# Parte I: El computador y el proceso de programación

- 1.Introducción a los computadores y su programación
- 2. Introducción al análisis y diseño de algoritmos
- 3. Introducción al análisis y diseño de programas
- 4. Verificación de programas



## Notas:

**1. Introducción a los computadores y su programación**

- Arquitectura básica de un computador.El software del sistema. Lenguajes de Alto Nivel. El proceso de compilación. El ciclo de vida del software.

**2. Introducción al análisis y diseño de algoritmos.**

- Diseño de un programa. Concepto de algoritmo. Descripción de algoritmos: el pseudolenguaje. Tiempo de ejecución de algoritmos. La notación O(n). Ejemplos de análisis.

**3. Introducción al análisis y diseño de programas**

- Actividades del ciclo de vida del software. Paradigmas de desarrollo de programas. Análisis y especificación. Diseño arquitectónico. Técnicas de diseño detallado.

**4. Verificación de programas**

- Importancia de la verificación. Estrategias de prueba. Depuración. Elección de datos para la prueba.

# 1.1 Arquitectura básica de un computador

## Un computador es una máquina que:

- acepta información de entrada
- la procesa ejecutando paso a paso una secuencia de instrucciones o programa
- y produce una información de salida.

## El computador está por tanto compuesto por un equipo electrónico (hardware) y un conjunto de programas (software)

## El computador puede realizar dos tipos de instrucciones:

- acciones
- decisiones



## Notas:

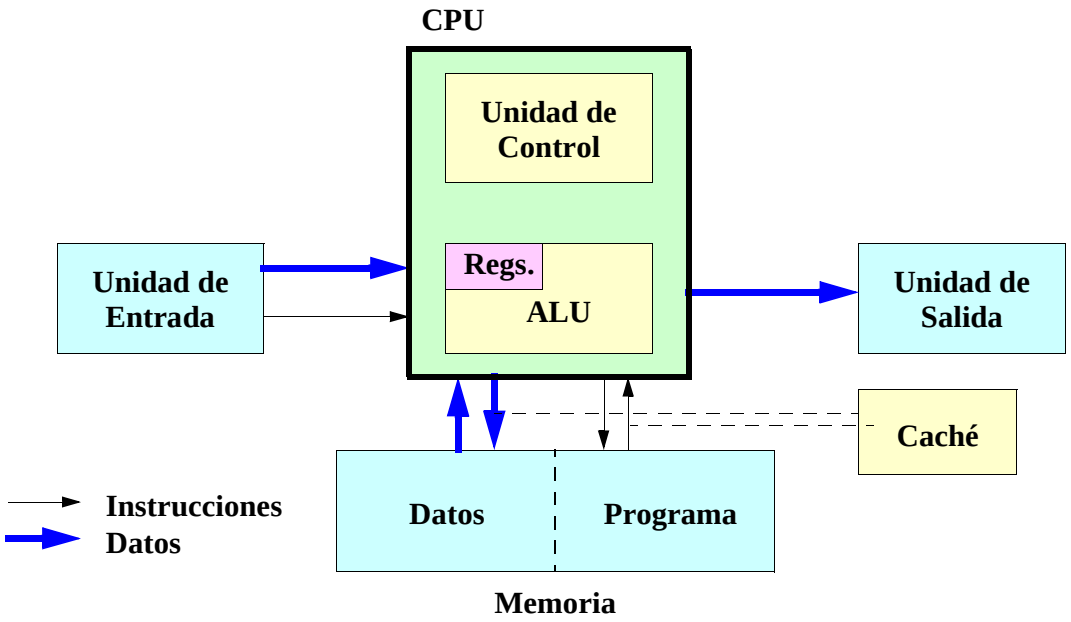
Un computador es una máquina que acepta información de entrada y la procesa ejecutando paso a paso una secuencia de instrucciones que posee almacenada, a la que llamamos programa, produciendo una información de salida.

Por ello, el computador está compuesto por un equipo electrónico, que denominamos hardware, y un conjunto de programas, que denominamos software.

Una de las características que distinguen al computador de una máquina automática es la capacidad de realizar dos tipos de instrucciones:

- **Acciones**. Tienen como consecuencia la modificación o intercambio de la información almacenada en el computador.
- **Decisiones**. Permiten comprobar el propio estado del computador y, en función del mismo, elegir uno de dos caminos alternativos.

# Estructura de un computador






## Notas:

Los computadores actuales son sistemas electrónicos digitales complejos capaces de realizar muchos millones de operaciones por segundo. Los computadores de propósito general presentan una estructura básica denominada “máquina de Von Neumann”, compuesta de cinco unidades funcionales:

- UNIDAD DE ENTRADA. Está compuesta por aquellos elementos a través de los cuales el computador recibe información de exterior, tanto en lo que se refiere a datos como a programas. Sistemas típicos que forman la unidad de entrada son: teclados, digitalizadores, instrumentos de medida, etc.
- UNIDAD DE SALIDA. Compuesta por el conjunto de dispositivos destinados a proporcionar al exterior la información generada por el computador. Ejemplos típicos son: pantalla, impresora, plotter, máquina herramienta, etc.
- MEMORIA. Está compuesta por el conjunto de circuitos y dispositivos electrónicos destinados a almacenar y retener los diferentes tipos de información que el computador utiliza o genera.
- UNIDAD DE CONTROL. Es la parte central del computador, encargada de interpretar las instrucciones del programa y ordenar su ejecución, generando las señales de control adecuadas que activan las diferentes unidades del computador.
- UNIDAD ARITMETICA Y LOGICA (ALU). Realiza las operaciones de transformación, comparación y transferencia de datos. La ALU y la unidad de control se integran, normalmente, en un sistema único denominado CPU (Central Processing Unit).

# La Unidad Central de Procesado (CPU)

## Integra la Unidad de Control y la Unidad Aritmética y Lógica

## Funciona sincronizada con un reloj

## La capacidad de cálculo de la CPU depende de ese reloj, y de la arquitectura interna

## La Unidad de Control controla las señales electrónicas internas del computador. Funciona repitiendo:

- a) lee de la memoria una instrucción del programa
- b) comprueba cuál es su significado
- c) organiza la ejecución de esa instrucción



## Notas:

La ALU y la unidad de control se integran, normalmente, en un sistema único denominado CPU (Central Processing Unit, o Unidad Central de Procesado). Funciona mediante un "*reloj*" que indica mediante una señal electrónica los instantes en los que se hace cada operación.

El reloj de la CPU funciona a una frecuencia que suele medirse en *Megahercios* (MHz). Por ejemplo, una CPU de 500 MHz tiene un reloj que oscila 500 millones de veces por segundo.

