

Introducción a la Programación

Grado en Ingeniería Informática

Teoría Contenido 1 - Curso 2015-2016

Contenido 1 - Introducción

Contenido del Tema

- 1.1.- Conceptos básicos y definiciones
- 1.2.- Programación de Ordenadores y Lenguajes de Programación
 - 1.2.1.- La Programación de ordenadores
 - 1.2.2.- Objetivos de la programación
 - 1.2.3.- Lenguajes de programación
- 1.3.- Traductores: Compiladores e Intérpretes
- 1.4.- Paradigmas de Programación

1.1 Conceptos básicos y definiciones

- **INFORMÁTICA** : INFORmación y autoMÁTICA.

Conjunto de conocimientos científicos y técnicas que hacen posible el tratamiento automático de la información por medio de ordenadores.

- **ORDENADOR**

Máquina capaz de aceptar unos datos de entrada, efectuar con ellos operaciones lógicas y aritméticas y proporcionar la información resultante a través de un medio de salida; sin intervención de un operador humano y bajo el control de un programa de instrucciones previamente almacenado

1.1 Conceptos básicos y definiciones

- **DATO** (Entrada/Salida)

Conjuntos de símbolos utilizados para expresar o representar un valor numérico, un hecho, un objeto o una idea; en la forma adecuada para ser objeto de tratamiento.

- **CODIFICACION**

Transformación que representa los elementos de un conjunto mediante los de otro, de forma que a cada elemento del primer conjunto le corresponda un elemento distinto del segundo.
(CODIGO BINARIO: 0 y 1)

1.1 Conceptos básicos y definiciones

- **BIT** : Unidad elemental de información (0 y 1)
- **BYTE** : Número de bits necesarios para almacenar un carácter (generalmente 8).

1 KB = 2^{10} bytes = 1024 bytes $\cong 10^3$ bytes

1 MG = 2^{10} KBytes = 1024 bytes $\cong 10^6$ bytes

1 GB = 2^{10} MBytes = 1024 bytes $\cong 10^9$ bytes

1 TB = 2^{10} GBytes = 1024 bytes $\cong 10^{12}$ bytes

- **HARDWARE**

Soporte físico de un ordenador, conjunto de circuitos electrónicos, cables, carcasas y otros elementos físicos.

- **SOFTWARE**

Conjunto de programas ejecutables por el ordenador. (Siendo un programa un conjunto ordenado de instrucciones)

1.2 Programación de ordenadores y Lenguajes de Programación

1.2.1.- La Programación de Ordenadores

Conceptos:

- **ALGORITMO:** descripción de los pasos necesarios para resolver un problema.
- **PROGRAMA:** algoritmo escrito en un lenguaje de programación.
- **PROGRAMACIÓN:** Tarea de desarrollo de programas
 - Pequeña escala: Programación
 - Gran escala: Ingeniería del Software

1.2 Programación de ordenadores y Lenguajes de Programación

1.2.2.- Objetivos de la Programación

- **Corrección:** Antes de desarrollar un programa debe especificarse con toda claridad cuál es el funcionamiento correcto del mismo.
- **Claridad:** Es fundamental que sus descripciones sean claras y fácilmente legibles.
- **Eficiencia:** una tarea de tratamiento de información puede ser programada de muy diferentes maneras. Los programas eficientes aprovecharán mejor los recursos disponibles.

Debe emplearse una metodología de programación apropiada que satisfaga estos objetivos

1.2 Programación de ordenadores y Lenguajes de Programación

1.2.3.- Lenguajes de Programación

- Un lenguaje de programación es un lenguaje que nos permite comunicarnos con un ordenador.
- Un lenguaje de programación es una notación para describir algoritmos y estructuras de datos.
- Un lenguaje de programación es una convención para escribir descripciones que pueden ser evaluadas.

1.2 Programación de ordenadores y Lenguajes de Programación

1.2.3.- Lenguajes de Programación

Clasificación de los lenguajes de Programación

- Con respecto a su nivel:
 - Lenguajes máquina
 - Lenguajes de bajo nivel o ensambladores
 - Lenguajes de alto nivel

1.2 Programación de ordenadores y Lenguajes de Programación

1.2.3.- Lenguajes de Programación

Clasificación de los lenguajes de Programación

- **Lenguajes máquina:**
- Las instrucciones son cadenas de ceros y unos. (código intermedio: octal o hexadecimal).
- Los datos se utilizan por medio de las direcciones de memoria donde se encuentran.
- Las instrucciones realizan operaciones muy simples.
- Existe muy poca versatilidad para la redacción de las instrucciones.
- El lenguaje depende y está ligado íntimamente a la CPU de la computadora.
- No pueden incluirse comentarios que faciliten la legibilidad del código.

1.2 Programación de ordenadores y Lenguajes de Programación

1.2.3.- Lenguajes de Programación

Clasificación de los lenguajes de Programación

- Lenguajes Ensambladores
- Instrucciones con códigos de operación en notación simbólica o nemotécnica, en vez de códigos numéricos.
- Direcciones simbólicas de memoria, en lugar de direcciones binarias absolutas.
- Permiten incluir líneas de comentarios entre líneas de instrucciones.

1.2 Programación de ordenadores y Lenguajes de Programación

1.2.3.- Lenguajes de Programación

Clasificación de los lenguajes de Programación

- Lenguajes de alto nivel
- Son lenguajes simbólicos orientados al problema y no a la máquina en la que se ejecutan.
- Operaciones expresadas en caracteres numéricos, alfanuméricos y especiales
- Es posible definir gran número de variables y elegir el identificador deseado
- Instrucciones potentes y funciones diversas
- Versatilidad en la definición de Instrucciones
- Permiten incluir comentarios
- Independencia de la CPU

1.3.- Traductores. Compiladores e Intérpretes

- Cualquier programa escrito en un lenguaje distinto del lenguaje máquina debe ser traducido para poder ser ejecutado en un ordenador.

TRADUCTORES:

- Compiladores
- Intérpretes

1.3.- Traductores. Compiladores e Intérpretes

- **COMPILADORES**

Programa fuente → Programa objeto

- Etapas de la Compilación:

Análisis lexicográfico

Análisis sintáctico

Análisis semántico

Generación de código intermedio

Optimización

Generación de código

1.3.- Traductores. Compiladores e Intérpretes

- **INTÉRPRETES**

- Ejecutan sentencia a sentencia tantas veces como estas se repitan
- Optimización no general
- Se traduce cada vez que se ejecuta

1.3.- Traductores. Compiladores e Intérpretes

- **COMPARACIÓN**

- Se usa un intérprete cuando el número de veces que va a ejecutarse es bajo y no hay problemas de velocidad
- Un programa bajo un intérprete puede ser interrumpido en cualquier momento y ser depurado. Continuará ejecutándose por la siguiente instrucción.
- Los compiladores suelen incorporar herramientas de depuración para facilitar la localización de errores.
- Los compiladores deben traducir nuevamente todo el programa cada vez que se corrige un error.
- Los lenguajes compilados consiguen más eficiencia ya que optimizan a nivel global
- Los intérpretes resultan más pedagógicos para aprender a programar.

1.4.- Paradigmas de Programación

- Imperativo: “Cómo”
- Declarativo: “Qué”

