



GUÍA DE TRABAJO VIRTUAL

ASIGNATURA: PROGRAMACION SEMANA DE TRABAJO: MARZO FEBERERO -

Guía elaborada por: Juan Manuel Henao

METAS DE APRENDIZAJE / COMPETENCIAS A DESARROLLAR

- Desarrollar algoritmos que permitan la apropiación de los conocimientos.
- Repasar los conceptos básicos de fundamentos de programación.

LECTURAS

LECTURA 1: ALGORTIMOS Y PSEUDOCODIGO

Un Algoritmo es una secuencia de pasos para resolver un problema y debe contar con las siguientes características:

- Preciso: cada paso debe ser claro y exacto en su construcción para que así determine puntualmente lo que se desea hacer.
- Definido: toda vez que se ejecute el algoritmo con los mismos datos de entrada, éste debe generar el mismo resultado.
- Finito: todo algoritmo debe tener un fin.

El algoritmo se construye usando palabras del idioma y debe poder ser entendido por cualquier persona.

Cuando se utilizan líneas de código se crea un programa y éstos se escriben usando lenguajes de programación. Si se emplea cuasi código en la construcción del programa, se tiene un Pseudocódigo.

Un algoritmo se construye en forma general usando los siguientes pasos:

1. Inicio.
2. Capturar, conocer o ingresar todos los datos que permitan resolver el problema.
3. Resolver, calcular o llevar a cabo los procesos con los datos capturados. Se sugiere en este punto colocar las ecuaciones necesarias.
4. Mostrar, visualizar o imprimir todos los resultados que el problema exija.
5. Fin.



GUÍA DE TRABAJO VIRTUAL

El siguiente ejemplo sencillo muestra un error clásico en la elaboración de un algoritmo:

Construya un algoritmo que encuentre y muestre el valor del área de un rectángulo.

PASOS

- 1 conocer los valores de altura y base
- 2 calcular el área
- 3 mostrar el valor del área hallado

El algoritmo no es preciso en sus pasos porque no indica a que figura geométrica hay que calcularle el área, por lo tanto, no se puede identificar la fórmula a aplicar.

Además, para mostrar que un algoritmo termina es conveniente incluir como último paso la palabra fin.

El algoritmo para resolver el ejemplo anterior sería entonces:

PASOS

- 1 inicio
- 2 conocer los valores de altura y base del rectángulo
- 3 calcular el área del rectángulo
- 4 mostrar el valor del área del rectángulo
- 5 fin

TIPOS DE INSTRUCCIONES

A partir de los algoritmos se pueden elaborar programas y para ello simplemente basta con "convertir" los pasos del algoritmo a sus instrucciones respectivas según el lenguaje de programación que se esté utilizando, pero, como existen diferentes lenguajes, inicialmente se construirán los programas usando una de las herramientas de programación: el Pseudocódigo.



GUÍA DE TRABAJO VIRTUAL

El Pseudocódigo se construye con instrucciones que no son específicas de un lenguaje de programación pero que reflejan un comportamiento similar; al tener un programa en pseudocódigo se puede convertir éste con mayor facilidad a un lenguaje de programación.



Las siguientes son algunas de las instrucciones que hacen parte del pseudocódigo.

Instrucción

Inicio / Fin

Se utilizan para dar comienzo y terminación al pseudocódigo, por cada programa escrito debe existir un solo Inicio y Fin.

A continuación, se muestran cómo se escriben las instrucciones en el pseudocódigo y a su vez, su representación en la segunda herramienta de programación: el diagrama de flujo.

PSEUDOCÓDIGO Instrucción	DIAGRAMA DE FLUJO
Inicio	
Fin	

El pseudocódigo es un lenguaje de alto nivel empleado para describir algoritmos, igual que los diagramas de flujo, aunque más cercano a la computadora. El objetivo último del pseudocódigo es facilitar la comprensión del algoritmo.

Concretamente, el pseudocódigo describe un algoritmo utilizando una mezcla de frases en lenguaje común, instrucciones de programación y palabras clave que definen las estructuras básicas. Su objetivo es permitir que el programador se centre en los aspectos lógicos de la solución a un problema. El pseudocódigo varía de un programador a otro, es decir, no hay una estructura semántica ni arquitectura estándar.



INSTITUTO UNIVERSITARIO DE CALDAS

"Dignificando la escuela transformamos el mundo"

GUÍA DE TRABAJO VIRTUAL

LECTURA 2 DIAGRAMAS DE FLUJO

Los diagramas de flujo u organigramas son esquemas que se emplean para representar gráficamente un algoritmo.

