinista).

1.-Inicio

2.- Iniciar un ciclo for donde el valor inicial de i sea n y que este vaya disminuyendo de 1 en 1 cada que este se repita y el límite sea menor o igual a 2.

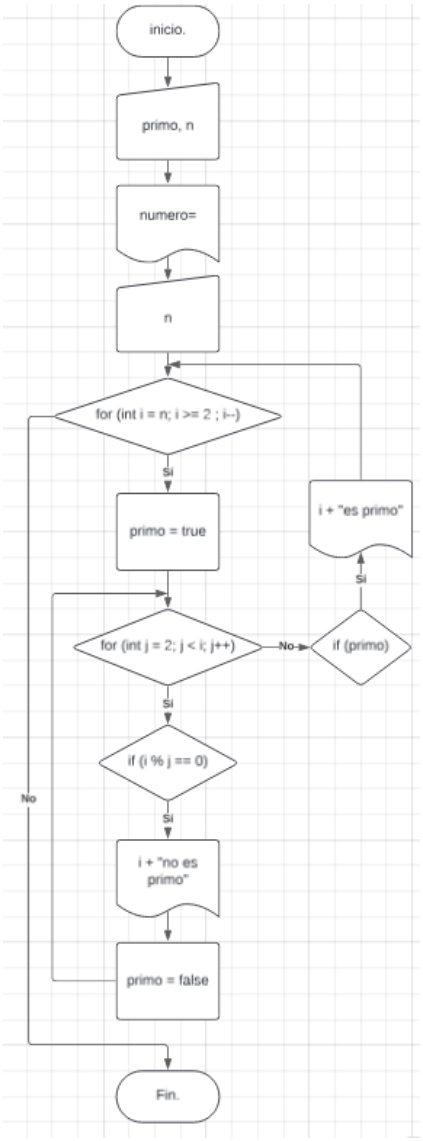
3.-agregar un ciclo anidado donde el valor inicial de j sea 2 y que el límite de este sea ser mayor que i y este vaya aumentando de 1 en 1.

4.- verificar con un if si i modulo de j es igual a 0 entonces imprimir la i e imprimir que este no es primo, y volver la variable primo a false y romper el ciclo

5.- verificar con un if fuera de el ciclo anidado si primo es true entonces imprimir la variable i e imprimir que es primo.

6.-Fin.

Diagrama de flujo:



Programa:

```java
import java.util.Scanner;
public class primosss {
    public static void main(String[] args) {
        int n;
        boolean primo = true;
        Scanner sc = new Scanner(System.in);
        System.out.print("Número= ");
        n = sc.nextInt();
        for (int i = n; i >= 2 ; i--) {
            primo = true;
            for (int j = 2; j < i; j++) {
                if (i % j == 0) {
                    System.out.println(i + " No es primo");
                    primo = false;
                    break;
                }
            }
            if (primo){
                System.out.println(i + " Es primo");
            }
        }
    }
}
```

Resultados:

```
Número= 7
7 Es primo
6 No es primo
5 Es primo
4 No es primo
3 Es primo
2 Es primo
```

```
Número= 11
11 Es primo
10 No es primo
9 No es primo
8 No es primo
7 Es primo
6 No es primo
5 Es primo
4 No es primo
3 Es primo
2 Es primo
```

8. Imprime la siguiente secuencia, por ejemplo, si lee 5, imprime:
x
x x
x x x
x x x x
x x x x x

Análisis del problema:

• Datos de entrada:

La variable núm, x y acumulador donde número serán las repeticiones que tendrá y x será la variable que se le ira agregando al acumulador

num: (dato tipo numérico con decimales).

acumulador: (dato tipo texto).

x: (dato tipo texto).

• Datos de salida:

La variable acumuladora pero esta ira aumentando

• Proceso:
establecer un ciclo for donde el valor inicial de i sea 1 y su limite sea la variable num y este vaya aumentando uno en uno por cada repetición cada que este se repita sumarle el valor de x a la variable acumulador el cual su valor es “x” para que este vaya aumentando 1 cada repetición e imprimir acumulador.

• Diseño de la solución:

Algoritmo (pasos para resolver un problema, tiene 3 características: finitud, precisión,

determinista).

1.-Inicio

2.- pedir al usuario el valor de n el cual será la cantidad máxima de x.

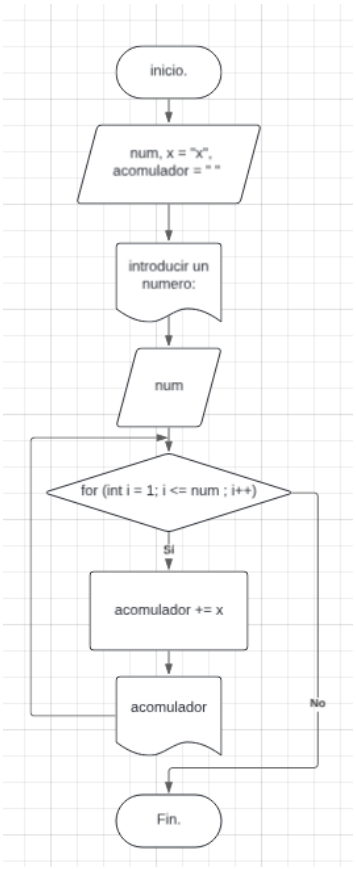
3.-agregar un ciclo anidado donde el valor inicial de i sea 1 y que el límite de este sea num y este vaya aumentando de 1 en 1.

4.- sumarle el valor de la variable x a la variable acumulador para que esta vaya aumentando de x cada que el for se repita

5.- imprimir acumulador

6.-Fin.

Diagrama de flujo:



Programa:

```java
import java.util.Scanner;
public class equis {
    public static void main (String[] args){
        Scanner s = new Scanner(System.in);
        int num;
        String x = "x";
        String acomulador ="";
        System.out.print("introducir un numero: ");
        num = s.nextInt();
        for (int i = 1; i <= num ; i++) {
            acomulador +=x;
            System.out.println(acomulador);
        }
    }
}
```

Resultados:

```
introducir un numero: 5
x
xx
xxx
xxxx
xxxxx
```

```
introducir un numero: 10
x
xx
xxx
xxxx
xxxxx
xxxxxx
xxxxxxx
xxxxxxxx
xxxxxxxxx
xxxxxxxxxx
```