Nombre del alumno:
Victor Hugo López Moreno

Nombre del profesor:
Andrés Alejandro Reyes Molina

Nombre del trabajo:
Supernota

Materia: Elementos de programación estructurada.

Grado: 4°

Programas y algoritmos

Un programa es un conjunto de pasos lógicos escritos en un lenguaje de programación que nos permite realizar una tarea específica. El programa suele contar con una interfaz de usuario, es decir, un medio visual mediante el cual interactuamos con la aplicación. Algunos ejemplos son la calculadora, el navegador de internet, un teclado en pantalla para el celular, etc.
Hoy encontramos programas o aplicaciones que pueden ejecutarse en una computadora, notebooks, tablets y celulares. Estas aplicaciones pueden ser escritas en diferentes lenguajes de programación. Como ejemplos encontramos C, Java, PHP, Python, entre otros. Estos programas corren sobre un sistema operativo, por ejemplo, Windows, Linux, Mac OS y Android entre otros.

Un algoritmo es en realidad un procedimiento por etapas. Es un conjunto de reglas que hay que seguir para realizar una tarea o resolver un problema.
Mucho antes de la aparición de los ordenadores, los humanos ya utilizaban algoritmos. Las recetas de cocina, las operaciones matemáticas o incluso las instrucciones para montar un mueble pueden considerarse algoritmos.
En el campo de la programación informática, los algoritmos son conjuntos de reglas que indican al ordenador cómo ejecutar una tarea. En realidad, un programa informático es un algoritmo que indica al ordenador qué pasos debe realizar y en qué orden para llevar a cabo una tarea específica. Se escriben utilizando un lenguaje de programación.

¿Qué es un algoritmo?

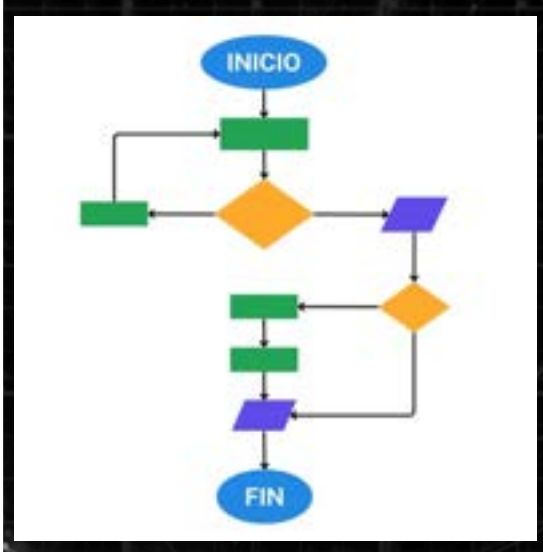
Un algoritmo es en realidad un procedimiento por etapas. Es un conjunto de reglas que hay que seguir para realizar una tarea o resolver un problema.

Tipos de algoritmos

Hay una gran variedad de algoritmos, clasificados según los conceptos que utilizan para realizar una tarea. Estas son las principales categorías.
Los algoritmos “divide y vencerás” permiten dividir un problema en varios subproblemas del mismo tipo. Estos problemas más pequeños se resuelven y sus soluciones se combinan para resolver el problema original.
Los algoritmos de fuerza bruta consisten en probar todas las soluciones posibles hasta encontrar la mejor. Un algoritmo aleatorio utiliza un número aleatorio al menos una vez durante el cálculo para encontrar la solución del problema.
Un algoritmo voraz encuentra la solución óptima localmente, con el objetivo de encontrar una solución óptima para el problema global. Un algoritmo recursivo resuelve la versión más simple de un problema y luego resuelve versiones cada vez más grandes hasta encontrar la solución del problema original.
Un algoritmo de vuelta atrás divide el problema en subproblemas que se pueden intentar resolver uno tras otro. Si no se encuentra la solución, basta con retroceder en el problema hasta encontrar la manera de seguir avanzando.
Por último, un algoritmo de programación dinámica permite descomponer un problema complejo en un conjunto de subproblemas más sencillos. Todos estos subproblemas se resuelven una vez, y su solución se almacena para su uso futuro. Esto evita tener que volver a calcular sus soluciones.

