



Curso 2023/24

Tema-01 Parte I

Introducción al desarrollo del software:
Hardware y Software.

Tema-01 (Parte I)

Contenido

1.- INTRODUCCIÓN	2
2.- HISTORIA DE LA INFORMÁTICA:	3
3.- DESARROLLO DEL SOFTWARE:.....	6
3.1.- Concepto de programa informático.	6
3.2.- Características de un programa informático	6
3.3.- Ejecución de programas informáticos en ordenadores. Instrucciones y datos	7
3.3.1. Datos y programas	8
3.3.2. Estructura funcional de un ordenador.....	14
3.3.3. Relación entre hardware - software	21
3.3.4. Tipos de software. BIOS. Sistema. Aplicaciones	23
3.3.5. Código fuente, código objeto y código ejecutable; máquinas virtuales	24
4.- LENGUAJES DE PROGRAMACIÓN:	26
4.1.- ¿Qué es un lenguaje de programación?	26
4.2.- Tipos de lenguajes de programación.....	26
4.3.- Paradigmas de programación.....	28
5.- INTRODUCCIÓN A LA INGENIERÍA DEL SOFTWARE:	30

1.- INTRODUCCIÓN

El módulo de **Entornos de desarrollo**, incluye la teoría necesaria para reconocer los elementos y las herramientas para el desarrollo de un programa informático, evaluar entornos integrados, verificar su funcionamiento mediante la realización de pruebas, optimizar el código, así como para generar diagramas de clases y de comportamiento.

Para ello se van a definir los conceptos previos más importantes que se deberían de tener claros, estos son:

- **Definición de Informática:** Es la ciencia que estudia el tratamiento automático y racional de la información.

Esto significa:

- Tratamiento automático: Porque son máquinas las que realizan los trabajos.
- Tratamiento racional: Porque el proceso lo siguen máquinas a través de programas que siguen el razonamiento humano.

El término informática se creó en Francia en el año 1962 bajo la denominación INFORMATIQUE y procede de la contracción de las palabras INFORmation autoMATIQUE. Posteriormente fue reconocido por el resto de países, siendo adoptado en España en 1968 bajo el nombre de informática que, como puede deducirse fácilmente, viene de la contracción de las palabras INFORmación autoMATICA.

- **Definición de Hardware (HW):** Es la parte física de un ordenador o Sistema Informático, es decir, todo lo que se puede ver y tocar, es visible y tangible. En cambio, el software es el programa para computadoras. Por ejemplo: La torre del PC, el teclado, la pantalla, la impresora, router, etc.
- **Definición de Software (SW):** Es la parte intangible de un ordenador o Sistema Informático, no se puede ver ni tocar. Por ejemplo: El Sistema Operativo, los programas o aplicaciones, los drivers, los protocolos, etc.
- **Definición de Sistema Informático (SI):** Es un sistema que permite almacenar y procesar información; es el conjunto de partes interrelacionadas: hardware, software y personal informático. El hardware incluye ordenadores o cualquier otro tipo de dispositivo electrónico, que consisten en procesadores, memoria, sistemas de almacenamiento externo, etc. El software incluye al sistema operativo, firmware y aplicaciones, siendo especialmente importante los sistemas de gestión de bases de datos. Por último, el componente humano incluye al personal técnico que apoya y mantienen el sistema (analistas, programadores, operarios, etc.) y a los usuarios que lo utilizan.

2.- HISTORIA DE LA INFORMÁTICA:

I. Generación Prehistórica

Dado que un ordenador es un aparato para contar, se puede afirmar que la idea de la computación es tan antigua como la misma civilización, ya que el primer aparato de computación podría haber sido un conjunto de piedras para representar medidas de rebaños de animales.

La palabra cálculo tiene su origen en el término latino *calculus*, que significa piedra, y que se utilizaba hace miles de años para denominar a unas pequeñas piedras que por medio de unas ranuras efectuadas en el suelo se utilizaban para contar.

II. Generación Cero

Posteriormente apareció **el ábaco**. El más antiguo se ha descubierto en el valle situado entre el Éufrates y el Tigris, (**data del año 3500 a. C**). Fue el **primer dispositivo manual de cálculo** que permitía representar números en el sistema decimal y contar, permitiendo realizar operaciones aritméticas sencillas.

En el **siglo XVII**, **John Napier** ideó un dispositivo basado en **varillas cifradas** que contenían números y era capaz de multiplicar y dividir de forma automática. **También ideó un calculador con tarjetas** que permitían multiplicar y dividir, **conociéndose como estructuras de Napier**.

Hacia el año **1623**, **Wilhelm Schickard** ideó una calculadora mecánica conocida como **reloj calculante**, que funcionaba con ruedas dentadas y era capaz de sumar y restar, pero no se pudo montar en aquella época, siendo construida posteriormente en el siglo XX por ingenieros de IBM. Se la consideró la primera máquina de calcular de origen mecánico.

Blaise Pascal inventó en 1642 la primera máquina automática de calcular sumas y restas a base de ruedas dentadas que simulaba el funcionamiento del ábaco. Al principio se la conoció como **Pascalina**.

En 1650, Patridge inventó la regla de cálculo, consistente en una pequeña regla deslizante sobre una base fija en la que aparecían varias escalas para realizar diferentes operaciones.

En 1666 Samuel Morland inventó otra máquina que realizaba sumas y restas a la que llamó **Máquina Aritmética de Morland**.

En 1672, G. Wilhelm von Leibnitz construyó una máquina a la que llamó **calculadora universal** capaz de sumar, restar, multiplicar, dividir y sacar raíces cuadradas. Esta máquina realizaba la multiplicación de forma directa y no mediante sumas sucesivas como la máquina de Pascal.

En 1833, Charles Babbage diseñó una máquina a la que llamó **máquina analítica**, capaz de realizar todas las operaciones matemáticas y con la posibilidad de programarla mediante tarjetas de cartón perforadas, siendo capaz de almacenar en su interior una gran cantidad de cifras. Esta máquina nunca llegó a construirse por problemas tecnológicos de la época, pero años después aparecieron algunas máquinas con su diseño, y por ello, a Babbage se le considera el padre de la Informática.

Augusta Ada Byron realizó programas para la máquina analítica de Babbage, por lo que se la consideró la primera programadora de la historia.

En 1854, George Boole desarrolló el álgebra de Boole, lo que permitió el **desarrollo del álgebra binaria** y con ella **la representación** de circuitos de conmutación y la aparición de la **“Teoría de los Circuitos Lógicos”**.

En 1885, H. Hollerith vio cómo se tardaban 10 años en realizar el censo de su país y observó que la mayoría de las respuestas del censo eran sí o no, lo que **le llevó a idear en 1886 unas tarjetas perforadas** con la información de las personas censadas **y una máquina capaz de leer y tabular esta información**.

En 1895, Hollerith incluyó a su máquina la operación suma para poder utilizarla para la contabilidad de los Ferrocarriles Centrales de New York, lo que constituyó el primer intento de realización automática de una aplicación comercial. En 1896 fundó la empresa Tabulating Machines Company, que fue el origen de la conocida International Business Machines (IBM).

En 1936, Alan M. Turing desarrolló la teoría de **una máquina capaz de resolver todo tipo de problemas con solución algorítmica**, llegando a la construcción de las máquinas de Turing. Con esta máquina comienza la teoría matemática de la computación.

