

U.D.1

Conceptos Básicos de Programación

(1ª parte)

PROGRAMACIÓN

ÍNDICE

1. Introducción.
2. Lenguajes de programación. Tipos.
3. Fases en la elaboración de un programa.
4. Traductores de Lenguajes.
5. Entornos integrados de desarrollo (IDE).

1. INTRODUCCIÓN

- Objetivo: Dirigir el comportamiento de un sistema informático.
- Programa: Conjunto de instrucciones que dirige el comportamiento del sistema.
- Periféricos intervinientes:
 - De entrada (captura de datos).
 - De salida (exposición de datos).
 - De entrada/salida.
 - De almacenamiento (permanente).

2. LENGUAJES DE PROGRAMACIÓN. TIPOS

- Herramienta para “entendernos” con el sistema, para comunicarnos con él, para darle órdenes y que las entienda: gramática y sintaxis (como cualquier lenguaje “natural”).
- Tipos:
 - Lenguajes de Bajo nivel:
 - Más próximos a la arquitectura de máquina.
 - Lenguaje máquina.
 - Lenguaje ensamblador.
 - Lenguajes de Alto nivel:
 - Más próximos al usuario o programador.

CARACTERÍSTICAS APLICABLES A LOS LENGUAJES DE PROGRAMACIÓN

- Universalidad (portabilidad).
- Detallabilidad (nº de pasos que hay que dar para ejecutar alguna acción).
- Generalidad (tipo de problemas que puede resolver).
- Usabilidad (facilidad de uso).

LENGUAJE MÁQUINA

- Combinación de ceros y unos (00011001001).
- Único inteligible directamente por ordenador.
- Propio del procesador.
- No es propiamente un lenguaje de programación.

LENGUAJE ENSAMBLADOR

Se basa en el uso de abreviaturas

Inconvenientes:

- Muy ligado a la máquina.
- No es transportable.
- Programas largos, difíciles de escribir y depurar.

Ventajas:

- Programas más rápidos.
- Ocupan menos espacio en memoria.
- Para desarrollo de programas de control de periféricos.

LENGUAJES DE ALTO NIVEL

- Más próximos al usuario o programador.
- Mayor nivel de abstracción.
- Independientes de arquitectura del ordenador.
- Facilidad para desarrollo, depuración y mantenimiento del programa.
- Necesidad de un traductor.

CARACTERÍSTICAS DE LOS LENGUAJES DE ALTO NIVEL

- Las instrucciones se expresan por medio de caracteres.
- Se pueden definir variables para recoger los datos a tratar.
- Sintaxis más flexible.
- Comentarios.
- Instrucciones potentes.
- Tiempo de codificación mucho menor.
- Más fáciles de corregir.
- Curva de aprendizaje de los programadores más corta.
- Los programas no pueden ejecutarse directamente.
- Tiempo de traducción.
- Ocupación mayor de memoria.
- Menor efectividad en el consumo de recursos y prestaciones.

ALGUNOS LENGUAJES CONOCIDOS

- Matemáticos: Fortran, Algol.
- Educativos: Logo, Basic, Pascal.
- Inteligencia Artificial: Lisp, Prolog.
- Contabilidad: Cobol.
- Programación concurrente: Ada.
- Propósito general: Pascal, C.
- Bases de datos: SQL/PLSQL.
- Orientación a objeto/evento: C++, Java, Visual Basic.
-

3. FASES DE ELABORACIÓN DE UN PROGRAMA (*ciclo de vida*)

1. Análisis.
2. Diseño.
3. Codificación y prueba.
4. Implantación.
5. Mantenimiento.

1ª ETAPA. ANÁLISIS

- Establecer requisitos que debe cumplir el producto.
- Especificación de Procesos y Estructuras de Datos.
- Comunicación entre usuario y analistas.
- Desarrollo de prototipos.
- Técnicas:
 - Diagrama de Flujo de Datos (DFD).
 - Modelo de Datos Entidad/Relación.
 - Diccionario de Datos.
 - Diagramas UML.
- ...

2ª ETAPA. DISEÑO

- Mayor detalle sobre el desarrollo de la aplicación.
- Teniendo en cuenta:
 - Recursos físicos del sistema (tipo de sistemas, periféricos, comunicaciones...).
 - Recursos lógicos del sistema (SO, BD...).

