

Introducción a los conceptos fundamentales

Índice

Introducción	1
Algoritmo	2
¿Qué es un Programa?	3
Variables	4
Características de una variable	4
Expresiones	5
Tipos de datos	6
Lenguaje de Programación. Lenguaje de Alto Nivel y Lenguaje de Bajo Nivel.	6
Pseudocódigo	7
Diagrama de flujo	8
Historia de los lenguajes de programación	9
Continuamos con la evolución de los lenguajes de programación	11
Bibliografía	13
Webgrafía	13

Introducción

Un programa es un conjunto de comandos escritos en un lenguaje de programación que se ejecutan en una computadora para realizar una tarea específica. El equipo solo puede procesar elementos binarios, por lo que necesitamos un lenguaje de programación para escribir de manera legible los comandos que la computadora debe ejecutar. Los programas se traducen después a un lenguaje de máquina a través de un compilador.

A continuación veremos algunos conceptos fundamentales a tener en cuenta al momento de realizar nuestra tarea como programadores en el ámbito de desarrollo de software.

Algoritmo

Un algoritmo constituye una lista bien definida, ordenada y finita de operaciones, que permite encontrar la solución a un problema determinado. Dado un estado inicial y una entrada, es a través de pasos sucesivos y bien definidos que se llega a un estado final, en el que se obtiene una solución (si hay varias) o la solución (si es única).

Un algoritmo es un conjunto de reglas definidas que permite solucionar un problema, de una determinada manera, mediante operaciones sistemáticas y finitas. Estas instrucciones, definidas y ordenadas en función de los datos, resuelven el problema o la tarea.

Todo algoritmo debe cumplir con estas características básicas:

- *Tienen inicio y fin*: todo algoritmo comienza en un estado inicial con una serie de datos específicos, y culmina con una solución o salida.
- *Funcionan en secuencia*: un algoritmo está compuesto por una serie de pasos ordenados.
- *Las secuencias son concretas*: cada paso es claro y no deja lugar a la ambigüedad.
- Los algoritmos son abstractos: son modelos o guías para ordenar procesos.
- La cantidad de pasos de un algoritmo es finita.

Un ejemplo de algoritmo en la vida cotidiana podría ser la realización de una recetas de cocina, en la cual se explican el paso a paso para crear una comida con una cantidad finita de ingredientes. El estado inicial serían los ingredientes sin procesar y el estado final la comida preparada.

La descripción de un algoritmo usualmente se realiza en **tres niveles**:

1. **Descripción de alto nivel.** El primer paso consiste en la descripción del problema; luego, se selecciona un modelo matemático y se explica el algoritmo de manera verbal, posiblemente con ilustraciones, pero omitiendo detalles.
2. **Descripción formal.** En este nivel se usa un pseudocódigo o diagrama de flujo para describir la secuencia de pasos que conducen a la solución.
3. **Implementación.** Por último, en este nivel se muestra el algoritmo expresado en un lenguaje de programación específico, o algún objeto capaz de llevar a cabo instrucciones.

¿Qué es un Programa?

Un algoritmo puede representarse en diversos lenguajes de programación, es decir, se trata de un ordenamiento previo a la tarea de programar.

Entonces, un **programa** no es otra cosa que una serie compleja de algoritmos ordenados y codificados mediante un lenguaje de programación para su posterior ejecución.

Un programa informático se define como un conjunto de instrucciones que, una vez ejecutado, realiza una o varias tareas en una computadora.

El conjunto general de programas que posee una computadora se denomina software, término que se utiliza para definir al equipamiento o soporte lógico de una computadora.

Un programa se escribe con instrucciones en un lenguaje de programación, el cual, a su vez, está definido por su sintaxis, que establece e indica las reglas de escritura (la gramática), y por la semántica de los tipos de datos, instrucciones, definiciones, y todos los otros elementos que constituyen un programa.

Algunos ejemplos de lenguajes de programación, son:

- Python.
- Java.
- C++.
- JavaScript.
- Kotlin.
- PHP.
- Ruby.
- Swift.

Si un programa está escrito en un lenguaje de programación comprensible para el ser humano, se le llama código fuente.

A su vez, el código fuente se puede convertir en un archivo ejecutable (código máquina) con la ayuda de un compilador, aunque también puede ser ejecutado de inmediato a través de un intérprete.

