

# Asignatura Introducción a la Programación

## Ingeniería Informática

### Guía de Aprendizaje

**Tema :** Introducción al lenguaje de programación Java

**Unidad didáctica:** Transcripción de algoritmos a lenguaje de programación Java. Estructura repetitivas y ciclos.

#### 1. Objetivos de la unidad didáctica:

- Caracterizar un algoritmo repetitivo o cíclico, bucle.
- Identificar el uso de estructuras repetitivas o cíclicas en el diseño de algoritmos.
- Caracterizar las estructuras repetitivas o cíclicas del lenguaje de programación Java

#### 2. Introducción a la unidad didáctica:

Cuando se diseña un algoritmo que sea capaz de resolver un determinado problema nos podemos encontrar que nuestro algoritmo puede ser:

- **Un algoritmo lineal o secuencial:** Algoritmo donde todas las instrucciones se ejecutan de forma secuencial una a continuación de otra si dejar de omitir ninguna instrucción.
- **Un algoritmo alternativo o con bifurcación:** Algoritmo donde un grupo de instrucciones pueden o no ejecutarse acorde a una expresión lógica booleana.
- **Un algoritmo repetitivo o cíclico:** Algoritmo donde un grupo de instrucciones se pueden ejecutar una cantidad de veces superior con respecto a otras instrucciones también presentes en el algoritmo.

En esta guía vamos a concentrarnos en los algoritmos repetitivos o cíclicos y como hacer su transcripción a un lenguaje de programación, en este caso Java. Comencemos por el siguiente ejercicio el cual fue enunciado en guías de aprendizajes anteriores.

*Diseñe un algoritmo que permita calcular y mostrar por pantalla el factorial de un número natural N. Tenga en cuenta que el algoritmo debe solicitar o pedir el valor N por pantalla. Utilice una de las técnicas para representar algoritmo que sea mas de su agrado.*

#### Análisis del algoritmo

Lo primero es convertir nuestro problema en el objetivo a resolver y cual es nuestro objetivo entonces. Diseñar un algoritmo que permita calcular el factorial de un número cuyo valor es introducido por el usuario. Vamos a recordar un poco que es el factorial de número. El factorial de un número es el resultado de la multiplicación de todos los números entre

naturales entre 1 y N. Por ejemplo el factorial de 4 es 24 porque es el resultado de  $1 \times 2 \times 3 \times 4$ .

Entonces una vez refrescado o aclarado que cosa es el factorial vayamos a determinar la cantidad de variables que necesitamos. Si se analiza creo que con tres variables no es suficiente: la primera para almacenar el valor de N del factorial de N mientras una tercera nos va servir para saber por cual número en el rango de 1 a N debemos multiplicar en cada momento para determinar el factorial. Es evidente que la instrucción de multiplicación se va repetir tantas veces como el valor de N. Hagamos el diseño de nuestro algoritmo.

