



REPUBLICA BOLIVARIANA DE VENEZUELA  
UNIVERSIDAD NACIONAL EXPERIMENTAL DE GUAYANA  
VICERRECTORADO ACADÉMICO  
DEPARTAMENTO DE CIENCIA Y TECNOLOGÍA  
COORDINACIÓN DE INGENIERIA EN INFORMÁTICA  
UNIDAD CURRICULAR FUNDAMENTOS DE LA INFORMÁTICA

PROFA. CLINIA CORDERO

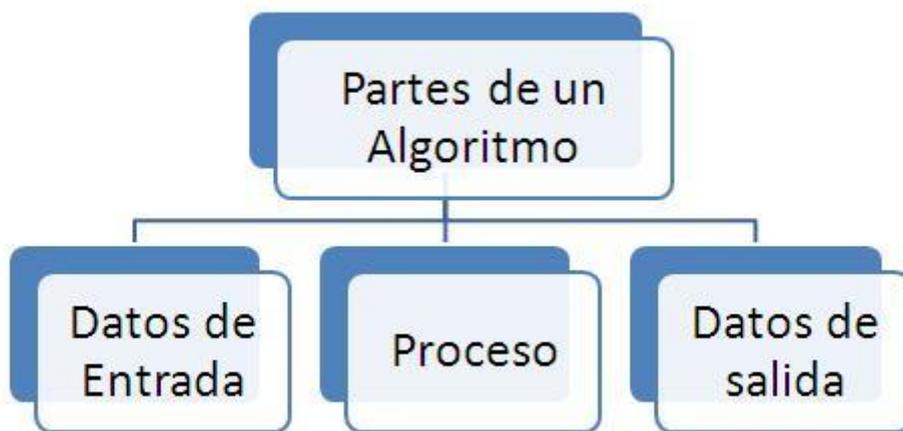
## ALGORITMOS Y DIAGRAMAS DE FLUJOS

### Introducción a los algoritmos

#### CONCEPTO DE ALGORITMO:

Es un conjunto de pasos secuenciales y ordenados que permiten lograr un objetivo. Que sean pasos secuenciales significa que deben ser ejecutados uno después de otro y que sean pasos ordenados quiere decir que deben llevar un orden casi obligatorio (u obligatorio en la mayoría de los casos). Como puede notar el algoritmo permite lograr un objetivo. O sea que la clave para hacer buenos algoritmos y llegar a la solución de un problema es tener claro el objetivo y no perderlo nunca de vista.

Debe ser independiente de la sintaxis de un lenguaje de programación puesto que es la concreción de un plan y un medio para comunicar ideas entre personas. Lo que sí es importante notar, es que de un algoritmo sale el código en un lenguaje de programación. Por cada línea del algoritmo se corresponden una o varias líneas de código en un lenguaje de programación.



### CARACTERÍSTICAS QUE DEBEN DE CUMPLIR LOS ALGORITMOS OBLIGATORIAMENTE:

- ♦ **Un algoritmo debe resolver el problema para el que fue formulado.**

Lógicamente no sirve un algoritmo que no resuelve ese problema. En el caso de los programadores, a veces crean algoritmos que resuelven problemas diferentes al planteado. Los algoritmos deben mostrar claramente cuáles son los datos iniciales y cuáles son los resultados.

- ♦ **Los algoritmos son independientes del computador.** Los algoritmos se escriben para poder ser utilizados en cualquier máquina.

- ♦ **Los algoritmos deben de ser precisos.** Los resultados de los cálculos deben de ser exactos, de manera rigurosa. No es válido un algoritmo que sólo aproxime la solución. Debe constar de pasos claros, precisos y no ambiguos.

- ♦ **Los algoritmos deben de ser finitos.** Deben alcanzar la solución correcta en un tiempo finito. No es un algoritmo válido aquel que produce situaciones en las que el algoritmo no termina.

- ♦ **Los algoritmos deben de poder repetirse.** Deben de permitir su ejecución las veces que sea necesario. No son válidos los que tras ejecutarse una vez, ya no pueden volver a hacerlo por la razón que sea. Debe tener la capacidad de resolver el problema aun cuando cambiemos los datos de entrada.

### CARACTERÍSTICAS ACONSEJABLES PARA LOS ALGORITMOS

- ♦ **Validez.** Un algoritmo es válido si carece de errores. Un algoritmo puede resolver el problema para el que se planteó y sin embargo no ser válido debido a que posee errores.

♦ **Eficiencia.** Un algoritmo es eficiente si obtiene la solución al problema en poco tiempo. No lo es si es lento en obtener el resultado.

♦ **Óptimo.** Un algoritmo es óptimo si es el más eficiente posible y no contiene errores. La búsqueda de este algoritmo es el objetivo prioritario del programador. No siempre podemos garantizar que el algoritmo hallado es el óptimo, a veces sí.

## ESTRUCTURA GENERAL DE UN ALGORITMO.

Aunque no existe una única forma de representar un algoritmo, la estructura general de éste debería ser como la siguiente:

```
Algoritmo <nombre_del_algoritmo>
Inicio
definición de constantes
declaración de variables
Sentencia 1
Sentencia 2
.
.
Sentencia n
Fin
```

## REPRESENTACIÓN DE ENTRADAS Y SALIDAS DE UN ALGORITMO.

### ENTRADAS

Entradas son los datos proporcionados al computador para procesar y generar la salida. Para que los resultados sean correctos, se tienen que dar los datos de entrada correctos. La entrada se puede representar de varias formas. Flujogramas y pseudocódigo tienen diferentes métodos de representar la instrucción de aceptar una entrada. Algunos ejemplos de representación de entradas son los siguientes:

En un Flujograma, la instrucción es representada usando el símbolo llamado “Entrada de datos”.

Los símbolos utilizados en flujogramas para representar las entradas son los siguientes: En un pseudocódigo, la instrucción se da a través de teclado.

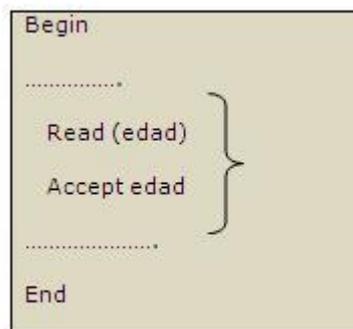
Existen varias formas de representar el pseudocódigo. Por ejemplo, para leer la edad de un estudiante en una variable llamada edad (Ver imagen 2).

Un computador requiere entradas para generar las salidas. Después de aceptar la entrada, el computador la procesa y luego despliega el resultado. Este resultado es llamado la salida.

**Imagen 1**

	Entrada de datos
	Entrada manual
	Entrada por tarjeta perforada

**Imagen 2**



### **SALIDAS:**

Al igual que las entradas, hay diferentes formas de representar la salida. En un Flujograma, se utiliza el símbolo “desplegar”.

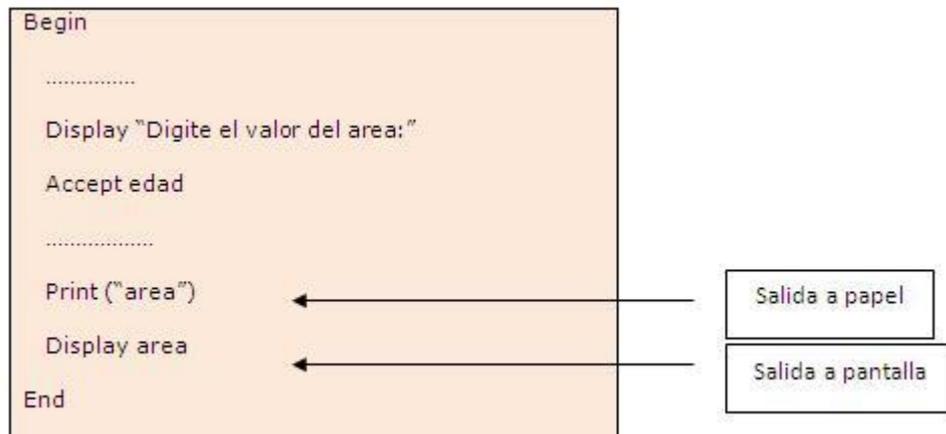
Los símbolos utilizados en flujogramas para representar las salidas se ven en la imagen 3.

En pseudocódigo, la instrucción “desplegar salida” es representada usando salida a pantalla o a papel que se ve en la imagen 4.

**Imagen 3**

	Salida de datos
	Salida a papel (documento)
	Salida a pantalla

**Imagen 4**



### Representación de entradas en algoritmos.

Los comentarios en un algoritmo se usan como una referencia para explicar la lógica del mismo. Es una forma de documentar el proceso, estos no tienen efecto en el código del programa (el computador los ignora), sino que sirven para que el programador comprenda mejor lo que se está representando en el algoritmo. Los comentarios se pueden representar ya sea en una sola línea o en varias líneas.

Para representar comentarios en una sola línea, se utilizan dos barras:

```
//Este es un comentario.
```

Para representar un comentario en varias líneas se utiliza la siguiente simbología:

```
/* Este
```

```
es un
```

comentario

en varias

Líneas \*/

## NOTACIONES PARA EL DISEÑO DE ALGORITMOS

Los algoritmos pueden ser representados en diferentes formas, entre las cuales tenemos:



### Representación en pseudocódigo

PSEUDO: Falso, imitación

CODIGO: instrucciones escritas en un lenguaje de programación

El pseudocódigo no es propiamente un código sino un tipo de descripción que incluye un poco de lenguaje natural y de instrucciones estandarizadas para los lenguajes de programación. Podemos decir que es un lenguaje intermedio entre el lenguaje natural (nuestra forma normal de expresarnos) y cualquier lenguaje de programación específico, como por ejemplo C, Fortran, Pascal, etc.

Es una mezcla de lenguaje natural, símbolos, términos y otras características comúnmente utilizadas en los lenguajes de programación. Además es una técnica para diseño de programas que permite definir las estructuras de datos, las operaciones que se aplicarán a los datos y la lógica que tendrá el programa de computadora para solucionar un determinado problema.