Variables

El objetivo general de un programa es transformar datos en resultados útiles para el usuario. Los datos están almacenados en la memoria principal o en la memoria secundaria, ya sea de manera temporal (durante toda la ejecución del programa o durante una parte del tiempo de ejecución) o de manera permanente. En la mayoría de los lenguajes de programación, los datos son de diferentes tipos, aparecen en expresiones o en las llamadas de funciones y se manejan a través del uso de variables.

Una **variable** es un espacio de memoria reservado en la computadora para almacenar un valor. Cada variable tiene un nombre, un valor y un tipo. El valor puede cambiar con el tiempo, y por eso es “variable”. Las variables se utilizan para almacenar datos que se utilizan en el programa. En Python, se pueden crear variables de la siguiente manera:

```
nombre = 'Juan'  
edad = 25
```



El nombre de una variable debe ser único y no ambiguo. La unicidad del nombre de la variable durante su ciclo de vida, asegura una semántica correcta de las operaciones (expresiones, órdenes o proposiciones) que implican a la variable. De esta forma, el nombre de una variable es un identificador diferente de cualquier palabra clave utilizada en el lenguaje o nombre de una función externa.

Características de una variable

Las variables en programación cuentan con tres características principales.

1. Asignación de memoria

Una variable en programación es un lugar en la memoria del ordenador, ya que, cuando este ejecuta un programa o una aplicación, la variable tendrá acceso a este bloque de memoria.

2. Declaración

Cuando hablamos de cómo declarar una variable en programación, nos referimos a asignarle un nombre y un tipo; de esta manera, el sistema le podrá otorgar una abstracción de la memoria para almacenar los datos y su valor.

Asimismo, esto dependerá del tipo de lenguaje de programación que se esté utilizando, ya que la clasificación de la variable antes de usarla es diferente en cada uno. Por ejemplo, en el caso de Javascript y Lenguaje C, sí se necesita declarar; sin embargo, en lenguajes de programación como Python, esto no es necesario.

3. Alcance

Según las características de las variables en programación, por medio del alcance se podrá determinar hasta dónde se puede leer o manipular la información o el valor de una variable.

Por ejemplo, en el caso de las variables globales, se tendrá mayor alcance, pues tienen la capacidad de funcionar a lo largo de todo el programa. Por otro lado, en cuanto a las variables locales, el alcance solo llega a su propia función.

Expresiones

Una expresión es una combinación de valores, variables y operadores que se evalúa como un único valor. Las expresiones se utilizan para realizar cálculos y tomar decisiones en el programa. Por ejemplo:

```
a = 5  
b = 3  
c = a + b
```



En este caso, la expresión $a + b$ se evalúa como 8 y se asigna a la variable c.

Una expresión es la traducción en lenguaje informático de un cálculo aritmético, lógico o de otra naturaleza.

La evaluación de una expresión se hace tomando en cuenta los valores que intervienen y aplicando los operadores. En las expresiones, los operadores tienen un orden de evaluación y prioridades. Una expresión contiene, entonces:

- Valores constantes
- Variables
- Operadores

Tipos de datos

Como mencionamos anteriormente, las variables son contenedores que almacenan datos que pueden cambiar a lo largo del programa. Cada variable tiene un nombre, un valor y un tipo. Los tipos de datos definen el tipo de valor que puede almacenar una variable, en programación es indispensable determinar a qué tipo o categoría corresponden los datos con los que se trabaja. Cada conjunto de datos de un tipo específico se manipula de diferente manera para obtener los resultados deseados.

Como tipos de datos de manera general, podemos nombrar:

Numérico

- Entero. Tipo de dato formado por una variable numérica que no cuenta con parte decimal.
- Real. Tipo de dato formado por una variable numérica que puede contar con parte decimal.

Texto

- Carácter. Tipo de dato formado por una unidad o símbolo que puede ser una letra, un número, una mayúscula o un signo de puntuación.
- Cadena. Tipo de dato formado por un conjunto de caracteres dispuestos de forma consecutiva que se representa entre comillas.

Lógico

- Boolean. Tipo de dato que puede representar dos valores: verdadero o falso.

Lenguaje de Programación. Lenguaje de Alto Nivel y Lenguaje de Bajo Nivel.

Lenguajes de programación se refiere a un conjunto de instrucciones que se utilizan para comunicarse con una computadora y darle órdenes. Estos lenguajes se pueden clasificar en dos tipos principales: lenguajes de bajo nivel y lenguajes de alto nivel.

- Los lenguajes de programación de bajo nivel están más cerca del lenguaje de la máquina y tienen un control directo sobre el hardware. Estos lenguajes no necesitan compiladores y se ejecutan directamente en el procesador. Además, son los tipos de lenguajes más difíciles de entender y aprender.