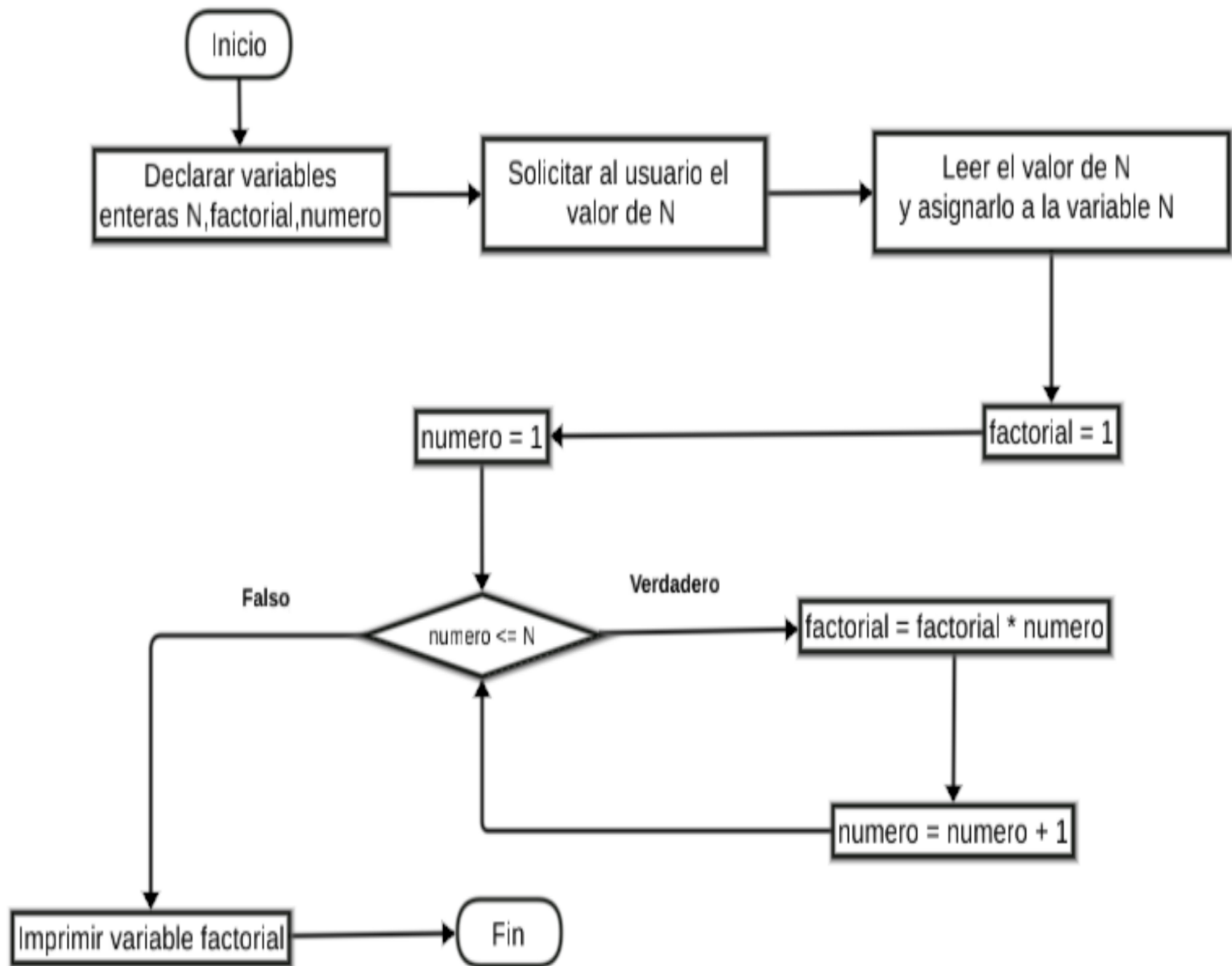
1. Declarar variables entera N, factorial y numero
2. Solicitar al usuario el valor de N
3. Leer el valor del coeficiente N y asignarlo a la variable N.
4. factorial = 1
5. numero = 1
6. Mientras numero  $\leq$  N:
  - 6.1 factorial = factorial \* numero
  - 6.2 numero = numero + 1
  - 6.3 volver a la instrucción 6

#### 7. Imprimir variable factorial

Ahora analicemos un momento el pseudocódigo anterior para ver algunos detalles. Lo primero es que hasta la instrucción 5 estamos en presencia de un algoritmo lineal ya que todas las sentencias se van ejecutar una a continuación de otra pero al llegar a la instrucción 6 la situación cambia completamente.

Por ejemplo la ejecución y la cantidad de ejecuciones de la instrucción 6.1 va depender de que sea cierta o no la instrucción 6. Observemos que si la instrucción 6.2 no existiera la condición de la instrucción 6 siempre sería verdadera por tanto nuestro algoritmo ejecutaría eternamente las instrucciones 6 , 6.1, 6.3 en ese orden realizando un ciclo infinito.

Ahora veamos el mismo algoritmo pero utilizando diagrama de bloque o flujo. Donde se debe reflejar mejor la situación que queremos explicar.



