

Tema 1 - Introducción

Ordenador

Unha máquina composta de elementos físicos de tipo electrónico, capaz de realizar una gran variedade de traballos a gran velocidade e con gran precisión, sempre que se lle proporcionen as instrucións adecuadas.

Unha máquina de **propósito xeral** sempre e cando teña o **programa** preciso para realizar a tarefa.



Exercicio: Pensa se as seguintes máquinas son ordenadores de acordo coa nosa definición:

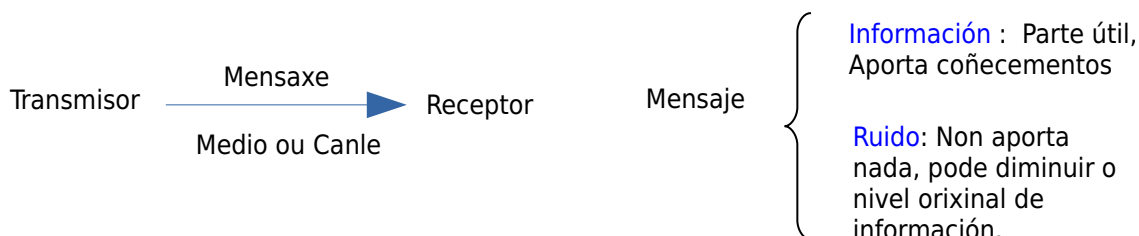
- Calculadora, lavadora, teléfono móbil.

Dato e Información

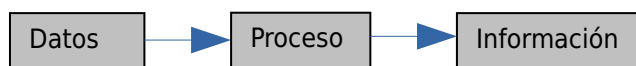
Os **datos** son magnitudes numéricas, valores cualitativos, feitos, premisas, etc., que non son suficientes por si mesmos para proporcionar un coñecemento efectivo do fenómeno tratado.

A **información** é o conxunto de **datos procesados** que aportan un coñecemento novo.

Transmisión de Información



Sistema de Información



Podemos considerar a información como o coñecemento derivado da análise dos datos mediante un proceso.

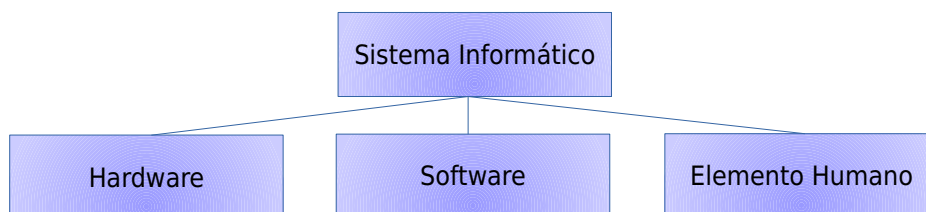
Ata a aparición dos ordenadores, o bloque de proceso realizábase de xeito manual. Pero as necesidades de procesar grandes cantidades de datos para obter Información eran cada vez maiores.

Sistema Informático

Un sistema informático non é mais que **automatizar o bloque de proceso dun Sistema de información.**

O termo informática se creouse en Alemaña no ano 1957, e procede da contracción das palabras información e automática. **A informática é polo tanto a ciencia que estuda o tratamento automático e lóxico da información.**

Elementos del sistema informático:



- O **hardware** é o conxunto de todos os elementos físicos, calquera que sea a súa natureza, (electrónica, mecánica ou electromecánica), que constitúen a parte tanxible ou material de un sistema informático.
- O **software** é o compoñente lóxico dun sistema informático. Son os programas que dan o equipo a capacidade para realizar calquera tipo de traballo. Son un conxunto de instrucións capaces de resolver un problema en termos que o ordenador poida entender.

Podemos facer unha pequena clasificación:

- **Software de base:** Sistema operativo
- **Software de programación:** Linguaxes e ferramentas de programación. Compiladores
- **Software de aplicacións:** Utilidades
- **El elemento humano** : Dependendo da súa relación co Sist. Informático, as persoas poden desempeñar diversos roles.
 - **Usuarios:** Persoa que sen grandes coñecementos de informática emprega diversas utilidades para realizar o seu traballo.
 - **Administrador do sistema** : Ten todos os privilexios. É responsable da administración dos recursos que posúe o sistema.
 - **Analista** : Persoa á que se lle propón unha necesidade do mundo real, e é a encargada de elaborar uns esquemas que conteñen a solución informática do problema proposto.
 - **Programador** : Person que se encarga de transcribir os esquemas descritos polo analista, nunha linguaxe comprensible polo ordenador.

O proceso de desenvolvemento de Software segue un proceso similar ao da construción dun edificio

Construción dun edificio

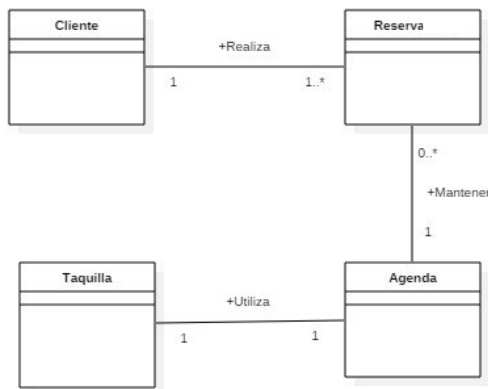


O arquitecto fai o proceso de deseño



Os obreiros en función dos planos constrúen o edificio

Desenvolvemento de software



O analista fai o deseño do programa

```
/**
 * Simple HelloButton() method.
 * @version 1.0
 * @author john doe <doe.j@example.com>
 */
HelloButton()
{
    JButton hello = new JButton( "Hello, wor
    hello.addActionListener( new HelloBtnList

    // use the JFrame type until support for t
    // new component is finished
    JFrame frame = new JFrame( "Hello Button"
    Container pane = frame.getContentPane();
    pane.add( hello );
    frame.pack();
    frame.show();           // display the fra
}
```

Os programadores codifican a solución nunha determinada linguaxe de programación

Os cambios na fase de análise son moito máis económicos que na fase de programación.

Para crear un programa:

- O código fonte escríbese nunha linguaxe de alto nivel: Java, Php, C#
- O Compilador xera un executable (binario) para una arquitectura.



Software Libre vs. Software propietario

- Software Propietario: Só nos entregan o código executable.
- Software Libre: Entrégannos ó código fonte e as seguintes liberdades.
 - Liberdade de executar o programa como desexemos.
 - Liberdade de estudar o código do programa e cambialo
 - Liberdade de redistribuír copias
 - Liberdade de distribuír copias de versións modificadas

Evolución Histórica de los ordenadores

Presentación

Sistemas de Numeración

Empregamos sistemas de numeración posicional.

$$N = \sum_{i=-1}^{i=0} a R^i$$

Decimal

- Base: 10
- Símbolos: 0,1,2,3,4,5,6,7,8,9
- $197_{10} = 1 \cdot 10^2 + 9 \cdot 10^1 + 7 \cdot 10^0$

Binario

- Base: 2
- Símbolos: 0,1
- $101_2 = 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 5_{10}$

Octal

- Base: 8
- Símbolos: 0,1,2,3,4,5,6,7
- Cada dígito octal equivale a tres dígitos binarios.

Hexadecimal

- Base: 16
- Símbolos: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
- Cada dígito hexadecimal equivale a cuatro dígitos binarios.

Decimal	Binario	Octal	Hexa
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Ejercicio:

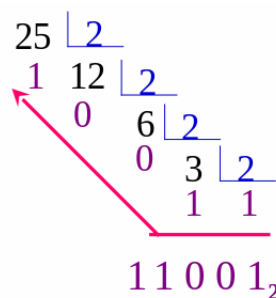
- Por que empregamos decimal?
- Por que empregamos binario?
- Por que empregamos Octal?
- Por que empregamos Hexadecimal?

Cita:

- Só hai 10 tipos de persoas, os que entenden binario e os que non.

Conversiones

- Decimal → Binario
- Octal → Binario
- Hexadecimal → Binario
- Binario → Decimal



Aritmética Binaria

A	B	A + B
0	0	0
0	1	1
1	0	1
1	1	10

Cando sumamos dous uns, o resultado é cero e prodúcese un acareo. Isto é moi importante cando sumamos números de varias cifras.

$$\begin{array}{r} 011_2 \quad 3_{10} \\ + \quad 101_2 \quad 5_{10} \\ \hline 1000_2 \quad 8_{10} \end{array}$$

Exemplo: Sumamos bit a bit, pero temos que ter en conta o acarreo para que o resultado final sexa correcto

Lóxica Binaria

A	B	A AND B	A OR B
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

A	NOT A
0	1
1	0

Arquitectura de Von Neumann

Os primeiros ordenadores de propósito xeral como o **ENIAC empregaban programación cableada**, iso facía moi custoso cambiar o programa, polo que finalmente empregábanse para unha única función: Cálculos balísticos.

En 1946 **John Von Neumann** propuxo unha serie de ideas que permitiron esquecer a programación cableada e construír auténticas máquinas de propósito xeral. A súa idea consistiu en conectar permanentemente as unidades, sendo coordinado o seu funcionamento por un control central.

Esta arquitectura aínda é a que se emprega nos ordenadores actuais. Basease en tres ideas claves:

1. Na memoria dun ordenador almacénanse simultaneamente datos e instrucións
2. Pódese acceder a información contida na memoria especificando o enderezo onde se atopa almacenada. A memoria non distingue seo que contén é un dato ou unha instrución.
3. A execución dun programa entendido como un conxunto de instrucións, realízase de xeito secuencial, pasando dunha instrución á seguinte, a non ser que explicitamente se modifique esta secuencia por un salto, sea condicional ou incondicional.

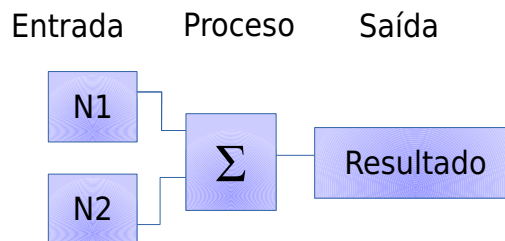
Podemos comprobar o seu funcionamento neste [simulador](http://lab.xitrus.es/VonNeumann/) (<http://lab.xitrus.es/VonNeumann/>)

Construcción dun Sistema Informático

O noso obxectivo é pensar o necesario para deseñar un sistema informático.

Dúas alternativas:

- **Configuración a medida para resolver un problema concreto.**



Isto sería **programación cableada**, pouco flexible

- **Configuración Flexible.** Considerar unha configuración de módulos aritméticos e lóxicos de propósito xeral. Este conxunto de elementos efectúa distintas funcións sobre os datos, dependendo das sinais de control recibidas.

Necesitaríamos como mínimo:

1. Un intérprete de instrucións, denominado **Unidade de Control (UC)**.
2. Un módulo de funcións lóxicas e aritméticas de propósito xeral, denominado Unidade Aritmético Lóxica (ALU)

Estas dúas unidades conxuntamente constitúen a Unidad Central de Proceso (CPU o Central Process Unit)

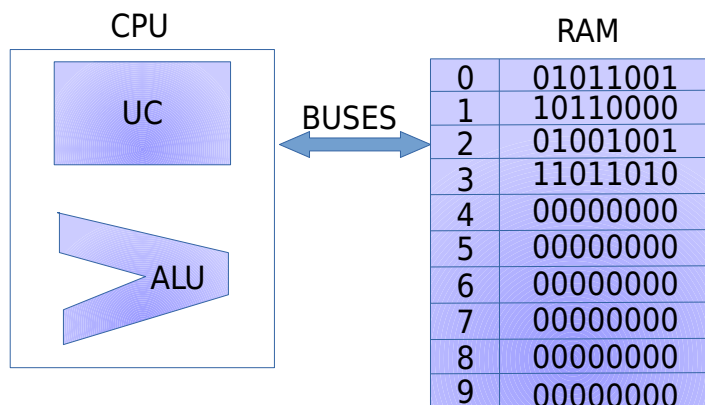
Para que el ordenador poda funcionar necesitaría compoñentes adicionais.

- Temos que introducir no sistema tanto os datos como as instrucións, para elo empregaríamos unha **unidade de entrada**. Este módulo terá que aceptar datos e instrucións e convertelos nun formato interno capaz de ser empregado polo sistema.
- Tamén precisa **unha unidade de saída** para sacar os resultados fora do sistema.

Ambas unidades constituirían as unidades de Entrada/Saída (E/S).

A unidade de Entrada introduce os datos de xeito secuencial, pero nun programa as instrucións non sempre seguen un orden secuencial, senón que poden producirse saltos, así que será necesaria una **unidade de memoria** que almacene tanto os datos como os programas (Arq. De Von Neuman).

Elementos funcionais dun sistema informático:



Unidade Central de proceso:

A CPU é o centro de operacións do ordenador, encargada de controlar e efectuar todas as operacións necesarias para executar un programa.

Dentro da CPU atopamos a unidade de control, a unidade aritmético-lóxica e os rexistros do procesador.

a) Unidade de Control:

O seu obxectivo é executar a instrución actual. Para elo terá que dirixir o que teñen que facer o resto de elementos que compoñen o ordenador.

Funcións:

- Traer de memoria a instrución a executar
- Descodificala
- Coordinar os elementos necesarios para a súa execución.

Por simplificación, supoñemos que cada instrución execútase nun ciclo. A UC posúe un reloxo que marca o ritmo o que se executan os ciclos.

A súa velocidade mídese en Hz. (1 Hz = 1 ciclo/seg)

A UC posúe varios rexistros (pequenas memorias para almacenamento temporal), entre eles o **Contador de Programa**, o cal contén en todo momento a dirección de memoria da seguinte instrución a executar.



Exercicio: O seguinte procesador **Intel Core i7-5820K 3.3Ghz**.

Cantas operacións é capaz de realizar por segundo?



Exercicio: Cal é máis rápido?:

- Intel Core i5-4460 3.2Ghz vs. Intel Core i7-5820K 3.3Ghz
- Intel Core i7-4770 3.4Ghz vs. Intel Core i7-5820K 3.3Ghz

b) Unidade Aritmético-Lóxica:

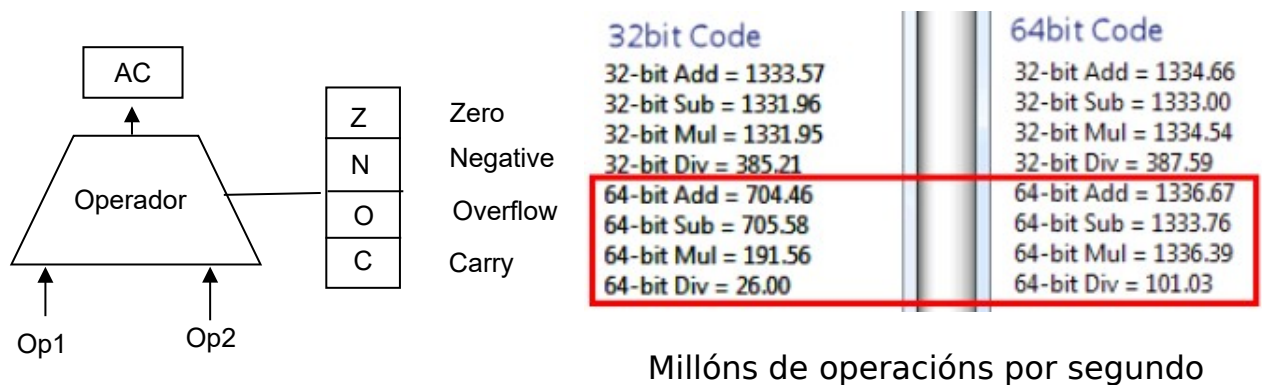
A UAL é un grupo de circuítos electrónicos que poden realizar as operacións aritméticas (sumas, restas, ...) e lóxicas (comparar, desprazar, ...) que requiran as instrucións a executar.

Posúe uns rexistros dun tamaño que pode ser de 32 ou 64 bits. Ese tamaño é o que caracteriza o tipo de CPU. Así, se dicimos que **unha CPU é de 32 bits, o que estamos indicando é que a súa UAL é capaz de procesar nunha soa operación datos de 32 bits.**

O tamaño dos rexistros inflúe:

- Na potencia de cálculo nun só ciclo
- O tamaño máximo de RAM que se pode direccionar

Posúe uns rexistros de estado para que sexa máis rápido consultar o resultado da operación anterior. Os máis importantes son acarreo e desbordamento.



Exemplo de Acarreo

$$\begin{array}{r} 9 \\ + 1 \\ \hline 10 \end{array}$$

“Lévome unha”

Exemplo de Desbordamento

$$\begin{array}{r} 99 \\ + 1 \\ \hline 1 \quad 00 \end{array}$$

“O resultado non cabe”



Exercicio: España gastou 420 millóns de euros en solucionar o efecto 2000 (Y2K). Que ten que ver coa ALU?

Memoria

A Unidade de Memoria consiste nun conxunto de posicións capaces de almacenar información, definidas polas súas direccións. Cada posición contén un número binario que pódese interpretar como un dato ou como unha instrución.

O obxectivo de calquera deseño dunha unidade de memoria é proporcionar a capacidade de almacenamento necesaria a un custo razoable e cun nivel de velocidade aceptable.

Localización

Dende este punto de vista pódense distinguir os seguintes tipos:

- **Memoria interna do procesador ou rexistros:** Empregados pola CPU como elementos de almacenamento temporal. Moi rápidos e pequenos.
- **Memoria principal:** Calquera programa para poder ser executado ten que estar na memoria principal.
- **Memoria secundaria:** Soe ter moita máis capacidade e ser máis lenta que a memoria principal. Emprégase fundamentalmente para o almacenamento de programas e datos que no están sendo continuamente empregados pola CPU. Exemplo: Discos Duros
- **Memoria auxiliar o masiva:** Soportes de información adicional como Cd-roms, Dvds, discos duros extraíbles, etc..

Capacidade

E a cantidade de información que pode almacenar. Mídese en múltiplos do byte. **A unidade mínima de almacenamento é o bit (binary digit) 0 ou 1.**

1 byte	8 bits
1 Kilobyte (KB)	2^{10} bytes = 1024 bytes
1 Megabyte (MB)	2^{20} bytes = 1024 KB = 1.048.576 bytes
1 Gigabyte (GB)	2^{30} bytes = 1024 MB = 1.073.441.824 bytes
1 Terabyte (TB)	2^{40} bytes = 1024 GB =
1 Petabyte (PB)	2^{50} bytes = 1024 TB
1 Exabyte (EB)	2^{60} bytes = 1024 PB
1 ZettaByte (ZB)	2^{70} bytes = 1024 EB
1 YottaByte (YB)	2^{80} bytes = 1024 ZB

Hai que ter cuidado porque en determinados contextos pode que non sexa o mesmo 1 KB que 1 Kb. O primeiro sería un kilobyte mentres que o segundo un kilobit (Exemplo: Velocidades de ADSL).

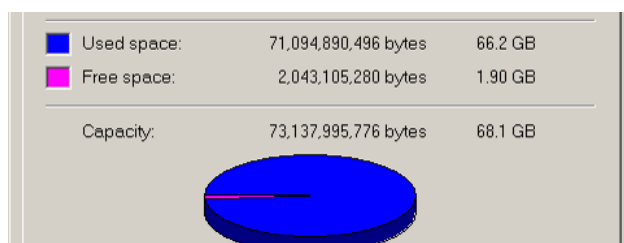
Problema: A notación é ambigua

capacidade	transmisión de información
1 KB= 1024 bytes	1 KB = 1000 bytes

Para solucionar esta ambigüidade, a Comisión Electrónica Internacional (CEI) en Decembro de 1998, decidiu engadir unha i ós prefixos cando falemos de múltiplos de 1024. Son coñecidos como os prefixos ICE.

capacidade	transmisión de información
1 MiB = 1024 KiB (Lese mibi e kivi)	1 MB = 1000 KB

De momento conviven as dúas nomenclaturas. En GNU/Linux xa se emprega a nova.



The screenshot shows the GParted application window for /dev/sda (149.05 GiB). The top bar includes menu items (GParted, Edit, View, Device, Partition, Help) and a toolbar with icons for New, Delete, Resize/Move, Copy, Paste, Undo, and Apply. Below the toolbar, a visual representation of the disk shows partitions /dev/sda2 (36.35 GiB) and /dev/sda3 (96.67 GiB). A table below lists all partitions with their details.

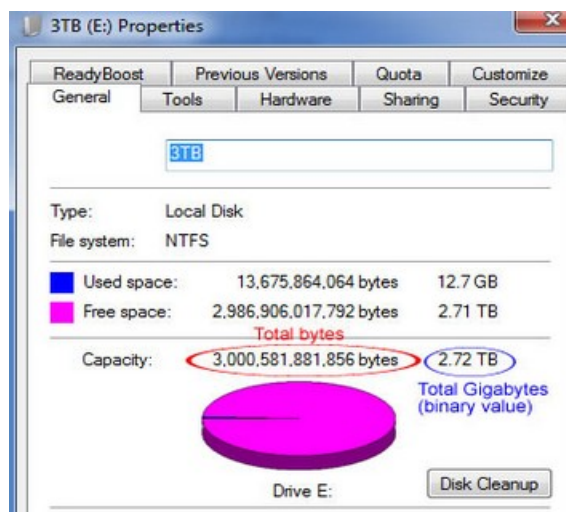
Partition	Filesystem	Mountpoint	Size	Used	Unused	Flags
/dev/sda1	fat16		39.19 MiB	---	---	
/dev/sda2	ntfs		36.35 GiB	---	---	boot
▼ /dev/sda4	extended		15.99 GiB	---	---	lba
/dev/sda5	ext3	/media/usbdisk-1	6.00 GiB	5.67 GiB	336.41 MiB	
/dev/sda6	ext3	/media/usbdisk-2	9.06 GiB	2.53 GiB	6.53 GiB	boot
/dev/sda7	fat16		502.00 MiB	---	---	lba
/dev/sda8	linux-swap		454.94 MiB	---	---	
/dev/sda3	ntfs	/media/usbdisk	96.67 GiB	44.76 GiB	51.91 GiB	

0 operations pending

The screenshot shows a file manager window titled 'BT 10.3% (72.28 MiB) ubuntu-6.10-alternate-i386.iso'. It displays the download progress of the file 'ubuntu-6.10-alternate-i386.iso' (697.07 MiB). The window includes a progress bar and detailed download statistics.

Time elapsed / estimated : 1 min 19 sec / 39 min 22 sec	
Download to: /home/omegatron/ubuntu-6.10-alternate-i386.iso	
Download rate: 319 kB/s	Downloaded: 76.02 MiB
Upload rate: 0 kB/s	Uploaded: 0.00 MiB
Share rating: 0.000	
connected to 0 peers with an average of 0.0% completed (total speed 1494 kB/s)	
connected to 35 seeds; also seeing 0.091 distributed copies	

Problema: Merquei un disco duro de 3 TiB e aparéceme menor capacidade!.



Os fabricantes de discos duros empregan os múltiplos do byte, non como múltiplos de 1024, senón como de 1000, a pesares de tratarse de capacidade de almacenamento.

Consecuencia: Perdemos un 0.7% entre o que mercamos e o que obtemos.



Exercicio: Realiza os seguintes cálculos.

- Cantos bits son 17 GiB?
- Cantos bits son 6 MiB?
- Ono ofrece ADSL 15 Mb/600 Kbps. Cantos bits temos de descarga e subida?

Método de Acceso

Xeito de acceder á información almacenada nesa memoria. Podemos distinguir dous tipos:

- a. **Acceso aleatorio:** Pode accederse ás informacións almacenadas nela en calquera orden, sendo o tempo de acceso independente da posición onde estea localizada a información. Este tipo de memoria denomínase RAM (Random Access Memory).

Exemplo: Memoria principal, discos duros, dvd.

- b. **Acceso secuencial:** Para acceder a unha determinada posición, ten que recorrer todas as anteriores. Polo tanto, o tempo necesario para acceder a unha información concreta é variable e depende da posición onde está almacenada. Exemplo: Cinta magnética.

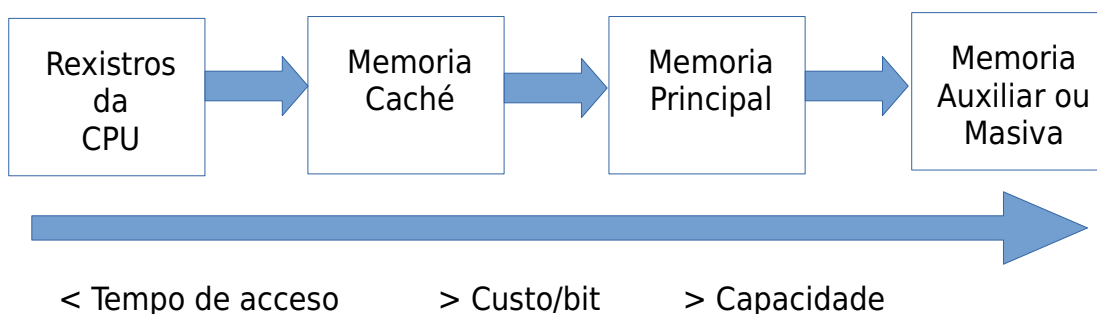
Características Físicas

- a. **Alterabilidade:** Posibilidade de alterar o contido dunha memoria
 - **ROM** (Read Only Memory) ou Só lectura: Non se pode modificar o seu contido.
 - **RAM:** Pódese modificar o seu contido
- b. **Permanencia da Información:**
 - **Volátil:** Os datos almacenados desaparecerán o deixar de ser alimentados por unha corrente eléctrica. Exemplo: Memoria Principal.
 - **Non Volátil:** Exemplo: Disco Duro.

Velocidade

Tempo de acceso á memoria é o tempo que tarda unha memoria en proporcionar un dato á CPU.

O ideal sería que toda a memoria do ordenador fose o máis rápida posible pero entón o custe sería moi grande. Por iso nos ordenadores existe o que se denomina unha **xerarquía de memorias**. Xa que se empregan diversos tipos de memorias, intentando lograr un compromiso entre capacidade, coste e velocidade.



Periféricos, dispositivos ou unidades de E/S:

Os elementos que o ordenador emprega para comunicarse co exterior.

Dentro dos dispositivos podemos distinguir os seguintes grupos:

- Unidades de entrada: Teclado, rato, escáner,...
- Unidades de saída: Impresora, pantalla,..
- Unidades de Entrada/Saída: Pantallas táctiles...

Xestión de E/S

É importante xestionar moi eficientemente as peticións E/S porque estánse producindo constantemente: rato, teclado...

- **Sondeo**

Cada certo tempo consultamos ó periférico por se necesita a nosa atención. Problema: Ineficiente, desperdiciamos moitos ciclos de CPU

- **Interrupcións: IRQ's**

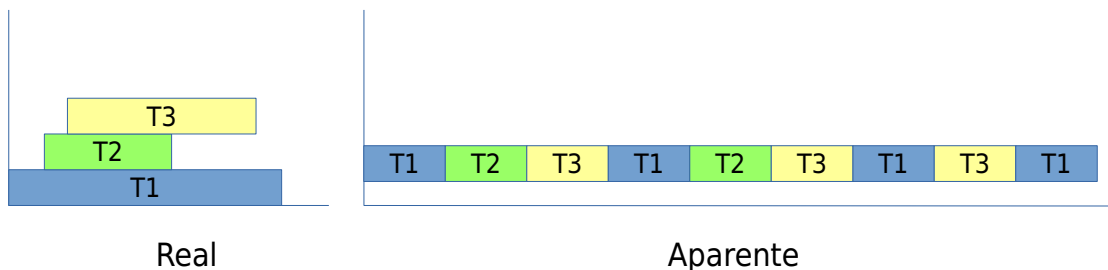
Cada periférico ten unha canle individual para avisar á CPU e interrompela. A CPU nese momento almacena o estado do programa que estaba executando, atende o periférico e cando remate, continua con a execución do programa no punto no que se atopaba.

A interrupcións proporcionan:

- Un xeito de xestionar a E/S
- Un xeito de interromper ó proceso activo
- Multitarefa:
 - Real: Temos tantas tarefas como Cpu's. As tarefas se executan simultaneamente.
 - Aparente: Temos máis tarefas que Cpu's, polo que a multitarefa é simulada.

En cada segundo execútanse unhas cantas instrucións do primeiro programa, do segundo... e así sucesivamente ata rematar todas as tarefas.

Isto é posible porque cada certo tempo prodúcese unha interrupción e se cambia a tarefa activa



Mecanismos de interconexión, Buses:

Os Buses son os cables que interconectan as unidades funcionais. Distinguimos tres tipos en función da información que transmiten:

- **Bus de Datos:** Transporta os datos
- **Bus de Direccións:** É empregado pola UC para transmitir as direccións de memoria ás que necesita acceder
- **Bus de Control:** Transmite sinais de control entre os diferentes compoñentes.

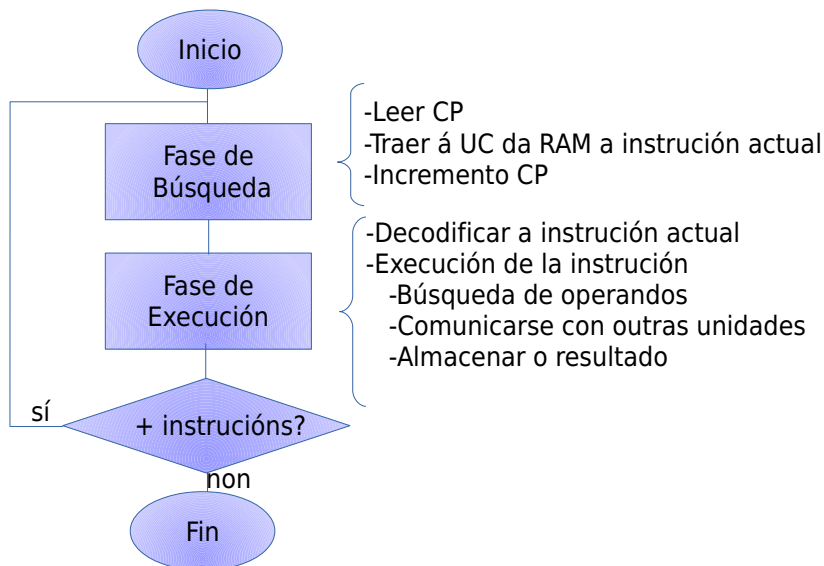
FUNCIONAMENTO DO ORDENADOR

A función básica que realiza un ordenador é a execución dun programa. Un programa consiste nun conxunto de instrucións almacenadas nunha unidade de memoria. A CPU é a encargada de executar as instrucións específicas no programa.

A secuencia de operacións realizadas pola CPU durante a execución dunha instrución constitúe o que se coñece como Ciclo de Instrución.

Consta de dúas fases:

- a) Fase o ciclo de Búsqueda
- b) Fase o ciclo de Execución



Representación da Información

Sabemos que os sistemas de memoria almacenan só números binarios. Pero os usuarios non só queremos almacenar números. Queremos almacenar texto, imaxes, vídeo....

Texto

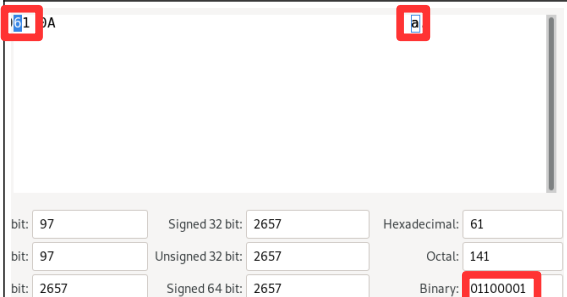
Para almacenar texto precisamos un código no que un conxunto de ceros e uns preestablecidos se correspondan a un símbolo. **Ese código é o ASCII.**

Caracteres de control ASCII			Caracteres ASCII imprimibles									ASCII extendido																	
DEC	HEX	Símbolo ASCII	DEC	HEX	Símbolo	DEC	HEX	Símbolo	DEC	HEX	Símbolo	DEC	HEX	Símbolo	DEC	HEX	Símbolo	DEC	HEX	Símbolo	DEC	HEX	Símbolo	DEC	HEX	Símbolo	DEC	HEX	Símbolo
00	00h	NULL (carácter nulo)	32	20h	espacio	64	40h	@	96	60h	`	128	80h	Ç	160	A0h	á	192	C0h	Ł	224	E0h	Ó						
01	01h	SOH (inicio encabezado)	33	21h	!	65	41h	A	97	61h	a	129	81h	Û	161	A1h	í	193	C1h	ł	225	E1h	ó						
02	02h	STX (inicio texto)	34	22h	"	66	42h	B	98	62h	b	130	82h	ê	162	A2h	ô	194	C2h	Ť	226	E2h	Ô						
03	03h	ETX (fin de texto)	35	23h	#	67	43h	C	99	63h	c	131	83h	â	163	A3h	û	195	C3h	Ŧ	227	E3h	Õ						
04	04h	EOT (fin transmisión)	36	24h	\$	68	44h	D	100	64h	d	132	84h	ä	164	A4h	ÿ	196	C4h	Ũ	228	E4h	ö						
05	05h	ENQ (enquiry)	37	25h	%	69	45h	E	101	65h	e	133	85h	å	165	A5h	ÿ	197	C5h	Ÿ	229	E5h	Û						
06	06h	ACK (acknowledgement)	38	26h	&	70	46h	F	102	66h	f	134	86h	ä	166	A6h	ä	198	C6h	Ž	230	E6h	ü						
07	07h	BEL (timbre)	39	27h	'	71	47h	G	103	67h	g	135	87h	ç	167	A7h	ö	199	C7h	Ž	231	E7h	ÿ						
08	08h	BS (retroceso)	40	28h	(72	48h	H	104	68h	h	136	88h	è	168	A8h	ï	200	C8h	Ł	232	E8h	ÿ						
09	09h	HT (tab horizontal)	41	29h)	73	49h	I	105	69h	i	137	89h	ë	169	A9h	©	201	C9h	Œ	233	E9h	Û						
10	0Ah	LF (salto de línea)	42	2Ah	*	74	4Ah	J	106	6Ah	j	138	8Ah	è	170	AAh	ŕ	202	CAh	Œ	234	EAh	Ü						
11	0Bh	VT (tab vertical)	43	2Bh	+	75	4Bh	K	107	6Bh	k	139	8Bh	ï	171	ABh	¼	203	CBh	Ŧ	235	EBh	Û						
12	0Ch	FF (form feed)	44	2Ch	,	76	4Ch	L	108	6Ch	l	140	8Ch	ì	172	ACH	½	204	CAh	Ŧ	236	ECh	ÿ						
13	0Dh	CR (retorno de carro)	45	2Dh	-	77	4Dh	M	109	6Dh	m	141	8Dh	í	173	ADh	¾	205	CDh	Ŧ	237	EDh	ÿ						
14	0Eh	SO (shift Out)	46	2Eh	.	78	4Eh	N	110	6Eh	n	142	8Eh	Ā	174	ACh	»	206	CEh	Ŧ	238	EEh	ÿ						
15	0Fh	SI (shift In)	47	2Fh	/	79	4Fh	O	111	6Fh	o	143	8Fh	Ā	175	AFh	»	207	CFh	Ŧ	239	EFh	ÿ						
16	10h	DLE (data link escape)	48	30h	0	80	50h	P	112	70h	p	144	90h	Ē	176	B0h	⋮	208	D0h	Ŧ	240	F0h	ÿ						
17	11h	DC1 (device control 1)	49	31h	1	81	51h	Q	113	71h	q	145	91h	æ	177	B1h	⋮	209	D1h	Ŧ	241	F1h	±						
18	12h	DC2 (device control 2)	50	32h	2	82	52h	R	114	72h	r	146	92h	Æ	178	B2h	⋮	210	D2h	Ŧ	242	F2h	¼						
19	13h	DC3 (device control 3)	51	33h	3	83	53h	S	115	73h	s	147	93h	ô	179	B3h	⋮	211	D3h	Ŧ	243	F3h	¾						
20	14h	DC4 (device control 4)	52	34h	4	84	54h	T	116	74h	t	148	94h	ò	180	B4h	⋮	212	D4h	Ŧ	244	F4h	Ŧ						
21	15h	NAK (negative acknowle.)	53	35h	5	85	55h	U	117	75h	u	149	95h	õ	181	B5h	⋮	213	D5h	Ŧ	245	F5h	§						
22	16h	SYN (synchronous idle)	54	36h	6	86	56h	V	118	76h	v	150	96h	ù	182	B6h	⋮	214	D6h	Ŧ	246	F6h	÷						
23	17h	ETB (end of trans. block)	55	37h	7	87	57h	W	119	77h	w	151	97h	û	183	B7h	⋮	215	D7h	Ŧ	247	F7h	ÿ						
24	18h	CAN (cancel)	56	38h	8	88	58h	X	120	78h	x	152	98h	ÿ	184	B8h	⋮	216	D8h	Ŧ	248	F8h	ÿ						
25	19h	EM (end of medium)	57	39h	9	89	59h	Y	121	79h	y	153	99h	Œ	185	B9h	⋮	217	D9h	Ŧ	249	F9h	ÿ						
26	1Ah	SUB (substitute)	58	3Ah	:	90	5Ah	Z	122	7Ah	z	154	9Ah	Œ	186	BAh	⋮	218	DAh	Ŧ	250	FAh	ÿ						
27	1Bh	ESC (escape)	59	3Bh	;	91	5Bh	[123	7Bh	{	155	9Bh	ø	187	BBh	⋮	219	DBh	Ŧ	251	FBh	ÿ						
28	1Ch	FS (file separator)	60	3Ch	<	92	5Ch	\	124	7Ch		156	9Ch	£	188	BCh	⋮	220	DCh	Ŧ	252	FCh	ÿ						
29	1Dh	GS (group separator)	61	3Dh	=	93	5Dh]	125	7Dh	}	157	9Dh	Ø	189	BDh	⋮	221	DDh	Ŧ	253	FDh	ÿ						
30	1Eh	RS (record separator)	62	3Eh	>	94	5Eh	^	126	7Eh	~	158	9Eh	x	190	BEh	⋮	222	DEh	Ŧ	254	FEh	ÿ						
31	1Fh	US (unit separator)	63	3Fh	?	95	5Fh	_				159	9Fh	f	191	BFh	Ŧ	223	DFh	Ŧ	255	FFh	ÿ						
127	20h	DEL (delete)																											
			elCodigoASCII.com.ar																										

Cada carácter ocupa 8 bits en código ASCII polo que temos 256 posibles símbolos.

Exemplo:

Creamos un arquivo con unha letra a, Se o examinamos cun editor hexadecimal:



Amosa o contido en:

- Hexadecimal: 61
- Código ASCII: a
- Binario: 01100001

Imaxes

Supoñamos que queremos almacenar unha imaxe en formato .bmp. Teremos que almacenar cada un dos píxeles que a forman.

Supoñamos que queremos almacenar a seguinte imaxe



- As súas dimensións son 2x2 píxeles
- Como é en branco e negro só necesitamos almacenar 1 bit por pixel

O formato do arquivo .bmp podería ser o seguinte

0002 _(h)	0002 _(h)	0001 _(h)	1	1	0	1
Res. Horiz	Res. Vert	Bits/pixel				



Exercicio: .

- Os campos resolución Horizontal e Vertical son de 16 bits. Cales son as dimensións máximas que podería alcanzar unha imaxe almacenada así?