

**Universidad Nacional de Loja**  
**Facultad de la Energía, las Industrias**  
**y los Recursos Naturales no**  
**Renovables**

**Carrera:** Computación

**Asignatura:** Teoría de la Programación

**Nombre completo:** Joaquin Moscol

Castillo

**Ciclo:** 1er Ciclo

**Unidad:** 1ra Unidad

**TEMA: Primer Acercamiento a la**  
**Construcción de Algoritmos con**  
**Estructuras Secuenciales en**  
**Pseudocodigo**



1859



Universidad  
Nacional  
de Loja

## Descripción de la(s) herramienta(s)

Herramienta seleccionada: Flowgorithm

### Características:

Flowgorithm es una herramienta gratuita y de interfaz amigable que permite diseñar algoritmos mediante diagramas de flujo.

Genera automáticamente pseudocódigo a partir del diagrama.

Soporta múltiples lenguajes de programación en su salida (como C++, Java, Python, etc.).

Es ideal para principiantes por su simplicidad visual.

### Ventajas:

Interfaz intuitiva y fácil de usar.

No requiere instalación compleja.

Ayuda a visualizar el flujo de un algoritmo de manera clara.

Permite ejecutar el diagrama paso a paso para depuración.

### Limitaciones:

No es colaborativo en línea.

Limitado en opciones avanzadas para proyectos complejos.

Solo disponible para Windows.

## Descripción del algoritmo seleccionado

### Escenario:

Diseño de un algoritmo que determine si un número ingresado por el usuario es par o impar.

### Objetivo:

Ilustrar el uso de estructuras secuenciales y condicionales básicas.

### i. Pseudocódigo

plaintext

Inicio

Definir número como entero

Escribir "Ingrese un número:"

Leer número

Si número mod 2 = 0 Entonces

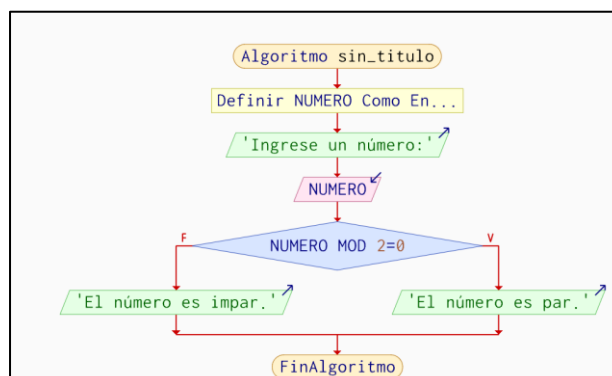
Escribir "El número es par."

Sino

Escribir "El número es impar."

Fin Si

Fin



### 3. CONCLUSIONES

El diseño de algoritmos mediante pseudocódigo y diagramas de flujo es fundamental en la resolución de problemas computacionales, ya que permite estructurar la lógica de programación de manera clara y ordenada antes de su implementación en un lenguaje específico. Herramientas como Flowgorithm facilitan este proceso al ofrecer un entorno visual e interactivo que mejora la comprensión y depuración de los algoritmos. Esta práctica refuerza habilidades de pensamiento lógico y es la base para el desarrollo de software eficiente y libre de errores.

### 4. BIBLIOGRAFÍA

[1] Flowgorithm. (2023). *Official Website*. Recuperado de: <http://www.flowgorithm.org/>

[2] IEEE. (2020). *IEEE Citation Reference*. Recuperado de: <https://ieeauthorcenter.ieee.org/wp-content/uploads/IEEE-Reference-Guide.pdf>