Es evidente ahora el ciclo o bucle se conforma entre la instrucción y las subinstrucciones de estas. Esta es la cuestión principal de los algoritmos repetitivos o con ciclos donde una o grupo de instrucciones van a ejecutarse una cantidad de veces diferentes a otras instrucciones durante la ejecución del programa. Lo siguiente es transcribir lo planteado anteriormente a lenguaje de programación Java. Lo cual nos quedaría de la siguiente manera:

```

import java.util.Scanner;

public class Main {

    public static void main(String [] args) {

        int N,numero,factorial;

        Scanner input = new Scanner( System.in );

        System.out.print("Entre el valor N: ");

        N = input.nextInt();

        factorial = 1;

        numero = 1;
  
```

```

        while (numero <= N){

            factorial= factorial * 1;

            numero = numero + 1; //Tambien se pudo poner lo siguiente numero++;

        }

        System.out.println("El factorial de "+N+" es: "+factorial);

    }

}

```

Es muy posible que lo anterior no hayas entendido en gran parte o en su totalidad pero con esta guía vamos a intentar que lo comprendas en su totalidad y que sea capaz no solo de diseñar algoritmos repetitivos sino transcribirlo utilizando el lenguaje de programación Java para que la computadora los realice.

### 3. Orientaciones para el estudio:

1. Le sugerimos que para realizar el estudio de una materia debe elegir un lugar agradable, limpio, ventilado, cómodo, iluminado y si necesita recursos tecnológicos deben estar disponibles.
2. Cree hábitos de estudio sistemático; para esto:
  - Planifique su tiempo y no deje para mañana lo que pueda hacer hoy; una buena planificación hace manejable las responsabilidades diarias que tenemos para con la vida, permite cumplir con todas las tareas programadas y así evita agobios de última hora.
  - Calendarice las fechas más importantes de estudio y entrega de tareas. Localice con antelación los materiales que necesita para realizar el estudio individual o colaborativo.
3. Utilice técnicas de estudio:
  - Elija un entorno de estudio que resulte agradable y sin elementos que lo que distraigan
  - Estudie activamente, para esto lea en voz alta, tome notas, elabore esquemas o mapas conceptuales, realice resúmenes.
  - Reflexione sobre lo que va aprendiendo, para esto relacione lo nuevo con lo anterior o conocido, asegúrese de que entiende y es capaz de aplicar lo que está aprendiendo antes de pasar adelante.
4. Interactúe con los materiales de estudio en tres fases:
  - Fase de aproximación: Revise el objetivo de la unidad y después la acción o acciones a lograr; busque los materiales sugeridos para desarrollar la tarea,

verifique cuál es el tiempo de que dispone.

- Fase de lectura profunda: aproxímese al material a través de una lectura ligera, poniendo especial interés en los títulos y subtítulos. Trate de relacionar lo que va leyendo con conocimientos adquiridos previamente.
- Fase de evaluación: Una vez realizada la lectura, intente realizar las actividades de auto-aprendizaje.

#### 4. Requisitos Previos:

Tener conocimientos básicos de computación: los estudiantes matriculados deben conocer los procedimientos básicos para el manejo de PC y/o dispositivos móviles, aplicaciones de ofimática; así como uso de navegadores de páginas Web y el correo electrónico.

#### 5. Actividades de auto-aprendizajes:

1. Cuales son las estructuras repetitivas o cíclicas que tiene el lenguaje de programación Java. Identifique algunas de dichas estructuras utilizados en el código presentado en la **Introducción a la unidad didáctica**.

2. De la estructura repetitiva o iterativa **while** defina su sintaxis en el lenguaje de programación Java. Funcionamiento de la misma.

3. De la estructura repetitiva o iterativa **do ... while** defina su sintaxis en el lenguaje de programación Java. Funcionamiento de la misma.

4. De la estructura repetitiva o iterativa **for** defina su sintaxis en el lenguaje de programación Java. Funcionamiento de la misma.

5. Elabore un cuadro comparativo de las ventajas, desventajas, diferencias de las estructuras repetitivas **for**, **while** y **do ... while**.

6. De las siguientes sentencias diga cuales son verdaderas y cuales son falsas. En caso de las falsas justifique.

- La estructura **while** se puede siempre sustituir por la estructura **for**.
- La estructura **while** se puede siempre sustituir por la estructura **do ... while**.
- La estructura **do ... while** se puede siempre sustituir por la estructura **for**.
- La estructura **do ... while** se puede siempre sustituir por la estructura **while**.
- La estructura **for** se puede siempre sustituir por la estructura **do ... while**.
- La estructura **for** se puede siempre sustituir por la estructura **while**.

#### 6. Actividades de evaluación:

1. Diseñe un algoritmo que simule el conteo de regresivo a partir de una valor introducido por el usuario hasta cero. Utilice una de las técnicas para representar algoritmo que sea mas de

su agrado. Transcriba dicho algoritmo a lenguaje de programación de Java. Ejemplo si el usuario introduce el valor 7. El programa debe generar la siguiente salida:

7 segundos ...

6 segundos ...

5 segundos ...

4 segundos ...

3 segundos ...

2 segundos ...

1 segundos ...

0 segundos ...

**2.** Diseñe un algoritmo que permita al usuario introducir una cantidad indeterminada de números. Por cada número introducido se debe imprimir el cuadrado del número. El programa debe deber terminar cuando el valor introducido sea 42. Utilice una de las técnicas para representar algoritmo que sea mas de su agrado. Transcriba dicho algoritmo a lenguaje de programación de Java.

**3.** Diseñe un algoritmo que permita determinar cuantos números existen en el rango [a,b] tal que terminen en 2 o 6 o 3. Los valores a y b son introducidos por el usuario y el valor de a es siempre menor igual que b. Utilice una de las técnicas para representar algoritmo que sea mas de su agrado. Transcriba dicho algoritmo a lenguaje de programación de Java.

**4.** Diseñe un algoritmo que permita dibujar un cuadrado (utilizando el símbolo asteriscos \*) del cual se conoce la longitud de su lado cuyo valor es introducido por el usuario. Utilice una de las técnicas para representar algoritmo que sea mas de su agrado. Transcriba dicho algoritmo a lenguaje de programación de Java. Si el usuario introduce el valor 4 el programa debe generar la siguiente salida.

\*\*\*\*

\*\*\*\*

\*\*\*\*

\*\*\*\*

**5.** Diseñe un algoritmo que permita dibujar un triangulo rectángulo e isóscele (utilizando el símbolo asteriscos \*) del cual se conoce la longitud de su lado (el lado que su longitud es igual a otro lado) cuyo valor es introducido por el usuario. Utilice una de las técnicas para representar algoritmo que sea mas de su agrado. Transcriba dicho algoritmo a lenguaje de programación de Java. Si el usuario introduce el valor 4 el programa debe generar la siguiente salida.

\*

\*\*

\*\*\*

\*\*\*\*

**6.** Volver a realizar los ejercicios anteriores pero cambiando la estructura repetitiva utilizada inicialmente por otra.

## **7. Resumen:**

La estructura repetitivas permiten que nuestros algoritmos sean capaces de ejecutar determinadas instrucciones una cantidad de veces que pueden variar dependiendo de una condición. Esto amplía las herramientas con que contamos para diseñar e implementar algoritmos.

Todos los lenguajes de programación cuentan con al menos una estructura repetitiva e iterativa dentro del arsenal de herramienta que le brinda a los programadores.

## **8. Glosario de términos:**

Consulte la bibliografía y defina el funcionamiento de las siguientes palabras reservadas:

break:

continue:

## **9. Bibliografía:**

Aprenda Java como si estuviera en primero. *Colectivo de autores*. Cap 2 epígrafes 2.3.4, 2.3.4.1, 2.3.4.2, 2.3.4.3, 2.3.4.4, 2.3.4.5.

Cómo programar en Java. *Deitel, Paul J. Y Harvey M. Deitel*. Cap 4 epígrafes 4.7, 4.8, 4.9, 4.10, 4.11, Cap 5 epígrafes 5.1, 5.2, 5.3, 5.4, 5.5, 5.6, 5.7, 5.8, 5.9

## **10. Recursos educativos digitales RED:**

**11. Próxima unidad:** Transcripción de algoritmos a lenguaje de programación Java.  
Ejercitación de los contenidos.