

2. Programa informático

Índice

- 1. Programa informático
- 2. Elementos de un programa

1. Definición de software

Conjunto de los componentes **lógicos** necesarios que hace posible la realización de **tareas** específicas, en contraposición a los componentes físicos que son llamados **hardware**. La interacción entre el software y el hardware hace operativo un ordenador (u otro dispositivo), es decir, el software envía **instrucciones** que el hardware **ejecuta**, haciendo posible su funcionamiento.

(Wikipedia, consultado en 25 de septiembre de 2023)

El software de **aplicación** es un tipo de software de computadora diseñado para realizar un grupo de funciones, tareas o actividades coordinadas para el beneficio del **usuario**.

(Wikipedia, consultado en 25 de septiembre de 2023)

2. Elementos de un programa

Todo programa está compuesto principalmente por dos tipos de elementos: **datos** (información o estados que reflejan un hecho) y **procesos** (operaciones y acciones que trabajan con los datos, almacenándolos, leyéndolos o modificándolos).

Datos

Los datos se almacenan en **memoria principal** (normalmente la RAM, que es una memoria volátil que depende del suministro eléctrico) o en **memoria secundaria** (el disco duro u otro almacenamiento externo como los CDs, que son medios de almacenamiento persistente).

Un dato se caracteriza por tener un **valor** y una **dirección** en la que se encuentra almacenado. Además, los lenguajes de programación categorizan los datos en los llamados **tipos**, que determinan dos cosas:

1. El **dominio** de posibles valores para ese dato.
2. Las **operaciones** que se pueden realizar con él.

Algunos ejemplos de tipos de datos comunes (llamados normalmente **primitivos**) son: número entero, número real (con decimales), carácter o *string* (cadena de caracteres), lógico (también llamado *boolean* en inglés, que tan solo puede ser de valor verdadero o falso), etc.

Subiendo un nivel de abstracción, los lenguajes suelen tener también definidas **estructuras de datos**, que son formas de componer y ordenar la información según ciertos criterios (como la indexación) para facilitar su manejo.

Dependiendo de si los datos pueden cambiarse o no una vez *declarados* (la forma habitual de referirse a su creación inicial), se les considera como **variables** o **constantes**, respectivamente.

Procesos

Los procesos son las operaciones que se realizan con los datos o cualquier otra acción que ejecute el programa con los recursos de la máquina. Existen operaciones de cálculo aritmético o lógico, de lectura o escritura, etc.

En un programa informático estas operaciones se indican en forma de **instrucciones**, que según la sintaxis definida del lenguaje en el que se esté codificando podrán ser:

- De asignación (dar valor a una variable o constante)
- Cálculo (sumas, restas, etc.)
- Entrada (leer un fichero o una pulsación de teclado o ratón)
- Salida (mostrar algo por pantalla o reproducir un sonido)

Es frecuente nombrar a las instrucciones como **expresiones** cuando se trata de operaciones más o menos complejas que se deben evaluar en un cierto orden para llegar al resultado final.

Todas ellas se construyen a partir de referencias a los datos (e.g., variables o constantes) y **operadores** que expresan el tipo de operación a realizar. Los operadores más comunes son:

- Aritméticos: Suma, resta, multiplicación, división, módulo, etc.
- Relacionales: Igual, distinto, mayor, menor, mayor o igual, menor o igual.
- Lógicos: Conjunción (*and*), disyunción (*or*) y negación (*not*).

Para el caso concreto de estos últimos, los operadores lógicos, que se basan en la lógica matemática, conviene tener presentes las tablas de verdad que definen su resultado:

A	B	A AND B	A OR B	NOT A
F	F	F	F	V
F	V	F	V	V
V	F	F	V	F
V	V	V	V	F

Estructuras de control de flujo

En la mayoría de los casos y aunque hay excepciones (por ejemplo cuando se programa que ocurra una acción en el futuro), un programa ejecuta sus instrucciones de manera secuencial, es decir, siguiendo el mismo orden en el que se escribieron.

Sin embargo, no todo el código escrito en un programa es siempre ejecutado cuando el usuario lo está utilizando, ya que depende de ciertas situaciones, casuísticas, lógica, acciones y decisiones que tomen tanto el usuario como el propio programa.

Para manejar esta multiplicidad de opciones, los programas disponen de la posibilidad de controlar el flujo de ejecución gracias a las estructuras de control, que se pueden clasificar en tres tipos:

- **Selectivas:** También llamados **condicionales** del tipo *SI-ENTONCES* o *SI-ENTONCES-EN CASO CONTRARIO*, evalúan una expresión lógica (generalmente expresada con un operador de

comparación y dos operandos a comparar) de manera que en caso de ser verdadero (*SI*) se ejecuta una rama del condicional (*ENTONCES*) y en caso de ser falso la otra (*EN CASO CONTRARIO*):

```
a = 1
b = 2

SI (a > b) ENTONCES
    "A es mayor que B"
EN CASO CONTRARIO
    "B es mayor que A"
```

- **Iterativos:** También llamados **bucles**, realizan la misma evaluación inicial que los condicionales pero en caso de cumplirse la expresión lógica como verdadera, ejecutarán el código que contienen y volverán a evaluarla hasta que sea falsa.

```
contador = 1

MIENTRAS (contador < 3)
    "Es el turno: " + contador

    contador = contador + 1
```

Debido a este comportamiento y para evitar que se e bucles infinitos se deben tener en cuenta manejar adecuadamente las tres partes de un bucle:

- Estado inicial
 - Condición de parada
 - Modificación del estado
- **De salto:** Permiten modificar el flujo normal definido por las otras dos anteriores, interrumpiendo una iteración de un bucle o el bucle completo.

Referencias

[Software](#)

[Software de aplicación](#)

[Operadores aritméticos](#)

[Operadores relacionales](#)

[Operadores lógicos](#)