Algoritmos. Definición

Un algoritmo se puede definir como una secuencia de instrucciones que representan un modelo de solución para determinado tipo de problemas. O bien como un conjunto de instrucciones que realizadas en orden conducen a obtener la solución de un problema.

Para realizar un programa es conveniente el diseño o definición previa del algoritmo. El diseño de algoritmos requiere creatividad y conocimientos profundos de la técnica de programación. Luis Joyanes, programador experto y autor de muchos libros acerca de lógica y programación nos dice "en la ciencia de la computación y en la programación, los algoritmos son más importantes que los lenguajes de programación o las computadoras. Un lenguaje de programación es sólo un medio para expresar un algoritmo y una computadora es sólo un procesador para ejecutarlo".

Loa algoritmos son independientes de los lenguajes de programación. En cada problema el algoritmo puede escribirse y luego ejecutarse en un lenguaje diferente de programación. El algoritmo es la infraestructura de cualquier solución, escrita luego en cualquier lenguaje de programación.

Características de los algoritmos

- Preciso. Definirse de manera rigurosa, sin dar lugar a ambigüedades.
- Definido. Si se sigue un algoritmo dos veces, se obtendrá el mismo resultado.
- Finito. Debe terminar en algún momento.
- Puede tener cero o más elementos de entrada.
- Debe producir un resultado. Los datos de salida serán los resultados de efectuar las instrucciones.

Loa algoritmos son independientes de los lenguajes de programación. En cada problema el algoritmo puede escribirse y luego ejecutarse en un lenguaje diferente de programación. El algoritmo es la infraestructura de cualquier solución, escrita luego en cualquier lenguaje de programación.

Características de los algoritmos

- Preciso. Definirse de manera rigurosa, sin dar lugar a ambigüedades.
- Definido. Si se sigue un algoritmo dos veces, se obtendrá el mismo resultado.
- Finito. Debe terminar en algún momento.
- Puede tener cero o más elementos de entrada.
- Debe producir un resultado. Los datos de salida serán los resultados de efectuar las instrucciones.

Se concluye que un algoritmo debe ser suficiente para resolver el problema. Entre dos algoritmos que lleven a un mismo objetivo, siempre será preferible el más corto (se deberá analizar la optimización de tiempos y / o recursos).

Etapas para la solución de un problema por medio del computador

1. Análisis del problema, definición y delimitación (macroalgoritmo). Considerar los datos de entrada, el proceso que debe realizar el computador y los datos de salida.
2. Diseño y desarrollo del algoritmo (se utiliza pseudocódigo, escritura natural del algoritmo, diagramas de flujo, etc.)
3. Prueba de escritorio. Seguimiento manual de los pasos descritos en el algoritmo. Se hace con valores bajos y tiene como fin detectar errores.
4. Codificación. Selección de un lenguaje de programación y digitación del pseudocódigo haciendo uso de la sintaxis y estructura gramatical del lenguaje seleccionado.

5. Compilación o interpretación del programa. El software elegido convierte las instrucciones escritas en el lenguaje a las comprendidas por el computador.
6. Ejecución. El programa es ejecutado por la máquina para llegar a los resultados esperados.
7. Depuración (debug). Operación de detectar, localizar y eliminar errores de mal funcionamiento del programa.
8. Evaluación de resultados. Obtenidos los resultados se los evalúa para verificar si son correctos. (Un programa puede arrojar resultados incorrectos aún cuando su ejecución no muestra erorres).

Algoritmos cualitativos y algoritmos cuantitativos

Un algoritmo es cualitativo cuando en sus pasos o instrucciones no están involucrados cálculos numéricos. Las instrucciones para armar un aeromodelo, para desarrollar una actividad física o encontrar un tesoro, son ejemplos de algoritmos cualitativos.

Trate de diseñar el algoritmo para estos casos

- Tomar mate
- Utilizar una guía telefónica
- Cocinar siguiendo una receta
- Cambiar una llanta de automóvil
- Buscar una palabra en el diccionario

Los algoritmos cuantitativos involucran cálculos numéricos.

Ejemplos:

- Solución de un factorial
- Solución de una ecuación de segundo grado
- Encontrar el mínimo común multiplicador.

Técnicas de representación

Para la representación de un algoritmo, antes de ser convertido a lenguaje de programación, se utilizan algunos métodos de representación escrita, gráfica o matemática. Los métodos más conocidos son:

- Diagramación libre (Diagramas de flujo)
- Diagramas Nassi-Shneiderman
- Pseudocódigo
- Lenguaje natural (español, inglés, etc.)
- Fórmulas matemáticas

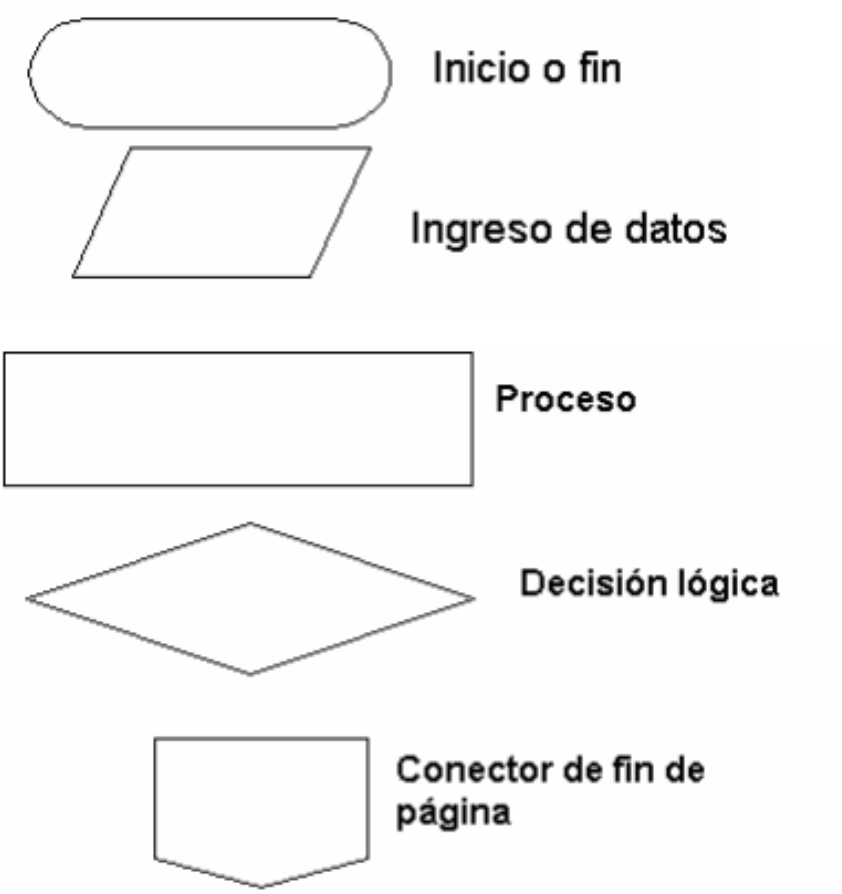
El lenguaje natural puede no ser suficientemente preciso, permitiendo ambigüedades, obteniendo una descripción no del todo satisfactoria. Las fórmulas, propias del lenguaje matemático, son un buen sistema de

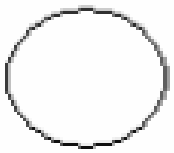
representación, pero no suelen ser fáciles de convertir en programas. Por lo tanto, trataremos en este curso los tres primeros modelos.

Diagramas de flujo.

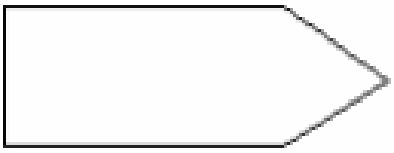
Es quizás la forma de representación más antigua. Algunos autores suelen llamarlos también como diagramas de lógica o flujogramas.

