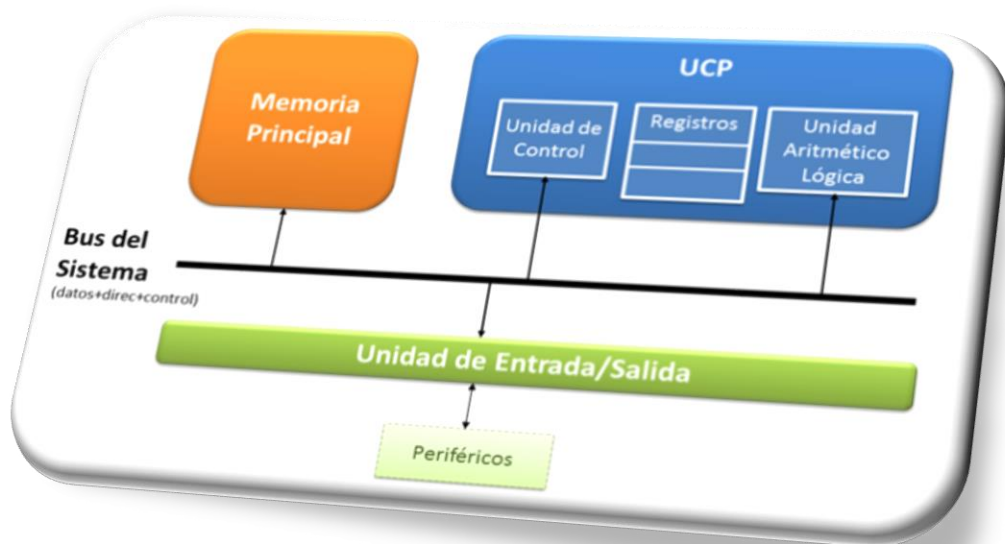


# SISTEMES INFORMÀTICS

---

## UD 1: INTRODUCCIÓ ALS SISTEMES INFORMÀTICS



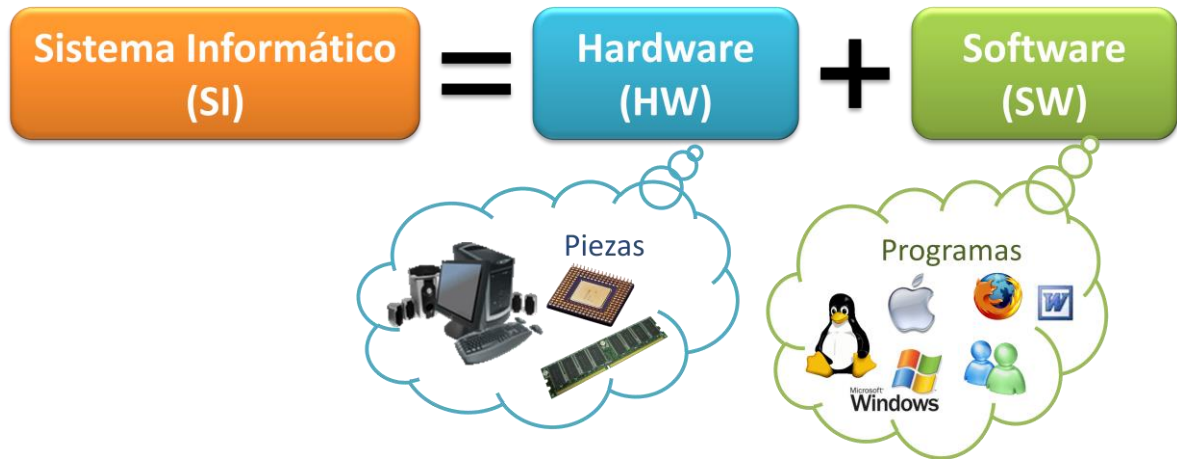
## ÍNDEX

<b>1.</b>	<b>EL SISTEMA INFORMÀTIC .....</b>	<b>3</b>
<b>2.</b>	<b>COMPONENTS FÍSICS: MAQUINARI .....</b>	<b>4</b>
2.1.	ARQUITECTURA VON NEUMANN .....	4
2.2.	UNITAT CENTRAL DE PROCÉS .....	4
2.3.	MEMÒRIA PRINCIPAL .....	4
2.4.	UNITAT D'ENTRADA/EIXIDA .....	5
2.5.	DISPOSITIUS PERIFÈRICS .....	5
2.6.	BUSOS.....	5
2.7.	FUNCIONAMENT. EXECUCIÓ D'UNA INSTRUCCIÓ .....	5
<b>3.</b>	<b>COMPONENTS LÒGICS: DADES I PROGRAMARI D'UN SI .....</b>	<b>6</b>
3.1.	DADES. REPRESENTACIÓ INTERNA. SISTEMES DE NUMERACIÓ .....	6
3.1.1.	SISTEMES DE NUMERACIÓ .....	6
3.1.2.	CANVIS DE BASE (CONVERSIONS) .....	8
3.1.3.	OPERACIONS EN BINARI .....	9
3.2.	PROGRAMARI: SO, APLICACIONS I LENGUATGES DE PROGRAMACIÓ .....	10

## 1. El Sistema informàtic

L'ordinador és l'eina que ens permet el tractament automàtic de la informació, és a dir, que ens permet organitzar, processar, transmetre i emmagatzemar la informació.

Un **sistema informàtic**, en major o menor mesura, és precisament això, un conjunt d'elements de **maquinari** i **programari** interconnectat per al tractament automàtic de la informació (d'ací la paraula informà-tica). D'altra banda, també es podria incloure l'element humà (per exemple, quan la persona que usa el sistema, intervé en ell introduint dades).



Per tant, un sistema informàtic es pot definir com una sèrie d'elements **físics (maquinari)** capaç de fer moltes tasques a gran velocitat i amb gran precisió. Perquè aquests elements físics realitzen un procés determinat, és necessari que en ell s'executen un conjunt d'ordres o **instruccions (programari)**, components no físics que posen en funcionament tots aqueixos components físics.

Podem fer una classificació dels SI en:

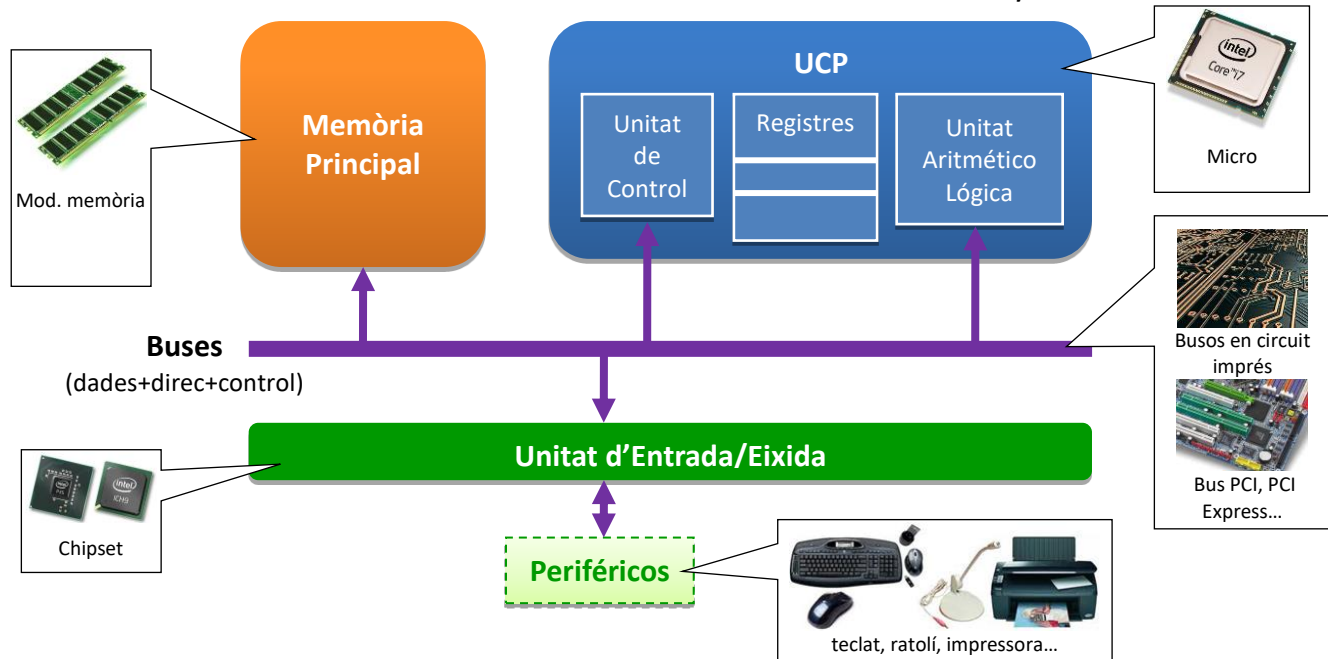


D'aquesta manera, trobem des dels **supercomputadors** utilitzats per grans organitzacions en la realització de tasques molt avançades de tipus científic, militar, tecnològic..., passant pels **macroordinadors** i **servidors** que proporcionen serveis diversos a altres equips, fins a arribar als **ordinadors personals** i **dispositius de butxaca** (mòbils moderns, PDA...) que usem en l'àmbit domèstic o professional.

## 2. Components físics: Maquinari

### 2.1. Arquitectura Von Neumann

El model d'arquitectura bàsica emprada en els ordinadors digitals va ser establert per Von **Neumann** en 1945. Aquesta màquina és capaç d'executar una sèrie d'instruccions o ordres elementals anomenades **instruccions màquina**, que han d'estar emmagatzemades en una unitat d'emmagatzematge anomenada **Memòria Principal**, per a poder ser llegides i executades per la **Unitat Central de Procés**. La comunicació amb l'exterior es fa a través de la **Unitat d'Entrada / Eixida**.



### 2.2. Unitat Central de Procés

La **Unitat Central de Procés** o microprocessador (UCP o CPU): s'encarrega de l'execució dels programes emmagatzemats en memòria i està formada per:

- La **Unitat de Control** (UC): decideix i controla les operacions que es realitzaran a cada moment. La seua funció és descodificar les instruccions del programa en execució i generar tots els senyals necessaris perquè puguin ser executades.
- La **Unitat Aritmètica Lògica**: realitza les operacions aritmètiques (sumes, restes...) i lògiques (And, Not, Or...).
- Els **Registres**: són un magatzem temporal durant l'execució d'una instrucció. Es tracta de memòries molt xicotetes d'accés molt ràpid.

### 2.3. Memòria principal

La **Memòria Principal**: conté les instruccions dels programes per a ser executats per la UCP i les dades a usar.

S'organitza com un conjunt de cel·les consecutives de longitud fixa (paraula), enumerades de manera que cada cel·la és directament accessible pel seu número denominat direcció. La unitat mínima d'informació que emmagatzema és el **bit** (amb dos estats: 0 i 1).

La seua capacitat se sol expressar en:

Mesura	Simbologia	Equivalència	Equival. (Bytes)	Mesura	Simbologia	Equivalència	Equival. (Bytes)
Byte	B	8bits		Terabyte	TB	1024 GB	$2^{40}$ B
Kilobyte	KB	1024 B	$2^{10}$ B	Petabyte	PB	2014 TB	$2^{50}$ B
Megabyte	MB	1024 *KB	$2^{20}$ B	Exabyte	EB	2014 *PB	$2^{60}$ B
Gigabyte	GB	1024 MB	$2^{30}$ B	Zettabyte	YB	1024 EB	$2^{70}$ B

## 2.4. Unitat d'Entrada/Eixida

La **Unitat d'Entrada/Eixida** permet la comunicació de la UCP i la Memòria Principal amb l'exterior: per exemple, impressores, ratolins i/o monitors. Entre les seues **funcions** destaquem:

- **Adreçament**: identifica els dispositius assignant-los una direcció única, anomenada **port**.
- **Sincronització**: sincronitza les diferents velocitats dels dispositius amb la UCP.
- **Transferència**: realitza l'intercanvi d'informació entre els dispositius i la UCP.

En l'actualitat, la **Unitat d'Entrada/Eixida** equival al **Chipset**.

## 2.5. Dispositius perifèrics

Es poden **classificar** segons la direcció de la informació amb la qual treballen en:

- **Entrada**: introdueixen informació a l'ordinador. Ex.: teclat, ratolí, escàner, micròfon, webcam...
- **Eixida**: extrauen informació de l'ordinador. Ex.: monitor, impressora, altaveus...
- **Entrada/eixida**: permeten introduir i extraure informació. Ex.: pantalla tàctil, impressores multifunció (impressora + escàner), memòries auxiliars (discos durs, lectors de CD o DVD, memòries flaix, \*pendrives...)

## 2.6. Busos

Els **Busos**: són els encarregats de la interconnexió entre els diferents elements que componen l'ordinador. Segons la informació que transporten es poden classificar en els següents **tipus**:

- **Dades**: porten informació (dades o instruccions) als diferents elements
- **Adreces**: porten la direcció de memòria de la qual es llegirà o emmagatzemar informació
- **Control**: porta senyals de control a cada element des de la UC, indicant com ha de posar-se en funcionament

## 2.7. Funcionament. Execució d'una instrucció

**Execució d'una instrucció:**

1. La UC **porta de memòria principal la pròxima instrucció a executar**, apuntada pel CP i s'incrementa aquest comptador perquè apunte a la següent instrucció.
2. **Descodifica** i interpreta la instrucció llegida per a determinar **el codi d'operació i les dades**.
3. Genera els senyals de control per a executar **la instrucció**.
4. Una vegada obtingut el **resultat, s'escriu en memòria**.

Activitat de classe: vegem un exemple i analitzem el funcionament fent clic [ací](#)

### 3. Components lògics: dades i programari d'un SI

Com s'ha indicat en el punt 1, la part **Programari d'un SI** són els **programes** i les **dades**.

El programa que s'executa quan s'arranca un ordinador és el **Sistema Operatiu (SO)**, el qual és el responsable del maneig i control del funcionament de l'equip. Una vegada carregat el SO ja és possible l'execució d'aplicacions.

Cada aplicació o **programa consta d'una** sèrie d'instruccions, la quals són carregades en MP i executades per la UCP tal com s'ha explicat en el punt anterior.

Un **programa** està format per un conjunt de fitxers escrits en algun **llenguatge de programació** (java, C#, javascript, Python, ...).

A continuació, vegem com es representen internament les dades.

#### 3.1. Dades. Representació interna. Sistemes de numeració

Un ordinador és una màquina capaç de processar informació, o més concretament, **dades** mitjançant l'execució d'una sèrie d'instruccions. Una dada és la representació de la informació de manera adequada per al seu tractament per un ordinador. Tots dos tipus d'informació, dades i instruccions, han d'estar **representats internament** d'una manera comprensible per al computador i, per tant, identificable pels seus circuits electrònics: el **sistema binari**, la unitat mínima del qual és el **bit** (binary digit) i només pot tindre dos valors diferents: **0 i 1**. Aquests valors es representen emprant la tensió elèctrica: normalment, el valor 0 indica absència de tensió elèctrica (0 V) i l'1 indica presència de tensió (+5 V).

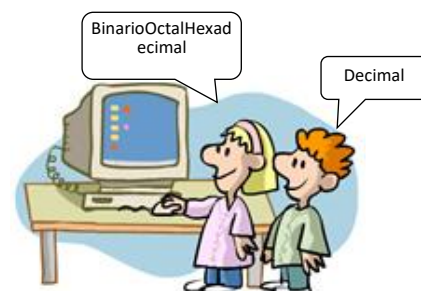
##### 3.1.1. Sistemes de numeració

Es defineix un **sistema de numeració** com el conjunt de símbols i regles que s'utilitzen per a representar dades numèriques.

Un sistema de numeració **en base b** utilitza, per a representar els números, un alfabet compost per b símbols diferents.

Els sistemes de numeració que es tractaran són **posicionals**, és a dir, són sistemes en els quals cada dígit dependrà del símbol utilitzat i de la posició que aquest ocupe en el número.

Els sistemes posicionals més usats són: Decimal (base 10), Binari (base 2), Octal (base 8) i Hexadecimal (base 16).



**Exemple:** En el sistema de numeració decimal, es té  $b = 10$  i l'alfabet està constituït per deu signes diferents  $\{0, 1, 2, \dots, 9\}$ .

Activitat de classe: trobar un exemple de dos números diferents del sistema decimal que usen els mateixos dígits

Els sistemes posicionals estan basats en el **teorema fonamental de la numeració (TFN)**, que relaciona una quantitat expressada en qualsevol sistema de numeració amb la mateixa quantitat expressada en el sistema decimal.

D'aquesta manera, si l'expressió d'un número N en base b és:

$$N_b = \dots n_3 n_2 n_1 n_0 n_{-1} n_{-2} n_{-3} \dots$$

Llavors, el seu valor decimal seria:

$$N_{10} = \dots + n_2 * b^2 + n_1 * b^1 + n_0 * b^0 + n_{-1} * b^{-1} + n_{-2} * b^{-2} + \dots$$

Per tant, per a la conversió d'un número en qualsevol base a decimal, s'utilitzarà aquest teorema.

En general, els ordinadors efectuen les operacions aritmètiques utilitzant una **representació** per a les dades numèriques basada en el **sistema binari**. També s'utilitzen com a codis **intermedis** els **sistemes octal i hexadecimal**, ja que requereixen menys símbols que el binari per a representar el seu valor i a més, la conversió entre aquests sistemes i el binari és directa.

### Sistema Binari:

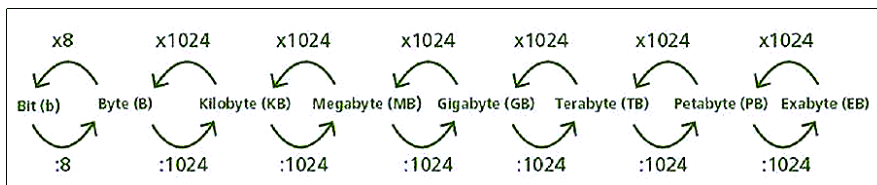
- Té base 2: dos signes diferents → 0 i 1. Cadascun d'aquests signes rep el nom de bit (binary digit)
- Avantatges: S'utilitza en els circuits electrònics per a caracteritzar els dos estats en què pot trobar-se.
- Inconvenient: utilitza més dígit que qualsevol altre sistema per a representar una quantitat.
- Múltiples del bit (b):
  - Byte (B): 8 bits
  - Kilobyte (KB):  $1024 B = 2^{10} B$
  - Megabyte (MB):  $1024 MB = 2^{20} B$
  - Gigabyte (GB):  $1024 MB = 2^{30} B$
  - Terabyte (TB):  $1024 GB = 2^{40} B$
  - Petabyte (PB):  $1024 TB = 2^{50} B$
  - Exabyte (EB):  $1024 * PB = 2^{60} B$
  - Zettabyte (ZB):  $1024 EB = 2^{70} B$
  - Yottabyte (YB):  $1024 * ZB = 2^{80} B$

Compte! no es el mateix **b** y **B**

- **b**: bit
- **B**: byte = 8 bits



Conversió  
d'unitats



### Sistema Octal:

Octal	Binari
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

- Té base 8: dígit → del 0 al 7
- Cada símbol en octal es representa per 3 símbols binaris.
- Avantatges: La conversió entre números de la base binaria y la octal es immediata, donat que  $8=2^3$ .

## Sistema hexadecimal:

Hexadecimal	Binari
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111

Hexadecimal	Binari
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

- Base 16: 16 signes distints  $\rightarrow 0..9A..F$
- Cada símbol en hexadecimal es pot representar mediant una combinació de 4 símbols binaris.
- Avantatges: La conversió entre números de la base binaria y la hexadecimal es immediata, donat que  $16=2^4$ .

### 3.1.2. Canvis de base (conversions)

#### Binari $\rightarrow$ decimal:

S'utilitza el TFN (teorema fonamental de la numeració), és a dir, es multiplica el valor de cada dígit binari per la potència de 2 de la posició que ocupe el dígit dins del número:

Exemples:

10010110<sub>2</sub> a decimal:

11001,01<sub>2</sub> a decimal:

$$11001,01_2 = 2^4 + 2^3 + 2^0 + 2^{-2} = 16 + 8 + 1 + 0.25 = 25,25_{10}$$

10010110<sub>2</sub> = 150<sub>10</sub>

#### Decimal $\rightarrow$ binari:

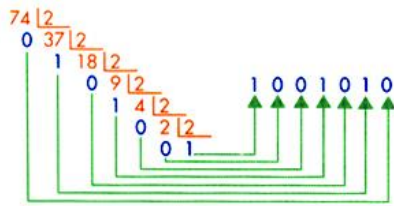
Per a convertir la **part sencera** de decimal a binari, aquest ha de ser dividit per dos i es repeteix el procés amb els seus quocients fins que el quocient prengui el valor 1. La unió de totes les restes escrites en ordre invers encapçalats per l'últim quocient ens donarà el valor expressat en binari.

Per a passar la **part fraccionària** de decimal a binari, es multiplica per 2 i es va afegint al número la part sencera obtinguda que es descartaria per a la següent multiplicació.



Exemple: 74,375<sub>10</sub> a binari

Primer passem a binari la part sencera:



Luego pasamos la parte fraccionaria:

$$0,375 \times 2 = 0,750 \rightarrow \text{Añado } 0$$

$$0,750 \times 2 = 1,50 \rightarrow \text{Añado } 1$$

$$0,5 \times 2 = 1,0 \rightarrow \text{Añado } 1$$

$$74,375_{10} = 1001010,011_2$$

**Binari ↔ Octal:**

La conversió entre la base binària i la base octal és immediata, donat que  $8=2^3$ .

Exemple:  $111011,001_2 = 73,1_8$

$\underbrace{111}_7 \underbrace{011}_3 \underbrace{,001}_1$

**Binari ↔ Hexadecimal:**

La conversió entre la base binària i l'hexadecimal és immediata, donat que  $16=2^4$ .

Exemple:  $11010111,11100011_2 = D7,E3_{16}$

$\underbrace{1101}_D \underbrace{0111}_7 \underbrace{,1110}_E \underbrace{0011}_3$

### 3.1.3. Operacions en binari

#### Operacions aritmètiques

SUMA	RESTA	PRODUCTE	DIVISIÓ
$0+0=0$	$0-0=0$	$0*0=0$	$0/0=\text{No definit}$
$0+1=1$	$0-1=1^1$ (préstec)	$0*1=0$	$0/1=0$
$1+0=1$	$1-0=1$	$1*0=0$	$1/0=\text{Impossible}$
$1+1=0^1$ (implique)	$1-1=0$	$1*1=1$	$1/1=1$

Cuando nos encontramos con tres unos, la suma da 1 y de acarreo 1

**Ejemplo de SUMA**

	1	1	1	1		Acarreos
	↓		↓	↓	↓	
	1	1	0	1	1	1 → 55
+	1	0	0	0	1	1 → 35
	1	0	1	1	0	1 → 90

· Cuando nos encontramos con el primer 0 – 1, el resultado es 1 y nos llevamos 1, que sumaremos al siguiente sustraendo.

· Si al sumar nos volvemos a llevar 1 (caso de sumar 1 de acarreo + 1 en sustraendo), ese 1 pasa al siguiente sustraendo, y así sucesivamente hasta que dé 0.

**Ejemplo de RESTA**

	1	1	0	0	1	0	1	→ 101
-	↓1	↓1		↓1				Acarreos
		1	1	0	1	1	→ 27	
	1	0	0	1	0	1	0	→ 74

## Operacions lògiques

A	B	NOT A	A AND B	A NAND B	A OR B	A NOR B	A XOR B
0	0	1	0	1	0	1	0
0	1		0	1	1	0	1
1	0	0	0	1	1	0	1
1	1		1	0	1	0	0

### Exemple:

A = 11101100 y B = 00011101

A AND B = 00001100

A OR B = 11111101

A NAND B = 11110011

A NOR B = 00000010

A XOR B = 11110001

**NAND:** es la negación de AND.  
**NOR:** es la negación de OR.  
**XOR:** el resultado será 0 cuando ambos valores sean iguales y 1 cuando sean diferentes

## 3.2. Programari: SO, aplicacions i llenguatges de programació

Segons la **funció** que realitza, podem classificar el programari en:

- Programari de base o **Sistema Operatiu**: programari que actua com a intermediari entre l'usuari i el maquinari, ocultant la seua complexitat amb una interfície fàcil d'usar. Els seus objectius són fer còmode l'ús de la computadora i utilitzar els recursos de l'ordinador de manera eficient.
- Programari d'aplicació (aplicacions): programes usats per a fer una tasca específica, com: Word, jocs, antivirus...
- Programari de programació: programes que serveixen per a desenvolupar altres programes. Per exemple, els entorns de desenvolupament NetBeans, Eclipse o Microsoft Visual Studio, els quals permeten construir programes utilitzant **llenguatges de programació**. Un llenguatge de programació és un conjunt de codis que seguint una sintaxi permet la introducció d'ordres en l'ordinador.



Components d'un Sistema Informàtic

D'altra banda, els **llenguatges de programació** poden ser:

- De baix **nivell**: pròxims al maquinari
  - ✓ Llenguatge **màquina**: únic comprensible directament per l'ordinador. Està format per zeros i uns.
  - ✓ Llenguatge **assemblador**: basat en el llenguatge màquina. Cada instrucció del llenguatge assemblador és equivalent a una instrucció del llenguatge màquina, però en comptes d'usar 0 i 1 s'utilitzen instruccions bàsiques. Les seues instruccions varien depenent del microprocessador.



- D'alt **nivell**: més comprensible per a les persones. Transportable d'un ordinador a un altre i independent del maquinari. Ha de ser traduït al llenguatge màquina perquè l'ordinador els entenga. Exemples: C++, Java o C#.

### **Bibliografia**

- Jiménez, I. (2012). "Sistemes Informàtics". Garceta.
- Stallings, W. (2005). "Sistemes Operatius". Pearson.