3ª ETAPA. CODIFICACIÓN Y PRUEBA

- Traducción a un lenguaje de programación.
- Pruebas para depurar errores:
 - Pruebas unitarias.
 - Pruebas integradas.
 - Pruebas de aceptación.

4ª ETAPA. IMPLANTACIÓN Y EXPLOTACIÓN

- Puesta en marcha del producto.
- Actividades:
 - Conversión de información al nuevo sistema.
 - Eliminación del sistema anterior.
 - Instalación de programas.
 - Formación al usuario.

5ª ETAPA. MANTENIMIENTO

- Una vez el producto está ya en marcha.

- Tipos:

- Correctivo

- Adaptativo

- Perfectivo

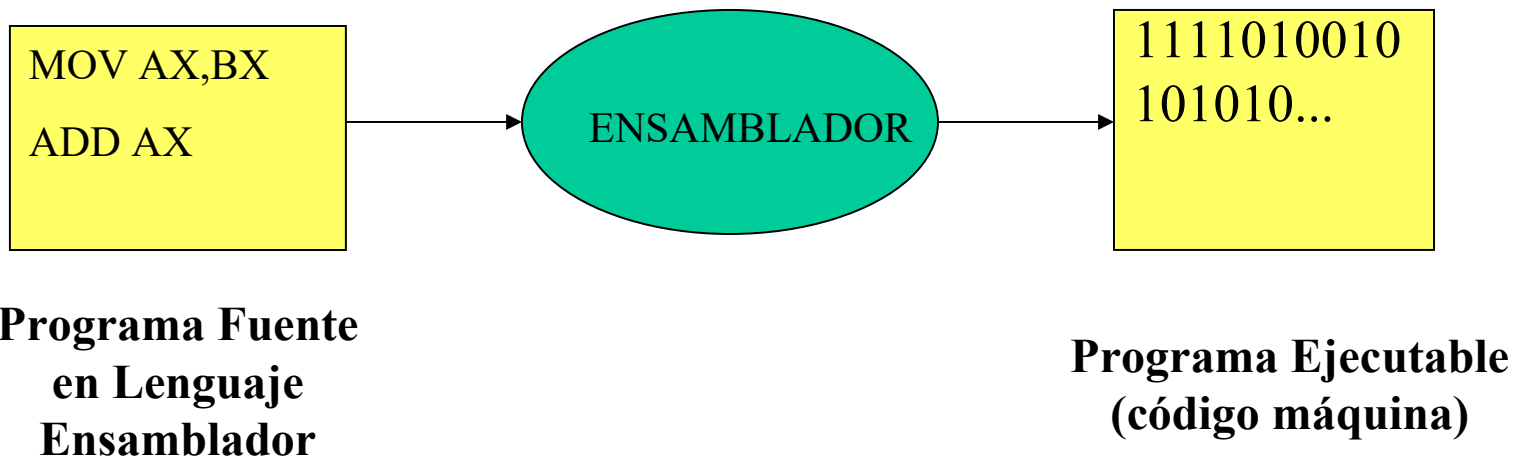
Reinician el ciclo de vida

4. TRADUCTORES DE LENGUAJES

- Programa fuente: Escrito en un lenguaje de programación concreto.
- El traductor se encarga de traducir el programa fuente a código binario (código máquina).
- Previamente comprobaba si la sintaxis es correcta e informará de los errores.
- Tipos:
 - Ensambladores.
 - Intérpretes.
 - Compiladores.