Diagrama de flujo

El diagrama de flujo o flujograma o diagrama de actividades es la representación gráfica de un algoritmo o proceso. Se utiliza en disciplinas como programación, economía, procesos industriales y psicología cognitiva.
En Lenguaje Unificado de Modelado (UML), es un diagrama de actividades que representa los flujos de trabajo paso a paso.
Estos diagramas utilizan símbolos con significados definidos que representan los pasos del algoritmo, y representan el flujo de ejecución mediante flechas que conectan los puntos de inicio y de fin del proceso.



El pseudocódigo

```
Procedimiento Ordenar (L)
    //Comentario: L = (L1, L2, ..., Ln) es una lista con n elementos//
    k ← 0;
    Repetir
        intercambio ← Falso;
        k ← k + 1;
        Para i ← 1 Hasta n − k Con Paso 1 Hacer
            Si Li > Li+1 Entonces
                intercambiar (Li, Li+1)
                intercambio ← Verdadero;
            Fin Si
        Fin Para
    Hasta Que intercambio = Falso;
Fin Procedimiento
```

El pseudocódigo es una forma de representar un algoritmo o programa de manera informal, utilizando un lenguaje natural y elementos similares al lenguaje de programación. Se utiliza para planificar el proceso de programación antes de pasar a la codificación, y tiene las siguientes características:

- No está destinado a ser ejecutado por un ordenador.
-
-
- Es una forma de mostrar el flujo lógico de un programa de manera que sea fácil de entender para cualquier persona.
-
-
- Permite centrarse en el proceso de pensamiento detrás del algoritmo, en lugar de en la sintaxis.
-
-
- Facilita la comunicación entre equipos de diseño y desarrollo.
-
-
- Es una herramienta fundamental para aprender a programar.

Comentarios

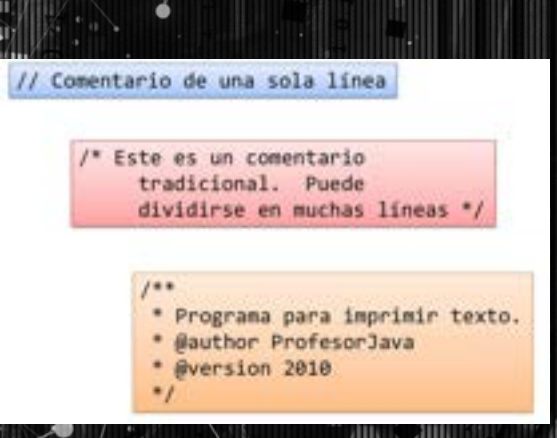
Los comentarios en programación son fragmentos de texto que se añaden al código fuente para facilitar su comprensión, mantenimiento y reutilización. El compilador los ignora y los trata como espacios en blanco.

Los comentarios son una herramienta útil para:

- Explicar el código a otros programadores
-
-
- Recordar por qué se tomaron ciertas decisiones
-
-
- Facilitar la depuración y revisión del código
-
-
- Generar documentación externa
-
-
- Integrar el código con sistemas de control de versiones
-
-

Para que los comentarios sean útiles, es importante que sean precisos, concisos y claros. Un comentario debe transmitir la intención del programador, pero no debe traducir el código a lenguaje natural.

Las reglas y la sintaxis para los comentarios varían según el lenguaje de programación. Por ejemplo, en C++ los comentarios se denotan con //, mientras que en Microsoft C se pueden utilizar comentarios de una línea precedidos por dos barras diagonales (//)

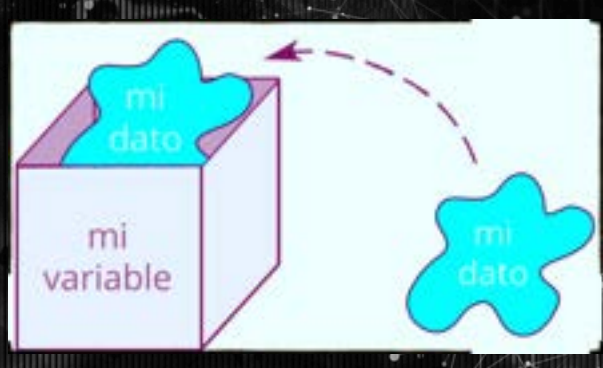


Las variables

En programación, una variable es un espacio en la memoria de la computadora que se utiliza para almacenar valores de distintos tipos. Las variables son elementos de datos con nombre que pueden cambiar de valor durante la ejecución de un programa.