- Para dos CPUs del mismo tipo, la velocidad de operación depende de la velocidad del reloj. Por ejemplo es más rápida una CPU de 500 MHz que una de 200MHz.
- Sin embargo, si dos CPUs son de distinto tipo, la velocidad ya no depende sólo del reloj. Por ejemplo, un Pentium-III a 300MHz puede ser más rápido que un Pentium a 350 MHz

La CPU tiene dos partes: La Unidad de Control, y la Unidad Aritmética y lógica.

- *Unidad de control*: controla mediante señales electrónicas el funcionamiento de todo el computador. Funciona repitiendo constantemente lo siguiente:
  - a) lee de la memoria una instrucción del programa.
  - b) comprueba cuál es su significado.
  - c) organiza la secuencia de acciones a realizar por cada elemento electrónico del computador para ejecutar la instrucción leída

# La Unidad Aritmética y Lógica

## Hace cálculos sencillos con números

## Manipula textos también mediante operaciones numéricas

## Tiene un conjunto pequeño de memorias: los registros de la CPU



## Notas:

*La unidad aritmética y lógica*:

- Hace todos los cálculos con números y textos. Por ejemplo, sumas, restas, multiplicaciones, divisiones, comparaciones, etc.
- Tiene un conjunto pequeño de memorias denominadas *registros de la CPU.*
  - En cada registro cabe un dato.
  - Se usan para guardar los datos que se están usando en ese momento

# La memoria

## Almacena dos tipos fundamentales de información:

- datos
  - numéricos
  - texto
- programas

## En un computador suele haber varios tipos de memoria:

- registros de la CPU
- memoria de caché
- memoria principal
- memoria secundaria o masiva



## Notas:

Existen dos tipos básicos de información:

- Datos, que se utilizan o generan en la ejecución del programa. Pueden ser datos numéricos o datos con información alfanumérica (texto).
- Programa, constituido por la secuencia de instrucciones a ejecutar.

La memoria de un computador se puede clasificar en varios niveles, segun su capacidad y tiempo de acceso:

- Registros de la CPU, internos a la misma, y de acceso muy rápido
- Memoria de caché: Situada entre la CPU y la memoria principal. Es una memoria electrónica de alta velocidad y poca capacidad. Se intenta colocar en ella los datos e instrucciones más frecuentemente usados por el computador.
- Memoria principal. Electrónica, directamente accesible en tiempos muy cortos, pudiendo a su vez ser memoria de lectura y escritura (RAM) o de solo lectura (ROM).
- Memoria masiva. Normalmente, en soportes magnéticos, con gran capacidad de almacenamiento pero con unos tiempos de acceso mucho mas altos que la anterior. Como ejemplo, los discos, diskettes, cintas, y CD-ROM. La información de la memoria masiva se almacena en “trozos” o unidades llamadas *ficheros* o *archivos*, que se distinguen de otros por su nombre. Los ficheros se guardan en *carpetas* o *directorios*. A su vez, una carpeta puede contener más carpetas, con otras carpetas o ficheros en su interior.

# La memoria: capacidad

## Existen diversas unidades de medida:

- **bit: una cifra binaria (0, 1)**
- **byte: 8 bits**
- **kilobyte (*Kb*): 1024 bytes**
- **megabyte (*Mb*): 1024*1024 bytes**
- **gigabyte (*Gb*): 1024*1024*1024 bytes**
- **terabyte (*Tb*): 1024 gigabytes**



## Notas:

La memoria es un elemento fundamental de un computador. Su capacidad y tiempo de acceso fijan, en gran medida, la potencia del mismo. La Capacidad de la memoria: se mide en bits, bytes, kilobytes (Kb), megabytes (Mb), y Gigabytes (Gb):

- *Bit*: Es la unidad más pequeña de información. Representa el estado de un circuito digital, que puede estar encendido o apagado. Esto se usa para representar la cifra numérica 0 o la cifra numérica 1. Con estas dos cifras se puede componer toda la información numérica o de texto del computador.
- *Byte*: Representa un dato pequeño, generalmente formado por 8 bits. Combinando 8 bits (es decir, 8 ceros o unos) se pueden representar números pequeños comprendidos entre 0 y 255. Estos números se pueden usar para representar caracteres alfanuméricos. Así, un texto con 2000 caracteres se puede almacenar en 2000 bytes (aunque hay representaciones de caracteres en distintos alfabetos que ocupan más bytes). Un número entero o real mayor necesita varios bytes para ser almacenado. Típicamente entre 2 y 8 bytes cada uno.
- *Kilobyte*: Son 1024 bytes (o aproximadamente 1000)
- *Megabyte*: son 1024x1024=1.048.576 bytes (o aproximadamente un millón de bytes)
- *Gigabyte*: son 1024x1024x1024=1.073.741.824 bytes (o aproximadamente mil millones de bytes)
- Terabyte: 1024 gigabytes, o aproximadamente un billón de bytes

# La memoria: tiempo de acceso y anchura de banda

## Otros de los factores importantes de una memoria:

- ***tiempo de acceso***: **tiempo que se tarda en promedio en obtener un dato de la memoria, o en modificarlo**
- ***anchura de banda***: **cantidad de datos que se pueden transferir por unidad de tiempo**



## Notas:

El tiempo de acceso a la memoria es el tiempo que se tarda en acceder a un dato en la memoria se suele expresar en segundos, *milisegundos* (milésimas de segundo), *microsegundos* (millonésimas de segundo), o *nanosegundos* (milmillonésimas de segundo).

La capacidad típica y la velocidad aproximada de las distintas memorias en computadores personales es

- *Registros de la CPU*: unas decenas o centenas de bytes, acceso en nanosegundos o menos
- *Memoria caché*: cientos de kilobytes, acceso en nanosegundos o menos
- *Memoria principal*: gigabytes, acceso en nanosegundos o decenas de nanosegundos, velocidad gigabytes por segundo
- *Memoria secundaria*:
  - Discos duros: centenares de Gb, acceso en varios milisegundos, velocidad 100 Mb/s
  - CD-ROM: 640 Mb, acceso en segundos
  - DVD: 4.3Gb, acceso en segundos, velocidad decenas de Mb/seg
  - Blu-ray: 46 Gb, acceso en segundos, velocidad: decenas de Mb/seg
  - Memorias Flash (memorias USB): 16 Gb, velocidad : decenas de Mb/s (lectura)

# Unidades de entrada/salida

**Los elementos de *memoria secundaria* (disquetes, discos duros, cintas, CD-ROM, etc.) también se pueden catalogar como elementos de entrada/salida**

