

U.T. 1: ARQUITECTURA Y COMPONENTES DEL PC (parte 1)

Objetivos:

- Conocer lo que es un sistema informático.
- Conocer que son los datos y como se codifican.
- Conocer los conceptos de programa e instrucción.
- Conocer la arquitectura de los ordenadores.

1.1.– INTRODUCCIÓN:

Desde hace mucho tiempo el hombre ha tenido la necesidad de transmitir la información de una forma continuada. Tradicionalmente se ha usado el correo convencional, el teléfono...Pero poco a poco se han ido creando herramientas que faciliten soluciones a estas necesidades.

Hace ya algunos años se construyen complejas herramientas para procesar y transmitir información que se han podido desarrollar gracias al gran avance tecnológico del siglo pasado, fundamentalmente del último cuarto de siglo.

El ordenador es la herramienta que permite procesar de forma automática la información, facilitando su organización su tratamiento, transmisión y almacenamiento.

La Informática en sus inicios fue definida como la ciencia que se encargaba de estudiar el tratamiento automático de la información.

En la actualidad esta palabra engloba muchos más conceptos, ya que se aplica a la mayoría de los campos técnicos y científicos.

Mediante la informática hoy en día se gestiona información en forma de programas y datos con los que es posible poner en marcha complejísimos sistemas electrónicos, mecánicos, etc. Por ejemplo controlar maquinaria, naves espaciales, ciclos de producción de cosechas, etc.

El espectacular desarrollo sufrido en las dos últimas décadas del siglo XX y la primera del siglo XXI la ha convertido en una herramienta imprescindible en multitud de campos tales como las comunicaciones, telefonía, Internet, vigilancia, control de tráfico, etc.

1.2.– EVOLUCIÓN HISTÓRICA.

Habitualmente se suele estudiar la evolución de la informática agrupada por distintas generaciones, la frontera entre una generación y otra no siempre está clara. Esta frontera suele decidirse en función de la aparición de alguna tecnología que hace avanzar el desarrollo de la informática.

1^a Generación (1940-1960): Se utilizaban válvulas de vacío (antiguas resistencias electrónicas). Estas computadoras eran máquinas programadas en lenguaje máquina puro. Eran de gran tamaño, elevado consumo de energía y muy lentas. Las operaciones se reducían a simples cálculos matemáticos.

2^a Generación (1960-1965): Aparecen los transistores, que se introducen dentro de la arquitectura de las computadoras. Desaparecen las válvulas de vacío, por lo que las computadoras se hacen más pequeñas, baratas, consumen menos y despiden menos calor.

Los encargados de la utilización del SI se dividen en categorías: perforador de tarjetas, operador de consola, etc. Aparece lo que se denomina el procesamiento por lotes, procesamiento que implicaba tres fases:

- Introducción de los datos en un soporte hardware, magnético o no.
- Llevar los datos en el soporte a la máquina encargada de procesar la información, que una vez procesada se almacenaba en otro soporte distinto.
- El soporte de salida con los resultados se llevaba a otra computadora que se encargaba de generar los resultados (impresión).

3^a Generación (1965-1975): Aparecen los circuitos integrados y se reduce considerablemente el tamaño y consumo de los ordenadores. Más baratos, rápidos, consumen menos energía y despiden mucho menos calor. Aparece el IBM360 como máquina de propósito general, con software básico para realizar varios procesos. Aparece Cobol, Fortran, Basic... y las memorias RAM, ROM, PROM y EPROM.

4^a Generación (1975-1981): El paso a esta generación lo produce la aparición de los microprocesadores. Básicamente, un microprocesador es una CPU integrada en una sola pastilla de circuito impreso. Son circuitos integrados de alta densidad, con una velocidad muy elevada.

