# **GUÍA INFORMÁTICA II**

**Definición: Algoritmo**

Podemos encontrar muchas definiciones de algoritmo en los textos de programación, todas ellas muy similares:

* Conjunto ordenado y finito de pasos que permite hallar la solución de un problema.
* Una secuencia de pasos que conducen a la realización de una tarea.
* Descripción exacta de la secuencia en que se ha de realizar un conjunto de actividades tendientes a resolver un determinado tipo de problema o procedimiento.
* Conjunto de sentencias / instrucciones en lenguaje nativo, los cuales expresan la lógica de un programa.
* Es un sistema por el cual se llega a una solución, teniendo en cuenta que debe de ser definido, finito y preciso.
* Toda receta, proceso, rutina, método, procedimiento, técnica, formula que resuelven un determinado problema.
* Conjunto de instrucciones concretas y detalladas mediante el cual se consigue una acción determinada.
* Conjunto de reglas que permiten obtener un resultado determinado a partir de ciertas reglas definidas.
* Descripción precisa de una sucesión de instrucciones que permite llevar a cabo un trabajo en un número finito de pasos.
* Un conjunto de símbolos y procedimientos usados en la realización de un cálculo.

Las definiciones más completas o formales:

* Secuencia finita de instrucciones, reglas o pasos que describen de forma precisa las operaciones de un ordenador debe realizar para llevar a cabo una tarea en un tiempo más finito. [Donald E. Knuth, 1968]
* Descripción de un esquema de comportamiento expresado mediante un reportorio finito de acciones y de informaciones elementales, identificadas, bien comprendidas y realizables a priori. Este repertorio se denomina léxico [Pierre Scholl, 1988]
* Un algoritmo es un conjunto finito de pasos definidos, estructurados en el tiempo y formulados con base a un conjunto finito de reglas no ambiguas, que proveen un procedimiento para dar la solución o indicar la falta de esta a un problema en un tiempo determinado. [Rodolfo Quispe-Otazu, 2004]

**Características:**

Las características fundamentales que debe cumplir todo algoritmo son:

* **Ser definido**: Sin ambigüedad, cada paso del algoritmo debe indicar la acción a realizar sin criterios de interpretación.
* **Ser finito**: Un número específico y numerable de pasos debe componer al algoritmo, el cual deberá finalizar al completarlos.
* **Tener cero o más entradas:** Datos son proporcionados a un algoritmo como insumo (o estos son generados de alguna forma) para llevar a cabo las operaciones que comprende.
* **Tener una o más salidas:** Debe siempre devolver un resultado; de nada sirve un algoritmo que hace algo y nunca sabemos que fue. El devolver un resultado no debe ser considerado como únicamente “verlos” en forma impresa o en pantalla, como ocurre con las computadoras. Existen muchos otros mecanismos susceptibles de programación que no cuentan con una salida de resultados de esta forma. Por salida de resultados debe entenderse todo medio o canal por el cual es posible apreciar los efectos de las acciones del algoritmo.
* **Efectividad:** El tiempo y esfuerzo por cada paso realizado debe ser preciso, no usando nada más ni nada menos que aquello que se requiera para y en su ejecución.

Un algoritmo puede ser escrito en lenguaje natural; pero esta descripción puede ser ambigua, por lo que se utilizan diferentes métodos de representación, que permiten evitar dicha ambigüedad y al mismo tiempo que sean fácilmente codificables. *Los métodos más usuales para la representación de algoritmos son:*

1. *Descripción narrada*
2. *Diagrama de flujo*
3. *Pseudocódigo*

**DESCRIPCIÓN NARRADA**

Es la forma más sencilla de describir o expresar un algoritmo. Consiste en hacer un relato de la solución en lenguaje natural. Por ejemplo:

Algoritmo (en descripción narrada) para calcular la suma de 2 números:

1. Obtener los 2 números a sumar.

2. Sumar los números.

3. Mostrar el resultado.

El uso del lenguaje natural provoca frecuentemente que la descripción sea imprecisa y poco confiable, por lo que este tipo de representación no es recomendable.

**DIAGRAMA DE FLUJO**

Es la representación gráfica de un algoritmo. Utiliza símbolos normalizados, con los pasos del algoritmo escritos en el símbolo adecuado y los símbolos unidos por flechas, denominadas “líneas de flujo”, que indican el orden en que los pasos deben ser ejecutados.