ENSAMBLADORES

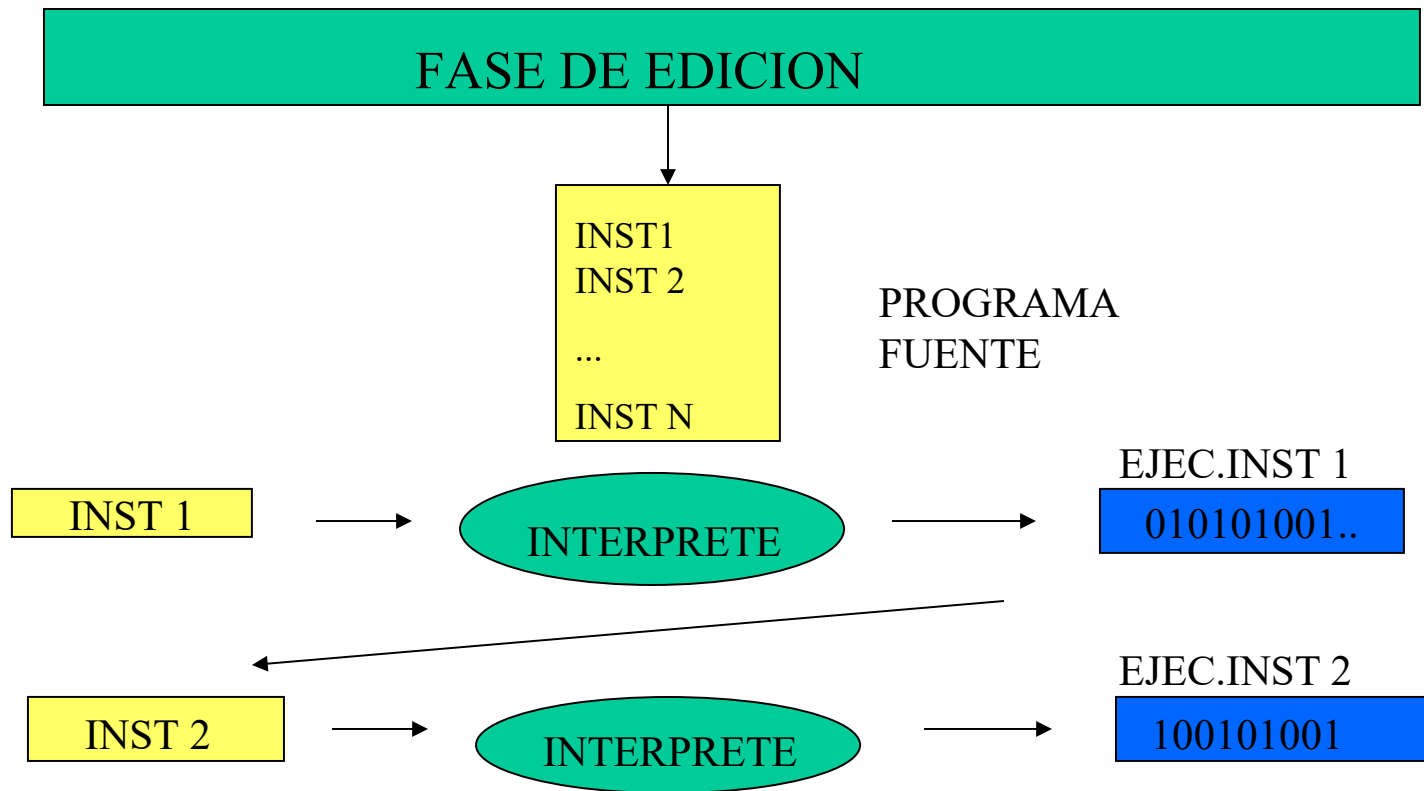
- Traducen programas escritos en lenguaje ensamblador.



INTÉRPRETE

- Traduce una instrucción y después, si su sintaxis es correcta, la ejecuta. En caso de error el programa se detendrá.
- El proceso se realiza instrucción a instrucción.
- Suelen incluir editor.
- Inconveniente: Lentitud.

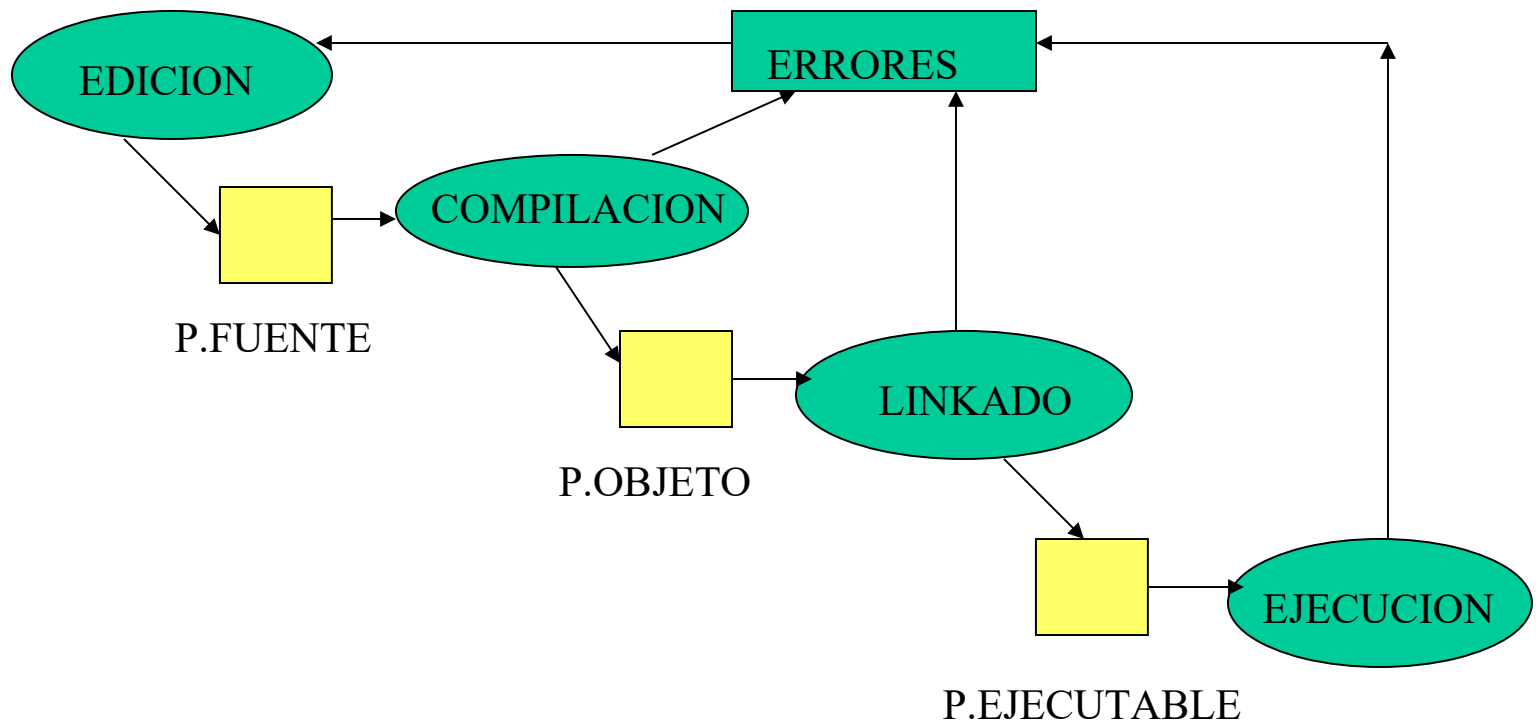
ESQUEMA INTÉRPRETE



COMPILADOR

- Traduce el programa fuente a código máquina, conectando todos los módulos del programa y de las librerías utilizadas.
- Ejecuta el programa, si no tiene errores de sintaxis. Informa de los errores de sintaxis si los hay.
- El programa compilado es mucho más rápido en tiempo de ejecución.
- Suelen incluir editor y depurador.

ETAPAS PROCESO COMPILACIÓN



COMPROBACIÓN DE ERRORES

- Análisis sintáctico:
 - Se analiza si el programa está escrito siguiendo las reglas sintácticas del lenguaje de programación.
 - Se realiza antes de ejecutar una instrucción (intérprete) o un programa (compilador).
- Análisis semántico:
 - Se analiza si el programa realiza la tarea prevista.
 - Se realiza en tiempo de ejecución.

5. ENTORNOS INTEGRADOS DE DESARROLLO (IDE)

- Es un programa informático compuesto por un conjunto de herramientas de programación.
- Puede dedicarse en exclusiva a un sólo lenguaje de programación o bien, poder utilizarse para varios.
- Pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes.
- Proveen un marco de trabajo “amigable” para el programador.

COMPONENTES DE UN IDE

- Editor de texto.
- Traductor (intérprete, compilador).
- Depurador de errores.
- Control de versiones.
- Asistentes para la construcción de interfaces gráficas de usuario.