Para usar una variable, se debe declararla y asignarle un valor. El nombre de una variable debe seguir un convenio de denominación que puede incluir letras, números y el signo de subrayado.

Las variables son útiles porque permiten a los programadores referirse a un valor con un nombre en lugar de tener que recordar el valor. Esto facilita tareas complejas, como realizar cálculos, operaciones de entrada y salida, comparaciones, y otros tipos de procesamiento de datos.



Algunos ejemplos de variables son:

- Variables auxiliares
- También conocidas como variables temporales, se utilizan para ejecutar algo de forma temporal.
-
-
- Variables de estado
- Se utilizan para conocer el estado de un objeto en cada momento de la programación.
-
-
- Variables numéricas
- Representan números y se pueden utilizar para realizar operaciones aritméticas.

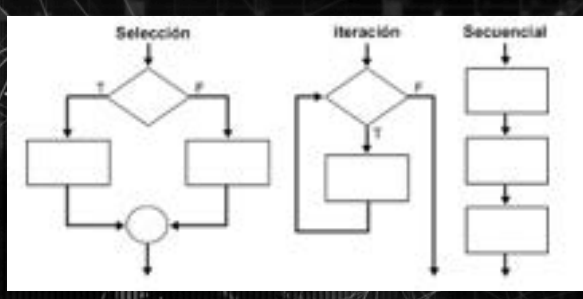
Estructuras de control

Las estructuras de control en programación son reglas que permiten controlar el orden en el que se ejecutan las instrucciones de un programa. Son fundamentales para escribir programas más complejos y sofisticados.

Las estructuras de control permiten:

- Omitir algunas sentencias mientras se ejecutan otras
-
-
- Repetir una o más sentencias mientras alguna condición sea verdadera
-
-
- Determinar la lógica y el orden en que ocurren las operaciones
-
-

La mayoría de los lenguajes de programación actuales utilizan las mismas estructuras de control o son muy parecidas. La diferencia entre ellos es la sintaxis con la que se escriben.



Un ejemplo de estructura de control es la estructura "Según" en PSeInt, que permite tomar decisiones complejas en función del valor de una variable o expresión.

Importancia de la programación de computadoras

La programación de computadoras es importante por varias razones, entre ellas:

- Desarrollo de habilidades
- La programación ayuda a desarrollar habilidades de resolución de problemas, creatividad e innovación.
-
-
- Automatización
- La programación permite automatizar tareas repetitivas, lo que mejora la eficiencia.
-
-
- Oportunidades laborales
- La programación abre oportunidades de empleo en un mundo cada vez más tecnológico.
-
-
- Desarrollo de tecnología
- La programación es fundamental para el funcionamiento de la tecnología y para el desarrollo de nuevas tecnologías.
-
-
- Mejora de la vida de las personas
- La programación permite desarrollar aplicaciones y sistemas que pueden resolver problemas sociales y mejorar la vida de las personas.
-
-
- Impulso a la competitividad
- La programación impulsa la innovación y la competitividad en el mercado global.

Clasificación de los lenguajes de programación

Los lenguajes de programación se pueden clasificar de varias formas, entre ellas:

- Por nivel
- En lenguajes de bajo nivel, los más cercanos a la computadora, y lenguajes de alto nivel, los más alejados. Los lenguajes de alto nivel son más fáciles de aprender y se utilizan para la programación web y de aplicaciones.
-
-
- Por propósito
- En lenguajes generales, que se utilizan para resolver múltiples problemas, y lenguajes específicos, que se utilizan para una tarea concreta.
-
-
- Por paradigma
- En paradigma por procedimientos o imperativo, paradigma declarativo, y paradigma funcional.
-
-
- Por interpretación
- En lenguajes interpretados, que no necesitan ser compilados antes de ejecutarse, y lenguajes compilados.