**PSEUDOCÓDIGO**

El pseudocódigo en un lenguaje de especificación de algoritmos que utiliza *palabras reservadas* y exige la *incensación*, o sea, sangría en el margen izquierdo de algunas líneas. Se concibió para superar las dos principales desventajas de los diagramas de flujo lento de crear y difícil de modificar sin un nuevo proceso de redibujo. Es una herramienta muy efectiva para el seguimiento de la lógica de un algoritmo y para transformar con facilidad los algoritmos a programas.

En el ámbito de las computadoras, los algoritmos se expresan como programas. Los programas son algoritmos codificados con un lenguaje no ambiguo cuya sintaxis y semántica "entiende" la computadora.

Así pues, si queremos que una computadora efectúe una tarea, primero debemos descubrir un algoritmo para llevarla a cabo; programar el algoritmo en la máquina consiste en representar ese algoritmo de modo que se pueda comunicar a una máquina. En otras palabras, debemos transformar el algoritmo conceptual en un conjunto de instrucciones y representar estas últimas en un lenguaje sin ambigüedad.

Gracias a la capacidad para comunicar nuestros pensamientos mediante algoritmos, podemos construir máquinas cuyo comportamiento simula, en alguna medida, la inteligencia. El nivel de inteligencia que simula la máquina está limitado por la inteligencia que podamos comunicarle por medio de algoritmos. Las máquinas sólo pueden realizar tareas algorítmicas. Si encontramos un algoritmo para dirigir la ejecución de una tarea, podemos construir una máquina para llevarla a cabo siempre que la tecnología haya avanzado lo suficiente. Si no encontramos un algoritmo, es posible que la ejecución esté fuera de las capacidades de las máquinas.

**CARACTERÍSTICAS DE LOS ALGORITMOS:**

Las características fundamentales que debe cumplir todo algoritmo son:

1. Un algoritmo debe ser preciso e indicar el orden de realización de cada paso.
2. Un algoritmo debe estar definido. Si se sigue un algoritmo dos veces, se debe obtener el mismo resultado cada vez.
3. Un algoritmo debe ser finito. Si se sigue un algoritmo se debe terminar en algún momento o sea, debe tener un número finito de pasos.

La estructura de un algoritmo debe estar integrada por tres partes: Entrada, Proceso y Salida.

La entrada se refiere a algo que existe y es utilizado por el algoritmo para transformarlo en los resultados que uno planifica.

**METODOLOGÍA DE SOLUCIÓN DE PROBLEMAS**

Antes de hablar de metodologías de solución de problemas debes tener muy claro el concepto de método, que a lo largo de la historia ha llamado la atención de gran cantidad de filósofos y científicos. El Diccionario Larousse define el término **método** como “el conjunto de operaciones ordenadas con que se pretende obtener un resultado”. La **metodología** es la ciencia que aplica este método. Existen muchos tipos de metodologías, como la metodología de investigación, metodología de enseñanza–aprendizaje, etc., así como la metodología de solución de problemas, que aplicamos constantemente en la vida diaria. Pero en este caso hablaremos específicamente de la solución de problemas que podemos resolver mediante el uso de la computadora.

La resolución de los problemas consta de *5 etapas* que garantizan la llegada correcta a la solución: identificación del problema, planteamiento de alternativas de solución, elección de una alternativa de solución, elección de una alternativa, desarrollo de la solución y evaluación de ésta.

**IDENTIFICACIÓN DEL PROBLEMA**

La identificación del problema es una fase muy importante en la metodología, pues de ella depende el desarrollo posterior en busca de la solución. Un problema bien delimitado es una gran ayuda para que el proceso general avance bien; un problema mal definido provocará desvíos conceptuales que serán difíciles de remediar posteriormente. En esta etapa es fundamental el análisis de la información inicial (entrada) con el fin de distinguir los datos pertinentes de los que no lo son, de manera que se pueda elegir la configuración más

Conveniente respecto a las soluciones posibles. También deben definirse los datos de salida que garanticen la continuidad del proceso para que sea más fácil eliminar las expectativas negativas.

**PLANTEAMIENTO DE ALTERNATIVAS DE SOLUCIÓN**

Después de la definición del problema y del análisis de los datos de entrada, el proceso continúa con el análisis de las alternativas de solución. Por lo general, la solución de un problema puede alcanzarse por distintas vías. Es útil tratar de plantear la mayor cantidad de alternativas posibles de solución, pues de esta forma las probabilidades aumentan a favor de encontrar la vía correcta. Se debe destacar que no es conveniente extender demasiado el número de alternativas, pues si el número de éstas es demasiado alto, se presentará una mayor dificultad para elegir la mejor de todas, que es en definitiva el objetivo del proceso.

**ELECCIÓN DE UNA ALTERNATIVA**

Después de tener todo el repertorio de alternativas, es necesario pasar a otra etapa: la elección de la mejor entre todas las posibilidades. Esta fase es muy importante porque de la elección realizada depende el avance final hasta la solución. La orientación hacia delante supone la irreversibilidad si la decisión es acertada o una “reversibilidad costosa”, pues si la decisión no es acertada, es necesario retroceder, lo que afecta la “optimidad” del proceso.

Por tanto, es necesario que cada alternativa sea bien analizada para que la toma de decisiones sea bien justificada. Deberá elegir la alternativa que sea la más adecuada para la solución de problemas, tomando en cuenta las características del problema y las características que deberá tener la solución, así como los elementos, datos o información con la que cuentas.

**DESARROLLO DE SOLUCIÓN**

Después de decidir cuál es la mejor alternativa de todas, se llega a la etapa de la solución. En esta fase, a partir de los datos relacionados con la alternativa seleccionada, se aplican las operaciones necesarias para solucionar el problema. La selección de los procesos también debe ser determinada en función de la optimidad, es decir, las operaciones deben llegar a la solución por el camino más corto para garantizar la mayor eficiencia en el funcionamiento. Si la alternativa es la óptima, llevará a la solución deseada que fue prevista en la identificación del problema.

**DEFINICIÓN**

Es la representación gráfica de flujo de un algoritmo o de secuencia rutinarias Se basan en la utilización de diversos símbolos 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 la operación.

**SÍMBOLOS UTILIZADOS**

Los símbolos que se utilizan para diseño se someten a una normalización, es decir, se hicieron símbolos casi universales, ya que, en un principio cada usuario podría tener sus propios símbolos para representar sus procesos en forma de Diagrama de flujo. Esto trajo como consecuencia que sólo aquel que conocía sus símbolos, los podía interpretar.

La simbología utilizada para la elaboración de diagramas de flujo es variable y debe ajustarse a las normas preestablecidas universalmente para dichos símbolos o datos. Los siguientes símbolos son los principales:



**DESCRIPCIÓN DE LOS SÍMBOLOS**

1. **Flujo de datos:** Indica la secuencia (flujo) de operación del diagrama, que a su vez indican el sentido y trayectoria.
2. **Inicio o fin:** Se utiliza para señalar el *comienzo* así como el *final* de un diagrama. Tradicionalmente se colocan las palabras "INICIO" o "FIN" dentro de la figura para hacerlo más explícito.
3. **Proceso:** Es el símbolo más comúnmente utilizado. Se usa para representar un evento o proceso determinado que es controlado dentro del diagrama de flujo en que se encuentra, sumar, restar y cualquier operación aritmética es un proceso.

**Decisión**. Se utiliza para representar una pregunta o interrogante que tiene al menos dos respuestas posibles, para cada una de las cuales hay un camino de continuación del proceso. Lo anterior hace que a partir de éste el proceso tenga dos o más caminos posibles.

* 1. **Conector:** Representa un punto de conexión entre procesos. Se utiliza cuando es necesario dividir un diagrama de flujo en varias partes o procesos dentro de la misma hoja, por razones de espacio o simplicidad. Se usa para conectar procesos contenidos dentro de la misma hoja. La mayoría de las veces se utilizan números en los círculos para hacer la referencia.
  2. **Entrada de datos:** Representa una operación de entrada. En este símbolo se indican los valores iníciales que deberá recibir el proceso. Esto se hace asignándoles letras o nombres de variables para cada uno de los valores y anotando estas letras en el interior de la figura.
  3. **Despliegue de resultados:** Este símbolo se utiliza para mostrar un resultado, el cual puede representar la solución al problema que se pretende resolver y que fue conseguida a través del resto del diagrama. Dentro de su interior se anotará la variable con el resultado final o el mensaje que represente el resultado del algoritmo.

**Símbolos gráficos**

1. Dentro de los símbolos fundamentales para la creación de diagramas de flujo, los símbolos gráficos son utilizados específicamente para operaciones aritméticas y relaciones condicionales. La siguiente es una lista de los símbolos más comúnmente utilizados:

|  |  |
| --- | --- |
| + | Sumar |
| - | Menos |
| \* | Multiplicación |
| / | División |
| ± | Mas o menos |
| = | Equivalente a |
| > | Mayor que |
| < | Menor que |
| ³ | Mayor o igual que |
| £ | Menor o igual que |
| ¹ o <> | Diferente de |
|  | Si |
|  | No |
|  | True |
|  | False |

**CARACTERÍSTICAS QUE DEBE CUMPLIR UN DIAGRAMA DE FLUJO**

1. Existe siempre un camino que permite llegar a una solución (finalización del algoritmo).
2. Existe un único inicio del proceso.
3. Existe un único punto de fin para el proceso de flujo (salvo del rombo que indica una comparación con dos caminos posibles).

**DESARROLLO DEL DIAGRAMA DE FLUJO**

***Las siguientes son acciones previas a la realización del diagrama de flujo:***

1. Identificar las ideas principales a ser incluidas en el diagrama de flujo.
2. Definir qué se espera obtener del diagrama de flujo.
3. Identificar quién lo empleará y cómo.
4. Establecer el nivel de detalle requerido.
5. Determinar los límites del proceso a describir.

***Los pasos a seguir para construir el diagrama de flujo son:***

1. Establecer el alcance del proceso a describir. De esta manera quedará fijado el comienzo y el final del diagrama. Frecuentemente el comienzo es la salida del proceso previo y el final la entrada al proceso siguiente.
2. Identificar y listar las principales actividades/subprocesos que están incluidos en el proceso a describir y su orden cronológico.
3. Si el nivel de detalle definido incluye actividades menores, listarlas también.
4. Identificar y listar los puntos de decisión.
5. Construir el diagrama respetando la secuencia cronológica y asignando los correspondientes símbolos.
6. Asignar un título al diagrama y verificar que esté completo y describa con exactitud el proceso elegido.

**VENTAJAS DE LOS DIAGRAMA DE FLUJO**

1. Favorecen la comprensión del proceso a través de mostrarlo como un dibujo. El cerebro humano reconoce fácilmente los dibujos. Un buen diagrama de flujo reemplaza varias páginas de texto.
2. Permiten identificar los problemas y las oportunidades de mejora del proceso. Se identifican los pasos redundantes, los flujos de los re-procesos, los conflictos de autoridad, las responsabilidades, los cuellos de botella, y los puntos de decisión.
3. Muestran las interfaces cliente-proveedor y las transacciones que en ellas se realizan, facilitando a los empleados el análisis de las mismas.
4. Son una excelente herramienta para capacitar a los nuevos empleados y también a los que desarrollan la tarea, cuando se realizan mejoras en el proceso.

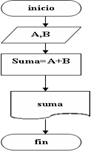
**ESTRUCTURAS ALGORÍTMICAS**

Las estructuras de operación de programas son un grupo de formas de trabajo, que permiten, mediante la manipulación de variables, realizar ciertos procesos específicos que nos lleven a la solución de problemas. Estas estructuras se clasifican de acuerdo con su complejidad en:

**Secuenciales**: Para dar solución a éste tipo de problemas, se emplea una serie de acciones ejecutadas repetidamente en un solo orden secuencial.

**Condicionales**: Para dar solución a éste tipo de problemas, se emplean acciones en las que la ejecución de alguna dependerá que se cumplan una o varias condiciones.

**Cíclicas**: Son aquellas que para dar una solución, es necesario utilizar el mismo conjunto de acciones que puedan ejecutarse más de una vez.

**Ejemplo de algoritmo**

**Algoritmo suma de dos números**

**Entrada: A, B (representan 2 números)**

**Proceso: Suma A+B**

**Salida: Resultado de la Suma**

**Inicio**

**Leer A, B**

**Sumar A+B**

**Salida de resultado Suma**

**Fin**