En 1937, Howard H. Aiken, con un equipo de científicos de su departamento e ingenieros de IBM, **construye el primer ordenador electromecánico**, basado en el uso de relés electromagnéticos, ruedas dentadas y embragues electromecánicos, a la que se la llamó **MARK-I**. Se terminó de construir en 1944 y tenía elementos de entrada (tarjetas y cinta perforadas), memoria central, unidad aritmética, unidad de control y elementos de salida.

En 1942, John Vicent Atanasoff y Clifford Berry construyeron **la primera máquina de calcular digital**, que operaba en binario.

En 1940, John W. Mauchly y John Presper Eckert construyeron **el primer ordenador electrónico** a la que llamaron **ENIAC** construido a base de válvulas de vacío, que empezó a funcionar en 1945. Esta máquina era programable y universal, es decir, podía realizar cualquier tipo de cálculo. Tenía un volumen de aproximadamente 111 metros cúbicos, ocupaba una superficie aproximada de 160 metros cuadrados y su peso era aproximadamente de 30 toneladas.

En 1944, John Von Neumann desarrolla **la idea del programa interno y describe la construcción de un ordenador electrónico que podía ser programado** no estando las órdenes cableadas en los circuitos de la máquina.

En 1951, John W. Mauchly construyó **el primer ordenador de serie** que se puso a la venta, la **UNIVAC I**.

III. Generación de los ordenadores

A partir de este momento, gracias a la evolución de la electrónica, se comenzaron a construir los ordenadores más potentes, baratos y pequeños, que nos permiten clasificar las computadoras en generaciones:

Primera Generación - (1940-1952)

Constituida por **ordenadores contruidos a base de válvulas de vacío** que utilizaban el lenguaje máquina como lenguaje de programación y tarjetas perforadas, cintas perforadas y líneas de demora de mercurio como memorias para almacenar información.

Segunda Generación - (1952-1964)

Se sustituyeron las válvulas de vacío por transistores. Los ordenadores **ganaron potencia y fiabilidad, perdiendo tamaño, consumo y precio.** Se **comenzaron a utilizar lenguajes de programación evolucionados** (Ensamblador, Fortran, Cobol y Algol), que hacía la programación más sencilla. Se empezó a utilizar como memoria interna los núcleos de ferrita y el tambor magnético y como memoria externa la cinta magnética y los tambores magnéticos

Tercera Generación - (1964-1971)

Aparecieron los circuitos integrados (encapsulamiento de una gran cantidad de elementos como resistencias, condensadores, diodos y transistores conformando uno o varios circuitos con una función determinada sobre una pastilla de silicona o plástico), que permitió la miniaturización de los ordenadores. El software evolucionó considerablemente con un gran desarrollo de los sistemas operativos y comenzaron a utilizarse memorias de semiconductores y discos magnéticos.

Cuarta Generación - (1971-1981)

En **1971 se fabricó el primer microprocesador** consistente en **la integración de toda la UCP** (Unidad de Control y Proceso) **de un ordenador en un solo circuito integrado**, que permitió la fabricación de microordenadores y ordenadores personales. **Se empezó a utilizar el disquete como unidad de almacenamiento externo** y aparecieron una gran cantidad de lenguajes de programación y las redes de transmisión de datos para la interconexión de computadoras.

Quinta Generación - (1981-1989)

Una de las principales características de esta generación es la **simplificación y miniaturización del ordenador**, además de mejor desempeño y **mayor capacidad de almacenamiento**. Todo eso, con los precios cada vez más accesibles. Su objetivo era el desarrollo de una nueva clase de ordenadores **que utilizaran técnicas y tecnologías de inteligencia artificial**, tanto en el plano del hardware como del software. La tecnología VLSI fue sustituida por la ULSI (Ultra Large Scale Integration).

Sexta Generación - (1990-Actualidad)

La sexta generación se podría llamar a la era de los **ordenadores inteligentes basados en redes neuronales artificiales o «cerebros artificiales»**. Son ordenadores que utilizarían superconductores como materia prima para sus procesadores, lo cual permitirían no malgastar electricidad en calor debido a su nula resistencia, ganando performance y economizando energía. Aparecen los procesadores que trabajan en paralelo, o sea, la división de tareas en múltiples unidades de procesamiento operando simultáneamente. Otra novedad es la incorporación de **chips de procesadores especializados en las tareas de vídeo y sonido**.

Comienzan a popularizarse las pantallas planas de tipo LCD.

Además de incluir un notable avance en los componentes de hardware, permitiendo que los usuarios puedan disponer de mayor potencia y velocidad de procesamiento. También se amplía la capacidad de almacenamiento de los dispositivos creados para ese fin.

3.- DESARROLLO DEL SOFTWARE:

3.1.- Concepto de programa informático.

Un **programa informático** es un conjunto ordenado, (simple o complejo) de instrucciones o pasos, comandos y procesos escritos en un lenguaje de programación, diseñados para realizar una tarea en particular en un ordenador o sistema de estos. En palabras sencillas, es una secuencia de órdenes que le indican a una computadora qué hacer.

Dichos programas pueden estar preinstalados en el ordenador, como el sistema operativo que domina todo el funcionamiento de éste, o pueden ser integrados de forma adicional por el usuario. Normalmente, **estos programas son desarrollados por personas que se dedican profesionalmente a la escritura de código**, los cuales se conocen como **programadores**.

Los programas poseen en su núcleo un código fuente, que es el **texto legible por los humanos y escrito en un lenguaje de programación**.

Para su diseño, los programadores usan, al mismo tiempo, **programas informáticos** que **funcionan para crear** dichos **códigos**, o sea, software creador para hacer software, y **se denomina** como **lenguaje de programación**. **Software creado para hacer software** es la herramienta que permite que, (un programador), pueda codificar un cúmulo de instrucciones lógicas, (el programa informático), para que un ordenador pueda ejecutarlo y que éste responda, llevando a cabo una serie determinada de acciones.

3.2.- Características de un programa informático

Un programa informático se caracteriza principalmente por ser:

- **Lógico**: Está constituido por un conjunto de procesos lógicos.

La lógica de programación: Es la organización coherente de las instrucciones del programa para que su objetivo sea alcanzado o logrado.

Para **poner en práctica** la Lógica de Programación se deben **desarrollar Algoritmos**, en los cuales **se utilizan una serie de elementos** que facilitan el entendimiento de cada paso a ejecutar, tales como: **Instrucciones. Datos. Variables. Constantes, Operadores**.

Para ser un excelente Programador de sistemas informáticos, se debe de tener totalmente claro de todas estas definiciones y las maneras o formas en que se utilizan, (por qué se utilizan y para qué se utilizan).

- **Intangible**: Es parte del segmento digital, (software), de un sistema informático, en contraposición al segmento físico, (hardware).
- **Funcional**: Es diseñado para cumplir una tarea o conjunto de tareas.
- **Preciso**: Cumple correctamente con lo programado.
- **Ejecutable**: Se encuentra en un formato para que se pueda accionar dentro una computadora.
- **Secuencial**: El código escrito se ejecuta en un determinado orden.

Diferencia entre software y programa informático:

Los términos programa informático y software son usados en muchos contextos como sinónimos. Sin embargo, en un sentido más estricto, tienen diferencias destacables.

Un ***programa informático es una cadena de instrucciones y comandos que sigue un patrón lógico para cumplir ciertas tareas*** (objetivos de procesamiento de datos); mientras que ***un software es un paquete informático que puede contener diversos programas informáticos, además de otros elementos***, como imágenes, textos, audios, y datos.

Por tanto, un ***programa informático se limita a la parte secuencial de instrucciones*** para llevar a cabo una función; y ***el software es el envoltorio que reúne programas informáticos y otros archivos*** que, de forma coordinada, ejercen un conjunto de labores.

3.3.- Ejecución de programas informáticos en ordenadores. Instrucciones y datos

La Ejecución en ingeniería informática y de software es el proceso mediante el cual un ordenador o máquina virtual, (explicado más adelante), lee y actúa según las instrucciones de un programa informático.

Cada instrucción de un programa es una descripción de una acción particular que debe llevarse a cabo para resolver un problema específico. La ejecución implica seguir repetidamente un ciclo '*buscar-decodificar-ejecutar*' para cada instrucción. A medida que la máquina ejecutora sigue las instrucciones, se producen efectos específicos de acuerdo con la semántica de esas instrucciones.

En Ciencias de la computación, la semántica denotacional, (inicialmente se le conoce como semántica matemática o semántica Scott-Strachey), es un enfoque para formalizar el significado de los lenguajes de programación mediante la construcción de objetos matemáticos. (llamados denotaciones), que describen los significados de las expresiones del lenguaje.

Los programas para un ordenador pueden ejecutarse en un proceso por lotes, sin interacción humana, o un usuario puede escribir comandos en una sesión interactiva de un intérprete. En este caso, los "comandos" son simplemente instrucciones de programa, cuya ejecución está encadenada.

3.3.1. Datos y programas

❖ Los **ELEMENTOS** de un **Programa Informático** son:

- Algoritmos e Instrucciones.
- Clases y Objetos.
- Lenguaje de Programación.
- Tipos de Datos.
- Métodos.
- La Instrucción de Retorno.
- La Instrucción de Asignación.
- La Instrucción de Llamada de un Método.

Un programa informático en ejecución, normalmente es tratado como algo diferente de los datos con los cuales opera. Sin embargo, en algunos casos esta distinción es ambigua, especialmente cuando un programa se modifica a sí mismo. El programa modificado es ejecutado secuencialmente como parte del mismo programa.

Los datos que va a utilizar el programa se operan (en una de sus formas más básicas) a través de las variables y constantes del programa.

La inicialización de variables significa especificar un valor inicial para asignarle (es decir, antes de que se use). Tenga en cuenta que una variable que no se inicializa no tiene un valor definido, por lo tanto, no se puede usar hasta que se le asigne dicho valor.

❖ Las **PARTES** de un **Programa Informático** son:

- Algoritmo.
- Cabecera.
- Declaración de funciones y procedimientos.
- Contadores.
- Salida de resultados.
- Acumuladores.

Nota-01: No confundir cuales son los elementos de un programa informático con las partes de éste.

Algoritmo

Está conformado por los comandos que cambian los elementos mediante un estado inicial hasta el final, dejando éstos asequibles dentro de la memoria central.

Los programas se incluyen dentro de la propiedad intelectual de los diseñadores, si bien en muchas situaciones son puestos de forma gratuita o bajo una licencia de copyleft, que son más permisivas y que no necesitan de pago por empleo.

Cabecera

Es vista como una sección obligatoria, puesto que tiene que verse en todos los programas. Tiene que empezar con la palabra reservada programa seguida del nombre del programa y también un «;».

Con esto ya se completarían los requisitos más elementales que tiene que tener una cabecera, pero se puede y es muy aconsejable añadir un comentario.

Dicho comentario se usa para documentar el programa que es algo que los usuarios suelen dejar en segundo lugar, pero es lo más esencial en la programación.

Dentro del comentario tiene que haber una mayor cantidad de componentes como autor del programa, la versión, fechas de ultimo cambio y de inicio del programa.

Declaración de funciones y procedimientos

Esta parte también es esencial, aunque para primeros programas no siempre son requeridos. En ella el programador puede declarar funciones o subprogramas y procedimientos para que sean empleados en otras funciones o en el cuerpo del programa. Todo lo concerniente a esto puede ser hallado en escritos sobre programación modular.

Es posible agregar las funciones y procedimientos que se quieran, inclusive cambiarse entre sí. Aunque es importante tener cuidado con el orden en que son escritos. Puesto, que, si una función llama a otra, debe ser declarado en primer lugar el llamado.

Contadores

Consiste en un campo de memoria cuyo valor aumenta en una, (o pueden ser más), cantidad negativa o positiva pero fija, usualmente vinculada a un bucle.

Tiene un valor previo al iniciar la función, y cada vez que se hace el suceso, aumenta el mismo.

Salida de resultados

Es un conglomerado de comandos que toman los datos finales de una memoria central y los trasladan a dispositivos de exterior.

Un comando o instrucción se conoce por el estado inicial y final del ambiente. El estado final de una instrucción es similar a la inicial de la que sigue. No ocurre que una instrucción cambia el entorno, pues su fin puede limitarse a una sencilla observación del mismo o a una variación en el orden de la realización de otras.

Acumuladores

Es un campo de memoria el cual tiene un valor que aumenta de forma sucesiva en diversas cantidades. Se emplea en las ocasiones en que se quiere conseguir el total que se acumula de un conglomerado de cantidades, siendo necesario empezarlo con el valor cero.

Asimismo, en casos en que se requiere conseguir un total como producto de diversas cantidades, se emplea un acumulador que comenzará con un valor cero.

❖ La **EJECUCIÓN** de un **Programa Informático**:

A **un programa en ejecución se le suele llamar también proceso**. El proceso de ejecución de un programa, escrito en un lenguaje de programación y llevado a cabo mediante un compilador. Tiene los siguientes pasos:

- Paso-01: La escritura del programa fuente con un editor, (programa que permite a una computadora actuar de modo similar a una máquina de escribir electrónica), y guardarlo en un dispositivo de almacenamiento.
- Paso-02: Introducir el programa fuente en memoria.
- Paso-03: Compilar el programa con el compilador.
- Paso-04: Verificar y corregir errores de compilación en el programa fuente, (fuente = origen).
- Paso-05: Obtención del programa objeto.
- Paso-06: Obtención del programa ejecutable. El enlazador, (o linker), obtiene el programa ejecutable.
- Paso-07: Se ejecuta el programa y, si no existen errores, se tendrá la salida del programa (resultado de la acción objetivo o funcionalidad que perseguía el desarrollo del mismo)

Los ordenadores trabajan básicamente con números binarios, las personas hablan sus idiomas nativos, castellano, inglés, francés, alemán, etc.... Por lo tanto los lenguajes de programación sirven para entablar la comunicación entre personas y computadoras.

Por ejemplo: Si alguien quiere resolver un problema y dijera directamente a un ordenador que se agregase 2 y 3, luego le restase 1, dudo mucho que el ordenador entienda algo, (tal vez en algún lenguaje de programación sí lo haría entendería y lo haría). Para poder resolverlo se debe traducir el problema a un formato de código fuente que el ordenador entienda, para ello, necesita un compilador, software que traduce un lenguaje de programación al llamado código de objeto. Pero el código objeto todavía no es el lenguaje que un ordenador entiende y pueda ejecutarlo directamente. Por lo tanto, necesita un vinculador que haga un archivo ejecutable que contenga instrucciones en el llamado lenguaje de máquina.

Un **lenguaje de máquina** es un **conjunto de operaciones codificadas en números binarios que el procesador comprende**. Todas las instrucciones binarias tienen su estructura y están publicadas por un fabricante de procesadores.

EL ENLAZADOR O LINKER:

En programación, *Enlazador* o *Linker*, es un módulo o programa que junta los ficheros de código objeto, (generados en la primera parte de la compilación), con la información de todos los recursos necesarios, (biblioteca), eliminando los recursos que no se necesitan y enlazando el código objeto con sus bibliotecas. Finalmente combina todo en un sólo archivo ejecutable, o a su vez archivo de biblioteca u otro archivo objeto.

Existen programas que se enlazan dinámicamente, esto significa que este proceso se hace en el momento en que se carga el programa.

Los enlazadores son parte de la teoría del compilador. Cuando compila un proyecto compuesto por más de un módulo, (archivo de código fuente), es común que el compilador genere un único archivo intermediario para cada módulo fuente. Esto tiene varios beneficios, uno de los cuales es, que, si sólo realiza cambios en un archivo, luego tiene que volver a compilarlo, no tiene que reconstruir todo el proyecto cuando solo ha realizado un cambio local.

Una vez que todo se ha compilado en archivos intermedios, el vinculador es lo que termina el trabajo. Revisa todos los archivos intermedios y los une en un binario final. Dado que está juntando cosas, conoce las direcciones reales de todas las funciones, por lo que puede reemplazar los tokens de referencia externos con instrucciones reales para las ubicaciones correctas en el binario.

EL CARGADOR:

Por otro lado, el cargador pertenece al sistema operativo, no al compilador. Su trabajo es cargar el binario, (ejecutable), en la memoria para que pueda ejecutarse y finalizar el proceso de vinculación, ya que el vinculador sólo puede resolver el código que conoce.

El cargador simplemente lee los datos de un archivo ejecutable en la memoria y luego comienza a ejecutarlos en alguna dirección.

En este ejemplo de sistema simple, el cargador sigue siendo una pieza de código bastante sencilla que, básicamente va recorriendo la lista de reubicaciones y agregando la dirección base a cada una.

EL TIEMPO DE EJECUCIÓN O RUNTIME:

Se denomina tiempo de ejecución o (runtime en inglés), al intervalo de tiempo en el que un programa de un ordenador se ejecuta en un sistema operativo. Este tiempo se inicia con la puesta en memoria principal del programa, por lo que el sistema operativo comienza a ejecutar sus instrucciones. El intervalo finaliza en el momento en que éste envía al sistema operativo la señal de terminación, sea ésta una terminación normal, en que el programa tuvo la posibilidad de concluir sus instrucciones satisfactoriamente, o una terminación anormal, en el que el programa produjo algún error y el sistema debió forzar su finalización.

Este término suele emplearse, en oposición a **tiempo de compilación**, para indicar si una acción o hecho sucede en uno u otro tiempo.

CODIFICACION:

Es cuando se implementa el algoritmo en un código escrito en un lenguaje de programación. Refleja las ideas desarrolladas en las etapas de análisis y diseño.

COMPILACION:

La compilación es el proceso y resultado de compilar código fuente. Traducción a lenguaje máquina, (código objeto), para que pueda ser ejecutado por el ordenador. La aplicación o la herramienta encargada de la traducción se llama compilador.

LIGAMENTO:

Una vez verificado que no existen errores, se procede a unificar un sólo proceso previo a su carga.

CARGA:

Recibe la configuración del equipo y la asignación de memoria y se carga el programa en el sistema.

RELOCALIZACION:

Sustituye referencias o nombres simbólicos de bibliotecas con direcciones usables reales dentro de la memoria antes de funcionar un programa. Relocaliza el código fuente antes de su ejecución por si existe algún cambio dentro de la carga.

EJECUCIÓN:

Aquí se inicia la carga del programa o de cualquier archivo ejecutable. En otras palabras, la ejecución es el proceso mediante el cual un ordenador lleva a cabo las instrucciones de un programa informático.

Ciclo de lectura (fetch) y ciclo de ejecución

La tarea básica que realiza el ordenador, es la ejecución de los programas. El programa a ejecutar consta de un conjunto de instrucciones almacenadas en memoria. El procesador lleva a cabo el trabajo, ejecutando las instrucciones especificadas en el programa.

Para alcanzar una mayor comprensión de esta función y de la manera en que los componentes principales del ordenador interactúan para ejecutar un programa, hace falta analizar con cierto detalle los elementos de la ejecución de un programa. EL punto de vista más sencillo es considerar que el procesamiento de instrucciones consta de dos pasos:

- Paso 1: El procesador trae las instrucciones desde la memoria, (cada vez de una en una).
- Paso 2: Ejecuta cada instrucción.

La **ejecución de un programa** consiste en la repetición de este proceso de lectura y ejecución de la instrucción.

La **ejecución de la instrucción** puede involucrar varias operaciones y depende de la naturaleza de la instrucción.

El procesamiento requerido para una instrucción simple se llama **ciclo de instrucción**.

Los **dos pasos** se llaman **ciclo de lectura (fetch)** y **ciclo de ejecución**. La ejecución del programa se detiene sólo si se apaga la máquina, ocurre algún tipo de error irreparable o se encuentra una instrucción en el programa que detiene el ordenador.

Lectura y ejecución de instrucciones

Al comienzo de cada ciclo de instrucción, el procesador lee una instrucción de la memoria.

Esta secuencia puede alterarse.

La instrucción leída se carga en un registro del procesador conocido como registro de instrucción (IR).

La instrucción está en forma de código binario que especifica cuál es la acción que el procesador llevará a cabo.

El procesador interpreta la instrucción y realiza la acción requerida.

3.3.2. Estructura funcional de un ordenador

Hay cuatro secciones principales de un ordenador: La Unidad de Control (CU), La Unidad Lógica-Aritmética (ALU), La Memoria, (una secuencia de celdas de almacenamiento con números en la cual cada una es una unidad de información conocida como bit), y los Dispositivos de Entrada y Salida.

El funcionamiento del ordenador se basa en la captura de datos que se van a procesar por medio de alguna unidad de entrada, su almacenamiento en la unidad central de procesamiento, la ejecución de un programa que transforma los datos de entrada en resultados y en la comunicación de esos resultados, (información), al exterior por medio de una unidad de salida. Tanto la captura de los datos como la salida de la información se ejecutan a partir de una unidad de almacenamiento. Las unidades encargadas de estas operaciones se denominan periféricos.

Si queremos iniciar un programa necesitaremos recursos hardware del ordenador, como son el procesador, la memoria RAM, dispositivos de Entrada/Salida (E/S), etc. Las instrucciones para inicializar el programa se cargan en la memoria principal y se ejecutarán en la Unidad Central de Proceso (CPU), (en inglés, Central Processing Unit).

Para entender el funcionamiento de los componentes que conforman la CPU del ordenador, tenemos que estudiar **la arquitectura Von Neumann o arquitectura Princeton**, la cual se desarrolla a continuación:

❖ **La Unidad de Control (CU):**

Se encarga de interpretar y ejecutar las instrucciones que se almacenan en la memoria principal y, además, genera las señales de control necesarias para ejecutarlas.

❖ **La Unidad Lógica-Aritmética (ALU):**

Es la que recibe los datos y ejecuta operaciones de cálculo y comparaciones, además de tomar decisiones lógicas, (si son verdaderas o falsas), pero siempre supervisada por la Unidad de Control.

❖ **Los Registros:**

son los que almacenan la información temporal, almacenamiento interno de la CPU. A continuación, vamos a ver los diferentes registros que posee la UC:

- **Contador de programa (CP):** Contendrá la dirección de la siguiente instrucción a realizar, su valor será actualizado por la CPU después de capturar una instrucción.
- **Registro de Instrucción (RI):** Es el que contiene el código de la instrucción, se analiza dicho código. Consta de dos partes: el código de la operación y la dirección de memoria en la que opera.
- **Registro de dirección de memoria (RDM):** Tiene asignada una dirección correspondiente a una posición de memoria que va a almacenar la información mediante el bus de direcciones.
- **Registro de intercambio de memoria (RIM):** Recibe o envía, según si es una operación de lectura o escritura, la información o dato contenidos en la posición apuntada por el RDM.

❖ **Decodificador de instrucción (DI):**

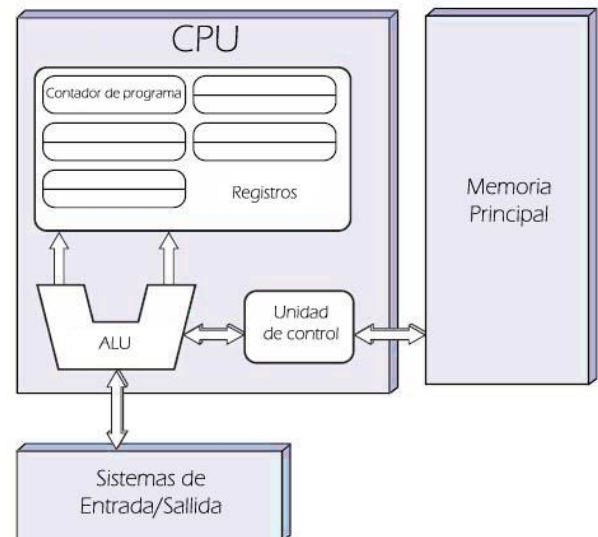
Extrae y analiza el código de la instrucción contenida en el RI y, además, genera las señales para que se ejecute correctamente la acción.

❖ **El Reloj:**

Marca el ritmo del DI y nos proporciona unos impulsos eléctricos con intervalos constantes a la vez que marca los tiempos para ejecutar las instrucciones.

❖ **El secuenciador:**

Son órdenes que se sincronizan con el reloj para que ejecuten correctamente y de forma ordenada la instrucción.



Cuando ejecutamos una instrucción podemos distinguir dos fases:

- **1ª Fase de búsqueda:** Se localiza la instrucción en la memoria principal y se envía a la Unidad de Control para poder procesarla.
- **2ª Fase de ejecución:** Se ejecutan las acciones de las instrucciones. Para que podamos realizar operaciones de lectura y escritura en una celda de memoria se utilizan el RDM, el RIM y el DI. El decodificador de instrucción es el encargado de conectar la celda RDM con el registro de intercambio RIM, el cual posibilita que la transferencia de datos se realice en un sentido u otro según sea de lectura o escritura.

❖ **Memoria Principal:**

La memoria principal en un ordenador se denomina memoria de acceso aleatorio. También es conocida como RAM. Ésta es la parte de la computadora que almacena software del sistema operativo, aplicaciones de software y otra información para la unidad de procesamiento central (CPU) y así tener acceso rápido y directo cuando sea necesario para realizar las tareas. Se llama "acceso aleatorio" porque la CPU puede acceder directamente a una sección de la memoria principal, y no debe emprender el proceso en un orden secuencial.

La RAM es uno de los tipos más rápidas de memoria, y tiene la capacidad de permitir que los datos sean leídos y escritos. Cuando el ordenador está apagado, todo el contenido almacenado en RAM se purga.

Las memorias principales actuales son de dos tipos:

- **Memoria (DRAM):** Es la memoria dinámica de acceso aleatorio.
- **Memoria (SRAM):** Es la memoria estática de acceso aleatorio.

La unidad central de proceso es uno de los componentes más importantes en el equipo. Es el lugar donde se realizan diversas tareas y se genera una salida. Cuando el microprocesador completa la ejecución de un conjunto de instrucciones, y está listo para realizar la siguiente tarea, recupera la información que necesita de RAM. Típicamente, las direcciones incluyen la dirección donde se encuentra la información que debe ser leída. La CPU envía la dirección al controlador de la RAM, que pasa por el proceso de localización de la dirección y la lectura de los datos.

La CPU se considera a menudo el elemento más importante en el rendimiento de una computadora personal. La RAM probablemente viene en un cercano segundo lugar. Tener una adecuada cantidad de RAM tiene un efecto directo sobre la velocidad de la computadora. Un sistema que carece de suficiente memoria principal para ejecutar sus aplicaciones debe confiar en el sistema operativo para crear recursos adicionales de memoria de la unidad de disco duro "intercambiando" los datos de entrada y salida. Cuando la CPU debe recuperar los datos del disco en lugar de RAM, se ralentiza el rendimiento del equipo. Muchos juegos, programas de edición de vídeo o gráficos requieren una gran cantidad de memoria para funcionar a un nivel óptimo.

Tener suficiente memoria principal en un equipo se inicia con el cumplimiento de la cantidad recomendada de memoria para el sistema operativo.

Formas de la memoria principal

Aunque las funciones de la memoria principal son básicas, hay muchas variaciones del hardware en el mercado hoy en día. Como memoria principal, la memoria ROM se muestra como el BIOS del ordenador, o Basic Input Output System. Todos los chips BIOS contienen información que permite el uso del teclado, monitor y drives. Muy pocos chips BIOS tienen la opción de sobrescribir pero algunos modelos utilizan impulsos eléctricos para borrar información, como el Electrically Erasable Programmable Read Only Memory, o EEPROM. Por otro lado, la memoria RAM tiene muchas variaciones y es completamente modificable. Revisa el manual de tu computadora para saber qué tipo de memoria RAM puedes utilizar.

¿Cuál es la diferencia entre memoria RAM y ROM?

La memoria RAM significa Random Access Memory o Memoria de Acceso Aleatorio. La memoria ROM significa Read Only Memory, o Memoria de Sólo Lectura.

Sus nombres ya dicen mucho de las diferencias de funcionamiento, aunque no aclaran el fondo.

¿Qué es y para qué sirve la memoria RAM?:

Como hemos visto, la memoria RAM es una memoria de acceso rápido y volátil. En ella se almacenan los datos necesarios para que los procesos ejecutados por el procesador funcionen correctamente. A más procesos abiertos, más memoria RAM estará ocupada tratando de tenerlos 100% listos en todo momento.

Este es el motivo por el que cuando abres muchos programas o aplicaciones, tu ordenador, portátil, tablet o móvil van más lentos, llegando a salir un mensaje parecido a este: “No hay espacio para más procesos en la memoria RAM”.

Como las aplicaciones cada vez son más completas y ocupan más RAM, los dispositivos van incorporando más y más memoria. Hasta hace poco era impensable ver PCs con 32 GB de RAM o móviles con 8 GB, pero ya están estandarizándose en él.

Es de acceso aleatorio porque no hay que leer los datos que almacena siguiendo un orden. El procesador puede recuperar información de distintos sectores de la memoria independientemente de cuáles sean.

¿Qué es y para qué sirve la memoria ROM?:

La memoria ROM responde a otro concepto, todos los ordenadores y móviles la tienen. En el caso de los smartphones han saltado a la fama porque entre sus características técnicas suele especificarse qué cantidad de ROM incorpora.

Esta memoria de sólo lectura no se escribe y reescribe constantemente como la RAM. Su fin es almacenar datos sensibles, clave para el funcionamiento del sistema. Es donde se instala Windows, Android o el sistema operativo que corresponda.

Memoria Secundaria

La memoria secundaria, (almacenamiento secundario), memoria auxiliar, memoria periférica o memoria externa, también conocida como almacenamiento secundario, es el conjunto de dispositivos y soportes de almacenamiento de datos que conforman el subsistema de memoria del ordenador, junto con la memoria primaria o principal.

Puede denominarse periférico de almacenamiento o “memoria práctica”, en contraposición a la “memoria central”, porque en ocasiones puede considerarse como periférico de Entrada/Salida. Un ejemplo muy claro de una memoria secundaria la tenemos en el disco duro de nuestro ordenador, el cual se usa para almacenar el sistema operativo y otros programas. Otras formas de memorias secundarias incluyen los CDs que usamos para grabar música o almacenar datos y también los DVDs. También podemos incluir las memorias flash, memorias USB a esta categoría.

¿Cómo se utiliza el almacenamiento secundario?: Normalmente la memoria secundaria viene en forma de hardware que debe ser insertado en la placa base o en alguno de los puertos que tiene el ordenador.

La memoria secundaria es un tipo de almacenamiento masivo y permanente, (no volátil), con mayor capacidad para almacenar datos e información que la memoria primaria, que es volátil, aunque la memoria secundaria es de menor velocidad.

No es accedida directamente por la unidad de procesamiento central o CPU.

Deben diferenciarse los “dispositivos o unidades de almacenamiento” de los “soportes o medios de almacenamiento”, porque los primeros son los aparatos que leen o escriben los datos almacenados en los soportes de almacenamiento.

Relación entre memoria principal y memoria secundaria

Hay que entender la diferencia entre la memoria principal y su contraparte, la memoria secundaria, para tener un entendimiento completo de toda la memoria del ordenador. Físicamente, la memoria principal procesa una menor cantidad de información, comparada con la memoria secundaria, la cual usualmente puede ser removida de tu computadora. La memoria principal trabaja más rápido que la memoria secundaria pero también tiene un mayor costo. La memoria principal está formada por un conjunto de chips de silicio mientras que la memoria secundaria puede ser el disco duro de metálico o memoria flash drive echa también de silicio, que almacena la información después de apagar el PC.

Por ejemplo: La memoria RAM no guarda la información a largo plazo, la pierde en cuanto apagas el ordenador. Aunque hay también memoria principal ROM, donde puedes guardar información, esta información no se puede modificar fácilmente, sólo unos pocos pueden modificarla con un aparato electrónico especial hecho para tal fin.

La memoria secundaria guarda la información aunque el ordenador esté apagado, haciendo que sea necesario que archive información a largo plazo.

Sistema Operativo

El sistema operativo (SO), es el programa o conjunto de programas más importantes de un ordenador o computadora. Es el que gestiona todos los recursos del equipo tanto a nivel de software (programas), como a nivel de hardware (disco duro, monitor, teclado, mouse, entre otros).

El sistema operativo facilita la interacción del usuario con el equipo en el que ha sido instalado y, en la actualidad, no solo están presentes en los ordenadores, sino también en los smartphones, tablets y en muchos de los dispositivos digitales que utilizamos diariamente.

La complejidad del sistema operativo dependerá de las funciones que debe realizar una vez haya sido instalado en el equipo.

¿Cuáles son las funciones del sistema operativo en un ordenador?

El sistema operativo trabaja para que los usuarios que no tienen conocimientos de programación ni conozcan los detalles del hardware puedan utilizar su equipo y periféricos sin ningún problema.

Debe facilitar al usuario la forma de acceder a sus programas y archivos, la manera de conectarse a Internet y todas sus tareas cotidianas. También debe ejecutar los programas y ayudarlos a que, de una manera sencilla, puedan acceder a los recursos que necesitan del sistema.

Tareas o funciones más importantes de un sistema operativo:

- **1. Administración del procesador:** Esta tarea consiste en asegurar que todas las aplicaciones y procesos reciben la cantidad de tiempo necesaria del procesador como para funcionar de forma correcta. Además de aprovechar la cantidad máxima posible de ciclos del procesador para que todo funcione en conjunto de manera adecuada. El sistema operativo utiliza procesos o subprocesos del procesador para llevar a cabo estas funciones y cambia de forma continua entre un proceso y otro a una velocidad de miles de procesos por segundo.
- **2. Administración de la memoria:** El sistema operativo debe garantizar que cada proceso tenga la cantidad de memoria suficiente para ejecutar la tarea que le corresponde, de modo que los procesos no se quiten memoria unos a otros. Además, debe hacer que cada tipo de memoria se utilice de forma correcta. Cuando un programa está en ejecución, el sistema operativo continúa trabajando. Por ejemplo, muchos programas necesitan acceso al teclado, la impresora o a la tarjeta de video. Todos esos accesos los permite el sistema operativo, que se mantiene activo siempre, ayudando a los programas que se están ejecutando.
- **3. Administración de almacenamiento secundario:** La memoria RAM es pequeña y volátil para todos los datos y programas que se deben guardar, por este motivo la mayoría de la información se almacena en discos. El sistema operativo es el responsable de asignar el espacio que ocupará esa información en los discos y de administrar el espacio libre. Además, debe decidir quién utiliza un recurso del disco cuando existe competencia por él.
- **4. Administración de dispositivos:** Cada pieza de hardware utiliza un controlador, un programa especial, para comunicarse con el sistema. El sistema operativo utiliza los controladores como traductores entre las señales eléctricas del hardware y el código de programación que se encuentra en las aplicaciones. El controlador toma datos del sistema operativo, los pasa al dispositivo y viceversa. El sistema operativo maneja este proceso llamando al controlador indicado cuando es necesario.

- **5.- Interfaz de aplicación:** Del mismo modo que el hardware tiene controladores, las aplicaciones tienen una interfaz de programación de aplicaciones ,(API, por sus siglas en inglés). Las API permiten a los programadores utilizar partes del sistema operativo y del equipo para llevar a cabo determinadas funciones. El sistema operativo contiene todas las API que son reconocidas por el equipo y juega el papel de intérprete enviando los datos necesarios para llevar a cabo la función.
- **6.- Interfaz de usuario:** Muchos sistemas operativos utilizan una interfaz gráfica de usuario, es decir, que utiliza imágenes e íconos para comunicarse con el usuario. El sistema operativo juega nuevamente el rol de intérprete para establecer la comunicación entre el usuario y el equipo en un idioma que ambos entiendan.

Tipos de Sistemas operativos: Según el equipo en el que deben funcionar, los sistemas operativos se dividen en:

- **Sistemas operativos para PC:** entre los más más conocidos se encuentran Microsoft Windows, Mac OS y Linux.
- **Sistemas operativos para Móviles y tablets:** Windows Phone, Android y iOS son los más populares.

3.3.3. Relación entre hardware - software

Como ya vimos en la introducción, un **sistema informático** es un **conjunto de partes que funcionan relacionándose entre sí con un objetivo preciso, procesar datos que han sido almacenados**. Sus partes son: **hardware, software y las personas que lo usan**.

Por ejemplo: Un ordenador, sus dispositivos, periféricos y la persona que la maneja, pueden constituir un sistema informático.

En un sistema informático, la información es introducida a través de los periféricos de entrada, luego es procesada y mostrada por los periféricos de salida. Como se puede ver la relación entre hardware y software se puede centrar desde dos puntos de vista:

- **Desde el punto de vista del sistema operativo:** El sistema operativo es el encargado de coordinar al hardware durante el funcionamiento del ordenador, actuando como intermediario entre éste y las aplicaciones que están corriendo en un momento dado.
- **Desde el punto de vista de las aplicaciones:** Una aplicación no es otra cosa que un conjunto de programas, y éstos están escritos en algún lenguaje de programación que el hardware del equipo debe interpretar y ejecutar. Hay multitud de lenguajes de programación diferentes, sin embargo, todos tienen algo en común: estar escritos con sentencias de un idioma que el ser humano puede aprender y usar fácilmente. Por otra parte, el hardware de un ordenador sólo es capaz de interpretar señales eléctricas (ausencias o presencias

de tensión) que en informática se traducen en secuencias de 0 y 1 (código binario). Tendrá que pasar algo (un proceso de traducción de código) para que el ordenador ejecute las instrucciones escritas en un lenguaje de programación.

Software

Recuerda que se conoce como software al equipamiento lógico o soporte lógico de una computadora digital; comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas, en contraposición a los componentes físicos del sistema, llamados hardware.

Hardware

Recuerda que hardware corresponde a todas las partes tangibles de un sistema informático; sus componentes son: eléctricos, electrónicos, electromecánicos y mecánicos. Son cables, gabinetes o cajas, periféricos de todo tipo y cualquier otro elemento físico involucrado.

Clasificación del hardware:

- **Hardware de procesamiento:** Corresponde a la Unidad Central de Procesamiento o CPU, el centro de operaciones lógicas de la computadora, en donde se interpretan y ejecutan las tareas necesarias para el funcionamiento del resto de los componentes.
- **Hardware de almacenamiento:** Como indica su nombre, hace referencia a todos los componentes cuya función es resguardar la información para que el usuario pueda acceder a ella en cualquier momento. El dispositivo principal en este caso es la memoria RAM (Random Access Memory), pero también está conformada por memorias secundarias, como los discos duros o las memorias SSD o USB.
- **Dispositivos periféricos:** Los dispositivos periféricos son dispositivos externos al ordenador que permiten la comunicación entre las personas y los ordenadores, como la entrada y salida de información desde o hacia el mismo ordenador. Se llaman periféricos porque se sitúan en la periferia de la computadora y sirven de memoria auxiliar de la memoria principal. Éstos se subdividen en 5 categorías:
 - Dispositivos periféricos de sólo entrada. Ejemplos: Teclado, ratón y micrófono.
 - Dispositivos periféricos de sólo salida. Ejemplos: Monitor, impresora y altavoces.
 - Dispositivos periféricos de entrada y salida. Ejemplos: Pantalla táctil, impresora multifunción.
 - Dispositivos periféricos de almacenamiento. Ejemplos: Disco duro, memoria flash y cinta magnética.
 - Dispositivos periféricos de comunicación. Ejemplos: Fax-módem, tarjeta de red y router.

Nota-02: Hay autores que engloban en el mismo grupo los dispositivos periféricos de E/S y almacenamiento.

3.3.4. Tipos de software. BIOS. Sistema. Aplicaciones

Tipos de software:

- **BIOS:** Corresponde a la Unidad Central de Procesamiento o CPU, el centro de operaciones lógicas de la computadora, en donde se interpretan y ejecutan las tareas necesarias para el funcionamiento del resto de los componentes.
- **Firmware:** El *firmware* o *soporte lógico inalterable*, es un bloque de instrucciones de máquina para propósitos específicos, grabado en una memoria, normalmente de lectura / escritura (ROM, EEPROM, flash, etc.), que establece la lógica de más bajo nivel que controla los circuitos electrónicos de un dispositivo de cualquier tipo. Es la parte intangible, (Software específico) de componentes del Hardware. Está fuertemente integrado con la electrónica del dispositivo siendo el software que tiene directa interacción con el hardware. En resumen, un firmware es el software que maneja al hardware.
- **Software de sistema:** Es el programa responsable de la ejecución de todas las aplicaciones necesarias para que un sistema opere correctamente. Suele confundirse con el sistema operativo, pero va más allá, ya que también incluye las herramientas de optimización, los controladores de dispositivo y los servidores, entre otros componentes.
- **Software de programación:** Son todas las herramientas que permiten el desarrollo de nuevo software. Por ello, no suelen ser utilizadas por el usuario final sino por los programadores. Incluye, por mencionar los más importantes, los siguientes:
 - Compiladores: Programas que analizan y ejecutan otros programas.
 - Intérpretes: Software que "traduce" los diferentes lenguajes de programación.
 - Editores de texto: Crean y modifican archivos de texto, lenguaje de programación o código fuente.
- **Software de aplicación:** Un software de aplicación es un programa diseñado para facilitar algunas tareas específicas para cualquier medio informático, ya sea en computadores, tablets o móviles.
- **Software malicioso:** Utilizado también en su forma original en inglés malicious software o simplemente malware, es el que se crea con fines ilícitos como obtener información privada del usuario o dañar el sistema.

3.3.5. Código fuente, código objeto y código ejecutable; máquinas virtuales

En la etapa de diseño construimos las herramientas de software capaces de generar un código fuente en lenguaje de programación. Estas pueden ser los diagramas de flujo o el pseudocódigo.

La etapa de codificación es la encargada de generar el código fuente y pasa por diferentes estados.

Tipos de código:

Cuando escribimos un código pasa por distintos estados hasta que se ejecuta:

- **Código Fuente:** Es el código realizado por los programadores usando algún editor de texto o herramienta de programación. Posee un lenguaje de alto nivel y para escribirlo se parte de los diagramas de flujo o pseudocódigos. No se puede ejecutar directamente en el ordenador.
- **Código Objeto:** Es el código que se crea tras realizar la compilación del código fuente. Este código no es entendido ni por el ordenador ni por nosotros. Es una representación intermedia de bajo nivel.
- **Código Ejecutable:** Este código se obtiene tras unir el código objeto con varias librerías para que así pueda ser ejecutado por el ordenador.

La **compilación** es el **proceso a través del cual se convierte un programa en lenguaje máquina** a partir de otro programa fuente de computadora escrito en otro lenguaje. La compilación **se realiza a través de dos programas: el compilador y el enlazador**. Si en el compilador se detecta algún tipo de error no se generará el código objeto y tendremos que modificar el código fuente para volver a pasarlo por el compilador.

Proceso:

Programa Fuente → Compilador → Programa Objeto → Enlazador + Librerías → Programa Ejecutable

Máquinas Virtuales

Una máquina virtual es un tipo de software capaz de ejecutar programas como si fuese una máquina real. Se puede decir que es tener un ordenador independiente dentro de otro ordenador. Se clasifican en dos categorías:

- **Máquinas virtuales de sistema:** Nos permiten virtualizar máquinas con distintos sistemas operativos en cada una. Un ejemplo son los programas VMware Workstation o Virtual Box que podremos usar para probar nuevos sistemas operativos o ejecutar programas.
- **Máquinas virtuales de proceso:** Se ejecutan como un proceso normal dentro de un SO y sólo soporta un proceso. Se inician cuando lanzamos el proceso y se detienen cuando éste finaliza. El objetivo es proporcionar un entorno de ejecución independiente del hardware y del sistema operativo permitiendo que el programa sea ejecutado de la misma forma en cualquier plataforma. Ejemplo de ello es la máquina virtual de Java.

Las máquinas virtuales requieren de grandes recursos por lo que hay que tener cuidado y ejecutarlas en ordenadores capaces de soportar los procesos que requieren dichas máquinas para que no funcionen lentas o se colapsen.

La máquina virtual de Java

Los programas que se compilan en lenguaje Java son capaces de funcionar en cualquier plataforma (UNIX, Mac, Windows, Solaris, etc.). Esto se debe a que el código no lo ejecuta el procesador del ordenador sino la propia **Máquina Virtual de Java (JVM)**. El **funcionamiento básico** de la máquina virtual es el siguiente:

- 1º)** El código fuente estará escrito en archivos de texto planos con la extensión .java.
- 2º)** El compilador javac generará uno o varios archivos siempre que no se produzcan errores y tendrán la extensión .class.
- 3º)** Este fichero .class contendrá un lenguaje intermedio entre el ordenador y el SO y se llamará bytecode.
- 4º)** La Java VM coge y traduce el bytecode en código binario para que el procesador de nuestro ordenador sea capaz de reconocerlo. Los ficheros .class podrán ser ejecutados en múltiples plataformas

Entre las **tareas que puede realizar la máquina virtual Java** están:

- La **reserva de espacio** para objetos creados y liberar aquella memoria que no se usa.
- **Comunicación con el sistema** en el que se ejecuta la aplicación para varias funciones.
- **Observar que se cumplen las normas de seguridad para las aplicaciones Java.**

Una de las desventajas de usar este tipo de lenguajes que se basan en una máquina virtual puede ser que son más lentos que los lenguajes ya compilados, debido a la capa intermedia. No obstante, cabe destacar que no es una desventaja demasiado crítica.

4.- LENGUAJES DE PROGRAMACIÓN:

4.1.- ¿Qué es un lenguaje de programación?.

❖ *Un lenguaje de programación se puede definir:*

- Se trata de un lenguaje artificial , creado por programadores.
- Permiten traducir las instrucciones de un programa a código comprensible por el ordenador.
- Son lenguajes mucho más complejos y precisos que el lenguaje máquina (01010101).
- Cada instrucción puede dar lugar a muchas instrucciones de lenguaje máquina.
- Se componen de un conjunto de símbolos (sintaxis) y reglas (semántica).

❖ *Conceptos importantes:*

- Lenguaje máquina: Lenguaje que sólo es comprensible por el ordenador (01010101).
- Programa: Conjunto de instrucciones escritas en un lenguaje de programación.
- Instrucción: Cada una de las sentencias u órdenes que forman parte de un programa.
- Palabra reservada: Cada uno de los símbolos (léxico) que componen la sintaxis de un lenguaje.
- Semántica: Reglas que definen la combinación de los símbolos de un lenguaje de programación.

4.2.- Tipos de lenguajes de programación.

❖ *Características:*

- Existen diferentes tipos de lenguajes de programación.
- Los lenguajes de programación han evolucionado a lo largo de los años.
- Actualmente, los lenguajes son más amigables y facilitan mucho la tarea del programador.
- A veces, la facilidad a la hora de programar implica crear programas más lentos y pesados.
- En la actualidad existen infinidad de lenguajes de programación:
Fortran, Cobol, Pascal, C, C++, Basic, Visual Basic, C#, Java, JavaScript, PHP, Python....

❖ **Tipos de lenguaje de programación – clasificación según la evolución temporal:**

- Lenguaje de bajo nivel – lenguaje máquina.
- Lenguaje de medio nivel – lenguaje ensamblador.
- Lenguajes de alto nivel – lenguajes naturales.

❖ **Lenguaje de bajo nivel - lenguaje máquina:**

- Posee instrucciones complejas e ininteligibles (01010101).
- Necesita ser traducido, (es el único lenguaje que entiende directamente el ordenador).
- Fue el primer lenguaje usado (muy dependiente del hardware).
- Difiere para cada procesador, (las instrucciones son diferentes de un ordenador a otro).
- En la actualidad sólo se usa para determinados módulos de un sistema operativo como driver.

❖ **Lenguaje de medio nivel - lenguaje ensamblador:**

- Facilita la labor de programación.
- Se centra en el hardware, pero usa mnemotécnicos (ADD...) comprensibles por el programador.
- Hay que compilarlos, (traducirlos) para que sean comprensibles por el ordenador.
- Trabaja con los registros del procesador y direcciones de memoria físicas.
- A pesar de estas mejoras, es difícil de entender y de usar.

❖ **Lenguajes de alto nivel - lenguajes naturales:**

- Poseen una forma de programar intuitiva y sencilla.
- Son más cercanos a los lenguajes humanos (IF , WHILE , DO...).
- Incorporan librerías y funciones predeterminadas que ayudan al programador en su labor.
- Suelen ofrecer frameworks, (marco de trabajo o estructura) , que incorporan funciones y componentes que facilitan el desarrollo.
- La mayoría de los lenguajes de programación actuales se engloban en esta categoría. **Por ejemplo:** Lenguajes C/C++, Java, Python, JavaScript y PHP.

4.3.- Paradigmas de programación.

Se denominan paradigmas de programación a las formas de clasificar los lenguajes de programación en función de sus características y estilo de programación.

❖ **Características:**

- Los lenguajes de programación pueden usarse de formas diferentes.
- Un paradigma de programación define la forma en la que se desarrolla el programa.
- Existen distintos tipos de paradigmas, su clasificación por características sería:
 - Programación estructurada o imperativa.
 - Programación funcional.
 - Programación orientada a objetos (POO).
 - Programación lógica.

❖ **Programación estructurada o imperativa:**

- Consiste en una secuencia ordenada y organizada de instrucciones.
- Es fácil de entender.
- Programas sencillos y rápidos.
- Posee tres estructuras básicas: *secuencia, selección e iteración*.
- *Inconveniente*: Un único bloque de programa (es inmanejable si crece).
- **Por ejemplo**: FORTRAN, Pascal, C, Ada, etc.

❖ **Programación funcional:**

- Consiste en descomponer el programa en funciones o módulos.
- Programa modular y estructurado.
- *Inconveniente* : El programa puede crecer en gran medida y ser complejo.
- **Por ejemplo**: Haskell, Miranda, Scala, SAP, etc.

❖ **Programación orientada a objetos (POO):**

- Consiste en representar entidades del mundo real.
- Permite reutilizar código.
- Principio de separación de responsabilidades.
- Patrones de diseño y paradigmas avanzados (MVC (Modelo-Vista-Controlador)...).
- Polimorfismo, herencia y encapsulación.
- Los más utilizados son: **Por ejemplo:** C++, Java, PHP, Python, Perl, etc.

❖ **Programación lógica:**

- Consiste en representar predicados y relaciones.
- Uso de lógica de predicados de primer orden.
- Muy utilizado en aplicaciones de inteligencia artificial.
- **Por ejemplo:** Prolog y Lisp.

5.- INTRODUCCIÓN A LA INGENIERÍA DEL SOFTWARE:

ESTE APARTADO SE DARÁ EN LA 2ª EVALUACIÓN