**Algunas unidades comunes:**

- **pantalla**
- **teclado**
- **ratón**
- **escáner**



## Notas:

Son de tipos muy variados. Algunas unidades de entrada/salida más comunes son:

- *Pantalla*: Suele ser un tubo de rayos catódicos (CRT), similar a un televisor, que presenta una imagen producida por una *tarjeta controladora de vídeo*. La controladora de vídeo usa memoria RAM para almacenar cada punto de la imagen, con sus colores. Por eso, para que la imagen tenga buena resolución (muchos puntos, con muchos colores posibles) necesita mucha memoria. Típicamente entre 2 y 64 Mb. Las resoluciones más habituales son:
  - VGA: 640 columnas (o puntos en cada línea horizontal) por 480 filas (o puntos en cada línea vertical)
  - SVGA: 800 por 600 puntos
  - XGA: 1024 por 728 puntos
- *Teclado*: Conjunto de teclas a las que se asocian caracteres alfanuméricos. Sirve para introducir texto.
- *Ratón*: Es un dispositivo apuntador, es decir, que sirve para apuntar en la pantalla a un objeto. Se utiliza en combinación con la pantalla para dar órdenes al computador, y también para dibujar
- Escáner: Permite convertir una foto, un gráfico o un texto impresos en papel a un formato digital, almacenado en la memoria del computador.

# Unidades de entrada/salida (cont.)

- **Impresora**
  - Matricial
  - Chorro de tinta
  - Láser
- **Terminal o consola**
- **Red de comunicación**
  - línea serie
  - línea paralelo
  - línea telefónica, con módem o ADSL
  - ethernet
  - líneas de fibra óptica
  - redes inalámbricas



## Notas:

- *Impresora*: Imprime sobre papel imágenes o textos. Las hay de distintos tipos:
  - Matricial o de agujas. Más lenta y anticuada, de poca calidad, pero de bajo coste de impresión
  - De chorro de tinta. Más rápida y de calidad media. Alto coste de impresión
  - Láser: Muy rápida y de gran calidad. Coste de impresión medio.
- *Terminal* o *consola*: es un dispositivo con pantalla y teclado desde el que podemos usar un computador.
- *Red de comunicación*. Hoy en día es uno de los elementos de entrada/salida más importantes. Permite la comunicación entre computadores. Así se puede trabajar desde un computador con la información que hay en otro. Existen diferentes tipos de redes de comunicación:
  - Línea serie: baja velocidad, corta distancia
  - Línea paralelo, velocidad media, muy corta distancia
  - Línea telefónica con módem. Velocidad baja, larga distancia. El módem es un dispositivo que convierte la información del computador en sonidos, que se transmiten por teléfono.
  - Línea telefónica ADSL. Velocidad media, larga distancia.
  - Red ethernet: muy alta velocidad, distancias medias (varios centenares de metros)
  - Red de fibra óptica: permite altísimas velocidades y largas distancias.

# La Internet

## Es una “red de redes” de ámbito mundial

## Cada computador tiene un *número de “IP”* que lo identifica, y un *nombre* perteneciente a un *dominio*.

## Protocolos de internet comunes:

- ftp
- mail
- telnet, ssh
- http



## Notas:

A través de una red, el computador se puede conectar a *Internet*, que es una red de redes mundial. A ella están conectadas la mayoría de las redes de computadores del mundo. A cada computador se le da un *número* (llamado número de IP, similar a un número de teléfono) y un nombre distinto. Así podemos acceder a cualquier computador conectado a Internet mediante su número o su nombre. El nombre suele ser compuesto por varias palabras separadas por puntos. Por ejemplo:

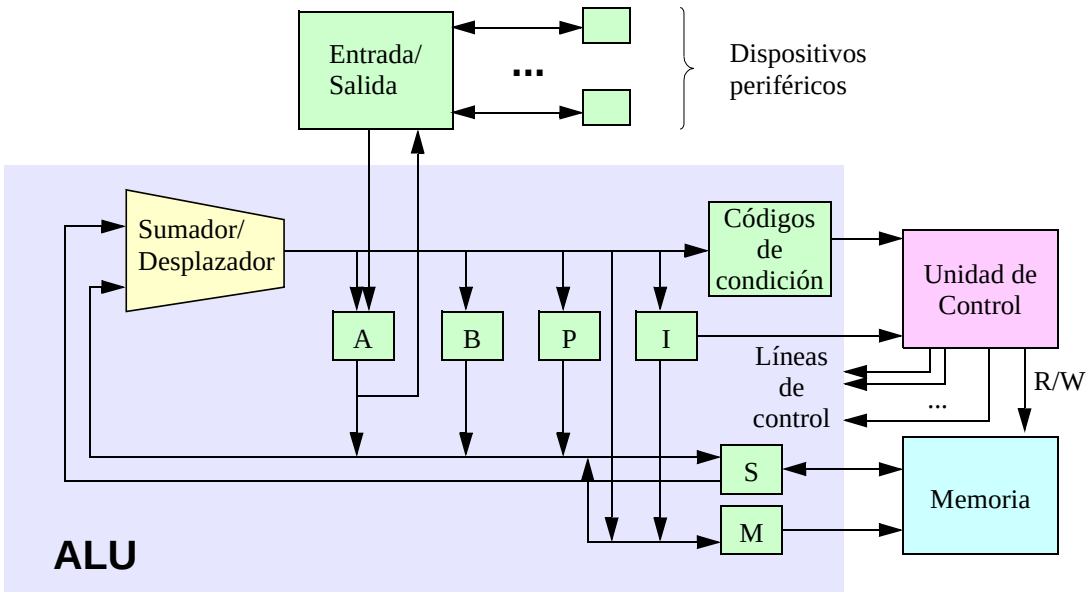
```
jupiter.unican.es
```

Jupiter es el nombre del computador. Unican es el nombre del dominio local. La última parte, “es” representa el dominio general (`.es` de España, `.com` de comercial, `.org` de organización, etc.)

Para usar la Internet es necesario hacerlo a través de uno de los protocolos de aplicación Internet:

- *ftp*: sirve para transferencia de ficheros con información
- *mail*: es el correo electrónico. Permite enviar mensajes entre distintos usuarios de Internet. Los mensajes pueden llevar “adheridos” ficheros con información
- *telnet*: permite conectarse a otro computador para trabajar con el desde un lugar remoto
- *http*: es el famoso “Web” (world wide web). Permite navegar, o acceder de forma simple a información almacenada en otros computadores, sin más que pulsar con el ratón sobre los “enlaces” que vemos en la pantalla, que son textos o gráficos.

# Computador de Von Neumann



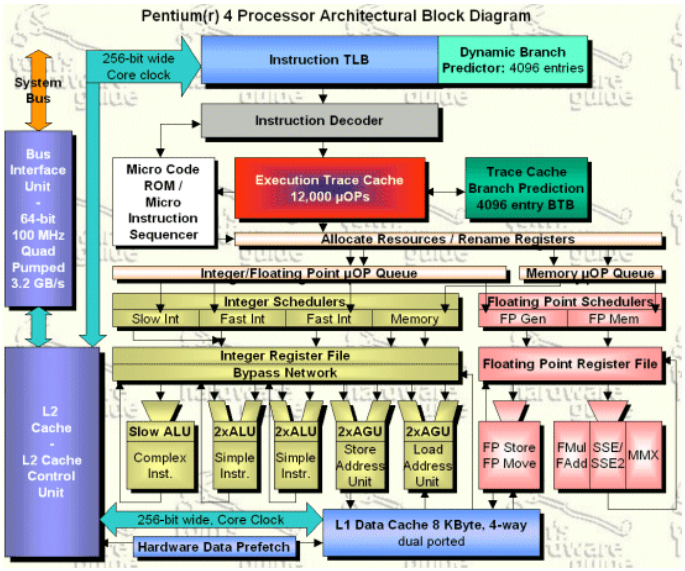



## Notas:

A modo de ejemplo primitivo, en la figura anterior se muestra la arquitectura básica de un computador simplificado, propuesto en 1946 por Burks, Goldstine y Von Neumann y cuyas características básicas son:

- Memoria :
  - 4096 = $2^{12}$ palabras de 40 bits en la memoria. Cada palabra identificada por dirección entre 0 y 4095. Tiempo de acceso 5-50 μs.
- Unidad Aritmética y lógica:
  - Sumador/desplazador. Dos acumuladores A y B.
  - Unidad de entrada/salida: Se accede a través del acumulador A. Permite intercambiar información con los periféricos externos.
- Unidad de control:
  - Instrucciones almacenadas en memoria
  - El registro P apunta a la siguiente instrucción.
  - La instrucción en ejecución almacenada en el registro I
  - La instrucción consta de dos campos: código de operación de 8 bits, y dirección del operando, de 12 bits.

# Computador Pentium IV






## Notas:

Desde el computador simplificado de Von Neumann hasta nuestros días el enorme avance en la tecnología de integración de circuitos ha hecho posible la integración de una CPU completa de 64 bits en un solo chip, o la construcción de complejísimos sistemas de cálculo, como los llamados supercomputadores, capaces algunos de ellos de hacer más de 1.000.000 millones de operaciones por segundo.

El procesador Pentium IV de Intel es un ejemplo de un computador moderno de altas prestaciones integrado en un solo “chip”. Sus características principales (año 2004) son:

- Arquitectura superescalar
- Capaz de almacenar hasta 126 instrucciones simultáneamente
- Tres niveles de caché
- Unidad aritmética y lógica de números reales de 128 bits
- Cinco unidades aritméticas y lógicas de números enteros, 4 de ellas a doble velocidad
- Predicción dinámica de saltos
- Reloj de hasta 3.2 GHz
- Tecnología de 0’13 micras

Hoy en día es común encontrar computadores con arquitectura multinúcleo, en el que hay dos o más CPUs con memorias caché sincronizadas.

# Clasificación de los computadores por su capacidad

| Categoría | Aplicación | Coste |
|---|---|---|
| Supercomputador | Cálculo numérico | > 1 M euros |
| “Mainframe” | Grandes bases de datos | 0.1-1M euros |
| Servidor | Comunicaciones y bases de datos | 5-100 K euro |
| Workstation | Diseño, ingeniería | 5-30 K euros |
| PC | Oficinas, uso doméstico | 500-5000 eur |
| Microcontrolador | Sistemas empotrados | > 1 euro |



## Notas:

El tipo de computador a emplear depende en gran medida de la aplicación a la que se destina. Existen varias categorías:

- **Supercomputador**. Aplicaciones científicas de cálculo numérico muy complejas. (IBM Roadrunner, IBM Blue Gene, Sun Ranger-SunBlade, Jaguar Cray XT4). Consultar: http://www.top500.org/
- **“Mainframe”**. Computador de propósito general grande, principalmente para aplicaciones de grandes bases de datos. (IBM System Z, Sun Fire E25K).
- **Servidor**. Computador de propósito general a mitad de camino entre un mainframe y una estación de trabajo. Da servicio a una red de computadores más pequeños. (IBM System P, Sun Spark Enterprise, Sun Blade).
- **Estación de Trabajo (“Workstation”)**. Computador de propósito general, de bajo costo, generalmente con capacidad multiproceso. (IBM Intellistation Power, Sun Ultra 40)
- **Computador Personal**. Computador con un microprocesador como CPU, normalmente configurado para un aplicación específica. (PC, Apple Machintosh).
- **Microcontrolador**. Computador de propósito especial basado en un conjunto integrado de microprocesador, memoria y dispositivos de entrada/salida, y que forma parte de un sistema mayor, tal como una máquina, instrumento, electrodoméstico, etc.

# 1.2 El software del sistema

## Junto al computador se ofrece el software del sistema

- un sistema de arranque
  - carga el sistema operativo en la memoria
- un sistema operativo
- soporte para lenguajes de programación
- un entorno de desarrollo
- programas de aplicación



## Notas:

Desde el punto de constitución física, el computador es una máquina compuesta de un entramado de circuitos electrónicos y dispositivos mecánicos de precisión. Sin embargo, el salto de máquina a computador es posible gracias a lo que constituye el **software**.

En el momento actual, un computador es un sistema electrónico muy complejo, capaz de realizar a muy alta velocidad una secuencia de operaciones, **de acuerdo con un programa previamente almacenado en sus elementos de memoria electrónica**.

El software del sistema proporciona un entorno que facilita la carga de programas en la memoria electrónica, así como la creación de programas nuevos.

# Programación del computador

## Las instrucciones de un programa son códigos numéricos almacenados en la memoria del computador

- la programación mediante códigos numéricos se conoce como lenguaje máquina
- es muy compleja

## Por ello se necesitan lenguajes de programación más cercanos a los programadores

- y herramientas para convertir programas a lenguaje máquina



## Notas:

Para programar y manejar un computador que aún no ha sido programado sería necesario conocer las operaciones básicas que puede realizar, así como la forma que debe ser codificada en la memoria electrónica del mismo. Esto es lo que se denomina la programación del computador en **lenguaje máquina**.

En la práctica, el lenguaje máquina resulta muy complicado para las personas. Se consigue una productividad mucho más alta con lenguajes de programación más cercanos a la persona, llamados lenguajes de alto nivel. Pero el computador lo único que entiende al final es lenguaje máquina, por los que se necesitan traductores de lenguajes de programación de alto nivel a lenguaje máquina.

En la sección 1.3 ampliaremos la discusión sobre los lenguajes de programación.

# El sistema operativo

**Controla el uso por parte de los programas de aplicación de todos los recursos del computador: memoria, CPU, unidades de entrada y salida**

**Independiza al programa de aplicación del hardware**

**Proporciona comunicación con otros computadores**

**Ejemplos de sistemas operativos:**

- **MS-DOS: monoproceso, sin protección**
- **Windows 95/98/ME: multiproceso, semi-protegido, un solo usuario**
- **UNIX, Windows NT/2000/XP/Vista: multiproceso, protegido, múltiples usuarios (según versión)**



## Notas:

El **sistema operativo** está constituido por una serie de programas que permiten utilizar de una forma conceptual los diferentes elementos que constituyen los recursos del computador (pantalla, teclado, diskettes, discos, impresoras, etc.).

Así, mientras el sistema operativo presente el mismo modelo conceptual al usuario, el manejo de equipos basados en sistemas electrónicos totalmente diferentes parecerá que es el mismo para el usuario.

Por ejemplo un PC de IBM y uno de otro fabricante se manejan de igual modo y pueden ejecutar los mismos programas, no porque tengan el mismo diseño electrónico, sino porque el sistema operativo que utilizan presenta el mismo modelo conceptual (Windows XP).

# Más software del sistema

## El entorno de desarrollo de programas suele constar de

- editores de texto
- herramientas CASE para análisis y diseño de programas
- depuradores
- herramientas de control de versiones

## Los programas de aplicación son muy variados, dependiendo de la aplicación concreta del computador:

- procesador de textos
- bases de datos y hojas de cálculo
- navegador de red Internet ("browser")
- programas de diseño gráfico (CAD), etc.



## Notas:

El entorno de desarrollo permite la implementación de programas de computador, cubriendo idealmente todas las fases del ciclo de vida del software. Herramientas características de un entorno de desarrollo son:

- Editor de textos: permiten crear un texto a partir de un teclado. Los caracteres tecleados se añaden al texto y, además, hay órdenes de control para gestionar la organización del texto. Normalmente todos los programas se escriben mediante un editor de texto
- Herramientas CASE para análisis y diseño de programas. Son herramientas avanzadas de ingeniería de software (CASE => Computer-Aided Software Engineering) que facilitan la labor de análisis y diseño del programa, previa a su codificación en un lenguaje de programación. Si no se dispone de estas herramientas el diseño puede hacerse manualmente, sobre papel.
- Depuradores: permiten ejecutar un programa en condiciones especiales que permiten su prueba. Permiten parar el programa en el punto deseado, consultar el estado de sus datos, continuarlo, etc.
- Herramientas de control de versiones. Son imprescindibles para gestionar los cambios incrementales en proyectos grandes de programación.

**Programas de aplicación o expertos**. Son programas desarrollados para resolver necesidades concretas dentro de un campo de interés humano que permite al usuario manejar un lenguaje muy próximo al campo especifico de que se trate. Programas de este tipo son, por ejemplo, los procesadores de texto, hojas de cálculo, bases de datos, simuladores, etc.

## Ejemplo de lenguaje máquina para el microprocesador 68000: suma de dos enteros:

```
Dirección Código Binario      Código Ensamblador Alto Nivel
$1000     0011101000111000    MOVE.W $1200,D5      Z=X+Y
$1002     0001001000000000
$1004     1101101001111000    ADD.W  $1202,D5
$1006     0001001000000010
$1008     0011000111000101    MOVE.W $D5,$1204
$100A     0001001000000100
```



## Notas:

El ejemplo de arriba hace las siguientes operaciones:

- Mueve número que está en la posición de memoria $1200 al registro D5
- Suma el número que está en la posición de memoria $1202 al registro D5
- Mueve el resultado de la suma, contenido en el registro D5, a la posición de memoria $1204

Como puede verse la programación en lenguaje máquina es muy poco entendible para las personas.

# Ensambladores, compiladores e intérpretes

**Son programas que traducen un programa de aplicación escrito en un lenguaje cómodo para los humanos a un programa en lenguaje máquina:**

- **lenguaje ensamblador: cada instrucción corresponde a una instrucción de lenguaje máquina; se traduce mediante un programa ensamblador**
- **lenguajes de alto nivel: son lenguajes independientes de la máquina; se traducen mediante compiladores e intérpretes**
  - **los compiladores traducen el programa de aplicación antes de que éste se ejecute**
  - **los intérpretes van traduciendo el programa de aplicación a medida que se va ejecutando**



## Notas:

Para hacer accesible el manejo de un computador a cualquier persona, **el fabricante** acompaña el equipo físico con un conjunto de programas de ayuda, desarrollados por él o por personas especializadas, que tienen como función presentar al usuario el computador de acuerdo con un modelo abstracto (**informático**) sencillo e independiente de su estructura electrónica interna:

- ensambladores, compiladores e intérpretes
- sistema operativo
- entorno de desarrollo
- programas de aplicación

Los ensambladores, compiladores e intérpretes traducen un programa escrito en un lenguaje más o menos cómodo para el usuario, a lenguaje máquina. Los lenguajes tienen dos categorías

- **Lenguaje ensamblador**. Cada instrucción de lenguaje ensamblador se corresponde con una instrucción de lenguaje máquina, pero en lugar de codificarse mediante números se codifica mediante símbolos alfanuméricos, más fáciles de recordar.
- **Lenguajes de alto nivel**.Permiten programar utilizando un lenguaje más próximo al humano, e independiente de la máquina. Ejemplos de este tipo de lenguajes son el FORTRAN, C, BASIC, Pascal, Ada, etc. La traducción a lenguaje máquina se hace mediante compiladores (que traducen el programa escrito en lenguaje de alto nivel de forma completa antes de su ejecución) e intérpretes (que traducen cada instrucción mientras se ejecuta el programa)

# Lenguajes de alto nivel

## Los lenguajes de programación de alto nivel son:

- una solución intermedia entre los lenguajes naturales y el lenguaje máquina
- son precisos, es decir, no ambiguos
- son relativamente simples (y por tanto poco expresivos y difíciles de usar)

## Ejemplos de lenguajes de programación:

- Fortran: 1956, para cálculo científico
  - estándares: 1966, 1977, 1990, 1997, 2003
- Cobol: 1960, para aplicaciones de gestión
  - estándar actual: 2002



## Notas:

Los lenguajes de programación de alto nivel han sido definidos como una solución intermedia entre los **lenguajes naturales humanos** y los lenguajes **máquina de los computadores**. Están bien definidos, en el sentido de que la tarea que se puede expresar con ellos no es ambigua y por lo tanto pueden ser traducidos en un programa máquina concreto, de forma automatizada y por el propio (aunque no necesariamente) computador que va a realizar la tarea.

Existen muchos lenguajes de alto nivel de propósito general, sus principales diferencias se encuentran en que poseen un conjunto de órdenes mas adecuado para expresar tareas de un tipo concreto de problema o porque corresponden a distintos niveles de evolución de los computadores:

**FORTRAN** (FORmula TRANslation). Su nombre evidencia la orientación matemática de uno de los lenguajes de alto nivel mas antiguos, que aún perduran. J. Backus lo desarrolló en 1956. Aunque ha perdido terreno frente a los lenguajes mas modernos, todavía es ampliamente utilizado en aplicaciones científicas de grandes cálculos numéricos, porque probablemente, es el lenguaje con mayor número de librerías, desarrolladas y comprobadas por mucha gente, a lo largo de su historia.

**COBOL** (COmmon Business Oriented Language). Se trata del lenguaje que ha alcanzado una mayor resonancia en las tareas de gestión. Su desarrollo fue promovido por el por el Departamento de Defensa de Los EEUU, en 1960. El lenguaje ha sufrido muchas extensiones, y ha sido actualizado recientemente.

# Ejemplos de lenguajes (cont.)

- **Lisp: 1959, para inteligencia artificial**
  - **estandarizado por ANSI (common LISP)**
  - **estandarizado por ISO en 1997 (ISLISP)**
- **Basic: 1964, para docencia, interpretado**
  - **Visual Basic 7.0, 2001 (Microsoft)**
- **Pascal: 1969, para docencia, programación estructurada**
- **C: 1972, para programación del software del sistema**
  - **estandarizado en 1990, y 1999**
- **Ada: 1983, para sistemas de alta integridad, incluyendo sistemas de tiempo real**
  - **estandarizado en 1983**



## Notas:

**LISP** (LISt Processing). El Massachusetts Institute of Technology creó, en 1959, este lenguaje de alto nivel orientado a aplicaciones de inteligencia artificial. La programación de procesos recurrentes (edificados sobre datos procesados en los pasos anteriores) es uno de los puntos fuertes del LISP.

**BASIC** (Beginners All-purpose Symbolic Instruction Code). Nació entre 1964 y 1965 en el Dartmouth College como una herramienta para la enseñanza. Con el tiempo han ido proliferando los dialectos y versiones, hasta el punto de que es raro el fabricante que no desarrolla un dialecto para sus propios equipos. Fue muy popular por su sencillez, pero tiene carencias importantes.

**PASCAL** (En honor del matemático francés Blaise Pascal). Es un lenguaje de programación desarrollado por el profesor Nicklaus Wirth, en 1969, en el Instituto Federal de Tecnología de Zurich partiendo de los fundamentos del ALGOL. Fue uno de los primeros lenguaje que incorporaron los conceptos de programación estructurada. Aunque fue muy popular, la dificultad para partir el programa en módulos y la falta de estandarización han hecho decaer su uso.

**C**. Es un lenguaje de programación desarrollado por la Bell Laboratories, en principio para trabajar con el sistema operativo UNIX. Quizás por ello, la popularidad del 'C' es muy alta. Es un lenguaje que, al mismo tiempo que permite una programación en alto nivel, permite una gran aproximación a la máquina. Muchos lo consideran un lenguaje intermedio entre alto y bajo nivel. Como estos últimos, presenta alta eficiencia y escasa fiabilidad. Es fácil cometer errores en C.

# Ejemplos de lenguajes (cont.)

## Lenguajes de programación orientada al objeto:

- Smaltalk: 1980, para programación orientada a objetos
  - estandarizado en 1998
- C++: 1987, extensión mejorada del C que incorpora programación orientada a objetos
  - estandarizado en 1998
- Java: 1995, para programación orientada a objetos en sistemas distribuidos (red Internet)
  - versión actual: Java 6 (2006)
- Ada 95y Ada 2005: versiones mejoradas del anterior, incluyendo programación orientada a objetos
  - estandarizado en 1995 y luego en 2005



## Notas:

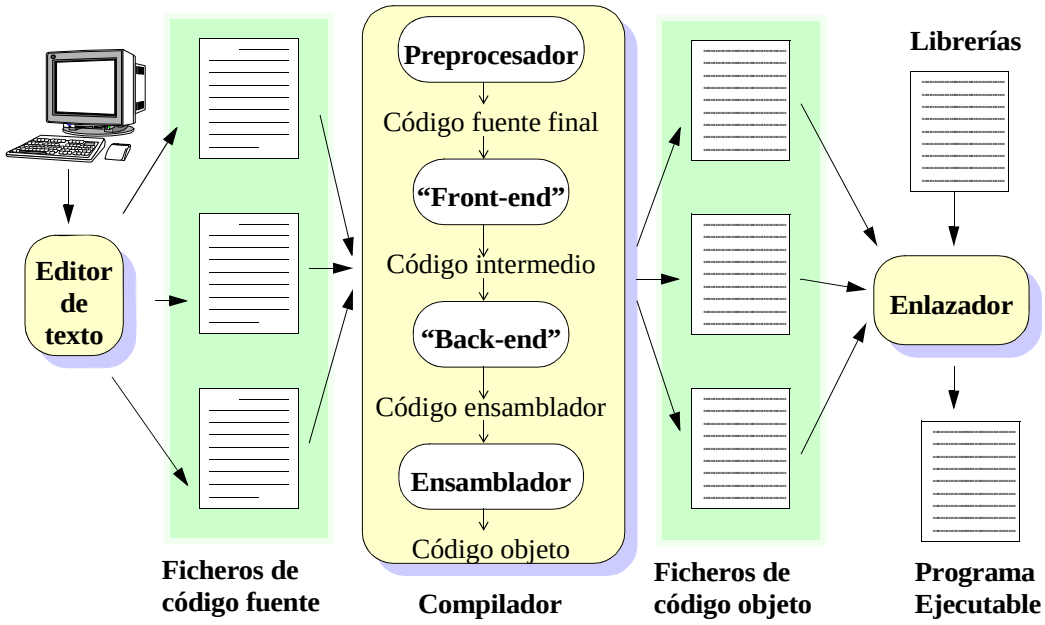
**SMALTALK**: Lenguaje de programación orientada a objetos puro. Es muy ineficiente con respecto a lenguajes procedurales como el C o el Ada, pero es cómodo de usar y de programar en él.

**JAVA**: Lenguaje derivado del C en cuanto a sintaxis, pero más parecido al Ada 95 en cuanto a las comprobaciones que hace el compilador y soporte de la programación concurrente. Está pensado para su ejecución en sistemas distribuidos (internet). Existe un código intermedio, bien definido, que puede intercambiarse entre computadores diferentes para luego ser traducido y ejecutado. Su popularidad está en fuerte ascenso.

**C++**: Extensión del lenguaje C que mejora algunos de sus inconvenientes, y añade construcciones de programación orientada a objetos. Entre las mejoras destacan una mayor comprobación de los tipos de datos por parte del compilador, las excepciones, y las plantillas genéricas.

**ADA** (En honor de Lady Augusta ADA Byron). El ADA es un lenguaje inspirado en el PASCAL, que fue promovido por el Departamento de Defensa de Los EEUU. El objetivo de su desarrollo era conseguir un lenguaje con posibilidades de convertirse en un estándar universal y que facilitara la ingeniería de software y el mantenimiento de los programas. Entre sus campos de aplicación se incluyen los sistemas de tiempo real y los sistemas de alta integridad. El 1995 se revisó el lenguaje para mejorarlo y para añadirle construcciones de programación orientada a objetos. En 2005 se finalizó una nueva versión.

# 1.4. El proceso de compilación





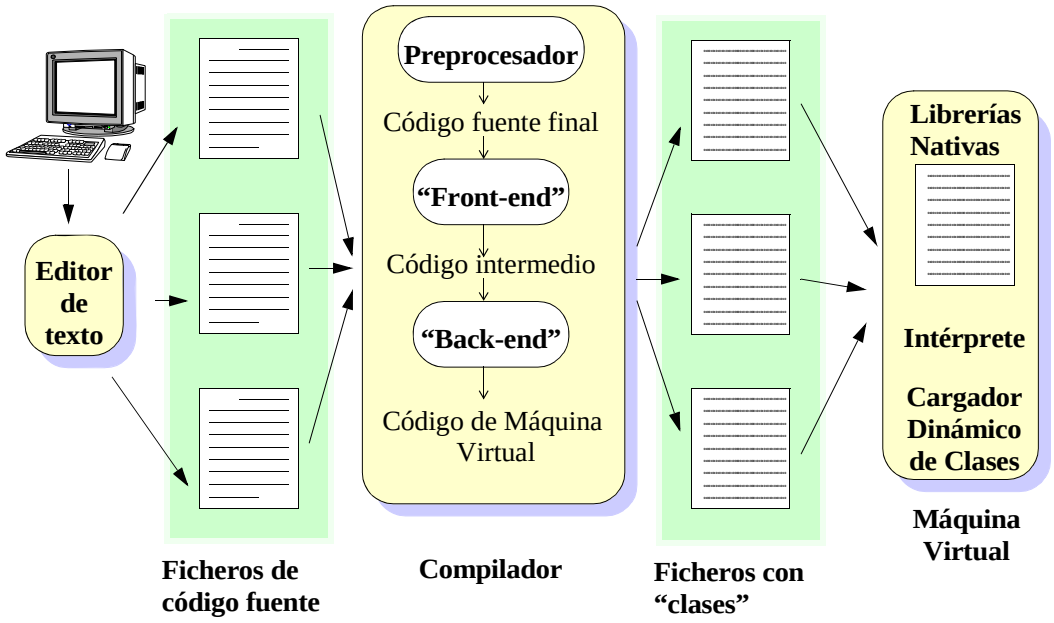

## Notas:

Un compilador puede definirse como una herramienta automática de traducción que lee un programa escrito en un lenguaje (el lenguaje fuente) y lo traduce a un programa equivalente en otro lenguaje (lenguaje objeto). En el proceso de traducción el compilador notifica al usuario de la presencia de errores en el programa fuente.

La variedad de compiladores que pueden aparecer es muy alta. Existen miles de lenguajes fuente. Igualmente ocurre con los lenguajes objeto: pueden ser otros lenguajes de programación, o el lenguaje máquina de cualquier computador entre un microprocesador o un supercomputador. En cualquier caso las tareas que debe realizar son las mismas.

Además del compilador también son necesarios otros programas para crear un programa ejecutable: el preprocesador, el ensamblador, el enlazador, y el cargador. En la figura de arriba se muestra un proceso de compilación típico.

# Compilación para máquina virtual






## Notas:

La arquitectura del entorno de ejecución de programas Java se basa en una máquina virtual, que se ejecuta en el computador para interpretar las instrucciones del programa del usuario descritas en un código intermedio especial, llamado código de máquina virtual Java (Java Byte Code).

La idea principal de esta arquitectura es la de "Escribir una vez, ejecutar en cualquier sitio". El compilador de Java no genera código máquina de un computador concreto, sino un código especial, que luego es interpretado por otro programa, llamado máquina virtual, que existe en cada computador en el que se desea ejecutar el programa Java. De este modo, un programa Java se puede ejecutar indistintamente en cualquier computador que disponga de esa máquina virtual, sin necesidad de recompilarlo.

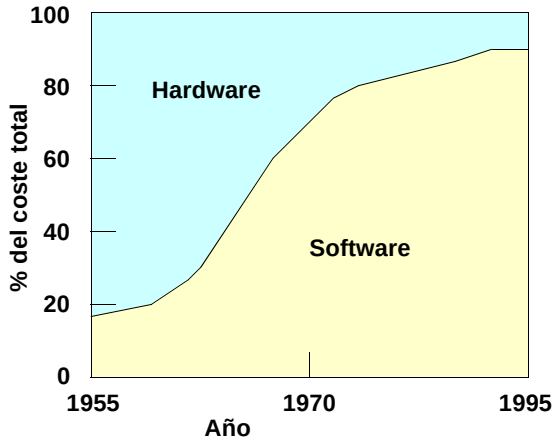
Adicionalmente, en la arquitectura Java los programas no se enlazan antes de su ejecución, sino que se utiliza un enlazado dinámico. Cuando se hace una llamada a una operación de un módulo (clase) que no está cargado en la máquina virtual, ésta se encarga de buscar ese módulo y cargarlo en ese momento en la máquina virtual.

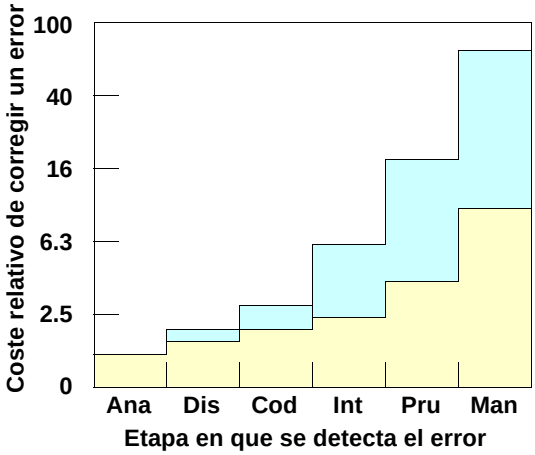
Desde el programa del usuario se pueden utilizar operaciones "nativas", suministradas por la máquina virtual, escritas generalmente en código máquina, y que pueden acceder a los dispositivos hardware del computador. El resultado es: programas muy portables, muy dinámicos, aunque poco eficientes.

# 1.5 El Ciclo de vida del software

## La mayor parte del gasto en sist. informáticos es el software

## Los errores software tienen un alto coste: efecto y corrección






## Notas:

Con el continuo incremento a gran escala de las prestaciones de los sistemas informáticos actuales, con un costo del hardware cada vez menor, la mayor parte del gasto total en sistemas informáticos es actualmente el software.

La complejidad actual del software puede ser tan elevada que la existencia de errores introducidos en la etapa de diseño es muy probable. Generalmente los errores del software implican un alto coste desde dos puntos de vista: coste del propio efecto del error, y coste de su corrección.

El efecto del error puede ser muy costoso en dinero (por ejemplo paralización de una actividad), e incluso en vidas humanas (por ejemplo, en el software de control de actividades críticas). La corrección de los errores software suele ser costosa, especialmente si se detectan en las últimas fases del diseño. El mayor esfuerzo de programación se realiza en la fase de test e integración.

Distribución del esfuerzo de la actividad software (sin tener en cuenta el mantenimiento):

- **Análisis** y **Diseño**:38%
- **Codificación**:20%
- **Integración** y **Prueba**:42%

El costo del **mantenimiento** depende del tiempo de vida del programa. En programas de tiempo de vida largo, el costo del mantenimiento puede ser típicamente del orden del 90% del costo total.

# Actividades del ciclo de vida del software

## Análisis de requerimientos

## Especificación funcional

## Diseño de la arquitectura

## Diseño detallado

## Codificación y desarrollo

## Integración y verificación

## Operación y mantenimiento



## Notas:

*Análisis y especificación*. En esta etapa se analiza la naturaleza del problema, y se establecen los requerimientos del sistema. En la especificación se desarrollan las descripciones de las funciones a realizar por el sistema, de sus restricciones, y de los recursos necesarios.

*Diseño de la Arquitectura*. A partir de las especificaciones funcionales, se diseña la estructura del sistema que resuelve el problema. Esta estructura representa las partes importantes del sistema y sus relaciones, así como las estructuras de datos más importantes.

*Diseño Detallado.* Las partes importantes del sistema se diseñan en detalle, describiendo los algoritmos y estructuras de datos concretas. Se detallan las interfaces entre las diferentes partes. Las etapas de diseño suelen requerir varios niveles de refinamiento.

*Codificación y Desarrollo*. Producción de una realización física del diseño.

*Integración y Verificación*. Suelen ser fases cíclicas, en las que de forma incremental se van probando e integrando las diversas partes del sistema.

*Operación y Mantenimiento*. Reparación de problemas, adaptación a nuevas condiciones, mejoras, etc.

# Actividades en cada etapa

| Etapa | Entradas | Salidas |
|---|---|---|
| Análisis | Necesidades iniciales, contexto, problemas del usuario | Definición de requerimientos |
| Especificación | Requerimientos, contexto del sistema | Especificaciones funcionales, diseño externo del sistema |
| Diseño de la Arquitectura | Especificaciones, contexto, experiencia previa | Definición de la estructura del programa con descomposición en módulos, y diseño de sus interfaces |
| Diseño Detallado | Descripción de la arquitectura, detalles del entorno | Descripción de las estructuras de datos y algoritmos a emplear en cada módulo |



## Notas:

Análisis:

- Identificación de las funciones más importantes, recolección de información y de restricciones.

Especificación:

- Conversión de las necesidades en funciones explícitas, selección de las restricciones que son operativas, descripción de las interfaces al usuario.

Diseño de la Arquitectura:

- Determinar la estructura del problema, identificar las partes importantes del sistema, establecer las relaciones entre las partes, abstracción, y descomposición.

Diseño Detallado:

- Abstracción, elaboración, selección de alternativas.

# Actividades en cada etapa (cont.)

| Etapa | Entradas | Salidas |
|---|---|---|
| Codificación y Desarrollo | Descripción de la arquitectura, diseño detallado, detalles del entorno | Código de unidades de programa y documentación |
| Integración y Verificación | Descripción de la arquitectura y código de las unidades de programa con su documentación | Código del programa o programas completos, ficheros de datos, sistema completo |
| Operación y Mantenimiento | Documentación del sistema, requerimientos de operación, peticiones de cambios | Sistema mejorado |



## Notas:

Codificación y Desarrollo:

- Codificación de algoritmos y estructuras de datos

Integración y Verificación:

- Depuración y verificación de unidades de programa y de prototipos del sistema global cada vez más elaborados, hasta llegar al sistema completo

Operación y mantenimiento:

- Reprogramación, mejora, depuración del rediseño, etc.

# Objetivos de la ingeniería software

## Ofrecer técnicas y herramientas adecuadas para el correcto diseño de programas eficientes, seguros, y a un coste mínimo

- **Principales hechos conseguidos:**
  - Control de configuración en proyectos grandes
  - Documentación adecuada
  - Procesos de desarrollo, herramientas CASE
  - Uso extensivo del lenguaje UML
- **Aún es preciso un desarrollo mayor, en áreas como:**
  - Herramientas y lenguajes más perfeccionados
  - Mayor utilización de (nuevos) métodos de desarrollo
  - Desarrollo basado en componentes
  - Sistemas automáticos de prueba y de programación



## Notas:

Principales hechos conseguidos:

- El control de configuración de software es de gran importancia en proyectos grandes que involucran a muchas personas, con objeto de tratar de forma adecuada los problemas de versiones y de desarrollo incremental de programas.
- La documentación adecuada, acompañada de métodos de representación del diseño es uno de los aspectos más importantes de un diseño software.
- Los procesos de desarrollo obligan a seguir una disciplina en los pasos a dar en el desarrollo del software, y ello redunda en una mayor productividad y en una mayor fiabilidad de los productos desarrollados. Se suelen basar en el uso extensivo de herramientas CASE (Ingeniería de software asistida por computador).