El primer microprocesador fue el INTEL 4004. Otro hecho importante de esta época fue la aparición de las pastillas de memoria de semiconductor, con lo que se abandonan las memorias de ferrita. Aparición de discos flexibles o floppy disc de 5 1/4

5^a Generación: (1981-1990): Los avances en la microelectrónica y la gran competencia entre compañías como Apple, Motorola, Cytrix, AMD, Intel o IBM provocan un continuo aumento de la integración y el desarrollo de nuevas arquitecturas computacionales, lo que contribuye a disponer de ordenadores cada vez más potentes y baratos.

En 1981 se presenta el PC IBM con su microprocesador Intel 80x88 de 8 bits, su aparición da paso a esta generación.

Macintosh de Apple con su procesador de 16 bits 68000 Motorola

Procesadores Intel como el 8086 de 16 bits. Se incorporan en un sistema varios microprocesadores, de 16 y 32 bits. Las memorias RAM alcanzan decenas de Megabytes

6^a Generación: (1991-Actualidad): La aparición de Internet como red de redes marca el paso a esta generación.

Actualmente existen dos líneas de investigación fundamentales:

- **HARDWARE.** El nivel de miniaturización que se está alcanzando prevé que en breve comiencen a no cumplirse las leyes de la física clásica, interfiriendo procesos cuánticos. Se está alcanzando el límite de integración, tal y como se realiza actualmente. Surgen varias líneas de trabajo complementarias:
 - Procesamiento en paralelo, mediante nuevas arquitecturas
 - Redes de área mundial o WAN (INTERNET), usando fibra óptica y satélites que alcanzan un ancho de banda impresionante
 - Sistemas de inteligencia artificial, holografía, transistores ópticos, nanotecnología.
- **SOFTWARE.** Manejo del lenguaje natural y sistemas de inteligencia artificial distribuida, teoría del caos, uso de sistemas difusos.

1.3.- EL SISTEMA INFORMÁTICO

Un SI puede definirse como un conjunto de partes interrelacionadas. Emplea un ordenador que usa dispositivos programables para capturar, almacenar y procesar datos. Dicho ordenador, junto con la persona o personas que lo manejan y los periféricos que lo envuelven, resultan de por si un ejemplo de SI.

Estructuralmente un SI se puede dividir en partes, pero funcionalmente es indivisible, ya que si se divide pierde alguna de sus propiedades esenciales.

Todo SI está compuesto por tres elementos básicos: Hardware, Software, el componente humano

1.3.1.- COMPONENTES FÍSICOS. HARDWARE

El ordenador se puede definir como una máquina compuesta de elementos físicos que llamamos Hardware, en su gran mayoría de tipo electrónico, capaz de realizar muchas tareas a gran velocidad y con gran precisión. Podemos decir que el HW es la parte tangible del ordenador, la que podemos tocar y ver.

En las primeras generaciones de ordenadores el HW tuvo una especial importancia debido a que no existían los medios electrónicos miniaturizados que usamos actualmente.

Por ejemplo el ENIAC que tan solo procesaba sencillas operaciones como una calculadora. Ocupaba el espacio de una habitación completa y su manejo era manual para conmutar operaciones, muy similar al de las centralitas de teléfono antiguas.

1.3.2.- COMPONENTES LÓGICOS. SOFTWARE:

Para que los componentes físicos de un ordenador sean capaces de funcionar y realizar un proceso determinado, hace falta que en él se ejecuten un conjunto de órdenes o instrucciones. Estas instrucciones ordenadas y agrupadas de forma adecuada constituyen un programa.

Los programas es lo que en general llamamos software.

Se puede ver el SW dividido en dos grandes grupos:

- **Software de sistema:** Maneja los procesos y recursos lógicos de funcionamiento del ordenador. En este tipo de SW se incluyen los SSOO.

Un S.O es el componente software capaz de hacer que los programas procesen información y datos sobre los componentes electrónicos. El SO facilita y permite de esta manera el trabajo de los usuarios.

- **Software de aplicación:** Es el conjunto de programas que tienen un trabajo definido, casi siempre en apoyo de las demandas del usuario. Suele estar constituido por otros subprogramas y pueden ser de muy diversa utilidad. Paquetes de ofimática, software orientado a la seguridad del sistema, programas para la reproducción de ficheros multimedia, etc.

