**T.1. INTRODUCCIÓN A LA PROGRAMACIÓN**

**1.1 Introducción**

**Informática**: Ciencia que estudia el tratamiento automático de la información. (Infor-mática)

Ej.: Calculadora, ábaco…

**Tratamiento informático**: Tratamiento que se aplica a unos datos de entrada, transformándolos en unos datos de salida.

ENTRADA -> TRATAMIENTO -> SALIDA

Puede ser:

- Operación aritmética (+, -, \*, /)

- Operación lógica (si, mientras, repetir)

- Operación relacional (=, >, <)

- Almacenamiento y recuperación de la información

* 1. **Lenguajes de programación**

Los lenguajes especifican las acciones que se quieren realizar sobre un conjunto de datos.

* **Lenguaje máquina**: está formado por 1 y 0. Determinan cada una de las instrucciones posibles sobre un ordenador.

Cada procesador tiene su **propio lenguaje máquina**. Esto quiere decir, que los programas escritos en un lenguaje máquina de un procesador no pueden ser usados en otro procesador con un lenguaje máquina distinto. (Portabilidad)

* **Lenguaje ensamblador**: Se traducen las instrucciones de 0 y 1 a caracteres alfabéticos parecidos al lenguaje natural. Estas abreviaturas se llaman nemónicos. Se caracterizan por:

- Los programas son igual de largos que en lenguaje máquina.

- Son más legibles y fácilmente revisables.

- Propio de cada procesador, por tanto no es portable.

* **Lenguaje de alto nivel:** Son independientes de la arquitectura subyacente. Los programas creados en estos lenguajes sí son portables. Se caracteriza por:

**-** Ser más cortos que el lenguaje máquina o ensamblador.

**-** Al separarse del lenguaje máquina se necesita de un traductor (compilador) que sea capaz de pasar las instrucciones a lenguaje máquina.

* 1. **Fases de elaboración de un programa**

Análisis 🡪 Diseño 🡪 Codificación 🡪 Explotación 🡪 Mantenimiento

* **Análisis**: Se toman todos los datos necesarios, las especificaciones de procesos y las estructuras de datos necesarias manteniendo sucesivas reuniones con el cliente.
* **Diseño**: Se diseña el sistema con una codificación lógica, empleando diagramas como; diagramas de clases, de estructuras, entidad-relación, de acción, etc.

En esta fase se determina cómo se realizan las especificaciones de la aplicación.

* **Codificación**: Traducción a un lenguaje de programación de los elementos generados en fase de diseño. En esta fase también se incluyen las pruebas necesarias para su funcionamiento.
* **Explotación**: Implantación de la aplicación sobre el entorno real. Incluye todas las pruebas con datos reales y verificación de la lista de requerimientos.
* **Mantenimiento**: Detección y corrección de errores sobre el programa. También se incluyen mejoras, adaptaciones y modificaciones.
  1. **Traductores**

Se encargan de traducir instrucciones de un lenguaje de programación a otro lenguaje de programación. Los traductores pueden ser:

* **Ensambladores**: traduce de lenguaje ensamblador a lenguaje máquina y ejecuta la instrucción.
* **Intérprete**: traduce instrucción por instrucción de un lenguaje de alto nivel a lenguaje máquina y la ejecuta.
* **Compilador**: traduce un programa completo de un lenguaje de alto nivel a lenguaje máquina y luego lo ejecuta. Un compilador comprende las siguientes fases:
  + **Edición**: Escritura **código fuente** en lenguaje de alto nivel.
  + **Compilación**: Traducción del programa fuente a un lenguaje intermedio. Genera el programa objeto.
  + **Linkado**: Unión de los programas objeto en un solo programa ejecutable.
* **Diferencias Compiladores-Intérpretes**

El código ejecutable generado por compiladores está libre de errores. Los intérpretes no garantizan que el resultado esté libre de errores.

Los compiladores emplean un traductor más sofisticado que los intérpretes. Pudiendo realizar operaciones de:

* Optimización de código.
  + Optimización de tiempo: Obtener programa más rápido, menos instrucciones.
  + Optimización de espacio: Obtener programa que necesite menos recursos. (Variables, estructuras de datos, ficheros…)
* Comprobación avanzada de errores en el código fuente.