

Introducción a los Fundamentos de Programación y Pseudocódigo Introducción a la Programación y Computación 1

Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Ciencias Y Sistemas

Agenda



RECORDATORIOS



¿Qué es la Programación?



Fundamentos de Programación



Elementos de un Lenguaje de Programación



Estructuras Básicas



¿Qué es el Pseudocódigo?



Reglas para Escribir Pseudocódigo



Ejemplos Prácticos de Pseudocódigo



Ventajas del Pseudocódigo



Conclusión y Recomendaciones

¿Qué es la Programación?

- Es el proceso mediante el cual se diseña y se crea un conjunto de instrucciones (código) que una computadora puede ejecutar.
- Permite automatizar tareas, resolver problemas y construir aplicaciones que interactúan con el usuario.
- Es una habilidad esencial en el mundo actual, transversal a diversas industrias y sectores.

Fundamentos de Programación

Los fundamentos de la programación constituyen los conceptos esenciales que todo programador debe conocer:

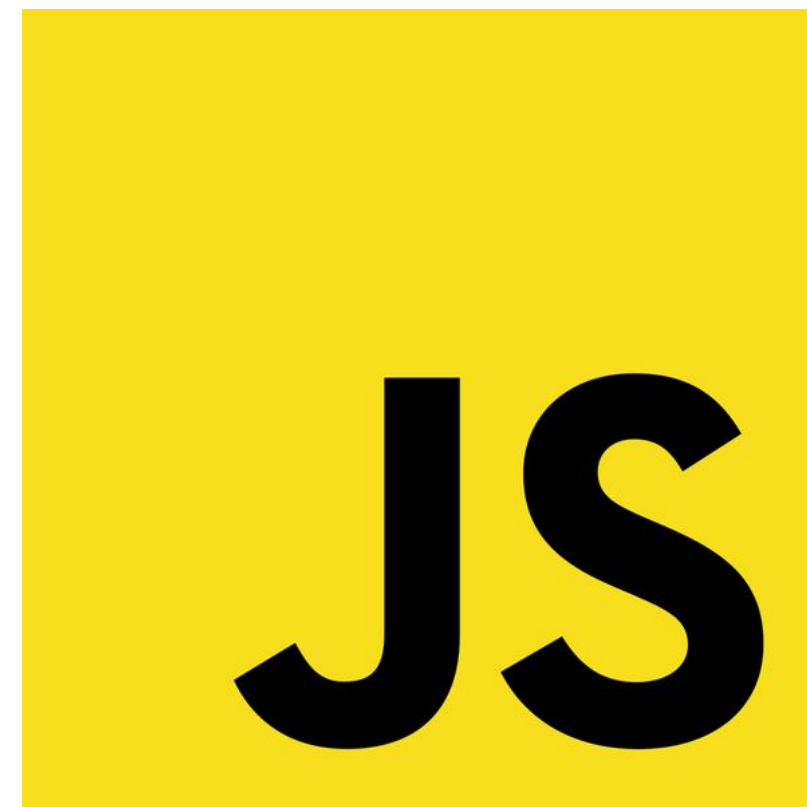
- Algoritmos: Secuencia de pasos finitos para resolver un problema.
- Variables y Tipos de Datos: Elementos para almacenar y manipular datos.
- Estructuras de Control: Decisiones (if, else) y repeticiones (for, while).
- Entrada y Salida: Comunicación con el usuario.
- Funciones o Procedimientos: Bloques de código reutilizables.



Elementos de un Lenguaje de Programación

- Sintaxis: Reglas que definen cómo se debe escribir el código.
- Semántica: Significado del código que se escribe.
- Compilación o Interpretación: Proceso mediante el cual el código fuente se traduce y ejecuta.

Ejemplos de lenguajes: Java, Python, C++, JavaScript.



Estructuras Básicas

- Secuencia: Ejecución ordenada línea por línea.
- Selección: Uso de condiciones (if, else) para ejecutar decisiones.
- Iteración: Repetición de bloques (for, while).

Inicio

Leer edad

Si edad \geq 18 Entonces

Escribir “Eres mayor de edad”

FinSi

Fin

¿Qué es el Pseudocódigo?



Simple

- Representación informal y estructurada de un algoritmo.
- Se utiliza para planificar la lógica de un programa sin enfocarse en la sintaxis de un lenguaje específico.
- Es útil para diseñar y comunicar ideas antes de codificar.

Reglas para Escribir Pseudocódigo

- Utilizar palabras clave comunes: Inicio, Fin, Leer, Escribir, Si, Entonces, Mientras, Repetir, Hasta que.
- Claridad y sencillez en cada paso.
- No emplear estructuras de código específicas de lenguajes de programación.
- Usar indentación para representar la jerarquía y el flujo.

Estructura General de un Pseudocódigo

- Inicio
 - Declarar variables
 - Entrada de datos
 - Procesamiento de datos
 - Salida de resultados
 - Fin
- Inicio
 - Leer base
 - Leer altura
 - $\text{área} \leftarrow \text{base} * \text{altura}$
 - Escribir "El área es: ", área
 - Fin

Ejemplos

- Inicio
- Leer número
- Si número MOD 2 = 0 Entonces
- Escribir "El número es par"
- Sino
- Escribir "El número es impar"
- FinSi
- Fin

- Inicio
- Leer N
- $\text{suma} \leftarrow 0$
- Para $i \leftarrow 1$ Hasta N Hacer
- $\text{suma} \leftarrow \text{suma} + i$
- FinPara
- Escribir "La suma es: ", suma
- Fin

Ventajas del Uso de Pseudocódigo

- Facilita el diseño y planificación de algoritmos.
- Permite concentrarse en la lógica sin preocuparse por la sintaxis del lenguaje.
- Facilita la comunicación entre personas sin conocimientos técnicos profundos.
- Útil como paso previo antes de la programación real.

Comparación: Pseudocódigo vs Código Real

Inicio

Leer número1, número2

suma \leftarrow número1 + número2

Escribir suma

Fin

```
import java.util.Scanner;
public class Suma {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int numero1 = sc.nextInt();
        int numero2 = sc.nextInt();
        int suma = numero1 + numero2;
        System.out.println(suma);
    }
}
```

El Pseudocódigo como Puente entre la Lógica y el Código

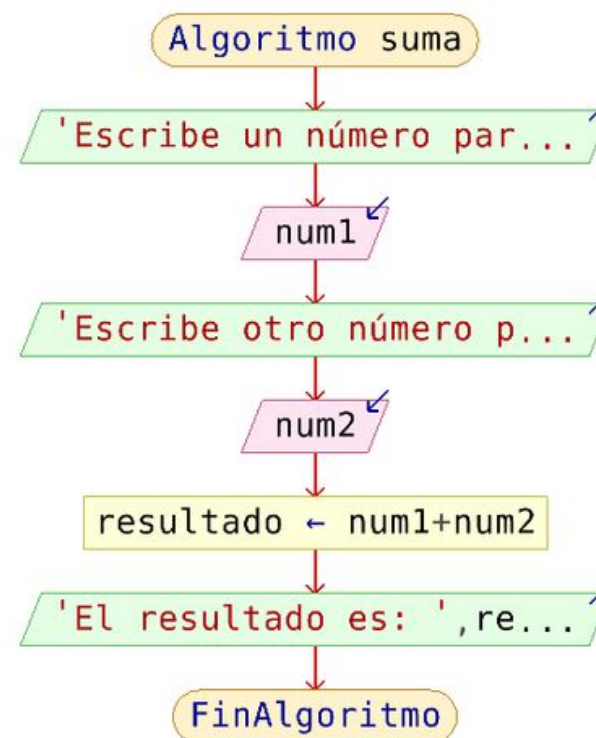
El pseudocódigo actúa como una herramienta intermedia entre el pensamiento lógico y la implementación en un lenguaje de programación. Cuando un estudiante escribe un pseudocódigo, no está limitado por reglas estrictas de sintaxis, lo cual permite que se enfoque en el proceso de resolución de problemas. Esto es especialmente valioso en etapas tempranas del aprendizaje, cuando el principal objetivo no es aprender un lenguaje específico, sino aprender a pensar como programador.

Pseudocódigo en el Aula: Una Herramienta Didáctica Fundamental

En ambientes educativos, el pseudocódigo ha demostrado ser una excelente herramienta para introducir a los estudiantes a los algoritmos sin sobrecargarlos con detalles técnicos. Muchas veces, los errores en los programas no son por mala sintaxis, sino por errores lógicos o de diseño. El pseudocódigo ayuda a aislar ese problema, permitiendo que los estudiantes se concentren en la lógica antes de pasar a la codificación. Por esta razón, los docentes suelen enseñar primero lógica de programación con pseudocódigo antes de introducir lenguajes como Java o Python.

Relación entre Pseudocódigo y Diagrama de Flujo

El pseudocódigo y el diagrama de flujo son dos herramientas que cumplen propósitos similares: representar la lógica de un algoritmo. Sin embargo, mientras que el diagrama de flujo se basa en elementos visuales (formas, flechas, símbolos), el pseudocódigo se basa en estructura textual secuencial, más cercana al código real. Muchos estudiantes prefieren el pseudocódigo porque pueden transcribirlo fácilmente a un lenguaje de programación. Otros encuentran en los diagramas una ayuda visual para comprender flujos complejos. Ambas herramientas se complementan entre sí.



Buenas Prácticas al Escribir Pseudocódigo

Al igual que en la programación real, escribir pseudocódigo de manera ordenada, clara y consistente es esencial para su efectividad. Se recomienda usar nombres de variables significativos, emplear comentarios que aclaren decisiones importantes, y estructurar el código con indentación adecuada. Estas prácticas, aunque parezcan simples, fomentan desde el inicio una mentalidad disciplinada y profesional, que será de gran utilidad cuando el estudiante avance hacia lenguajes más formales.

Recomendaciones Finales para Iniciarse en Programación

Antes de aprender un lenguaje de programación, es recomendable dominar primero los principios fundamentales de lógica computacional. Comprender qué es un algoritmo, cómo se estructura una decisión, cómo se representa una repetición, y cómo se recibe y entrega información, es el verdadero núcleo de la programación. El pseudocódigo es el primer paso hacia ese entendimiento. Una vez interiorizados estos principios, el aprendizaje de cualquier lenguaje será mucho más natural.

**¡GRACIAS
ATENCIÓN!**

POR

SU



DUDAS

RECUERDA QUE TENEMOS NUESTRO FORO SEMANAL DONDE PUEDES CONSULTAR CUALQUIER DUDA QUE
TE SURJA EN LA SEMANA