Lenguaje de programación de bajo nivel

Un lenguaje de programación de bajo nivel es un tipo de lenguaje que se escribe de forma similar a las instrucciones del procesador de una computadora. Se considera un lenguaje de primera generación y se caracteriza por:

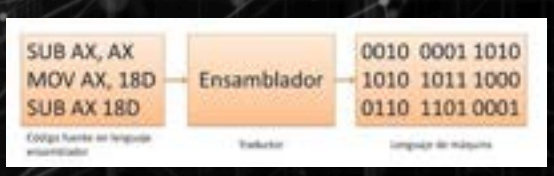
- Ser cercano al hardware y sus componentes
- Tener poca o ninguna abstracción del microprocesador
- Proporcionar un control directo sobre el hardware.
- Estar condicionado por la estructura física de las computadoras

Algunos de los lenguajes de bajo nivel más conocidos son el código máquina y el lenguaje ensamblador.

Los programas escritos en lenguajes de bajo nivel tienen algunas ventajas, como:

- Son rápidos y eficientes en el uso de la memoria
- Son esenciales para comprender y programar dispositivos electrónicos modernos
- Son optimizables para un rendimiento máximo

Sin embargo, también tienen algunas desventajas, como: Son complicados de leer o escribir, Son difíciles de desarrollar, Están demasiado ligados al hardware, Es necesario prestar especial atención para no cometer errores

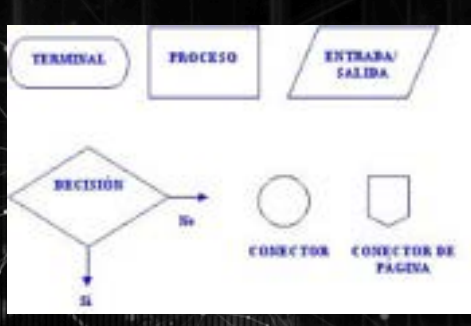


Diseño de algoritmos

El diseño de algoritmos es un método para crear un modelo matemático que resuelva un problema específico en ciencias de la computación. Es un área fundamental en la programación, la ingeniería del software y la investigación de operaciones.

Los algoritmos son un conjunto de instrucciones ordenadas que se siguen para resolver un problema o realizar una tarea. Para diseñar algoritmos, es importante:

- Analizar el problema para comprenderlo y obtener una idea general de lo que se debe hacer.
- Cumplir con requisitos como tener un principio y un fin, ser precisos, tener una secuencia clara y ser repetibles.
-



Para crear un algoritmo, se pueden seguir los siguientes pasos:

1. Definir el problema.
2. Analizar el problema para buscar posibles soluciones.
3. Diseñar el algoritmo, traduciendo la solución a pasos ordenados

El diseño de un algoritmo implica la representación de las instrucciones en orden lógico. Para la representación se utilizan herramientas de diagramación con figuras o con texto. Para la representación de un algoritmo se puede utilizar diagrama de flujo o pseudocódigo.

Referencias

https://www.lucidchart.com/pages/es/que-es-un-diagrama-de-flujo

https://openwebinars.net/blog/que-es-pseudocodigo/

https://ebac.mx/blog/variable-en-programacion

https://www.galileo.edu/pdh/historias-de-exito/por-que-es-importante-aprender-a-programar/#:~:text=A%20trav%C3%A9s%20de%20la%20programaci%C3%B3n,de%20vista%20de%20la%20eficiencia.

https://www.hackaboss.com/blog/tipos-lenguajes-programacion#:~:text=Un%20lenguaje%20de%20programaci%C3%B3n%20de,y%2Fo%20al%20lenguaje%20ensamblador.

https://mdm.usta.edu.co/remos_downloads/oev/logica_de_programacion/temas/algoritmos_concepto.htm#:~:text=El%20dise%C3%B1o%20de%20un%20algoritmo,diagrama%20de%20flujo%20o%20pseudoc%C3%B3digo.