No existe una notación formal o estándar de pseudocódigo, sino que, cada programador puede utilizar la suya propia.

Usted puede detectar errores cuando escribe pseudocódigo porque cada paso del pseudocódigo puede ser revisado durante la creación.

Se concibió para superar las dos principales desventajas de los flujogramas: requiere mucho trabajo para elaborarlo y que no se puede modificar tan fácilmente sin tener que hacer un nuevo redibujo.

### **VENTAJAS DEL PSEUDOCÓDIGO.**

- Es más fácil y rápido de elaborar en comparación con el Flujograma.
- Es más fácil detectar errores y hacer cambios.
- No necesita ser reescrito si se hacen cambios, ya que cada paso es independiente y puede ser modificado sin alterar los otros pasos.
- Es fácil su traducción a cualquier lenguaje de programación. Esto no se logra con los flujogramas y tablas de decisión, ya que el formato usado por el pseudocódigo es similar a un programa. Ambos contienen un grupo de instrucciones secuenciales usando un grupo de instrucciones definidas.
- Es un lenguaje algorítmico similar al español u otro idioma
- Es una imitación de las instrucciones reales para una computadora
- Permite representar en forma fácil operaciones repetitivas complejas.
- Al seguir las reglas se pueden observar claramente los niveles que tiene cada operación.

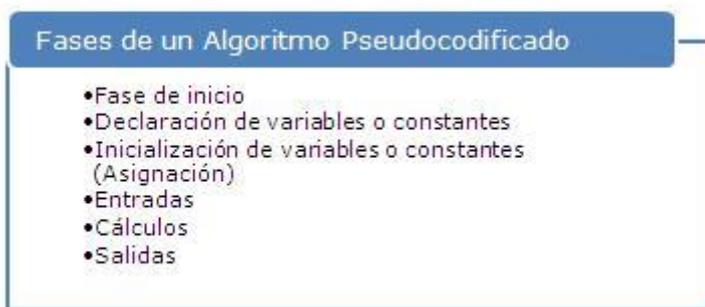
### **DESVENTAJAS DEL PSEUDOCÓDIGO.**

- No provee una representación gráfica del algoritmo, por lo que puede ser difícil entender la lógica compleja de un pseudocódigo.
- Si el pseudocódigo contiene demasiadas condiciones anidadas puede ser difícil de entender.

El pseudocódigo puede ser diseñado en español o en inglés. Cuando se diseña en inglés, se tiene un grupo de instrucciones más parecidas a un lenguaje de programación.

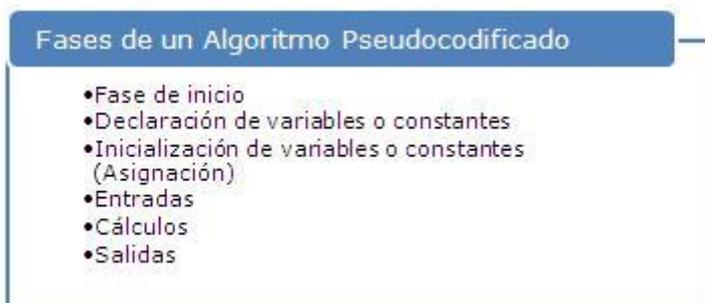
Entre las instrucciones utilizadas en el pseudocódigo inglés tenemos las siguientes:

- Begin .... End/ Start.... Stop. Estas instrucciones son usadas para iniciar y finalizar.
- Accept, read, input: Estas instrucciones son usadas para obtener una entrada de un usuario.
- Display, write, print: Estas son usadas para presentar un resultado o una salida.
- If... else: Son usadas para hacer decisiones.



## EJEMPLO

Elaborar el pseudocódigo para encontrar la superficie de un círculo para un radio cualquiera.



## Diseño de flujogramas

Continuando con el estudio de algoritmos, estudiaremos la representación en Diagramas de Flujo o llamados también Flujogramas.

## FLUJOGRAMAS

Son la representación gráfica de la secuencia de actividades de un proceso en los algoritmos. Consiste en símbolos para representar los pasos de un algoritmo. Cada símbolo tiene un significado que representa una acción a ser seguida, correspondiente a un paso del algoritmo. Cada símbolo se conecta a través de flechas, denominadas líneas de flujo, que indican el orden en que los pasos deben ser ejecutados.

Puesto que un Flujograma es la representación gráfica de un algoritmo, también debe tener: ENTRADA – PROCESO – SALIDA.

Para comprender mejor los diagramas de flujo, se tiene que respetar las reglas de construcción. Al realizar una prueba manual, se debe tomar un conjunto de datos significativos de entrada y comenzar a recorrer el Flujograma de arriba hacia abajo y de izquierda a derecha; según sea la forma representada, para ver cómo se comporta el Flujograma y si los resultados obtenidos son correctos y coherentes.

#### EJEMPLO.

En ejemplos anteriores, diseñamos un pseudocódigo para encontrar la superficie de un círculo para un radio cualquiera.

El Flujograma que representa a dicho ejemplo es el siguiente:

