Introducción a la Programación. Conceptos

1.- PROGRAMA

1.1.- Algoritmos

2.- LENGUAJES

2.1.- Lenguajes de bajo nivel

2.1.1.- Código máquina

2.1.2.- Lenguaje ensamblador

2.2.- Lenguajes de alto nivel

2.2.1.- Lenguajes tradicionales

2.2.2.- Programación orientada a objetos y Programación Visual

2.2.3. Lenguajes orientados a la construcción de páginas Web

3.- ESTILOS DE PROGRAMACIÓN

3.1.- Programación estructurada

3.2.- Programación modular

3.3.- Programación Visual

4.- EJECUCIÓN DEL PROGRAMA. Código fuente vs Archivo ejecutable. Compiladores vs Intérpretes

1.- PROGRAMA

Un programa es un conjunto de instrucciones, escritas en un lenguaje determinado, que se ejecutan en un cierto orden, con el objetivo de realizar una tarea

Recordamos que, en un PC, es el microprocesador la parte encargada de leer las instrucciones de la memoria y de ejecutarlas.

1.1.- Algoritmo

Un algoritmo es simplemente un conjunto de instrucciones genéricas, que se ejecutan en un cierto orden, con el objetivo de realizar una tarea.

Para hacer un programa, lo primero es pensar en el algoritmo (es decir, cómo haré para resolver el problema que me han encomendado).

Una vez que tenga el algoritmo y por tanto sepa cómo resolver el problema, traduciré las instrucciones al lenguaje de programación que sea. Observa que el mismo algoritmo me sirve para cualquier lenguaje de programación.

LO DIFÍCIL NO ES ESCRIBIR INSTRUCCIONES EN UN LENGUAJE DETERMINADO, SINO EL ALGORITMO

2.- LENGUAJES

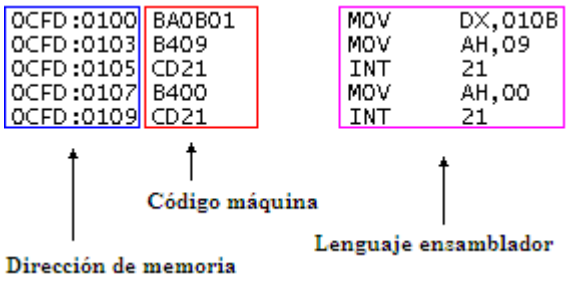
2.1.- Lenguajes de bajo nivel

2.1.1.- Lenguaje máquina o Código máquina

- Las instrucciones son números que se corresponden con operaciones que realizará el microprocesador
- Está claro que un programa escrito en código máquina es dificilísimo de leer (es todo números)
- Sólo es entendible por el microprocesador para el que fue hecho y hay que conocer detalladamente la arquitectura del microprocesador

2.1.2.- Lenguaje ensamblador

- Son las mismas instrucciones que el código máquina pero expresadas mediante palabras (mnemónicos) más fáciles de recordar
- Las herramientas para programas en ensamblador permiten usar etiquetas, constantes, macros, comentarios, lo que facilita la programación
- Hará falta un programa ensamblador que traduzca nuestro programa escrito en ensamblador a código máquina (también existen programas desensambladores que traducen de código máquina a lenguaje ensamblador



2.1.3.- Resumen de los lenguajes de bajo nivel

- Su uso actual es principalmente para la manipulación directa de hardware o si se precisa un código supereficiente
- Recordamos que está hecho para un microprocesador determinado

2.2.- Lenguajes de alto nivel

Su principal característica es que las instrucciones son genéricas y mucho más fáciles de comprender.

Como las instrucciones son genéricas, un programa escrito en un lenguaje de alto nivel podría servir (en principio) para cualquier ordenador con cualquier microprocesador (claro, previamente a ejecutar el programa habrá que traducirlo a código máquina)

Al traducir a código máquina el código que generan no es tan eficiente como si se hiciera en lenguaje ensamblador, pero como los microprocesador son cada vez más rápidos y los discos más grandes esto ya da igual

2.2.1.- Lenguajes tradicionales

- Pascal: Ejemplo representativo de lenguaje estructurado (ya veremos qué es esto)
- C: Es un lenguaje que produce un código máquina bastante eficiente y directo. Se ha hecho famoso porque con él se escribió UNIX. Evolucionó a C++ (C orientado a objetos) y a C# (versión de C++ de Microsoft con la que se escribe Windows)
- BASIC: (Beginner All-purpose Symbolic Instruction Code). Lenguaje sencillo que apareció para facilitar la programación cuando aparecieron los primeros ordenadores personales y se hizo muy popular. Era sencillo pero tuvo mala fama entre los programadores profesionales porque empezó numerando las instrucciones y utilizando muchos saltos con GOTO <número-de-linea>. Luego evolucionó estructurándose y pasó a Visual Basic y a Visual Basic .NET
- COBOL (COmmon Bussines Oriented Languaje): Lenguaje orientado a los negocios. Todavía se usa en grandes servidores de empresas y bancos. Especializado en el proceso de gran cantidad de datos
- LOGO: Empleado con fines didácticos. Es famoso por hacer programas para hacer dibujo con una tortuga.
- FORTRAN: (Formule Translation): Orientado al cálculo numérico
- LISP: LISt Processing y PROLOG (PROgramación LOGica): Orientados a la inteligencia artificial
- ADA: Para sistemas en tiempo real, concurrentes y fiables (en los que hay no puede haber fallos y las respuestas garantizadas en un tiempo)
- Lenguaje de script: No es un lenguaje, sino que son una serie de comandos agrupados en un archivo de texto y que contienen varias ordenes de la consola del sistema operativo. Pueden ser de Linux o de Windows(a través de la consola de MS-DOS emulada por el programa cmd.exe). En el caso de Windows forman los archivos con extensión BAT

2.2.2.- Programación orientada a objetos y Programación Visual

Aquí se trabaja con objetos. Ya estudiaremos en su momento detalladamente la Programación Orientada a Objetos.

Como ejemplo: Un botón es un objeto, tiene propiedades tales como Color, tamaño, texto, dibujo, … A este botón le puede ocurrir un evento (alguien hace click en él, pulsa ENTER cuando está marcado, …).También tiene métodos, es decir, cosas o código que puede mandan ejecutar el botón.

Como puedes ver, la programación orientada a objetos se puede aplicar muy bien a la programación visual, en la cual el programador va colocando sobre un FORM diversos elementos tales como botones, cajas de texto, botones de selección,, menús, y posteriormente escribe el código de respuesta a acciones (eventos) que haga el usuario del programa (¿Qué hará el programa si el usuario pulsa el botón_1?¿Qué hará si pulsa el botón_2 y en la caja_1 hay escrito un cero?...)

Hoy en día la Programación es *Orientada a Objetos* y *Conducida por Eventos*

Ejemplos:
- Delphi (Visual Pascal)
- Visual Basic, Visual Basic .NET
- C++, C#, Visual C++, Visual C#

2.2.3. Lenguajes orientados a la construcción de páginas Web

- HTML: Hiper Text Markup Languaje. Su filosofía es mostrar texto, imágenes y enlaces, formateando todo mediante etiquetas.

- Javascript: Es un lenguaje de programación interpretado que va incrustado dentro del código HTML de las páginas web y que lo ejecuta el navegador del cliente

- ASP (Active Server Pages): Es una tecnología de Microsoft para la realización de páginas web dinámicas (que se construyen en el PC-servidor-Web) de forma que al cliente sólo le llega la página web con la respuesta. Es usado frecuentemente para el acceso a bases de datosa través de internet
  Ejemplo: Pregunto a través de una web que pueblos hay a menos de 50km de donde estoy, y el servidor hace una página web con la respuesta

- PHP: Podríamos decir que es como el ASP pero libre.

- JSP: Como ASP y PHP pero realizado con tecnología Java

- Java: Lenguaje orientado a objetos cuyo código 'semicompilado' en ByteCode. Este código semicompilado es leído por una máquina virtual de java que es la que realmente ejecuta el código La idea es que genero un único código ByteCode que puede ser ejecutado en CUALQUIER MÁQUINA ya que aunque sean máquinas distintas cada una tendrá una máquina virtual de java propia que será la que ejecute el código. Es libre. Suelen ser pequeños programas llamados Applets

- CGI: Es una tecnología que permite a un cliente web solicitar datos a un programa ejecutado en un servidor web. Ese programa puede estar en cualquier lenguaje, pero suelen usarse lenguajes de script y el lenguaje Perl (éste permite una muy buena manipulación de textos y cadenas de caracteres y búsquedas)

2.2.4.- Otros

- SQL: Structured Query Languaje: Es un lenguaje usado para consultar y manipular bases de datos

3.- ESTILOS DE PROGRAMACIÓN

3.1.- Programación estructurada: Estilo de programación en la que sólo se usan 3 estructuras:

- Secuenciación: Cada instrucción se ejecuta en orden, de arriba abajo, no hay saltos

- Selección: Solo se ejecuta la instrucción si se cumple una condición.

- Iteración: Repetición de unas instrucciones mientras se cumplan una condición

Este estilo de programación crea programas fáciles de entender y seguir.

3.2.- Se debe combinar con la programación modular (división del programa en partes más pequeñas y reutilizables

Relacionado con la programación modular está el uso de Bibliotecas, que son subprogramas y funciones ya escritos por el usuario en otra ocasión, por otro usuarios o por los fabricantes de software.

3.3.- Actualmente los programas son todos gráficos, por lo que se desarrolló el estilo de programación visual, en que los fabricantes de software crean un IDE (Entorno de Desarrollo Integrado), que consta de:

- Un editor de textos
- Un compilador
- Un interprete
- Un depurador (debugger) para localizar y eliminar errores de programación
- Herramientas para facilitar la creación de programas gráficos

4.- EJECUCIÓN DEL PROGRAMA. Código fuente vs Archivo ejecutable. Compiladores vs Intérpretes

En sí, el conjunto de instrucciones en un lenguaje determinado constituye un archivo de sólo texto llamado Código Fuente. Pero esto es texto no entendible por el microprocesador, que es el que ejecuta las instrucciones en el lenguaje máquina. Para convertir que el programa se ejecute tenemos dos opciones:

- Utilizar un Compilador, que traduce de una sola vez todas las instrucciones del Código Fuente creando un Archivo Ejecutable (archivo binario con instrucciones en código máquina, usualmente con extensión .EXE). Posteriormente lo que se ejecutará es ese archivo ejecutable.

- Utilizar un Intérprete, que lee una instrucción del Código Fuente, la traduce a Código Máquina y la ejecuta; luego hace lo mismo con la siguiente instrucción del Código Fuente; y así todas las instrucciones