

Algoritmos y programas

Sitio:

[Agencia de Aprendizaje a lo largo de la Vida](#)

Curso:

Tecnicas de Programación 1° G

Libro:

Algoritmos y programas

Imprimido por:

Sebastian Puche

Día:

martes, 13 de agosto de 2024, 22:18

Tabla de contenidos

1. Algoritmos - Punto de partida

- 1.1. ¿Qué es un algoritmo?
- 1.2. Las características de los algoritmos son:

2. ¿Qué entendemos como programa?

- 2.1. Dos pasos para la construcción de un programa y su ciclo de vida
- 2.2. Codificar un algoritmo

1. Algoritmos - Punto de partida

Seguramente escuchaste hablar muchas la palabra algoritmo... y durante esta materia (¡y toda la carrera!) lo vas a escuchar muchísimas veces más. Pero lo más importante es que los algoritmos formarán parte de tu quehacer diario como desarrollador/a de software.



Tomemos como punto de partida tu propio conocimiento sobre algoritmos. En un papel, la computadora o el celular, escribí lo que vos creés que es un algoritmo.

¿Listo/a? Ahora te invitamos a ver este video:



Volvé a leer lo que anotaste sobre algoritmos antes de ver el video.

- ¿Coincide con la definición del video?
- ¿En qué se parece?
- ¿En qué se diferencia?
- ¿Se modificó tu idea inicial?

No alcanza con preguntarnos qué es un algoritmo... la otra gran cuestión es: ¿para qué sirve?, ¿cuál es su utilidad?

En esta presentación vas a encontrar algunas respuestas:

IMPORTANCIA DE LOS ALGORITMOS EN LA VIDA Y EN LA INFORMATICA

EM De Edward Martinez
18 de Febrero de 2014

Presentar

Ver en **Prezi.com**



Para seguir pensando

- ¿Conocías todos esos usos del algoritmo en la vida cotidiana y en la informática?
- ¿Cuál creés que será su impacto en tu futura profesión?

1.1. ¿Qué es un algoritmo?

El **algoritmo** es un concepto más general que **programa**, que implica un procedimiento o serie de pasos independiente de la computadora y/o del lenguaje de programación en que está expresado.

En términos precisos podemos decir que un algoritmo es un **conjunto finito de reglas que suministra una secuencia ordenada de acciones básicas con el propósito de resolver un tipo específico de problema.**

La programación, entonces, se basa en el uso de algoritmos, es decir, los programas funcionan sobre algoritmos.

Como mencionamos antes, el algoritmo consiste en reglas pero, **¿Qué tipo de reglas sigue un algoritmo?** Te invitamos a descubrirlo con esta presentación:

El algoritmo debe cumplir las siguientes normas.



Debe finalizar luego de una cantidad finita de pasos.



Cada paso debe estar perfectamente definido.



Puede tener o no datos de entrada



Debe tener uno o más datos de salida



Debe ser efectivo (cada uno de sus pasos se deben poder ejecutar con precisión en un tiempo finito)





Recordá: Un algoritmo es un conjunto finito de reglas o pasos, que suministra una secuencia lógica ordenada de acciones con el propósito de resolver un tipo específico de problema.

1.2. Las características de los algoritmos son:



Entonces, un algoritmo:

- Debe finalizar luego de una cantidad finita de pasos.
- Los pasos deben estar perfectamente definidos.
- Pueden tener o no datos de entrada ingresados por el usuario.
- Puede tener o no datos de salida para mostrar al usuario.

Aunque no lo sepamos, en la vida cotidiana estamos ejecutando algoritmos continuamente. Es decir, seguimos un conjunto de instrucciones implícitas que alguna vez aprendimos, y lo hacemos prácticamente de modo automático.

Por ejemplo, cada vez que cocinamos algún plato de pasta, sabemos que debemos seguir una cantidad de instrucciones que ya conocemos, pero que pertenecen a una receta.