Se basan en la utilización de diversos símbolos convenidos para representar operaciones específicas. Se les llama diagramas de flujo porque los símbolos utilizados se conectan por medio de flechas para indicar la secuencia de operación.

Es recomendable (especialmente en problemas complicados o muy largos), a la hora de establecer el problema y antes de comenzar a programar, expresar mediante diagramas de flujo el algoritmo. Los diagramas de flujo se dibujan generalmente usando símbolos estándares. No obstante, símbolos especiales pueden ser introducidos ad hoc siempre y cuando sean requeridos. Los símbolos más comunes se muestran a continuación:



GUÍA DE TRABAJO VIRTUAL







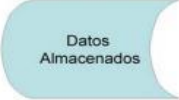







	
	
	
	
 Operación lógica OR	 Operación lógica AND
	 Líneas de flujo
	

Tabla 1. Símbolos estándar empleados en diagramas de flujo.

Algunas reglas para dibujar diagramas de flujo son

Análisis Numérico: Lógica de programación y diagramas de flujo

1. Los Diagramas de flujo deben escribirse de arriba hacia abajo, y/o de izquierda a derecha.
2. Los símbolos se unen con líneas, las cuales tienen en la punta una flecha que indica la dirección que fluye la información, se deben de utilizar solamente líneas de flujo horizontal o verticales (nunca diagonales).



GUÍA DE TRABAJO VIRTUAL

3. Se debe evitar el cruce de líneas, para lo cual se quisiera separar el flujo del diagrama a un sitio distinto, se pudiera realizar utilizando los conectores. Se debe tener en cuenta que solo se vaya a utilizar conectores cuando sea estrictamente necesario.
4. No deben quedar líneas de flujo sin conectar.
5. Todo texto escrito dentro de un símbolo debe ser legible, preciso, evitando el uso de muchas palabras.
6. Todos los símbolos pueden tener más de una línea de entrada, a excepción del símbolo final.
7. Solo los símbolos de decisión pueden y deben tener más de una línea de flujo de salida.

RECURSOS

RECURSO 1

Video Algoritmos y Seudocódigo

https://www.youtube.com/watch?v=2UvW_ziPRIM&ab_channel=Genesisjccm



GUÍA DE TRABAJO VIRTUAL

RECURSO 2

Video Diagramas de flujo

https://www.youtube.com/watch?v=Lub5qOmY4JQ&ab_channel=JuanRaGarciaMontes

ACTIVIDADES

ACTIVIDAD 1

Hacer un resumen de las lecturas 1 y 2 donde quede en el cuaderno registrado los conceptos principales de algoritmos, Seudocódigo y diagramas de flujos.

ACTIVIDAD 2

Crear 10 algoritmos.

Crear el pseudocódigo y los diagramas de flujos para estas actividades.

Ejemplo:

Ejercicio No 1: Desarrolle un algoritmo que permita leer dos valores distintos, determinar cuál de los dos valores es el mayor y escribirlo

Pseudocódigo	Diagrama de Flujo
<ol style="list-style-type: none">InicioInicializar variables: $A = 0$, $B = 0$Solicitar la introducción de dos valores distintosLeer los dos valoresAsignarlos a las variables A y BSi $A = B$ Entonces vuelve a 3 porque los valores deben ser distintosSi $A > B$ Entonces Escribir A, "Es el mayor"De lo contrario: Escribir B, "Es el mayor"Fin_SiFin	

EVALUACIONES



INSTITUTO UNIVERSITARIO DE CALDAS

"Dignificando la escuela transformamos el mundo"

GUÍA DE TRABAJO VIRTUAL

EVALUACIÓN 1

A través del correo electrónico del docente.

Fotos o pantallazos a través de whatsapp

Entrega en medio físico (según disposiciones de la institución educativa)

NOTA: El correo debe de enviarlo con la siguiente estructura: En el **Asunto** escriba Nombres, apellidos y el grado correspondiente. **Ejemplo:** Juan Manuel Henao 10.4 Diplomado

CRITERIOS DE EVALUACIÓN Y PLAZOS DE ENTREGA

INFORMACIÓN DE CONTACTO

DOCENTE 1

- Nombre: Juan Manuel Henao
- Grupos: 10.A
- Correo Electrónico: juanma1715@hotmail.com