Un diagrama de flujo utiliza cajas estándar tales como las que se muestran en las figuras 1, 2 y 3:

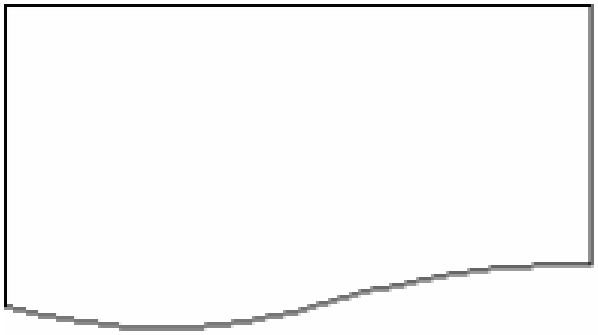




Conector dentro de página



Repetición



Imprimir datos de salida.



Llamada a subproceso o subrutina.

PROGRAMACIÓN ORIENTADA A OBJETOS

La programación Orientada a objetos se define como un paradigma de la programación, una manera de programar específica, donde se organiza el código en unidades denominadas clases, de las cuales se crean objetos que se relacionan entre sí para conseguir los objetivos de las aplicaciones.

La programación Orientada a objetos (POO) es una forma especial de programar, más cercana a como expresaríamos las cosas en la vida real que otros tipos de programación.

Al programar orientado a objetos tenemos que aprender a pensar cómo resolver los problemas de una manera distinta a como se realizaba anteriormente, en la programación estructurada. Ahora tendremos que escribir nuestros programas en términos de clases, objetos, propiedades, métodos y otras cosas que veremos rápidamente para aclarar conceptos y dar una pequeña base que permita soltarnos un poco con los conceptos de este tipo de programación.



Lenguajes de alto nivel para programación y desarrollo de software

LENGUAJE DE ALTO NIVEL JAVA

Qué es Java, uno de los lenguajes de programación más populares, de propósito general y capaz de adaptarse la mayor gama de proyectos y aplicaciones.

Java es un lenguaje de programación con el que podemos realizar cualquier tipo de programa. En la actualidad es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito de Internet como en la informática en general. Fue creado inicialmente por la compañía Sun Microsystems con gran dedicación y siempre enfocado a cubrir las necesidades tecnológicas más punteras. Actualmente se encuentra en propiedad de Oracle, después que ésta adquiriera a Sun.

Una de las principales características por las que Java se ha hecho muy famoso es que es un lenguaje independiente de la plataforma. Eso quiere decir que si hacemos un programa en Java podrá funcionar en cualquier ordenador del mercado. Es una ventaja significativa para los desarrolladores de software, pues antes tenían que hacer un programa para cada sistema operativo, por ejemplo Windows, Linux, Apple, etc. Esto lo consigue porque se ha creado una Máquina virtual de Java para cada plataforma, que hace de puente entre el sistema operativo y el programa de Java y posibilita que este último se entienda perfectamente.

La independencia de plataforma es una de las razones por las que Java es interesante para Internet, ya que muchas personas deben tener acceso con ordenadores distintos. Pero no se queda ahí, Java está desarrollándose incluso para distintos tipos de dispositivos además del ordenador, como móviles, agendas y en general para cualquier cosa que se le ocurra a la industria.

Además, el hecho que Java fuera elegido por Google como lenguaje nativo para la programación de aplicaciones Android, ha provocado una nueva juventud de la tecnología y aún mayor demanda en el ámbito profesional.

ORIENTADO A OBJETOS

La primera característica, orientado a objetos (“OO”), se refiere a un método de programación y al diseño del lenguaje. Aunque hay muchas interpretaciones para OO, una primera idea es diseñar el software de forma que los distintos tipos de datos que usen estén unidos a sus operaciones. Así, los datos y el código (funciones o métodos) se combinan en entidades llamadas objetos. Un objeto puede verse como un paquete que contiene el “comportamiento” (el código) y el “estado” (datos). El principio es separar aquello que cambia de las cosas que permanecen inalterables. Frecuentemente, cambiar una estructura de datos implica un cambio en el código que opera sobre los mismos, o viceversa. Esta separación en objetos coherentes e independientes ofrece una base más estable para el diseño de un sistema software. El objetivo es hacer que grandes proyectos sean fáciles de gestionar y manejar, mejorando como consecuencia su calidad y reduciendo el número de proyectos fallidos.