Las instrucciones para cocinar un plato de pasta podrían ser:

1. Poner el agua a hervir.
2. Agregar sal al agua.
3. Agregar un poco de aceite.
4. Esperar a que hierva el agua.
5. Agregar los fideos.
6. Esperar a que pasen 10 minutos.
7. Apagar la hornalla.
8. Poner manteca en en la olla.
9. Colar los fideos.
10. Servir los fideos en el plato.

Recordemos que en este caso, el problema era cocinar los fideos. Teniendo en cuenta este ejemplo, podemos ver que si la receta está escrita en español, inglés o italiano, no es importante, ya que el algoritmo plantea la solución del problema sin importar en qué lenguaje o idioma está escrito.



Basta que quien deba ejecutarlo entienda ese lenguaje para poder realizar las acciones indicadas.

2. ¿Qué entendemos como programa?

Escuchamos todos los días que la programación es cada vez más necesaria. Antes, leíste que la programación se basa en algoritmos, es decir, es una unidad mayor que un conjunto de reglas o pasos. Pero es necesario definir mejor el concepto.



¿Qué se entiende como programa?

En el ámbito de la informática, es un conjunto de instrucciones u órdenes capaces de ser obedecidas por una computadora de forma tal que, al ejecutarlas, realice una determinada tarea en un tiempo finito. Todo programa tiene un objetivo bien definido.

Un programa es el resultado de una acción compuesta, que fue desmenuzada en acciones elementales capaces de ser ejecutada por la computadora.

2.1. Dos pasos para la construcción de un programa y su ciclo de vida

Definimos a la programación como una disciplina cuyo objetivo esencial es la resolución de problemas mediante la especificación de los pasos necesarios para llegar a la solución, codificándolos en un lenguaje que pueda ser interpretado por una computadora.

Estos problemas pueden ser resueltos cumpliendo una metodología preestablecida de cuatro etapas:

*Metodología preestablecida
de cuatro etapas*



En su ciclo de vida involucra dos pasos Fundamentales:

Paso 1. Diagramación : este proceso consiste en la construcción de un esquema o dibujo (diagrama) que muestre la secuencia lógica de los pasos a seguir para resolver el problema planteado.

El proceso de diagramación puede descomponerse en 4 fases:

- **Fase 1:** enunciación y comprensión del problema. Definición de las necesidades.
- **Fase 2:** análisis del problema.
- **Fase 3:** obtención de un algoritmo que lo resuelva y su representación en forma de diagrama.
- **Fase 4:** etapa de diseño.



Paso 2. Codificación: consiste en expresar el diagrama utilizando la sintaxis de algún lenguaje de programación, de forma que pueda ser traducido por el programa compilador y posteriormente ejecutado por el procesador.

2.2. Codificar un algoritmo

¿Cómo se escribe o codifica un algoritmo?

Al momento de escribir o codificar un algoritmo, se hace en un lenguaje coloquial con determinadas reglas de acuerdo al lenguaje empleado.

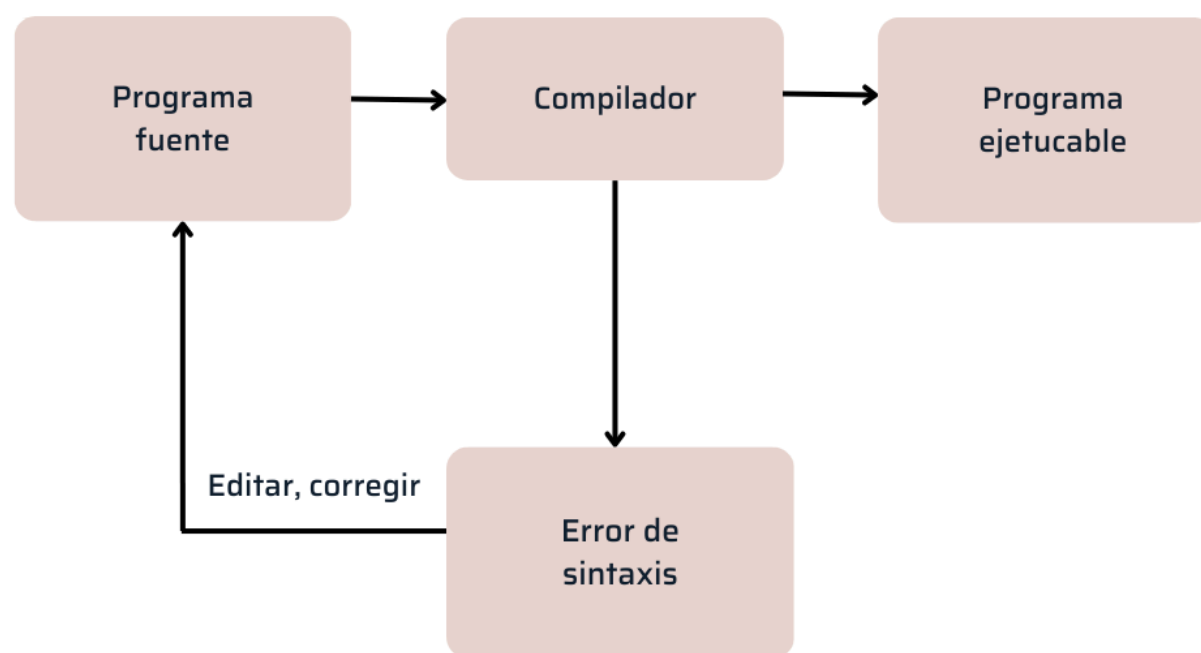
Pero la computadora sólo entiende **un sistema binario**, sólo comprende ceros y unos, con lo cual, **necesitamos un programa llamado compilador** que es capaz de, a partir de nuestro algoritmo generado, chequear que la sintaxis utilizada es la correcta y no presenta errores de escritura y luego lo traduce al lenguaje binario que la máquina pueda comprender y ejecutar.

A la codificación o escritura de nuestro programa la llamamos **programa fuente**.

Una vez escrito nuestro programa fuente, se pasa por el proceso de compilación.

- **Si el compilador** detecta algún error, por ejemplo de sintaxis, volvemos a editar nuestro fuente y volvemos a repetir el paso de la compilación. Cuando no haya más errores, entonces podemos ejecutar nuestro programa y probaremos que cumpla con los objetivos propuestos.

Ejecutar un programa es el proceso mediante el cual una computadora lleva a cabo las instrucciones que le hayamos indicado en nuestro programa fuente.



La tarea que realiza la computadora obedeciendo al programa recibe el nombre de proceso.

La actividad fundamental del/la programador/a es resolver problemas empleando la computadora como herramienta. Para la resolución de un problema hay que plantear un algoritmo.

Los programas realizan básicamente dos tipos de acciones:

- Acciones que actúan sobre los datos.

- Acciones que actúan sobre el programa mismo, determinando el camino que seguirán las acciones dentro del programa.

A esto lo denominamos [flujo de control](#).

Un algoritmo puede ser diseñado correctamente con la combinación de estas dos acciones. Toda acción debe tener un único punto de entrada y un único punto de salida.

Dentro del paradigma de la [programación estructurada](#), cualquier problema puede ser resuelto mediante una combinación de tres estructuras básicas que veremos en detalle más adelante:

1. Secuencia de sentencias
2. Bifurcación condicional simple y múltiple
3. Ciclo o iteración

¿[A qué se le llama sentencia](#)? Se llama *sentencia* a cada una de las líneas de código o instrucción que se le da al programa.

Estas estructuras pueden combinarse entre sí, incluirse unas a otras ya sean del mismo u otro tipo de manera secuencial y anidadas (decisiones y repeticiones) y siempre cumpliendo con la premisa de tener un único punto de entrada y un único punto de salida.