**TEMA 1: INTRODUCCIÓN AL DESARROLLO DE SOFTWARE.**

1. HARDWARE.

Es la parte tangible de un sistema informático. Un ordenador se basa en la arquitectura de Von Newmann.

* UNIDAD DE CONTROL: controla la ejecución de todas las operaciones y dirige el funcionamiento de todos los demás componentes para trabajar de manera conjunta.
* UNIDAD ARITMÉTICO-LÓGICA: Es la parte encargada de realizar las operaciones aritméticas y lógicas.
* MEMORIA PRINCIPAL: memoria de almacenamiento interno, opera a gran velocidad, aquí se encuentran los programas (las instrucciones y los datos sobre los que actúan).
* DISPOSITIVOS DE ENTRADA/SALIDA: ratón, teclado, monitor, impresora, etc.

EL CICLO DE UNA TRANSMISION.

1. La U.C. indica a la M.P. la instrucción que quiere ejecutar.
2. La U.C. recibe la instrucción, analizándola para determinar las operaciones sobre las que actúan y su localización.
3. Bajo las instrucciones de la U.C. la U.A.L. realiza la operación y guarda el resultado en su destino.
4. Una vez ejecutada la instrucción el contador se incrementa y pasa a realizar la siguiente ejecución.

\* Los ordenadores digitales utilizan internamente código binario (0 y 1).

1 byte = 8 bits. Lenguaje Máquina.

2. SOFTWARE.

Es la parte intangible de un sistema informático, el software es el encargado de traducir las instrucciones del USUARIO a LENGUAJE MÁQUINA.

CARACTERÍSTICAS.

* Es lógico, intangible.
* Se desarrolla, no se fabrica.
* No se estropea, una copia suya da lugar a un clon idéntico.
* Se puede construir a medida o enlatado.

TIPOS DE SOFTWARE.

1. De Sistemas: proporciona al usuario una interfaz de alto nivel y fácil de usar.
2. De programación: proporciona herramientas al usuario para el desarrollo de programas informáticos. (Editores de texto, IDEs…).
3. De aplicación: permite al usuario realizar una o varias tareas específicas.

3. LENGUAJES DE PROGRAMACION.

* Todos los programas están escritos en un lenguaje de programación.
* Los lenguajes de programación permiten comunicarnos con el hardware del equipo.
* Todos los lenguajes de programación tienen: una sintaxis, conjunto de normas y palabras reservadas.

TIPOS DE LENGUAJES DE PROGRAMACION.

1. LENGUAJE MÁQUINA.

* Instrucciones complejas.
* No necesita ser traducido (ya que es el único lenguaje que entiende el ordenador.
* Es diferente en cada ordenador, es decir, el conjunto de las instrucciones no es portable de un equipo a otro.

1. LENGUAJE DE NIVEL MEDIO O ENSAMBLADOR.

* Sustituye al lenguaje máquina para facilitar la labor de programar.
* Utiliza mnemotécnicos, instrucciones complejas en lugar de 0 y 1.
* Necesita ser traducido a LM para poder ejecutarse.
* Trabaja con registros del procesador y con direcciones físicas.
* Difícil de comprender y programar.

1. LENGUAJE DE ALTO NIVEL.

* La mayoría de lenguajes pertenecen a esta categoría.
* Utilizan sentencias y ordenes derivadas de otros idiomas, inglés.
* Tiene librerías y funciones predefinidas que facilita la tarea del programador.
* Suelen trabajar con abstracción y la orientación a objetos, haciendo más fácil la reutilización y encapsulación de los componentes.

TIPOS DE LENGUAJES DE PROGRAMACION (SEGÚN SU EJECUCION).

1. LENGUAJES COMPILADOS.

* Necesitan de un “compilador” para pasar el código fuente a máquina.
* Lenguajes con mejor tiempo de ejecución.
* Necesitan un Linker para unir el código objeto del programa con librerías.

1. LENGUAJES INTERPRETADOS.

* Requiere de intérprete (programa) para interpretar y ejecutar las instrucciones.
* Solo traduce lo que se va a ejecutar.

1. LENGUAJES VIRTUALES.

* Son los mas portables ya que generan código intermedio o bytecode.
* Lenguajes con mayor tiempo de ejecución.
* Su código puede ser interpretado por una maquina virtual instalada en un equipo.

4. CODIFICACION.

* TRADUCTORES: son programas cuya finalidad es traducir lenguajes de alto nivel a lenguajes de bajo nivel.
* INTÉRPRETE: traduce el código fuente línea a línea, tiene que estar en memoria para poder realizar este proceso.
* COMPILADOR: traduce el código fuente a código máquina. El compilador esta en la maquina de desarrollo. El código generado solo funcionara en una máquina.