Se dividen en dos tipos: lenguaje de máquina y ensamblador.

El lenguaje máquina utiliza unos y ceros para dar instrucciones a las máquinas y está compuesto por una serie de instrucciones para indicarle a una máquina qué debe de hacer, cuándo debe realizarlo. Los programas en código máquina están compuestos por una serie de instrucciones que son ejecutadas de forma secuencial. Al ser un lenguaje de bajo nivel está directamente ligado a la máquina por lo cual, cada máquina tendrá un lenguaje máquina diferente.

Los lenguajes de bajo nivel tienen algunas ventajas, como la capacidad de interactuar directamente con el hardware sin necesidad de utilizar compiladores ni intérpretes adicionales, lo que les da una gran velocidad y rendimiento.

- Los lenguajes de programación de alto nivel son más fáciles de entender y aprender, y se utilizan para escribir programas complejos. Estos lenguajes son traducidos al lenguaje máquina mediante un intérprete o compilador. La escritura de código en los lenguajes de alto nivel se aproxima más al lenguaje humano. Ejemplos de lenguajes de programación de alto nivel incluyen Python, JavaScript, PHP y Perl.

Pseudocódigo

Un pseudocódigo está formado por una serie de palabras con un formalismo muy sencillo, que permite describir el funcionamiento de un programa. Este describe un algoritmo utilizando una mezcla de frases en lenguaje común, instrucciones de programación y palabras clave que definen las estructuras básicas. Su objetivo es permitir que el programador se centre en los aspectos lógicos de la solución de un problema.

El pseudocódigo utiliza expresiones matemáticas, expresiones lógicas y la noción de variable (sencilla, arreglo, pila, cola, conjunto, etcétera). El pseudocódigo se puede extender para expresar tipos complejos y operaciones entre variables y constantes de este nuevo tipo.

El pseudocódigo, al ser independiente del lenguaje de programación, permite que su uso se pueda aplicar utilizando diferentes lenguajes y permitiendo que el programador no tenga que ser la misma persona que escribió el pseudocódigo.

Todo algoritmo en pseudocódigo tiene la siguiente estructura general:

```
Proceso SinTitulo
  acción 1;
  acción 2;
  ...
  acción n;
FinProceso
```

Por ejemplo, dado este problema: Leer el radio de un círculo y calcular e imprimir su superficie y su circunferencia.

El pseudocódigo podría ser:

```
//Leer el radio de un círculo y calcular e imprimir su superficie y su circunfe
//Análisis
//Entradas: Radio del círculo (Variable RADIO).
//Salidas: Superficie del círculo (Variable SUPERFICIE) y Circunferencia del c
//Variables: RADIO, SUPERFICIE, PERIMETRO de tipo REAL

Proceso Círculo
  Definir radio,superficie,perimetro como Real;
  Escribir "Introduce el radio de la circunferencia:";
  Leer radio;
  superficie <- PI * radio ^ 2;
  perimetro <- 2 * PI * radio;
  Escribir "La superficie es ",superficie;
  Escribir "El perímetro es ",perimetro;
FinProceso
```

Diagrama de flujo

Un diagrama de flujo es una forma visual la cual nos permite representar un algoritmo o proceso. En un diagrama de flujo se usan diferentes elementos y conexiones las cuales nos permiten representar el algoritmo de una manera visual, estructurada y organizada.

Es ideal, que antes de programar un algoritmo, lo representemos en un diagrama de flujos, diseñando así su lógica.

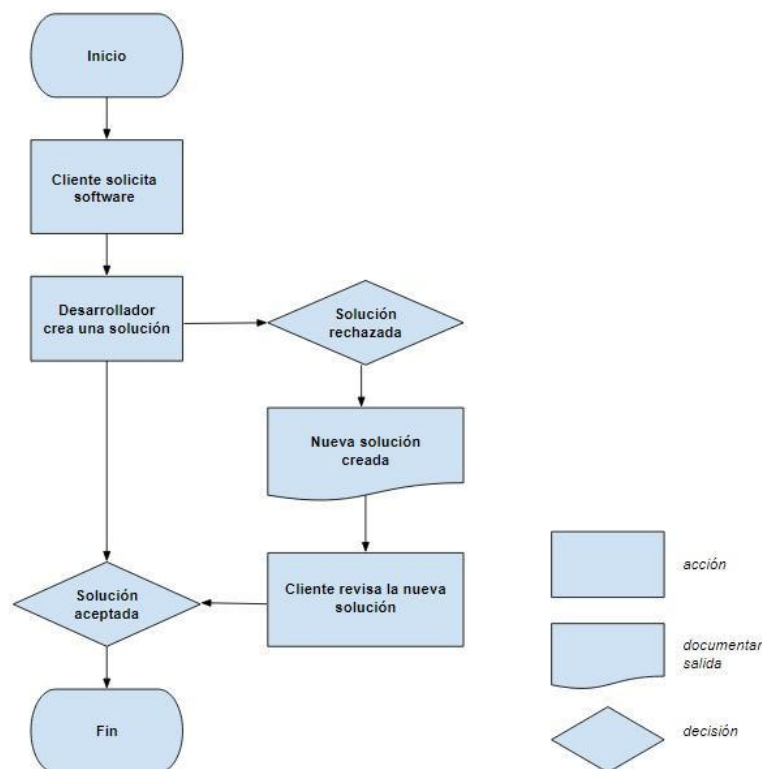
Un diagrama de flujos está formado por diferentes elementos que nos permiten dar forma a la idea del algoritmo:

- Línea o flechas del flujo: Indica la instrucción que se va a realizar, o la dirección del flujo del proceso.

- Inicio y final: Es la forma en la cual se indica el “inicio del proceso” y “final del proceso”. Su icono suele ser un rectángulo con las esquinas redondeadas.
- Asignación/ definición: Permite asignar un valor o definir una variable, son los procesos o actividades que se necesitan para resolver el problema. En este caso, suele estar representado con un rectángulo.
- Datos de entrada o de salida: Representa la impresión de datos en la salida. Se representa con un recuadro con las esquinas inclinadas y una flecha hacia fuera.
- Decisión: Indica que des del punto que nos encontramos, puede haber más de un camino para seguir, según la condición dada. En este caso se usa un rombo.

Sin duda, saber estructurar un diagrama de flujo en el mundo de la programación nos permitirá seguir una lógica estructurar en el proceso de desarrollo.

Ejemplo



Historia de los lenguajes de programaci3n

El primer lenguaje de programaci3n

En 1801, un inventor franc3s, Joseph Marie Jacquard fue el inventor del telar programable. Jacquard cre3 unas tarjetas programables que se introducían en un telar que leía el c3digo, que contenía las instrucciones y, de esta forma, se automatizaba el proceso.

Esto permitió que cualquier persona que tuviera un telar y las tarjetas perforadas, pudieran crear diferentes diseños de tejidos sin ser expertos en ello.

Ada Lovelace & Alan Turing

En 1842, Ada Lovelace, una mujer que había recibido formación exclusivamente en matemáticas, se propuso traducir las memorias del matemático italiano Luigi Menabrea. Mientras realizaba esta labor, creó un lenguaje de programación a mano para calcular la serie de números de Bernoulli con la máquina de Babbage. Escribió el primer algoritmo informático de la Historia de la programación.

Hacemos un salto hasta 1936, en el que Alan Turing, un matemático e informático teórico que revolucionó el mundo de la computación con la Máquina de Turing. Esta, demostraba que, con un algoritmo, podían resolver cualquier problema matemático.

1957: En este año, **John W. Backus**, crea el primer lenguaje de programación de alto niveles y, el primero que usaron programadores más cercanos a lo que conocemos hoy en día: **Fortran**.

Fortran, se trata del lenguaje de programación más antiguo y que continúa en uso hoy en día. Este lenguaje se sigue utilizando en la industria automovilística, gubernamental, aeroespacial y en la investigación. Por ejemplo, resulta muy eficaz para pronosticar el tiempo.

1959: Cobol

Cobol (Common Business Oriented Language) se encuentra detrás de la mayor parte de sistemas de transacciones de empresas, como los procesos de las tarjetas de crédito, telefonía, gobierno, señales de tráfico, entre otros.

Este lenguaje de programación, desarrollado por Grace Murray Hopper, es un lenguaje uniforme para transacciones empresariales. El objetivo de su creación era desarrollar un lenguaje que fuera compatible para cualquier ordenador.

1964: Basic

Desarrollado por alumnos de Dartmouth College, Basic fue diseñado como un lenguaje simplificado para aquellos que no contaban con grandes conocimientos técnicos o matemáticos. Una versión modificada de Basic, elaborada por Bill Gates y Paul Allen, fue el primer producto de Microsoft. En 1977, Basic fue integrado en el Apple II para su puesta en marcha.

1969: C

Este lenguaje de programación se desarrolló entre 1969 y 1973, por Dennis Ritchie para los laboratorios Bell Telephone, con el objetivo de utilizarlos en el sistema Unix. C llegó a ser tan potente que gran parte del núcleo de Unix se reescribió con C.

Continuamos con la evolución de los lenguajes de programación

1970 (Pascal)

Este lenguaje de programación se creó por Niklaus Wirth, que le puso en nombre “Pascal” en honor a Blaise Pascal, el inventor de las calculadoras matemáticas. Este lenguaje se desarrolló como una herramienta de enseñanza y formación, aunque su uso es comercial. Por ejemplo, es el lenguaje de programación que se utiliza en Skype.

1983: C++

En los mismos laboratorios Bells, Bjarne Stroustrup modificó el lenguaje C, por lo que surgió C++. Este lenguaje de programación es uno de los más populares, y se encuentra en el Top Ten de los lenguajes de programación desde 1986. Hoy en día se utiliza en Firefox, Adobe PDF Reader y Office.

1987: Perl

Desarrollado por un programador de Unix, Larry Wall, Perl como solución a las flaquezas de Unix. Actualmente, se utiliza en CraigList.

1991: Python

Guido Van Rossum fue el creador de este lenguaje de programación mundialmente reconocido. Fue creado para resolver lenguajes en el lenguaje ABC y, hoy en día, se sigue utilizando con esta finalidad y también, para más usos. La Nasa, Google y Youtube son algunas de las empresas que lo utilizan.

1993: Ruby

Este lenguaje fue creado por Yukihiro Matsumoto, que utilizó partes de sus lenguajes de programación preferidos: Perl, Smalltalk, Eiffel, Ada y Lisp. Ahora se utiliza por Basecamp.

1995: PHP, Java, JavaScript

Este lenguaje de programación fue desarrollado por Rasmus Lerdoff para reemplazar unos scripts de Perl utilizados para el mantenimiento de su web personal. En la actualidad PHP es parte de una arquitectura web integrada en veinte millones de sitios web. Facebook utiliza este lenguaje hoy en día.

En el mismo año, también se creó Java, por un grupo de trabajadores de Sun Microsystems, con el objetivo de poner en marcha los decodificadores para una televisión interactiva.

Actualmente, Java está presente en más de mil millones de ordenadores en todo el mundo y, muchos sitios webs, no pueden funcionar sin Java.

Por otro lado, se desarrolló JavaScript. Aunque compartan parte del nombre Java y JavaScript no están relacionados, ya que tienen una semántica muy diferente. Este último fue creado por Brendan Eich, de Netscape y, hoy en día, de este lenguaje de programación depende AJAX.

2001: c#

La llegada de los 2000, llegaron nuevos programas de programación, como C# y Visual Basic. Junto a los años noventa, C# culmina como el mejor momento de la historia respecto a los lenguajes de programación. C# es uno de los lenguajes de programación más importantes para crear programas de todo tipo.

2009: Go, de Google

Se trata, nada más y nada menos, que el lenguaje de programación creado por Google. 2009, Go, de Google. Se inspira en C, pero es más complejo. Sigamos descubriendo más lenguajes que forman parte de la evolución de los lenguajes de programación.

2012: KOTLIN

Hoy en día, este es uno de los lenguajes de programación de moda, no porque sea el más sencillo, sino que Google lo ha señalado como el lenguaje de programación óptimo para android.

Aunque hay lenguajes no mencionados, estos son algunos clave que marcaron la historia y la evolución de los lenguajes de programación.

Bibliografía

Mihaela Juganaru Mathie. Introducción a la Programación. Grupo Editorial Patria, 21 de Octubre de 2014. Cap. 1

Webgrafías

<https://concepto.de/algoritmo-en-informatica/#ixzz7wpjmMzgl>

<https://www.significados.com/algoritmo/>

<https://www.crehana.com/blog/transformacion-digital/que-es-variable-programacion/>

<https://concepto.de/dato/#ixzz7xHvqvvp0>

<http://progra.usm.cl/apunte/materia/tipos.html>

<https://www.diarlu.com/lenguajes-de-programacion>

<https://miformacion.eu/blog/lenguaje-bajo-nivel-en-programacion>

<https://www.tecnologia-informatica.com/pseudocodigo/>

<https://www.epitech-it.es/diagrama-de-flujo-programacion/>

<https://kanbantool.com/es/guia-kanban/diagramas-de-flujo>

https://www.ciberaula.com/cursos/java/historia_programacion.php