Se llaman aplicaciones porque sus tareas están muy definidas, y porque están orientadas a aplicar dichas tareas para cubrir una demanda del usuario.

Son poco modificables por parte del usuario, no se puede alterar su forma de trabajo ni su orientación original, aunque se pueden configurar algunas características de su funcionalidad, para personalizarlas.

1.3.3.- COMPONENTE HUMANO

Está constituido por las personas que participan en la dirección, diseño, desarrollo, implantación y explotación de un SI.

1.3.4.- FRONTERA ENTRE EL SW Y EL HW. FIRMWARE.

Situado en la frontera entre el SW y el HW existe otro concepto importante:

El Firmware, Es un SW capaz de proporcionar un control de bajo nivel específico para los componentes HW.

Por ejemplo el SW con el que está programadas las memorias ROM. Este tipo de SW no es modificable fácilmente, una vez que se graba en el componente queda prácticamente invariable a lo largo de la vida del equipo.

Otro tipo de Firmware es el SW con el que se configuran dispositivos de comunicaciones como enruteadores o comutadores.

1.3.5.- CODIFICACIÓN DE LA INFORMACIÓN:

En la era digital la unidad mínima de información es el Bit (1 ó 0) (binary digit) que representa la ocurrencia o no de un suceso.

Debido a que la memoria del ordenador es básicamente un conjunto de biestables en los que puede o no haber corriente eléctrica es usual usar el sistema binario para representar estas dos situaciones:

Presencia de corriente (1)

Ausencia de corriente(0).

1.3.6.- SISTEMAS DE NUMERACIÓN

Un sistema de numeración es el conjunto de símbolos y reglas que se utilizan para representar cantidades o datos numéricos.

Cuando nosotros realizamos operaciones aritméticas utilizamos el sistema de numeración decimal en el cual hay 10 símbolos diferentes: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Al haber 10 símbolos diferentes se dice que se está trabajando en base 10.

Este es un sistema posicional, es decir, que cuando tenemos un número representado en base 10, estará formado por cierta cantidad de cifras contribuyendo cada una de ellas con un valor que depende:

- a) De la cifra en sí
- b) De la posición que ocupa dicha cifra dentro del número

Por ejemplo, el número 325 expresado en base decimal puede obtenerse como una suma de productos de la siguiente manera: $325 = 3 \cdot 10^2 + 2 \cdot 10^1 + 5 \cdot 10^0$

El número 687'53 sería: $687'53 = 6 \cdot 10^2 + 8 \cdot 10^1 + 7 \cdot 10^0 + 5 \cdot 10^{-1} + 3 \cdot 10^{-2}$

Por lo tanto, cada posición tiene un peso en el valor final del número.

1.3.7.- SISTEMA BINARIO:

En este sistema la base es $b=2$ lo que implica que hay 2 símbolos diferentes que son el 0 y el 1. Éste es el sistema que utilizan internamente los ordenadores para efectuar operaciones aritméticas.

1.3.8.- SISTEMA OCTAL:

En este sistema la base es $b=8$ lo que implica que hay 8 símbolos diferentes que son 0, 1, 2, 3, 4, 5, 6, 7.

1.3.9.- SISTEMA HEXADECIMAL:

En este sistema la base es $b=16$ lo que implica que hay 16 símbolos diferentes que son 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

1.3.10.- CAMBIOS DE BASE.

Todos los sistemas posicionales están basados en el **Teorema Fundamental de la Numeración (TFN)** que sirve para relacionar una cantidad expresada en cualquier sistema de numeración con la misma cantidad expresada en el sistema decimal, es decir, en el sistema que utiliza base 10. Según este teorema, si tenemos un número X expresado en base b, para obtener su representación decimal (base 10) haríamos lo siguiente:

$$\text{NUM} = X_n \cdot b^n + X_{n-1} \cdot b^{n-1} + \dots + X_1 \cdot b^1 + X_0 \cdot b^0 + X_{-1} \cdot b^{-1} + X_{-2} \cdot b^{-2} + \dots + X_{-n} \cdot b^{-n}$$

Donde X_i = Cifra en la posición i-esima en la expresión del número en base b.

Para calcular la expresión en base 10 (decimal) del número 101010 en base 2 (binario) aplicamos el TFN:

$$\text{NUM} = 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 32 + 0 + 8 + 0 + 2 + 0 = 42$$

Para calcular la expresión en base 10 (decimal) del número 77 en base 8 (octal) aplicamos el TFN:

$$\text{NUM} = 7 \cdot 8^1 + 7 \cdot 8^0 = 56 + 7 = 63$$

Para hacer la transformación inversa, es decir de la base decimal a cualquier otra base se sigue el siguiente algoritmo:

Se toma el número en decimal y se hace la división entera por la base ; a continuación, se toma el cociente obtenido y se vuelve a realizar con él la división entera por la base; con el cociente obtenido repetimos el proceso y así se sigue sucesivamente hasta obtener un cociente igual a 0. El número escrito en la nueva base estará formado por los restos de las sucesivas divisiones empezando por el de la última hasta llegar al resto de la primera de las divisiones.

Ejemplos:

Para pasar a base 2(binario) el número 63 en base 10 (decimal) :

$$63/2=31+1 \quad 31/2=15+1 \quad 15/2=7+1 \quad 7/2=3+1 \quad 3/2=1+1 \quad 1/2=0+1 \quad 111111$$

Para pasar a base 2(binario) el número 64 en base 10 (decimal) :

$$64/2=32+0 \quad 32/2=16+0 \quad 16/2=8+0 \quad 8/2=4+0 \quad 4/2=2+0 \quad 2/2=1+0 \quad 1/2=0+1 \quad 1000000$$

Para pasar a base 8(octal) el número 63 en base 10 (decimal) :

$$63/8=7+7 \quad 7/8=0+7 \quad 77$$

1.3.11.- MEDIDAS DE LA INFORMACIÓN

Bit: Es la unidad más pequeña de información. Con un bit se puede representar solo dos posibles valores diferentes, ejemplo: cero o uno, falso o verdadero, blanco o negro, abajo o arriba, no o sí, etc.

Nibble: Es una colección de 4 bits. No sería un tipo de dato interesante a excepción de que con un nibble se presenta un número BCD y también que un nibble puede representar un dígito hexadecimal.

Byte: Es una colección de 8 bits. Habitualmente es el número mínimo de bits con el que se representa un carácter.

1 Byte	B=8 bits.
1 Kilobyte	KB= 1024 Bytes
1 Megabyte	MB= 1024 KB
1 Gigabyte	GB = 1024 MB
1 Terabyte	TB = 1024 GB
1 Petabyte	PB = 1024 TB
1 Exabyte	EB = 1024 PB
1 Zetabyte	ZB = 1024 EB
1 Yottabyte	YB = 1024 ZB

Para hacernos una idea del tamaño de estas medidas podríamos decir que según un estudio de 2012 toda la información disponible en Internet por esa época ocupaba aproximadamente unos 2,75 Zetabytes y se piensa que se duplica cada 3 años.

Puesto que la información se almacena en electrones, si contáramos el número de electrones, que se necesitan para representar toda la información que hay en Internet su peso, en 2012, sería de unos 60 gramos, más o menos el peso de una cereza.

1.3.12.- SISTEMAS DE REPRESENTACIÓN DE LA INFORMACIÓN

En los computadores la información que representamos puede ser de todos los siguientes tipos:

Textos	<ul style="list-style-type: none">▪ BCD de 6 bits▪ EBCDIC▪ ASCII▪ UNICODE		
Datos Numéricos	Enteros	Dígitos decimales codificados en Binario (BCD)	<ul style="list-style-type: none">▪ Empaqueado▪ Desempaqueado
		Representación Binaria - Coma Fija -	<ul style="list-style-type: none">▪ Módulo y Signo▪ Complemento a 1▪ Complemento a 2▪ Exceso a 2 elevado a N-1
	Reales	Coma Flotante <ul style="list-style-type: none">▪ Notación exponencial▪ Normalización IEEE754	
Sonidos	WAV, MIDI, MP3		
Imágenes	Mapa de Bits		BMP, TIFF, JPEG, GIF, PNG
	Mapa de Vectores		DXF, IGES, EPS, TrueType

1.3.13.- SISTEMAS DE REPRESENTACIÓN NUMÉRICA

Se pueden considerar varios casos a la hora de representar información numérica:

Representación de números naturales, representación de números enteros y representación de números reales

1.3.14.- REPRESENTACIÓN DE NÚMEROS NATURALES

Para la representación de números naturales se usa su representación en binario puro.

Es decir a cada número natural se le representa por su equivalente en el sistema binario.

1.3.15.- REPRESENTACIÓN DE NÚMEROS ENTEROS, NÚMEROS CON SIGNO

1.3.15.1 MÓDULO Y SIGNO.

Se destina el bit más significativo para el signo (0 positivo y 1 negativo y el resto representa el módulo del número (su valor absoluto)

Rango para un byte: $-2^{n-1} + 1 \leq X \leq 2^{n-1} - 1$

Ventaja rango simétrico, desventaja el cero tiene dos representaciones 00000000 positiva y 10000000 negativa.

1.3.15.2 DECIMAL DESEMPAQUETADO

Se almacena cada cifra en un byte, como para representar cada cifra del sistema decimal es suficiente con 4 bits, los bits que sobran (el cuarteto de la izquierda de cada byte, bits de zona) se ponen a 1, el signo se marca en el cuarteto de zona primero de la derecha para el positivo 1100 y para el negativo 1101.

-419

1111 0100 1111 0001 1101 1001

1.3.15.3 DECIMAL EMPAQUETADO

Desaparecen los bits de zona. Se usa un cuarteto para cada cifra y el primer cuarteto de la izquierda se usa para el signo, para el positivo 1100 y para el negativo 1101.

-419

1101 0100 0001 1001

1.3.16.- REPRESENTACIÓN DE NÚMEROS REALES

Para representar número reales normalmente se usa la **representación en coma flotante**:

$$Nl = \text{mantisa} * \text{base}^{\text{exponente}}$$

$$123,56 = 0,12356 * 10^3 \quad \text{es decir: mantisa}= 0,12356 \quad \text{base}=10 \quad \text{exponente}= 3$$

De esta forma solo hace falta almacenar la parte decimal de la mantisa y su signo (número entero) y el exponente con su signo (número entero).

Se debe decidir que precisión se usa para almacenar la mantisa y para el exponente, es decir, se debe decidir un tamaño en bytes para la mantisa y para el exponente.

1.3.17.- REPRESENTACIÓN DE DATOS ALFANUMÉRICOS

La información alfanumérica es la que está formada por los símbolos de los alfabetos de los idiomas que usamos los humanos, incluidos números y otros símbolos, como @, /, (,), #, +, - etc...

El código ASCII ha sido el más utilizado, puede representar 127 caracteres, usando 7 bits para codificar.

El ASCII extendido se usa un octavo bit para representar 128 caracteres más.

TABLA ASCII

Binario Hex\Dec	000 0	001 1	010 2	011 3	100 4	101 5	110 6	111 7
0000 0	NUL 0	DLE 16	SP 32	0 48	@ 64	P 80	` 96	p 112
0001 1	SOH 1	DC1 17	! 33	1 49	A 65	Q 81	a 97	q 113
0010 2	STX 2	DC2 18	" 34	2 50	B 66	R 82	b 98	r 114
0011 3	ETX 3	DC3 19	# 35	3 51	C 67	S 83	c 99	s 115
0100 4	EOT 4	DC4 20	\$ 36	4 52	D 68	T 84	d 100	t 116
0101 5	ENQ 5	NAK 21	% 37	5 53	E 69	U 85	e 101	u 117
0110 6	ACK 6	SYN 22	& 38	6 54	F 70	V 86	f 102	v 118
0111 7	BEL 7	ETB 23	' 39	7 55	G 71	W 87	g 103	w 119
1000 8	BS 8	CAN 24	(40	8 56	H 72	X 88	h 104	x 120
1001 9	HT 9	EM 25) 41	9 57	I 73	Y 89	i 105	y 121
1010 A	LF 10	SUB 26	* 42	:	J 74	Z 90	j 106	z 122
1011 B	VT 11	ESC 27	+	;	K 59	I 75	k 91	{ 107
1100 C	FF 12	FS 28	' 44	< 60	L 76	\ 92	l 108	 124
1101 D	CR 13	GS 29	- 45	= 61	M 77	l 93	m 109	} 125
1110 E	SO 14	RS 30	. 46	> 62	N 78	^ 94	n 110	~ 126
1111 F	SI 15	US 31	/	?	O 79	- 95	o 111	DEL 127

Otro código muy utilizado es UNICODE que utiliza hasta 32 bits para representar un carácter en cualquier idioma. Tiene la ventaja que usa el mismo conjunto de códigos para diferentes idiomas.

ASCII/8859-1 Text

A	0100 0001
S	0101 0011
C	0100 0011
I	0100 1001
I	0100 1001
/	0010 1111
8	0011 1000
8	0011 1000
5	0011 0101
9	0011 1001
-	0010 1101
1	0011 0001
0	0010 0000
t	0111 0100
e	0110 0101
x	0111 1000
t	0111 0100

Unicode Text

A	0000 0000 0100 0001
S	0000 0000 0101 0011
C	0000 0000 0100 0011
I	0000 0000 0100 1001
I	0000 0000 0100 1001
/	0000 0000 0010 0000
天地	0101 1001 0010 1001
天地	0101 0111 0011 0000
一	0000 0000 0010 0000
フ	0000 0110 0011 0011
フ	0000 0110 0100 0100
フ	0000 0110 0011 0111
フ	0000 0110 0100 0101
フ	0000 0000 0010 0000
フ	0000 0011 1011 0001
フ	0010 0010 0111 0000
フ	0000 0011 1011 0011

1.3.18.- PROGRAMAS E INSTRUCCIONES:

Un lenguaje de programación no es más que un conjunto de símbolos y reglas para representar de forma lógica una serie de instrucciones y datos para desarrollar algoritmos.

Los lenguajes de programación se usan para construir programas.

Un programa podríamos entenderlo como un conjunto ordenado de instrucciones para realizar una o varias tareas de forma automatizada.

Dependiendo de lo cerca que se esté de la forma de pensar humana o de lo cerca que se esté del funcionamiento de la máquina se dirá que el lenguaje es de alto o bajo nivel.

- **Lenguaje máquina:** es un sistema de códigos directamente interpretable por un circuito microprogramable, como el microprocesador.
Está compuesto por un conjunto de instrucciones que determinan acciones a ser tomadas por la máquina. Se escribe en binario. Es el más cercano a la máquina y difícil de programar.
Ejemplo de instrucción en lenguaje máquina: 00001101 11001000 11111011
- **Lenguaje ensamblador:** implementa una representación simbólica del lenguaje máquina. Aún es cercano a la máquina.
Es un lenguaje basado en mnemotecnias que representan instrucciones de código máquina, registros del procesador o posiciones de memoria.

Ejemplo de instrucciones en ensamblador: MOV AL, 061h (muestra el valor 061h al registro AL).

- **Lenguajes de alto nivel:** los lenguajes de alto nivel son muy cercanos (cada vez más) a la forma de pensar humana, por lo que están muy lejos de la máquina. Generan un código más sencillo y comprensible. Normalmente se puede escribir código válido para diversas máquinas y sistemas operativos.

Ejemplos de lenguajes de alto nivel: Pascal, Visual, C, C++, Basic..., Java

Ejemplo de instrucciones de un lenguaje de alto nivel:

```
Procedure Nace_oso ( oso: Animal)
Begin
    new animal(oso)
    oso.fecha_nacimiento:=hoy()
    oso.raza:=pardo
    oso.peso=200
End
```

1.3.19.- ORGANIZACIÓN DE LOS DATOS:

La información que maneja un sistema informático puede ser de distinto tipo dependiendo del tratamiento que se le dé.

Además del conjunto de instrucciones existen una serie de datos necesarios para la realización de dichas tareas estos son los **datos de entrada**.

Como resultado de la realización de estas tareas, esto es como **resultado del proceso**, se producen una serie de **datos de salida**.

Además generalmente mientras se realizan estas tareas también se producen una serie de datos intermedios o de trabajo.

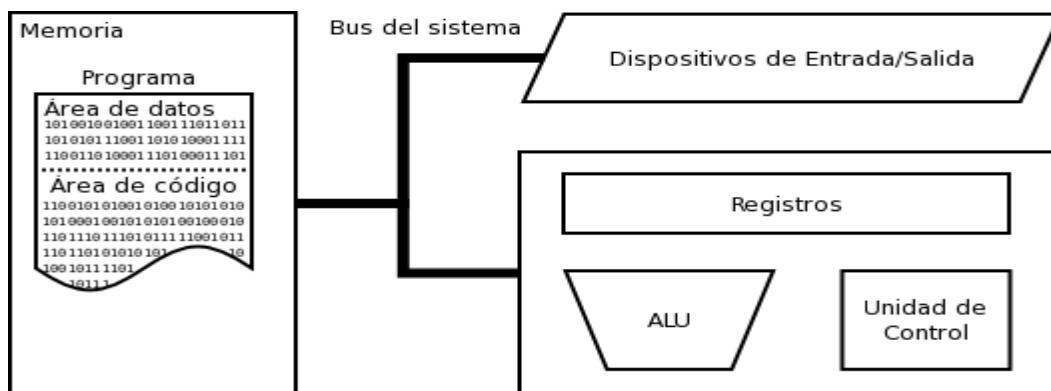
1.3.20.- ARQUITECTURA DE VON NEUMANN

El origen de la arquitectura Von Neumann surge a raíz de una colaboración en el proyecto ENIAC del matemático de origen húngaro, John Von Neumann. Este trabajaba en 1944 en el Laboratorio Nacional Los Álamos cuando se interesó por el problema de la necesidad de recablear la máquina para cada nueva tarea.

Desarrolló la solución a este problema colocando la información sobre las operaciones a realizar en la misma memoria utilizada para los datos, escribiéndola de la misma forma, es decir en código binario. Se habla desde entonces de la arquitectura de Von Neumann.

Esta idea de colocar datos y programas en una memoria única simplificaba la labor de reprogramar la máquina y además hacia independiente del HW la solución de cualquier problema. La información sobre las operaciones a realizar en la memoria con los datos es el embrión de lo que hoy conocemos como Software (los programas).

En esta arquitectura, todas las partes de un ordenador están conectadas permanentemente. La coordinación de su funcionamiento corre a cargo de una unidad de central de proceso.



Según la arquitectura Von Neumann, un computador está formado por:

- **Unidad central de proceso: (CPU)** :Es el cerebro del ordenador: controla y gobierna el sistema. Hoy en día es un circuito microscópico que interpreta las instrucciones de los programas almacenados en memoria y que tomando los datos de las unidades de entrada, procesándolos y enviándolos a las unidades de salida. Es decir se ocupa del control y proceso de datos. Está formada a su vez por tres elementos:
 - **Unidad de control (UC):** Interpreta y ejecuta las instrucciones almacenadas en la memoria y genera las señales de control necesarias para que se realicen las modificaciones sobre los registros o sobre la memoria.
 - **Unidad Aritmético lógica (ALU):** Recibe los datos para las operaciones de cálculo y comparación, toma decisiones lógicas (usa el álgebra de Boole) y devuelve el resultado de las operaciones, todo bajo el control de la UC.
 - **Registros de trabajo:** Pequeños espacios de memoria donde se almacena información temporal, dichos registros son el almacenamiento interno de la CPU. La UC y la ALU se encargan de utilizar estos registros.

- **Memoria:** Formada por los elementos que permiten almacenar y recuperar la información. Está dedicada a almacenar programas y datos de la ejecución de estos programas.
- **Sistemas de E/S:** Se encargan de la comunicación con los dispositivos periféricos con los que se comunica el usuario.
- **Bus del sistema:** Que proporciona un medio de transporte de datos e instrucciones entre las distintas partes.

Esta arquitectura es la que se utiliza con algunos matices en la mayoría de los computadores actuales.

1.3.20.1 ARQUITECTURA DE LOS PROCESADORES

Cada procesador (CPU) tiene un repertorio de instrucciones que puede ejecutar. Dependiendo de como es ese repertorio de instrucciones se pueden clasificar los procesadores en:

- **RISC (Reduced instruction-set computer):** Son instrucciones relativamente simples y no muy numerosas que usa el procesador. Disminuye la complejidad a la hora de diseñar la CPU. Como inconveniente tiene que para hacer algunas tareas se usan muchas instrucciones.
Ejemplo: SPARC de Sun Microsystem , Power PC APPLE (G4,G5,G6)
- **CISC (Complex instruction-set computer):** El procesador tiene un repertorio numeroso de instrucciones complejas, por lo que es más rápido procesando código complejo y los programas son más cortos, pero a cambio desaprovecha el uso de algunas instrucciones.

Ejemplos: familia 80x86 de Intel (Pentium incluidos) o AMD (Athlon, Duron)

1.3.20.1.1. FASES DE EJECUCIÓN DE UNA INSTRUCCIÓN:

- **Carga, búsqueda o lectura (fetch):** La UC envía a memoria principal la dirección de la instrucción a ejecutar, que está almacenada en el registro contador de programa (CP) y activa las señales de control necesarias para que la memoria le entregue la instrucción.
- **Decodificación:** La UC recibe la instrucción, la analiza y lee los operandos de la memoria principal, enviando la dirección de los operandos y activando las correspondientes señales de control.
- **Ejecución:** La UAL, bajo las ordenes de la UC, realiza la operación sobre los operandos y, si es necesario, graba el resultado en la memoria principal o en un registro.
- **Incremento del contador de programa (CP):** Se incrementa el CP, con lo que se puede pasar a ejecutar la instrucción siguiente. Hay instrucciones que pueden modificar el contenido del CP.

1.3.20.1.2. EL JUEGO DE INSTRUCCIONES. DIRECCIONAMIENTO

Los modos de direccionamiento son la forma en que el procesador determina el lugar donde se almacena un operando o la dirección a la que se debe acceder ante la instrucción que se está ejecutando.

Los fines que persiguen las técnicas de direccionamiento son:

- Dar versatilidad, para poder utilizar arrays, índices, etc.
- Reducir el número de bits del campo operando.

Algunos tipos de direccionamiento utilizados:

- **Inmediato.** El dato está en la propia instrucción.
- **Directo.** La instrucción contiene la dirección de memoria en la que se encuentra el dato. Puede ser absoluto o relativo.
- **Indirecto.** La instrucción contiene una dirección de memoria que a su vez contiene la dirección de memoria en la que se encuentra el dato
- **Indexado.** La dirección de memoria en la que se encuentra el dato se obtiene sumando a la dirección que hay en la instrucción, una cantidad que se encuentra en un registro especial llamado índice de direccionamiento indexado.