

TEMA 1.

PROGRAMACIÓN DE UN COMPUTADOR

1. CONCEPTO DE PROGRAMA
2. LENGUAJES DE PROGRAMACIÓN
 - 2.1. LENGUAJE MÁQUINA
 - 2.2. LENGUAJE ENSAMBLADOR
 - 2.3. LENGUAJE DE ALTO NIVEL
3. ALGORITMOS. REPRESENTACIÓN
 - 3.1. PSEUDOCÓDIGO
 - 3.2. DIAGRAMAS DE FLUJO.
4. EJEMPLOS.

✓ PROGRAMA

- × Para que un computador realice procesos útiles necesita tener almacenadas en memoria instrucciones precisas y exactas que le indiquen las operaciones que debe hacer.
- × Un programa es el conjunto de instrucciones necesarias para resolver un problema en un computador, ordenadas en una secuencia adecuada.
- × El conjunto de programas de un computador se denomina *software*, en contraposición al *hardware*, que se refiere a la parte física.
- × El software de un computador se puede dividir en:
 - software básico (sistema operativo)
 - software de usuario

Lenguaje C

2. Lenguajes de programación

✓ **Los programas de un computador se realizan utilizando *lenguajes de programación*.**

✓ **Evolución de los lenguajes de programación**

- × **Lenguaje máquina**
 - Es el lenguaje usado directamente por el computador y compuesto de instrucciones codificadas en binario.
 - En este lenguaje una instrucción es una cadena de unos y ceros que permite a la unidad de control reconocer una operación elemental y ejecutarla.
 - Viene dado por el fabricante del computador.
- × **Lenguaje ensamblador**
 - Es un lenguaje de bajo nivel (cercano a la máquina) en el que se utilizan nemotécnicos para representar las instrucciones del lenguaje máquina para un computador concreto.
 - Un *ensamblador* es un programa que lee, como datos de entrada, un programa escrito en lenguaje ensamblador y produce, como resultado, un programa en lenguaje máquina.

Fundamentos de Informática

3

ISA-UMH © T-99-026V1.0

Lenguaje C

2. Lenguajes de programación

✓ **Proceso de ensamblaje**

ADD A
 JMP 4
 MOV B
 ...

→

ENSAMBLADOR

→

00101000
 10001001
 11000001
 11100000
 ...

Programa en
ensamblador

Ensamblaje

Programa en código
máquina

Fundamentos de Informática

4

ISA-UMH © T-99-026V1.0

2. Lenguajes de programación

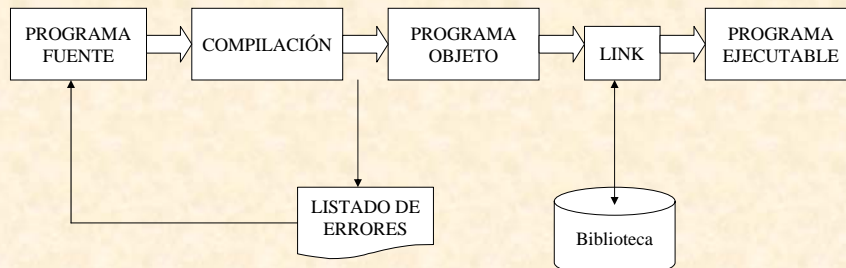
- × El desarrollo de los ensambladores supuso un paso importante, pero los programadores estaban aún forzados a pensar en términos de instrucciones máquina individuales.
- × El lenguaje máquina y el lenguaje ensamblador son “lenguajes orientados a la máquina” y se denominan *lenguajes de bajo nivel*, porque son dependientes de la arquitectura del computador que los soporta.

✓ Lenguajes de alto nivel

- × Son lenguajes de programación más cercanos a los lenguajes naturales, tales como el inglés, y no tan dependientes de las características de los computadores.
- × Ejemplos: C, C++, Pascal, Basic, COBOL, FORTRAN, ...
- × Los programas escritos en estos lenguajes se traducen a instrucciones en lenguaje máquina mediante programas especiales llamados *compiladores e intérpretes*.

2. Lenguajes de programación

- × Un *compilador* analiza el programa, comprobando su sintaxis e indicando los errores, si los hay, y luego genera el programa en lenguaje máquina.
 - *Programa fuente*: es el programa en lenguaje de alto nivel.
 - *Programa objeto*: es el programa en lenguaje máquina generado por el compilador a partir del programa fuente.
 - El programa objeto necesita otro proceso además de la compilación, el enlazado (“linkado”).



- × Un *intérprete* es un programa que analiza y ejecuta un programa sentencia a sentencia.

Lenguaje C

2. Lenguajes de programación

× Si escribimos un programa en un lenguaje de alto nivel, podemos ejecutarlo en cualquier computador que tenga el compilador apropiado.

```

graph LR
    A[PROGRAMA EN C] --> B[COMPILADOR 1  
(en el computador 1)]
    A --> C[COMPILADOR 2  
(en el computador 2)]
    B --> D[PROGRAMA OBJETO 1  
(en el computador 1)]
    C --> E[PROGRAMA OBJETO 2  
(en el computador 2)]
    D --> F[RESULTADO]
    E --> F
  
```

× En las prácticas de Fundamentos de Informática vamos a utilizar un entorno de programación llamado *Dev C++*, que incluye un compilador de C y C++.

Fundamentos de Informática
ISA-UMH © T-99-026V1.0

7

Lenguaje C

3. Algoritmos

✓ Un **algoritmo** es una descripción concisa (no ambigua) y ordenada de la secuencia de un número finito de instrucciones necesarias para la resolución de un problema.

✓ Ejemplos:

- × El procedimiento que utilizamos para ordenar una lista de números.
- × El procedimiento que utilizamos para calcular el factorial de un número.
- × El procedimiento que utilizamos para calcular el m.c.d de dos números.

✓ Para que un computador resuelva un problema, debemos indicarle los pasos sucesivos que debe seguir para resolverlo.

✓ El algoritmo se traducirá posteriormente a un lenguaje de alto nivel y se compilará, obteniendo finalmente el programa objeto que será ejecutado para resolver el problema.

Fundamentos de Informática
ISA-UMH © T-99-026V1.0

8

Lenguaje C

3. Algoritmos

✓ **Representación de un algoritmo**

- × Para diseñar un algoritmo independientemente del lenguaje de programación se suelen utilizar distintas técnicas de representación, entre ellas el *lenguaje algorítmico* o *pseudocódigo* y los *diagramas de flujo* o *flujogramas*.
- × Lenguaje algorítmico o pseudocódigo
 - Se utiliza un vocabulario restringido y unas reglas para construir las frases.
 - Comienzo y fin de algoritmo:


```

          ALGORITMO nombre

          FIN
          
```
 - La secuencia de operaciones se expresa mediante la enumeración en líneas sucesivas.

Fundamentos de Informática
ISA-UMH © T-99-026V1.0

9

Lenguaje C

3. Algoritmos

- Selección

si condición
instrucción1
si no
instrucción2
fsi

si condición
instrucción
fsi

Fundamentos de Informática
ISA-UMH © T-99-026V1.0

10

× Repeticiones o bucles

(1) CON CONDICIÓN INICIAL (2) CON CONDICIÓN FINAL

mientras condición
instrucción
fmientras

hacer
instrucción
mientras condición

(3) CON CONTADOR

para valor inicial **hasta** valor final
instrucción
fpara

× En todas las estructuras de control descritas, la palabra “instrucción” hace referencia a una o varias sentencias.

✓ Diagramas de flujo

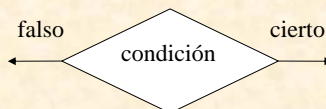
× Símbolos principales del tratamiento



PROCESO
operaciones
de tratamiento



operación de
entrada / salida

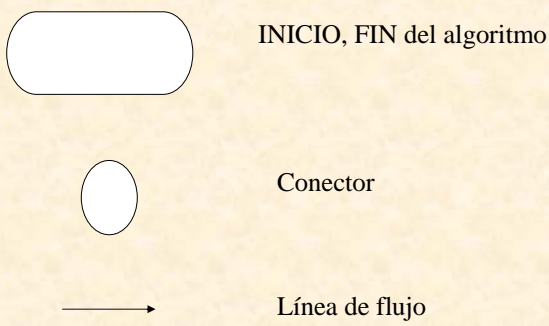


COMPARACIÓN
toma de decisiones

Lenguaje C

3. Algoritmos

× Símbolos de carácter secundario y de enlace



INICIO, FIN del algoritmo

Conector

Línea de flujo

Fundamentos de Informática

ISA-UMH © T-99-026V1.0

13

Lenguaje C

4. Ejemplos de algoritmos

✓ **Ejemplo 1: Escribir un algoritmo que, dado un número real, obtenga su valor absoluto**

- × Para almacenar el número necesitamos lo que en programación se denomina una variable, que en este caso será de tipo real.
- × También necesitamos otra variable para almacenar el resultado, que también será de tipo real.
- × El algoritmo en pseudocódigo puede ser:

```
ALGORITMO absoluto
variables reales x, xAbs
leer x
si (x >= 0)
    xAbs = x
si no
    xAbs = -x
fsi
escribir xAbs
FIN
```

Fundamentos de Informática

ISA-UMH © T-99-026V1.0

14

* Otra versión de este algoritmo que utiliza una variable:

```
ALGORITMO absolutoBis  
  
  variables reales x  
  
  leer x  
  si (x < 0)  
    x = -x  
  fsi  
  escribir x  
  
FIN
```

✓ Diagrama de flujo de la segunda versión:

