

Aspectes tecnològics dels sistemes informàtics

Antoni Martínez Ballesté
Gregorio Robles Martínez

PID_00150290



Universitat Oberta
de Catalunya

www.uoc.edu



Els textos i imatges publicats en aquesta obra estan subjectes –llevat que s'indiqui el contrari– a una llicència de Reconeixement-NoComercial-SenseObraDerivada (BY-NC-ND) v.3.0 Espanya de Creative Commons. Podeu copiar-los, distribuir-los i transmetre'ls públicament sempre que en citeu l'autor i la font (FUOC. Fundació per a la Universitat Oberta de Catalunya), no en feu un ús comercial i no en feu obra derivada. La llicència completa es pot consultar a <http://creativecommons.org/licenses/by-nc-nd/3.0/es/legalcode.ca>

Índex

Introducció	5
Objectius	6
1. Perspectiva històrica de la informàtica	7
1.1. L'electrònica i els primers computadors	8
1.2. L'ordinador personal	10
1.3. La informàtica, a tot arreu i per a tothom	10
2. Tractament de la informació	12
2.1. El món binari	12
2.1.1. Els nombres en binari	13
2.1.2. Els caràcters	15
2.1.3. Els múltiples del byte i del bit	15
2.2. El tractament lògic de la informació	16
3. Components del sistema informàtic	18
3.1. L'estructura del maquinari	18
3.1.1. El microprocessador	19
3.1.2. La memòria	20
3.1.3. La placa base	22
3.1.4. Dispositius d'entrada i sortida d'informació	22
3.1.5. Dispositius d'emmagatzematge d'informació	23
3.2. El programari	24
3.2.1. Algoritmes	26
3.2.2. Disseny del programari	27
3.2.3. Evolució dels llenguatges de programació	29
3.3. Tipus de programari	33
3.3.1. El programari de sistemes: el sistema operatiu	34
3.3.2. Programari per a usuari final	37
4. Tipus d'ordinador	39
4.1. Els ordinadors centrals	39
4.2. Els microcontroladors	40
4.3. Els superordinadors	40
4.4. Multicomputadors	40
4.5. Els ordinadors portàtils	41
5. Sistemes d'informació	42
Resum	45

Activitats	47
Exercicis d'autoavaluació	47
Solucionari	49
Glossari	50
Bibliografia	52

Introducció

En aquest mòdul estudiarem els aspectes tecnològics dels ordinadors i els sistemes informàtics. L'ordinador ha esdevingut l'element que vehicula el coneixement dins la societat de la informació. Els sistemes informàtics estan formats per un o diversos ordinadors, de diferents tipus i amb diferents propòsits (ordinadors personals per a feines d'oficina, computadors que formen part d'entorns industrials etc.). Sobre aquests ordinadors s'executen una gran diversitat de programes. En molts sistemes informàtics, el programari inclou sistemes d'informació per a l'emmagatzematge i gestió de dades.

Els termes *computació* i *informàtica* defineixen el propòsit inicial dels ordinadors: el càlcul i el tractament automàtic de la informació. Tot i ser termes sinònims, la paraula *informàtica* té avui dia un sentit més generalista que no pas computació, ja que la informàtica ha permès l'accés al públic en general a tota una sèrie de potents eines que faciliten multitud de tasques, i permeten fins i tot la comunicació sense barreres i l'accés a quantitats ingents d'informació.

Les paraules *computador* i *ordinador* també tenen significats lleugerament diferents. Un computador tindria una utilitat no tan generalista com un ordinador. Normalment, els computadores designen els primers ordinadors (destinats a fer còmputos i càlculs), mentre que el terme *ordinador* s'aplicaria als computadores actuals. Tanmateix, en anglès el terme *computer* serveix per a definir tant computadores com ordinadors.

Els ordinadors van començar com a eines de càlcul per a facilitar tasques rutinàries com ara operacions aritmètiques, o bé grans processos de càlcul, com ara la gestió de censos. En l'apartat 1 del mòdul, coneixerem les fites més importants en la història de la informàtica i els ordinadors. En l'apartat 2 del mòdul, veurem com tracten els ordinadors la informació en el món digital.

En l'apartat 3 del mòdul, distingirem les diferents parts que formen un sistema informàtic. Bàsicament veurem que hi ha una part tangible, anomenada *maquinari*, que serveix de suport a la part intangible de la informàtica: els programes i la informació. També definirem aquest component flexible dels ordinadors: el programari (o *software*). Veurem què és el programari i aprendrem com es crea. Així, sabrem com es transforma un programa des de l'especificació fins a un programa que es pugui executar en un ordinador.

Finalment, en l'apartat 4 del mòdul trobareu els tipus d'ordinadors actuals que podem trobar i, en l'últim apartat, els sistemes d'informació que s'utilitzen de manera generalitzada en el món empresarial i governamental.

Objectius

Els objectius que l'estudiant haurà assolit en finalitzar aquest mòdul són:

- 1.** Tenir una perspectiva històrica de la informàtica.
- 2.** Comprendre els fonaments del tractament de la informació per part dels ordinadors.
- 3.** Distingir les parts fonamentals d'un sistema informàtic i entendre quins són els seus objectius.
- 4.** Entendre els conceptes més importants relacionats amb el programari, incloent-hi totes les etapes de creació des de l'especificació en algoritmes fins a la implementació utilitzant un llenguatge de programació d'alt nivell.
- 5.** Conèixer l'evolució dels llenguatges de programació.
- 6.** Conèixer els diferents tipus de programari: el programari de sistemes (el més significatiu del qual és el sistema operatiu) i el programari d'usuari final, fent especial èmfasi en els sistemes d'informació.

1. Perspectiva històrica de la informàtica

Els inicis de la informàtica són més aviat difosos. El que avui coneixem com a ordinadors i informàtica és el resultat de l'avenç de diferents disciplines, en especial les matemàtiques, l'electrònica i la física. Tanmateix, els esforços de la humanitat per a tenir ginyes que els ajudin amb els càlculs es remunten a èpoques antigues.

Els inicis de la informàtica van lligats a la invenció de diferents estris i ginyes dedicats a facilitar el càlcul matemàtic.

Uns dels més antics és l'àbac, originari de la Xina del segle v aC. Es tracta d'un marc de fusta amb eixos sobre els quals es fan lliscar boles. Els àbacs faciliten la realització de sumes, restes, multiplicacions i divisions. Aquest instrument fou adaptat i utilitzat per moltes altres cultures.

Durant els segles XVII i XVIII l'avenç de la mecànica propicia el disseny i fins i tot la producció de diferents màquines de calcular. Una de les més cèlebres és la Pascalina, feta l'any 1645 per Blaise Pascal i de la qual se'n van produir una cinquantena de còpies.



La Pascalina, una de les primeres calculadores. Exposada al Musée des Arts et Métiers de Paris

Al segle XIX té lloc el primer disseny del que es pot considerar un computador programable i no una mera màquina de calcular: la màquina analítica, de Charles Babbage.

La màquina analítica de Babbage estava pensada per a poder implementar petits programes de càlcul, mostrar la sortida i desar dades en una memòria.

Malauradament, el seu inventor no la va poder veure realitzada per una sèrie de problemes, sobretot perquè l'enginyeria de l'època encara no estava preparada per a portar-la a la pràctica. Ada Lovelace, filla de Lord Byron, va estudiar el funcionament d'aquesta màquina i fins i tot va proposar mètodes per a definir procediments a partir de les operacions que la màquina permetria fer. Ada Lovelace és considerada la primera persona que va treballar en els conceptes de programació d'ordinadors.

Aquí comença el que es considera la història dels ordinadors. Tot seguit en veurem les fites més importants. Abans, però, descriurem una divisió clàssica de les etapes de la història dels ordinadors que els agrupa en diferents generacions:

- La primera generació comprèn els grans ordinadors electromecànics en què l'electrònica es controlava per mitjà de vàlvules i els circuits eren cablejats.
- La segona generació s'inicia amb la introducció del transistor i la reducció consegüent de la mida dels equips. Un mòdul basat en components electrònics de la mida d'una bombeta, ara, es podria dissenyar amb components no més grans que una mosca.
- La tercera generació porta uns ordinadors encara més petits gràcies a l'aparició dels microxips. Així, doncs, un circuit cablejat de dimensions considerables comparables a la mida d'una taula de menjador es podia reduir a la mida d'un segell de correus.
- La quarta generació dels ordinadors comprèn els ordinadors personals. La miniaturització iniciada amb la tercera generació encara és més evident, ja que els circuits integrats ja poden incloure diferents circuits, fins i tot sobreposats en diferents capes per a poder realitzar diverses tasques.

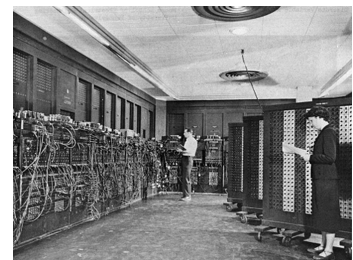
Es considera que la generació actual d'ordinadors és la cinquena. En aquesta generació s'introdueixen conceptes com ara el multiprocessador i la memòria cau.

1.1. L'electrònica i els primers computadors

L'arribada de l'electrònica a mitjan el segle XX va propiciar la materialització de dissenys que, amb tecnologies purament mecàniques, no eren realitzables. Així, doncs, durant la dècada dels anys 1940 van veure la llum els primers computadors electromecànics. Aquests ordinadors, grans reeiximents de l'enginyeria de l'època, eren ben diferents dels ordinadors que coneixem avui dia: eren màquines de grans dimensions, alimentades per una potència elèctrica considerable i generadores de soroll i altes temperatures. Per a controlar els voltatges i intensitats que regulaven els mòduls de processament d'informació es feien servir vàlvules, com les que hi havia dins les ràdios antigues.

L'any 1941 es va construir l'ordinador Z3 i es va presentar a Berlín. Malauradament, aquest equip va quedar destruït en un bombardeig durant la Segona Guerra Mundial. És per això que la popularitat la va guanyar el Mark I, dissenyat i construït a la Universitat de Harvard. Fou presentat l'any 1944. El va seguir l'ENIAC¹, presentat l'any 1946 a la Universitat de Pennsilvània. Totes dues màquines tenien grans dimensions i ocupaven una sala sencera. Elevaven la temperatura ambient i necessitaven una gran quantitat d'energia per a arrencar. La diferència entre l'ENIAC i el Mark I és que amb el primer s'hi interactuava per mitjà de cables i taulers de commutació, mentre que el segon

⁽¹⁾ENIAC és l'acrònim d'*electronic numerical integrator and calculator*, en català calculadora i integrador numèric electrònic.



L'ENIAC, un dels primers grans ordinadors

disposava de lectors de cinta perforada. Aquesta tècnica d'emmagatzematge ja s'emprava per a les partitures de les pianoles automàtiques i primeres calculadores com ara la màquina analítica.

Les màquines de calcular analògiques

Tot i que la gran majoria d'ordinadors d'aquestes èpoques es basen en el tractament digital de la informació, durant el segle XX també es van desenvolupar màquines de calcular analògiques. En aquestes, per exemple, per a fer una operació de suma se sumaven dos voltatges diferents. El resultat es mostrava en un voltímetre.

Durant els anys 1950 diferents empreses fabricaren grans ordinadors com a producte comercial. El primer computador comercial fou l'UNIVAC² I. D'aquestes màquines, que ocupaven uns 30 m² cadascuna, se'n van vendre unes quaranta còpies a un milió de dòlars. IBM va entrar en el mercat dels ordinadors fent màquines considerablement més petites i, en conseqüència, més econòmiques. Un dels primers models populars a partir de l'any 1954 fou l'IBM 650, amb uns 900 kg de pes i que necessitava un espai d'uns 3 m³.

⁽²⁾En anglès *universal automatic computer*, computador automàtic universal.

Al final d'aquesta dècada, l'aparició del transistor i els circuits integrats facilitaren la reducció de la mida dels computadors, alhora que en minvaren el consum energètic. Paral·lelament, van apareixent equips per a l'entrada i sortida de dades, ja que, com hem esmentat anteriorment, els primers computadors s'havien de programar per mitjà de cables i endolls (com és el cas de l'ENIAC) i la sortida es representava en un panell de bombetes. Les pantalles de televisió i les màquines d'escriure es van adaptar per a facilitar la interacció dels operadors amb les màquines.

Als inicis de la informàtica, cada fabricant utilitzava els seus sistemes de representació d'informació i els seus sistemes de programació d'ordinadors. Durant els anys seixanta i setanta del segle XX, es van fer esforços perquè els ordinadors es miniaturitzessin i fessin servir estàndards tant a nivell de programari, com de representació de la informació o, fins i tot, a nivell de comunicació.

Vegeu també

Les xarxes d'ordinadors s'introduiran en el mòdul "Aspectes tecnològics de les xarxes i Internet".

En aquest sentit, veuen la llum els sistemes UNIX, els llenguatges de programació generalistes per a fer programes i, fins i tot, apareixen els primers intents d'interconnexió d'ordinadors en una Internet incipient.

Al final dels anys 1960 apareix l'enginyeria informàtica, especialitzada en el disseny de sistemes informàtics per a gestionar els processos d'informació. En la mateixa època, la complexitat creixent de la gestió de la informació fa que s'ideïn eines d'enginyeria per al disseny de programes, començant el que anomenem *enginyeria del programari*. Els conceptes teòrics sobre els models matemàtics i de processament d'informació relacionats amb els computadors donaren lloc a les ciències de la computació.

1.2. L'ordinador personal

Els anys 1980 tenen com a estrella la irrupció de la informàtica personal gràcies a l'aparició de la quarta generació d'ordinadors. Fins aleshores les empreses grans i mitjanes ja disposaven d'ordinadors centrals o *mainframes* on diversos treballadors podien accedir des dels seus terminals. Els ordinadors personals IBM PC³ i els Apple van fer possible que moltes petites empreses i particulars s'endinsessin en el món de la informàtica, per afició o bé per millorar la productivitat.

⁽³⁾De l'anglès *personal computer*.

Els IBM PC utilitzaven xips fabricats per Intel i un sistema operatiu (allò que fa possible l'execució dels programes) fabricat per Microsoft. Totes dues empreses esdevindrien de les més importants en el món de la informàtica. La companyia Apple es va fer popular amb el model Macintosh, el qual aprofitava unes idees que tenien el seu origen en investigacions de l'empresa Xerox: el ratolí i les finestres. Aquests conceptes serien portats al món dels IBM PC per Microsoft per mitjà del sistema operatiu Windows.

Durant els vuitanta i els noranta del segle xx, els ordinadors van evolucionar a una velocitat vertiginosa. D'una banda, van aparèixer microordinadors com ara el ZX Spectrum, que van popularitzar la informàtica a un cost assequible. Aquestes màquines no eren compatibles amb els PC d'IBM (és a dir, els programes fets per aquests ordinadors no podien funcionar en els altres). Moltes companyies van dissenyar els microordinadors sense tenir en compte, en general, la compatibilitat entre productes de diferents marques.



Microordinador de la casa Spectrum. Com molts ordinadors de la seva època, s'havia de connectar al televisor de casa.

D'altra banda, Intel va permetre que altres companyies fessin servir els seus xips, amb la qual cosa centenars d'empreses van fabricar ordinadors personals compatibles amb el d'IBM i sensiblement més econòmics que l'original. Així, doncs, després d'uns quants anys en què multitud de marques i models ocupaven una part important del mercat de la informàtica domèstica, els ordinadors compatibles amb els IBM PC van passar a ser els més venuts.

1.3. La informàtica, a tot arreu i per a tothom

L'evolució dels ordinadors PC quant a velocitat i capacitat, l'aparició dels ordinadors portàtils, les impressores de qualitat fotogràfica, les càmeres digitals, etc. són històries prou conegudes per nosaltres donat que són recents i les podem recordar.

Els ordinadors personals han adquirit una popularitat notable, en part gràcies al seu abaratiment i també per les possibilitats que ofereix Internet com a peça integrant de qualsevol sistema informàtic. L'ordinador ja no és una eina exclusivament dedicada a millorar la productivitat: l'oci personal, la gestió de continguts multimèdia, el comerç electrònic, etc. són fets del dia a dia vehiculats pels ordinadors personals.

Els ordinadors portàtils han esdevingut un complement de l'ordinador de sobretaula, i en molts casos l'han substituït. La miniaturització dels ordinadors ha permès la fabricació i popularització de telèfons mòbils (que en el fons són petits ordinadors), aparells digitals d'enregistrament de vídeo (que també ho són), cotxes controlats i gestionats per un sistema informàtic de bord, etc.

Persones de qualsevol nivell cultural tenen accés a les eines informàtiques que els permeten formar part de la societat de la informació. L'adquisició d'informació i el seu tractament automàtic ha esdevingut universal en tots els àmbits..., si més no en el que anomenem *el primer món*.

One Laptop Per Child Association

Diverses associacions, com ara One Laptop Per Child Association, promouen la fabricació i distribució d'equips informàtics per a països en vies de desenvolupament.

2. Tractament de la informació

Els ordinadors, i la gran majoria d'aparells electrònics, treballen en el que s'anomena **món digital**. En contraposició, el món real és analògic. En el món analògic, les forces i senyals que trobem (com ara la gravetat, l'electricitat, el so, els colors, etc.) són continus, en contraposició al món digital, en què les mesures són discretes.

Defugint d'una descripció basada en les matemàtiques, posarem un exemple. Suposem que volem mesurar la longitud d'una taula. Si utilitzem un sistema continu de mesura, utilitzarem tota la precisió necessària per a calcular la mida exacta de la taula.

En el món analògic, totes les taules tenen llargades diferents, encara que a vegades difereixin de milionèsimes de mil·límetre.

En un món digital, les mesures i els senyals són discrets, és a dir, tenen una precisió determinada que especifica un conjunt de valors. Per tant, no hi pot haver infinits valors diferents a part dels que defineixen una determinada precisió.

En un món digital amb una precisió de centímetres, totes les taules que en el "món analògic" tenen 1.200, 1.201, 1.202, 1.203..., 1.209 mil·límetres es consideraran de 120 centímetres.

En aquest apartat veurem com es tracta la informació en el món digital.

2.1. El món binari

Els circuits electrònics en què es basen els computadors funcionen a partir d'intensitats de corrent elèctric. Així, doncs, un transistor pot estar emetent corrent o bé pot estar sense emetre'n. De la mateixa manera, els sistemes primitius d'emmagatzematge d'informació en targetes perforades utilitzaven forats per a desar les dades: o bé hi ha forat o bé no n'hi ha.

Tot això respon al que s'anomena *àlgebra de Boole*, ideada pel matemàtic George Boole al segle XIX. Les operacions de càlcul complexes es poden reduir a **operacions lògiques binàries**, que treballen sobre dos valors possibles (per exemple, el zero o l'u; el forat o l'absència de forat). En els primers ordinadors, aquestes operacions es duïen a terme amb diferents mòduls de càlcul que es basaven, en general, en les senzilles operacions lògiques de l'àlgebra de Boole. La representació de nombres en el sistema binari, és a dir en zeros i uns, va

Els ordinadors quàntics

Els ordinadors quàntics, actualment en fase d'investigació, permetrien treballar amb més valors que no pas els dos de l'àlgebra de Boole.

fer possible que es pogués tractar automàticament la informació amb aquests mòduls de càlcul. Avui dia, els ordinadors tenen la mateixa base de funcionament.

Els ordinadors tracten la informació fent servir el sistema binari, format per uns i zeros. Un **bit** és un dígit binari, és a dir un d'aquests dos valors. En conseqüència, un bit és la quantitat mínima d'informació que pot tractar un ordinador.

Tanmateix, els mòduls de càlcul i els xips actuals no fan els càlculs només amb 1 bit. Normalment es fan servir grups de bits, per exemple 4 bits, 8 bits, 16 bits, 32 bits o 64 bits. El més habitual històricament va ser treballar amb grups de 8 bits, anomenats *bytes*.

2.1.1. Els nombres en binari

Qualsevol nombre es pot representar en binari. Per exemple, vet aquí tots els nombres naturals que es poden representar amb tres bits:

Nre. binari	Nre. natural
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

La quantitat de nombres naturals que es poden representar dependrà del nombre de bits. En concret, amb n bits es poden representar 2^n nombres diferents, del 0 al $2^n - 1$.

El byte

Com que els mòduls de càlcul podien treballar amb la informació a base de "menjar-se-la a mossegades de 8 bits", el grup de 8 bits s'anomena *byte*, que sona igual que *bite* ('mossegada' en anglès).

El nombre decimal que representa un nombre en binari

Per a obtenir el nombre decimal que representa un nombre en binari, cal multiplicar el dígit (zero o u) per un valor que correspon a la base (en aquest cas, dos per ser el sistema binari) elevada al nombre que indica la posició del dígit, entenent que la posició de més a la dreta és la posició zero. Per entendre-ho millor, vegem un exemple. En el cas dels 3 bits, els valors de cada un d'esquerra a dreta són $4(2^2)$, $2(2^1)$ i $1(2^0)$. Per tant, el número binari 100 correspon al número decimal 4, ja que $1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 4$. De la mateixa manera, el número 111 correspon al número 7, ja que $1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 7$.

Fixem-nos que això també passa amb el sistema decimal que utilitzem els humans, en què la base és 10. Per exemple, el número 2009 es pot descompondre en diferents potències: $2 \cdot 10^3 + 0 \cdot 10^2 + 0 \cdot 10^1 + 9 \cdot 10^0$.

En el cas dels nombres enters (és a dir els que poden ser positius o negatius), hi ha diferents alternatives per a representar-los en binari. Una d'elles seria utilitzar un dels bits per a dir si el nombre és positiu o bé negatiu. En el cas anterior, es necessitaria 1 bit de més. Fixeu-vos que apareix el cas estrany en què hi ha dues representacions possibles del zero!

Nombres positius		Nombres negatius	
Nre. binari	Nre. decimal	Nre. binari	Nre. decimal
0000	0	1000	-0
0001	1	1001	-1
0010	2	1010	-2
0011	3	1011	-3
0100	4	1100	-4
0101	5	1101	-5
0110	6	1110	-6
0111	7	1111	-7

Gràcies a la representació de nombres en el sistema binari, va ser possible crear màquines digitals de càlcul, des dels primers grans ordinadors fins a les calculadores de butxaca.

El sistema binari no és l'únic que es fa servir en informàtica. També és habitual l'anomenat *sistema hexadecimal*, en què la base és 16. L'objectiu d'aquest sistema és representar nombres binaris de manera més compacta utilitzant menys dígits que si els representem amb zeros i uns.

Nombre decimal	Nombre binari	Nombre hexadecimal	Nombre decimal	Nombre binari	Nombre hexadecimal
0	0000	0	8	1000	8
1	0001	1	9	1001	9
2	0010	2	10	1010	A

Origen del sistema decimal

Els antropòlegs afirmen que l'origen del sistema decimal és en els deu dits que sempre ens han servit de base per a comptar.

Nombre decimal	Nombre binari	Nombre hexadecimal	Nombre decimal	Nombre binari	Nombre hexadecimal
3	0011	3	11	1011	B
4	0100	4	12	1100	C
5	0101	5	13	1101	D
6	0110	6	14	1110	E
7	0111	7	15	1111	F

2.1.2. Els caràcters

Els primers ordinadors estaven dedicats a fer grans càlculs, però la transmissió i emmagatzematge d'informació digital en format text van necessitar codificar aquesta informació. Per exemple, els textos enviats per un sistema de transmissió s'havien de codificar amb bits.

Un dels sistemes de codificació de caràcters, i el més important en l'actualitat, és l'ASCII⁴. Un codi ASCII de 8 bits pot representar 256 símbols diferents. Per exemple, la lletra *A* té assignat el codi 65 (que en binari és 01000001), la xifra 1 té el codi 49, l'interrogant té el codi 63, etc.

La codificació de caràcters

La frase *la vida es bella* ocupa en l'ordinador un total de 16 bytes (tretze lletres més els tres espais en blanc). El número 2009, si es tracta com una cadena de caràcters, ocuparà 4 bytes (32 bits), un per a cada xifra, però si el número es tracta com un nombre enter ocuparà 11 bits, ja que es farà servir el seu valor en binari: 11111011001.

Amb l'augment de la transmissió d'informació amb diferents llengües amb el codi ASCII apareix un problema: no es poden representar tots els sistemes d'escriptura. Actualment, s'utilitza el codi UNICODE (de l'anglès *Universal Code*) de 16 bits on es poden representar fins i tot els símbols de les llengües asiàtiques.

2.1.3. Els múltiples del byte i del bit

Els sistemes informàtics de tercera i quarta generació són capaços de tractar i emmagatzemar milions de bytes d'informació. D'aquesta manera s'han definit múltiples del byte:

- Un **quilobyte** o KB són 1.024 bytes. També s'anomena *ka*.
- Un **megabyte** o MB són 1.024 KB o 1.048.576 bytes. També s'anomena *mega*.
- Un **gigabyte** o GB són 1.024 MB o 1.073.741.824 bytes. També s'anomena *giga*.
- Un **terabyte** o TB són 1.024 GB o 1.099.511.627.776 bytes. També s'anomena *tera*.

Codi Morse

L'ús de la codificació és anterior als mateixos ordinadors. Un exemple el trobem en el codi Morse, en què diferents símbols (lletres i xifres, bàsicament) es representen en un codi de punts i ratlles.

⁽⁴⁾De l'anglès *American standard code for information interchange*, codi estàndard americà per a l'intercanvi d'informació.

Exemples sobre quant poden ocupar alguns elements habituals

Una carta feta amb un processador de textos pot ocupar 25 KB. No solament s'hi desen els caràcters que la formen, sinó també informació addicional com ara el tipus de lletra, el color del text, possibles imatges integrades, etc. Una foto d'alta qualitat pot ocupar 2 MB. Una cançó per a ser escoltada en un reproductor MP3 ocupa uns 4 MB. Una pel·lícula en DVD pot ocupar uns 6 GB.

Per què múltiples de 1.024 i no pas de 1.000? Doncs per raons estrictament relacionades amb el sistema binari i la construcció dels ordinadors. Tanmateix, per a l'especificació de la capacitat de suports d'emmagatzematge, com ara els discos durs i els DVD, es fan servir múltiples de 1.000. Així, doncs, s'esperaria poder desar 5.046.586.572 bytes en un DVD de 4,7 GB, quan realment n'hi caben 4.700.000.000 bytes.

Quan es parla de **velocitats de transferència d'informació**, parlem de múltiples de bit per segon. Per exemple, tindrem els quilobits per segon (kbit/s) o els megabits per segon (Mbit/s). En aquests casos, es fan servir múltiples de 1.000.

2.2. El tractament lògic de la informació

Un cop hem vist com es representen els valors numèrics i els caràcters en els sistemes digitals o informàtics, veurem alguns conceptes bàsics sobre com es dissenyen els mòduls de càlcul que ja hi havia en els primers computadors.

Hem vist que l'àlgebra de Boole es fonamenta en una sèrie de funcions lògiques que treballen amb bits. Usant aquestes funcions, es poden fer mòduls sumadors, restadors, multiplicadors, etc. Les tres funcions lògiques més importants són:

- La funció **no**, anomenada *NOT*, canvia el valor del bit que hi entra. Per exemple, el valor 0 es converteix en un 1 i l'1 en un 0.
- La funció **i**, anomenada *AND*, la qual té el valor 1 de sortida quan tots els valors d'entrada són 1, i 0 en cas contrari.
- La funció **o**, anomenada *OR*, que té valor 1 quan alguna de les entrades val 1, i 0 si totes les entrades valen 0.

Aquestes funcions tenen un símbol que les representa i una **taula de veritat** associada. En la figura següent es mostra el símbol per a les tres funcions lògiques acabades de definir, com també les seves taules de veritat.

Vegeu també

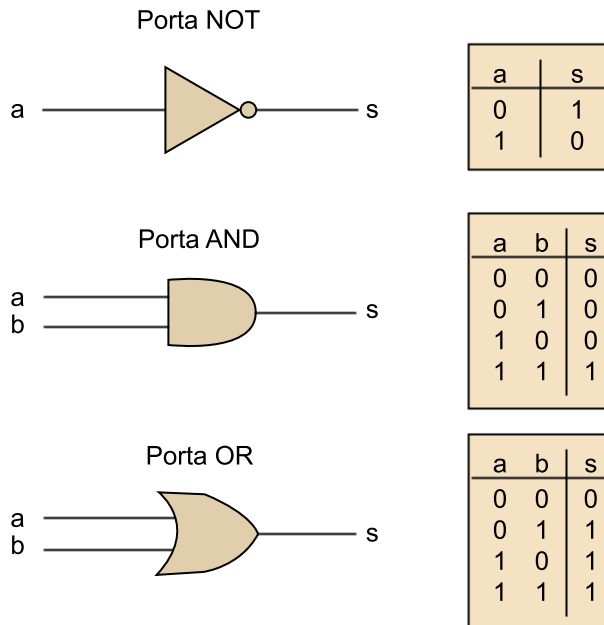
Els formats multimèdia (imatge, vídeo i so) s'explicaran en el mòdul "El World Wide Web".

Vegeu també

Tractarem de les velocitats de transferència d'informació en el mòdul "Aspectes tecnològics de les xarxes i Internet".

Taula de veritat

Una taula de veritat especifica, per les diferents combinacions dels bits d'entrada, quina és la sortida d'una funció lògica o un circuit de càlcul.



A dalt, porta lògica NOT, al centre la porta AND, i a sota la porta OR.

Altres funcions utilitzades són la XOR, la NAND o la NOR. La funció XOR té 1 com a sortida només si una de les dues entrades és 1. La NAND i la NOR són els resultats de "negar" les sortides de les funcions AND i OR respectivament. Totes aquestes funcions es poden implementar físicament en un circuit mitjançant la combinació de dispositius electrònics, com ara transistors. Encadenant aquests dispositius s'aconsegueixen funcions més complexes dissenyades a partir de les bàsiques i, en conseqüència, es poden fer mòduls per a càlculs complexos.

Els mòduls de càlcul estan dissenyats per a treballar amb un nombre concret de bits, per exemple, 8 bits. Encadenant mòduls d'aquests, s'aconsegueixen mòduls capaços de processar 16 bits, 32 bits o 64 bits.

Per a fer operacions més complexes que la suma o la resta, per exemple, una divisió o una arrel quadrada, hi ha dues alternatives:

- Dissenyar blocs específics per a dur a terme aquestes operacions. Això implicaria tenir molts mòduls que sovint no s'estarien fent servir. Alguns dels primers grans ordinadors seguien aquesta alternativa.
- Fer un únic mòdul capaç de realitzar operacions "complexes" a partir de la definició de procediments que utilitzin un conjunt reduït de mòduls de càlcul senzills; per exemple, fer la resta a partir del mòdul suma amb el segon operand canviat de signe, fer una divisió a base de restes, etc. Aquesta és la tendència que ha marcat el disseny dels ordinadors moderns. Aquest mòdul de càlcul s'anomena *processador*.

3. Components del sistema informàtic

Un cop hem vist com els ordinadors i els sistemes digitals tracten la informació, estudiarem els components del sistema informàtic. Un sistema informàtic està format per dos tipus de components:

- **Components de maquinari**⁵. Es tracta de tots els components físics i electrònics que configuren els sistemes informàtics. Dins aquests components trobem, entre altres, el processador, el teclat, el lector de DVD, etc.
- **Components de programari**⁶. En formen part tots els programes que funcionen en l'ordinador i les dades que manegen. Es tracta, doncs, de la part "intangible" del sistema informàtic.

⁽⁵⁾En anglès, *hardware*.

⁽⁶⁾En anglès, *software*.

Un element important del programari és el **sistema operatiu**, el qual crea una "capa" intermèdia entre el maquinari i el programari amb l'objectiu de facilitar la interacció del programari i l'usuari amb el maquinari. Un cas concret de programari és el que es troba enregistrat en els mateixos xips del maquinari, per exemple, en els dispositius mòbils com telèfons o GPS, que rep el nom de **firmware**.

3.1. L'estructura del maquinari

L'estructura del maquinari de l'ordinador es basa en l'anomenada *arquitectura Von Neumann*. John Von Neumann, un matemàtic prolífic, va proposar que els ordinadors haurien d'estar governats per una **unitat de control**, la qual s'encarregaria de transportar informació des de la memòria cap a una **unitat de càlcul** (anomenada *unitat aritmeticològica*) i tornar-la a la **memòria** un cop acabades les operacions. Va definir, a més, uns **mòduls per a l'entrada i sortida** d'informació, com també tres camins (o **busos**) per on circularien diferents tipus d'informació: les dades amb què cal treballar, el nom de la instrucció a executar per la unitat **aritmeticològica** i la posició en memòria on s'ubica la informació i es desaran els resultats. Les instruccions que conformen el procediment d'operacions a realitzar també s'emmagatzemen en la memòria.

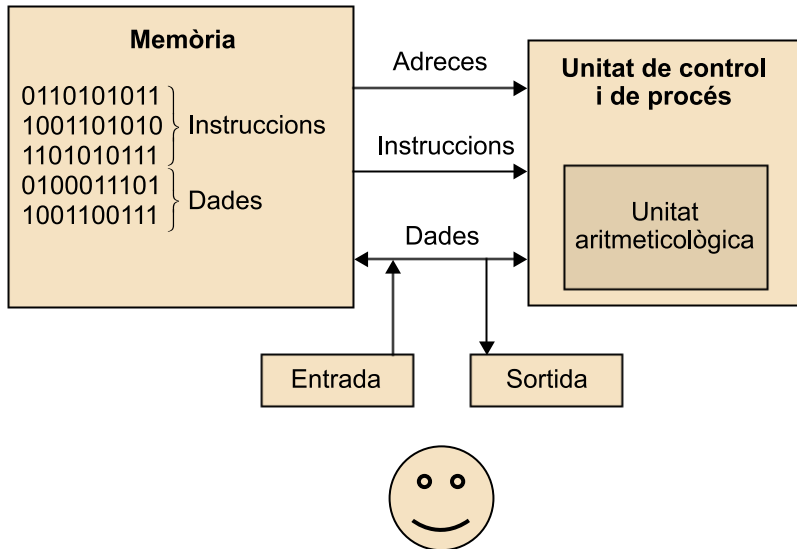
Vegeu també

Es definirà amb més detall el programari en el subapartat 3.2 d'aquest mòdul.

Per mitjà de les instruccions, els programadors defineixen el que han de fer els ordinadors durant l'execució dels programes.

La figura següent mostra un esquema d'aquesta arquitectura.

Arquitectura Von Neumann



Fins i tot els ordinadors actuals es basen en aquesta proposta, malgrat haver evolucionat i haver-s'hi afegit nous elements.

3.1.1. El microprocessador

La unitat de control i de processament⁷ és l'element central dels **microprocessadors**. El microprocessador és l'encarregat de la preparació del maquinari per a l'execució de tasques, com també de controlar-ne l'execució i ubicar els resultats en el lloc especificat en les instruccions. Un dels factors a valorar sobre el rendiment d'un ordinador s'obté mesurant les prestacions del microprocessador.

⁽⁷⁾En anglès, *central processing unit* (CPU).

Entre els factors que permeten avaluar el **rendiment d'un microprocessador** destaquem la velocitat, el nombre de bits amb què és capaç de treballar o el nombre d'operacions sobre nombres decimals capaç de fer en un segon.

A continuació en fem una descripció més detallada. La **velocitat del microprocessador** depèn de la freqüència de funcionament del rellotge que incorpora. Aquesta velocitat es mesura en hertz i durant decennis no es va arribar a superar el megahertz (o MHz, és a dir, el milió d'hertz). En la darrera dècada del segle XX es va superar la barrera del gigahertz (o GHz) i les freqüències actuals presenten dificultats tecnològiques per a ser superades.

El **nombre de bits de treball** també és un factor determinant del rendiment.

Exemple sobre el nombre de bits de treball

Un processador de 8 bits pot fer operacions amb nombres de 8 bits (per exemple, nombres enters entre -127 i 128). Si es vol fer una operació amb un nombre més gran, cal dividir-lo en dos nombres de 8 bits i, en conseqüència, el microprocessador esmerçarà el doble de temps que a fer l'operació amb un únic nombre de 8 bits. Els processadors actuals treballen amb grups de fins a 64 bits.

Finalment, el **nombre d'operacions amb nombres decimals per segon**⁸ també és un factor important. Avui dia els microprocessadors arriben a ultrapassar els gigaflops (milers de milions d'operacions per segon).

⁽⁸⁾En anglès, *floating point operations per second* (flops).

La taula següent mostra alguns dels microprocessadors de la casa Intel usats en els ordinadors personals, amb l'any d'introducció al mercat. També s'hi mostra el nombre de transistors que el formen, per a donar idea de com s'ha incrementat la complexitat en el disseny dels microprocessadors.

Any	Nom	Bits de treball	Megahertz ⁹	Transistors
1978	8086	16	8	29.000
1982	80286	16	12,5	134.000
1988	80386	32	25	275.000
1989	80486	32	50	1.200.000
1993	Pentium	32	66	3.100.000
1997	Pentium II	32	300	7.500.000
1999	Pentium III	32	800	28.000.000
2003	Pentium IV	32	2.400	42.000.000

⁽⁹⁾De cada processador hi ha hagut versions amb diferents velocitats. Nosaltres n'agafem un valor mitjà.

A partir del 2003, en tenir dificultats tecnològiques per a augmentar la velocitat de rellotge, es va pensar de dissenyar microprocessadors que inclouen, en realitat, més d'un microprocessador.

3.1.2. La memòria

Si el microprocessador és un element important a considerar en els ordinadors, també ho és la memòria. Tot seguit es detalla els tipus de memòria que es pot trobar en un ordinador.

La **memòria ROM**¹⁰ és una memòria que no s'esborra. Aquesta mena de memòries serveixen, per exemple, per a emmagatzemar procediments i dades fonamentals per al funcionament de determinats components de l'ordinador o aparells digitals. Noteu que el contingut d'aquest tipus de memòria no es pot modificar, únicament es pot consultar.

⁽¹⁰⁾En anglès, *read-only memory*, és a dir memòria tan sols de lectura.

La **memòria RAM**¹¹ és la més important. Es tracta del conjunt de xips que serveixen perquè l'ordinador hi emmagatzemi els processos que s'hi executen (és a dir, les instruccions que conformen els programes que s'estan executant) i les dades que fan servir aquests programes. Com més memòria RAM hi hagi i més ràpid sigui l'accés a aquesta, millor podrà ser el rendiment de l'ordinador.

⁽¹¹⁾De l'anglès, *random access memory*, memòria d'accés aleatori.

Aquesta memòria RAM es buida quan s'apaga l'ordinador. Per tant, és necessari disposar de memòries capaces de retenir la informació que contenen. El **disc dur** és el dispositiu d'emmagatzematge de memòria en què s'emmagatzema el sistema operatiu, el programari que es fa servir i els documents amb què treballarem. Quan s'inicia l'ordinador, el sistema operatiu es carrega des del disc dur a la memòria RAM. Tot i això, alguns equips fan servir memòries tipus ROM per a desar-hi aquest programari (per exemple, els telèfons mòbils o els ordinadors lleugers).

Memòria principal i memòria secundària

Normalment, la memòria principal de l'ordinador està formada per mòduls de memòria RAM. Els discos durs formen la denominada *memòria secundària*.

La velocitat del disc dur també és un factor important per a avaluar el rendiment d'un sistema informàtic: com més ràpid es pugui anar a llegir o escriure la informació, millor.

En general, la memòria RAM té molta menys capacitat que el mateix disc dur. Per exemple, el disc dur ofereix capacitats un centenar de vegades superior a la memòria RAM. Es podria pensar per què no s'utilitzen discos durs en comptes de memòries RAM si tenen més capacitat. La raó és el temps d'accés. La tecnologia utilitzada en els discos durs fa que la velocitat d'accés sigui molt inferior a la velocitat de la memòria RAM.

Una solució utilitzada per a augmentar la memòria principal de l'ordinador és fer servir part de l'espai de disc dur com a **memòria virtual** i desar la informació que s'està fent servir però no cap en la RAM.

Exemple d'ús de la memòria virtual

Si tenim un gestor de correu obert però estem retocant una fotografia, el més probable és que el programa de retoc i la imatge siguin en la RAM, mentre que el programa de correu està "descansant" en el disc dur. Un cop activem el gestor de correu, aquest passarà automàticament a la memòria RAM (i, en conseqüència, potser la imatge que estem retocant es desarà en la memòria virtual del disc dur).

Finalment, la **memòria cau**¹² permet aproximar al microprocessador la informació que s'utilitza més sovint. Es troba ubicada dins la mateixa CPU, al costat del processador, de manera que accedir a informació de la memòria cau porta menys temps que accedir a la informació que es troba en la RAM (i, evidentment, menys temps que per a accedir a la memòria virtual del disc dur!).

⁽¹²⁾També coneguda com a *memòria cache*.

Exemple d'ús de la memòria cau

Seguint amb l'exemple anterior, la porció de la imatge que estem retocant podria ser en la memòria cau, mentre que la resta de la imatge es troba en la RAM.

També hi ha altres dispositius d'emmagatzematge com ara els DVD o les memòries flaix.

3.1.3. La placa base

El microprocessador i la memòria es troben acoblats a la placa base¹³. Aquesta placa conté una ubicació per a cada element bàsic del sistema informàtic: bancs per a ubicar xips de memòria RAM, connexions per al disc dur, ports per als dispositius d'entrada i sortida, etc.

3.1.4. Dispositius d'entrada i sortida d'informació

Esmentem ara els dispositius més importants d'entrada i sortida d'informació. Gràcies a ells podem introduir informació en el sistema informàtic (cas dels dispositius d'entrada) o bé veure la informació (cas dels de sortida).

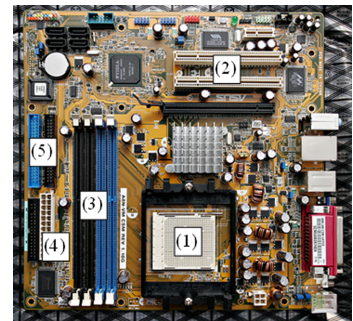
El **teclat** és un dispositiu d'entrada que permet introduir text, per exemple, en un document, al navegador, etc. Disposa de tecles distribuïdes en àrees (àrea numèrica, alfanumèrica, tecles de navegació, etc.), tot i que alguns teclats més moderns disposen d'altres botons i tecles per a fer tasques diverses (pujar o baixar el volum del so, apagar l'ordinador, etc.). Encara que el més habitual és disposar d'un teclat que es connecta a la unitat central amb un cable, podem optar per un teclat sense fil. La idea és que el que va connectat a l'ordinador no és el mateix teclat sinó un receptor.

El **ratolí** és un dispositiu d'entrada que es va inventar al final dels seixanta i va ser oblidat en un racó, fins que la casa Apple el va popularitzar en els seus ordinadors. El ratolí permet moure un punter per la pantalla tot movent-se per una superfície plana. Podem activar accions fent clic amb els seus botons o bé movent una rodeta de desplaçament. En podem trobar de dos tipus, com en el cas del teclat: ratolins amb cable i ratolins sense cable. D'altra banda, segons la tecnologia que fan servir per a detectar el moviment, tenim els ratolins mecànics (de bola) i els ratolins òptics. L'avantatge d'aquests sobre els primers és que no s'embruta el mecanisme intern i sempre gaudirem d'un moviment precís, mentre que amb els ratolins de bola sovint haurem de treure la bola i netejar els eixos interns.

Un **escàner** és un dispositiu d'entrada que permet digitalitzar (passar a l'ordinador) qualsevol document en paper, com ara una fotografia, un dibuix, una diapositiva, etc.

El darrer dispositiu d'entrada que descrivim és la **webcam** o **càmera web**, que va connectada a l'ordinador, la qual cosa fa que el camp que pot abastar per a la presa d'imatges sigui força limitat. S'utilitza bàsicament en comunicacions de videoconferència.

⁽¹³⁾També coneguda com a placa mare, segons el nom en anglès *motherboard*.



Esquema d'una placa base habitual. 1. Espai per al microprocessador. 2. Ranures per a plaques d'extensió per a connectar-hi dispositius. 3. Espai per a xips de memòria RAM. 4. Connector per a la font d'alimentació elèctrica. 5. Connectors per al disc dur

La **pantalla** o monitor és el dispositiu de sortida més bàsic i important de l'ordinador. La seva dimensió es mesura, com tots els televisors, amb el nombre de polzades en diagonal. Així, doncs, trobem monitors de 15", 17", etc. La pantalla és una graella que conté milers de píxels per a representar imatges i textos. El nombre de píxels que poden representar és un factor a tenir en compte: com més files i columnes tingui, més elements alhora s'hi podran visualitzar i més còmode serà treballar amb determinades tasques.

La **impressora** és un dispositiu de sortida que permet treure en paper tot allò que fem amb l'ordinador. Hi ha diferents tecnologies d'impressió entre les quals destaquen la d'injecció de tinta (la més habitual en el mercat domèstic), la làser (per a grans volums d'impressió) i la matricial (sorollosa, però utilitzada en casos específics).

3.1.5. Dispositius d'emmagatzematge d'informació

Finalment, descriurem alguns dels dispositius dedicats a l'emmagatzematge de la informació.

El **disc dur**, generalment ubicat a l'interior de l'ordinador, permet l'accés ràpid a la informació que s'hi emmagatzema. Està format per diversos discos tancats en una carcassa que els protegeix.

Els discos durs externs se solen utilitzar com a extensió del disc dur, per a desar els documents que no ens hi caben o bé com a suport per a fer còpies de seguretat.

Les **unitats òptiques**, com ara els lectors/gravadors de CD-ROM i DVD, permeten la lectura i enregistrament d'aquesta mena de discos. Els CD-ROM¹⁴ tenen capacitats d'enregistrament d'uns 600 MB, mentre que amb els DVD arribem a uns quants GB d'espai d'emmagatzematge. Inicialment, a part de com a suport d'àudio i vídeo digital, aquests dispositius s'han fet servir com a suport per a les còpies de seguretat dels documents i programes. Tanmateix, la proliferació i abaratiment dels discos durs externs, han fet que els CD i els DVD es releguin a la multimèdia i a la distribució de programari.

Els disquets

Els **disquets** són uns dispositius àmpliament utilitzats als anys vuitanta i noranta del segle XX. Malgrat que alguns ordinadors encara inclouen de sèrie una disquetera, les funcionalitats dels disquets han quedat relegades amb les memòries flaix i fins i tot l'enviament d'informació per Internet.

Finalment, les **memòries flaix** són un dispositiu utilitzat per a transportar informació d'un equip a un altre. Permeten l'emmagatzematge d'una quantitat considerable d'informació (actualment de l'ordre dels gigabytes) però no se

Vegeu també

Entrarem més en detall sobre com es representen les imatges en format digital en el mòdul "El World Wide Web".

Els píxels

Una imatge digital es dibuixa en la pantalla a base de píxels, de l'anglès *picture element*, en català *element d'imatge*.



Interior d'un disc dur, on es mostra el braç lector/escriptor.

⁽¹⁴⁾De l'anglès *compact disc read only memory*, memòria tan sols de lectura en un disc compacte.

solen fer servir com a suport de còpia de seguretat per la seva baixa fiabilitat a llarg termini i perquè la quantitat d'informació que es pot arribar a emmagatzemar en el disc dur és considerablement més elevada.

3.2. El programari

El programari és un **conjunt d'instruccions** que indiquen al processador el que ha de fer. Aquestes instruccions s'utilitzen per a manejar les dades, codificades en binari com hem vist, de manera convenient per a resoldre problemes.

És interessant fer notar que el programari també es troba codificat en binari i emmagatzemat en memòria juntament amb les dades.

En realitat, no pot ser de cap altra manera, ja que, com hem explicat, els ordinadors només entenen conceptualment d'uns i zeros. Per tant, com en codi ASCII es codifica un caràcter en una sèrie d'uns i zeros per a emmagatzemar-lo en memòria, es fa el mateix amb una instrucció que s'ha d'executar (per exemple, la instrucció *sumar*).

Si tot és binari i tot està emmagatzemat en memòria, com sap un ordinador què és una instrucció i què és una dada? Es tracta d'una bona qüestió, perquè en realitat no ho sap. En un dispositiu d'emmagatzematge, per exemple, un disc dur, només es poden identificar uns i zeros i no es pot distingir quins corresponen a instruccions i quins a dades. La clau està en l'ordre en què es passen els bits al processador.

El processador es construeix de manera que cada instrucció es representa amb un conjunt de bits. A més, cada instrucció accepta un conjunt determinat d'arguments després de la instrucció que identifiquen les dades.

Imaginem una situació en què vulguem sumar dos nombres amb un processador que té la instrucció següent:

SUMAR DADA1, DADA2, RESULTAT

Seguint l'ordre predeterminat el processador espera primer el codi binari de la instrucció i, a continuació, l'adreça de memòria on s'emmagatzema la primera xifra seguida de la segona. Finalment, rep l'adreça en què es desarà el resultat. D'aquesta manera, el processador, depenent de l'ordre en què arriben els bits, de vegades els interpretarà d'una manera (instruccions) i altres vegades d'una altra (dades).

Si, per qualsevol raó, les coses no s'envien al processador en l'ordre correcte, donarà un comportament erroni o un error. Fixeu-vos que, normalment, cada processador té el seu conjunt d'instruccions codificades d'una manera concreta. Per això, el programari creat per a una arquitectura específica (de la qual el processador és una part important) no funciona en cap altra arquitectura. Un programari per a l'arquitectura i386 de la família Intel (la dels IBM PC tradi-

cionals) no funcionarà en una arquitectura Motorola, utilitzada en els antics Macintosh de Apple. La solució consisteix a adaptar el programari perquè segueixi les pautes de cada arquitectura.

La característica més important que diferencia el programari del maquinari és la **flexibilitat**. El maquinari, i no solament el d'ordinador, és específic per a cada tasca. Per exemple, la pascalina que ja coneixem permetia fer (alguns) càlculs però res més, igual com una màquina espremedora de fruita només espremer fruita. Es pot intentar fer altres coses amb la màquina espremedora com, per exemple, fregir un ou, però generalment el resultat no serà satisfactori.

El programari és molt versàtil perquè la seva tasca és manejar (prendre, transformar, desar) dades binàries. En altres paraules, amb un mateix ordinador i diferent programari podem resoldre un gran conjunt de problemes. I és que una vegada que un problema s'ha aconseguit transformar en dades binàries, podem resoldre el problema aplicant una sèrie d'instruccions sobre aquest conjunt de dades.

Els edificis intel·ligents

Els edificis intel·ligents disposen d'ordinadors que prenen dades de l'edifici (persones que entren i surten de l'edifici, temperatura, humitat, lluminositat, etc.) mitjançant sensors i altres dispositius. Aquestes dades són convertides en binari, i posteriorment són emmagatzemades i tractades. Així, es pot conèixer fàcilment el nombre de persones que han entrat i sortit, per exemple, en les hores punta o ajustar la temperatura o la lluminositat que hi ha d'haver en cada espai per a estalviar energia. I això solament és el principi, ja que se'ns podrien ocórrer moltes tasques més, encara que els dissenyadors originals del programari ni les haguessin pogut imaginar. Per exemple, es podria afegir funcionalitat perquè l'ordinador central, dissenyat en la dècada dels vuitanta quan encara no existia la tecnologia de telefonia mòbil, permetés reservar sales de reunions enviant un missatge SMS.

Però no tot són avantatges. La flexibilitat també fa que el programari sigui cada vegada més complex, ja que cada vegada es volen programes que facin tasques més complicades.

La complexitat del programari

En el cas dels edificis intel·ligents, si al principi només es volia conèixer el nombre de persones que entraven i sortien d'un edifici, més endavant es va desenvolupar un sistema per a ajustar de manera intel·ligent la temperatura segons l'ocupació de l'edifici, per acabar volent un sistema que accepti reserves de sales mitjançant missatges de telefonia mòbil.

Tot això fa que el programari es faci cada vegada més complex: el nombre de factors a tenir en compte creix, com també les situacions i les regles... Amb tantes coses a tenir en compte, crear programari s'ha convertit en una de les activitats que requereixen més enginy humà. Si no fos així, probablement el crearien màquines de manera automàtica.

3.2.1. Algoritmes

Com es crea el programari? Ja hem indicat que el programari està pensat principalment per a resoldre problemes. La manera en què podem resoldre aquests problemes és mitjançant l'ús d'**algoritmes**.

Un **algoritme** és un conjunt finit de passos que serveixen per a resoldre un problema o dur a terme una tasca.

Un algoritme és com una mena de recepta, com les de la cuina però en aquest cas informàtiques. Així, imaginem que es volen cuinar unes exquisides torradetes de Santa Teresa per a degustar-les a l'hora del berenar. Anant a un llibre de receptes de cuina qualsevol, ens trobaríem amb una recepta com la que segueix:

- 1) Barrejar 2 ous batuts lleument amb una culleradeta de vainilla, mitja de canyella i una tassa de llet.
- 2) Mullar 6 llesques de pa en la barreja.
- 3) Fregir en una mica de mantega fins que es daurin.
- 4) Servir el pa amb una mica de xarop de caramel.

Els algoritmes comparteixen amb les receptes de cuina que estan escrits en **llenguatge natural**. Això és un avantatge, perquè una vegada que s'ha après a llegir i es tenen uns coneixements bàsics dels ingredients que se solen trobar en la cuina tradicional, es pot entendre una recepta sense gaires problemes. Però, al seu torn, l'ús del llenguatge natural resulta problemàtic. És un problema perquè el llenguatge natural és **ambigu**. Així, quan es demana batre lleument dos ous, a què es refereix amb *lleument*? quina velocitat significa *lleument*? També hi ha els detalls relacionats amb les quantitats que tots coneixem de les nostres experiències culinàries sobre *l'afegiu una cullerada de sal...*, es tracta d'una cullerada sopera? o d'una de cafè?

En general, els humans tenim certa capacitat (encara que limitada) per a dirimir aquests problemes associats a l'ambigüitat del llenguatge. I aquesta capacitat es veu augmentada en el cas de la cuina, perquè se sol tastar el producte mentre es cuina o just en acabar. Però les màquines no tenen aquesta capacitat. Les màquines no entenen bé *lleument* o *una mica*, llevat que se sigui més concret. I és aquí on, en el programari, entren els programes. Els programes s'han de veure com una **descripció formal d'un algoritme**. Per a aquesta fi-



En la imatge es poden veure unes torradetes de Santa Teresa, a punt per menjar, també conegudes com a *torrades franceses*. Font: Viquipèdia. Llicència: Creative Commons Atribució 2.0.

nalitat, en comptes de valer-nos del llenguatge natural, fem servir una sèrie de llenguatges pseudomatemàtics anomenats **llenguatges de programació**, que explicarem més endavant.

3.2.2. Disseny del programari

Quan es crea un programari el primer pas sol correspondre a la descripció de l'algoritme en llenguatge natural. Generalment les persones que volen fer servir el programari no saben de llenguatges de programació. D'altra banda, els enginyers de programari saben programar, però moltes vegades no entenen el problema a què s'enfronten.

Per aclarir aquest punt, imaginem un xef que vol una màquina de fer torradetes de Santa Teresa. És evident que el xef sabrà fer les torradetes, però és improbable que tingui coneixements avançats en programació de programari. Per tenir un programa que faci les torradetes cridaria algú amb coneixements de programació. D'altra banda, és molt probable que aquesta persona experta en programació no hagi fet una torradeta de Santa Teresa en la seva vida.

El mitjà que el xef i l'enginyer de programari tindran per a entendre's serà un algoritme com el que hem vist anteriorment, especificat en llenguatge natural. El xef i l'enginyer de programari quedaran per indicar les especificacions i requisits del programa. El nom específic que rep l'enginyer de programari en aquest cas sol ser el d'**analista de programari**, ja que la seva tasca consisteix a realitzar una anàlisi dels requisits i de com afrontar-los per a codificar-los convenientment en el programa que vol el xef.

Generalment, l'etapa següent consisteix a prendre els requisits de l'anàlisi del programari (plasmats en un document) i crear un document de disseny del programa. Bàsicament això consisteix a prendre els algorismes de l'etapa d'anàlisi i organitzar-los de manera que el programari sigui senzill de realitzar sense perdre, al seu torn, flexibilitat perquè en el futur es puguin fer millores. I és que com més ben dissenyat estigui, menys errors tindrà el programa i més fàcil serà modificar-lo en el futur, incloent-hi noves funcionalitats. Com a conseqüència d'aquest treball es crearà un altre document en què els algorismes estaran organitzats convenientment.

Serà tasca d'un programador transformar aquests algorismes en un programa de programari utilitzant un llenguatge de programació. Si la tasca d'elaborar un algoritme no és senzilla, tampoc no ho és la transformació en un llenguatge de programació. S'ha d'eliminar l'ambigüitat, ja que els programes han de ser precisos. En cas que no es faci correctament, el programari contindrà errors. Imaginem les tràgiques conseqüències d'haver indicat malament la mida de les cullerades de sucre per a les nostres torrades. Seria un desastre!

Errors del programari

D'errors informàtics se'n poden trobar a tot arreu, però els aeroespacials són dels més coneguts. Per exemple, el 1972 la sonda *Mariner 1* es va desviar del curs establert i va haver de ser destruïda perquè una fórmula escrita a mà es va traslladar erròniament al llenguatge de programació. Un altre cas, aquest el 1999, va tenir com a protagonista la *Mars Climate*, que es va estavellar contra el terra de Mart a causa d'un error en la conversió de milles a quilòmetres.

Com veiem, és comú que el programari tingui errors i per a aplicacions una mica complexes –gairebé totes les que utilitzem a dia d'avui– és impossible assegurar que un programari no en tingui cap. Per això la creació del programa també consisteix a **provar-lo** –veure si té errors– i **validar-lo** –veure que al final del procés s'obté el que volíem al principi i no una altra cosa. Abans que un programari es publiqui definitivament, sol passar per un període en què es comprova que fa el que ha de fer. En aquest sentit, la programació s'assembla molt a la tasca habitual de la cuina, en què tota recepta es prova després..., i les receptes més innovadores se solen provar primer en cercles íntims abans de presentar-les en grans ocasions.

I, tot i així, és comú que amb el pas del temps apareguin errors que no s'havien detectat o que es vulgui que el programari tingui noves funcionalitats que no s'havien inclòs. És per això que l'última etapa de la vida d'un programa és la de **manteniment**. Durant tota aquesta fase, els enginyers de programari gestionen el programa eliminant errors i afegint noves funcionalitats. Es tracta de l'etapa més llarga, ja que, mentre que les anteriors se solen dur a terme en qüestió de mesos o uns quants anys, un programari pot estar en fase de manteniment durant molts anys, en alguns casos fins i tot dècades.

Exemple de programari longeu

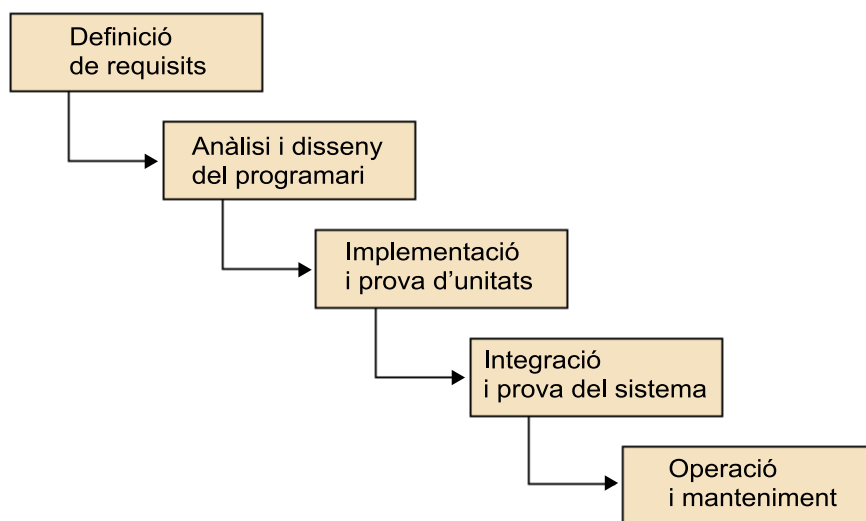
El cas més conegut de programari longeu es pot trobar en el sector de la banca, en què hi ha una gran quantitat de programari de la dècada dels setanta. Un exemple de funcionalitat que es va haver d'afegir a aquest programari és la conversió de les antigues pessetes a l'euro, la nova moneda única europea. Amb aquest fi, els programadors encarregats del manteniment van haver de modificar els programes i provar una altra vegada les noves versions per a constatar que no tinguessin errors.

El model de desenvolupament del programari rep el nom de **model de desenvolupament en cascada**, perquè les fases es realitzen una darrere l'altra sense solapaments. Aquest és el model que s'utilitza de manera tradicional en la gran indústria del programari.

- La definició de requeriments és el que en el nostre cas ha fet l'analista de programari amb el xef en un algoritme.
- Prenent els requeriments com a punt de partida, un dissenyador de programari crea un disseny sobre paper, com un arquitecte dibuixa uns plànols. Generalment és una bona pràctica dividir el problema general en subproblemes, més manejables i abordables. Aquests subproblemes reben el nom d'*unitats*.

- En la tercera etapa es prendrà aquest disseny i s'implementarà mitjançant l'ús d'un llenguatge de programació. La divisió en unitats permet provar cada unitat de manera aïllada i comprovar que cada una fa el que volem.
- En la quarta etapa es prenen totes les unitats i s'integren en un únic programa. És ara quan cal provar el sistema complet per a comprovar que és correcte.
- Després de provar-lo, es lliura a l'usuari (entra en operació) i es corregeixen els errors que aquest pugui notificar.

Model en cascada pur o seqüencial per al cicle de vida del programari



Font: Viquipèdia. Llicència: Creative Commons Atribució Compartir Igual

El model en cascada és senzill de dur a terme i d'entendre, però també té els seus aspectes negatius. El primer és que des de la primera fase fins que s'obté el programa pot passar molt temps. I només aleshores podem lliurar el programa definitiu al client. Si es dona la circumstància que en qualsevol baula de la cadena la informació no ha fluït de manera convenient o s'ha desviat del propòsit inicial, es podria tenir un programari que funciona però que no fa el que el client volia. Per a evitar aquesta situació, hi ha models de desenvolupament més flexibles que obtenen resultats intermedis abans, de manera que el client pot veure l'evolució i dictaminar si li convé o no.

3.2.3. Evolució dels llenguatges de programació

Al principi, el treball d'implementació dels programadors era molt ardu, perquè les màquines només entenen uns i zeros. Així, el treball de transformació consistia a prendre l'algorisme desitjat i transformar-lo en **codi màquina**, que és el nom que rep la seqüència d'uns i zeros dels programes.

Treballar d'aquesta manera amb uns i zeros directament, com us podeu imaginar, és una tasca tediosa. És difícil veure què es fa i és difícil identificar si hi ha algun error. En programes escrits directament en codi màquina, la tasca

Exemple de codi màquina

El codi màquina de dos nombres que se sumen podria ser el següent: 1100 0100 0010 1001 0100 0100.

de trobar un error (coneguda com a **depuració**) es converteix en una tasca titànica. El fet de treballar amb uns i zeros es coneix com a *treballar al nivell de màquina*, ja que és com si estiguéssim conversant a la seva mateixa altura.

Per no haver de treballar al nivell de màquina, es va inventar un **nivell d'abstracció** superior. Aquest nivell d'abstracció fa que la tasca de programació sigui més senzilla: en comptes d'uns i zeros es treballa amb representacions més abstractes de les instruccions. Així, va sorgir el **llenguatge ensamblador**.

Exemple de llenguatge ensamblador

La instrucció per al codi màquina de dos nombres que se sumen (1100 0100 0010 1001 0100 0100) podria ser la següent: *add \$1 \$2*.

Haureu vist que es tracta d'un pseudollenguatge mig matemàtic mig anglès, que sembla indicar que es tracta d'una operació de suma (*add*) de dos nombres. En comptes d'indicar els dos nombres directament, s'indiquen les posicions en què estan emmagatzemats aquests dos nombres en memòria (en 1 i 2). Fixeu-vos que no s'indica l'adreça en què s'ha de desar el resultat. Alguns llenguatges ensamblador desen el resultat de l'operació en el primer operand. També hi haurà altres instruccions com, per exemple, una operació per a introduir un nombre en una posició de memòria *store \$1, valor*. Per exemple, amb aquesta instrucció es podria introduir el valor -4 en l'adreça \$1.

Tot i així, el llenguatge ensamblador té una filosofia similar al codi màquina; simplement és més llegible. Els informàtics dirien que es tracta d'un llenguatge molt proper a la màquina, ja que la transformació en binari és immediata. Això té els seus punts positius, però també els seus aspectes negatius. L'avantatge és que és molt eficient, ja que fa servir directament el conjunt d'instruccions que utilitza el processador. En treballar en un nivell d'abstracció tan proper a la màquina, es poden optimitzar les operacions, obviar les redundàncies i aprofitar el coneixement dels detalls del processador. Tanmateix, encara que és millor que treballar al nivell d'uns i zeros, no deixa de ser una manera molt tediosa d'indicar seqüències d'instruccions; continua essent molt fàcil cometre errors. A més, diferents processadors –per exemple, de diferents fabricants– poden tenir un conjunt d'instruccions diferents, per la qual cosa és possible que el programa ensamblador només serveixi per a un processador específic.

Programa ensamblador

Per a aclarir una mica més els aspectes relatius al llenguatge ensamblador i sense entrar en detalls, en la figura següent es mostra una part d'un programa ensamblador amb el codi màquina corresponent. El codi màquina es representa en hexadecimal per a simplificar l'exemple. Podeu veure que les primeres cinc línies corresponen a instruccions en llenguatge ensamblador, però de la setena a l'onzena els uns i zeros no corresponen a instruccions, sinó a dades. Es tracta d'un text codificat en ASCII.

Llenguatge màquina i llenguatge ensamblador per a l'arquitectura Intel 8088

Codi màquina		Codi ensamblador		
-u 100 1a				
OCFD:0100	BA0B01	MOV	DX,010B	
OCFD:0103	B409	MOV	AH,09	
OCFD:0105	CD21	INT	21	
OCFD:0107	B400	MOV	AH,00	
OCFD:0109	CD21	INT	21	
-d 10b 13f				
OCFD:0100				
OCFD:0110	20 65 73 74 65 20 65 73-20 75 6E 20 70 72 6F 67			48 6F 6C 61 2C
OCFD:0120	72 61 60 61 20 68 65 63-68 6F 20 65 6E 20 61 73			
OCFD:0130	73 65 6D 62 6C 65 72 20-70 61 72 61 20 6C 61 20			
OCFD:0140	57 69 68 69 70 65 64 69-61 24			
				Hola,
				este es un prog
				rama hecho en as
				sembler para la
				wikipedia\$
Adreces de memòria		Codi màquina		

Font: Viquipèdia. Llicència: GFDL i Creative Commons Atribució CompartirIgual 3.0.

A causa de la laboriositat del codi ensamblador, amb el pas del temps van començar a sorgir **llenguatges de programació** d'un nivell d'abstracció més alt, és a dir, més propers a l'humà (al programador). Fixeu-vos que com més s'allunyen de la màquina (dels uns i zeros) i més s'apropen al llenguatge natural, més alt és el nivell d'abstracció. D'una banda, es guanya en facilitat d'implementar els algorismes però, d'una altra, es perd l'eficiència que es guanyava en ajustar els algorismes a la màquina. Llevat de qüestions molt concretes que requereixin molta eficiència com, per exemple, tasques del sistema operatiu, avui en dia gairebé totes les aplicacions es realitzen en llenguatges d'alt nivell.

Exemples de llenguatge de programació

Hi ha molts llenguatges de programació. Els més populars d'avui dia són Java, C, C++, COBOL, Pascal..., però hi ha milers de llenguatges de programació més.

Els llenguatges de programació permeten al programador centrar-se en la implementació de l'algorisme i oblidar-se de les particularitats de la màquina.

Mitjançant aquests llenguatges es creen programes menys difícils de llegir i d'escriure i, al mateix temps, menys propensos a errors, la qual cosa permet tenir aplicacions més potents –en el sentit que poden realitzar tasques més complexes. Tanmateix, són programes que no estan optimitzats com ho podrien estar si s'haguessin programat en llenguatge ensamblador. Però, en la majoria dels casos, l'optimització no és una prioritat per als programadors. Això és perquè, en els últims trenta anys, la potència dels ordinadors (la velocitat de processament, la capacitat d'emmagatzemament, etc.) s'ha duplicat aproximadament cada divuit mesos. Això vol dir que no té gaire sentit optimitzar un programa utilitzant el tediós codi ensamblador tenint en compte la potència de les màquines actuals. Si aquesta optimització fos necessària, seria més rendible comprar una màquina més potent. En altres paraules: el cost de comprar una màquina més potent és molt menor que el de tenir programadors optimitzant els programes.

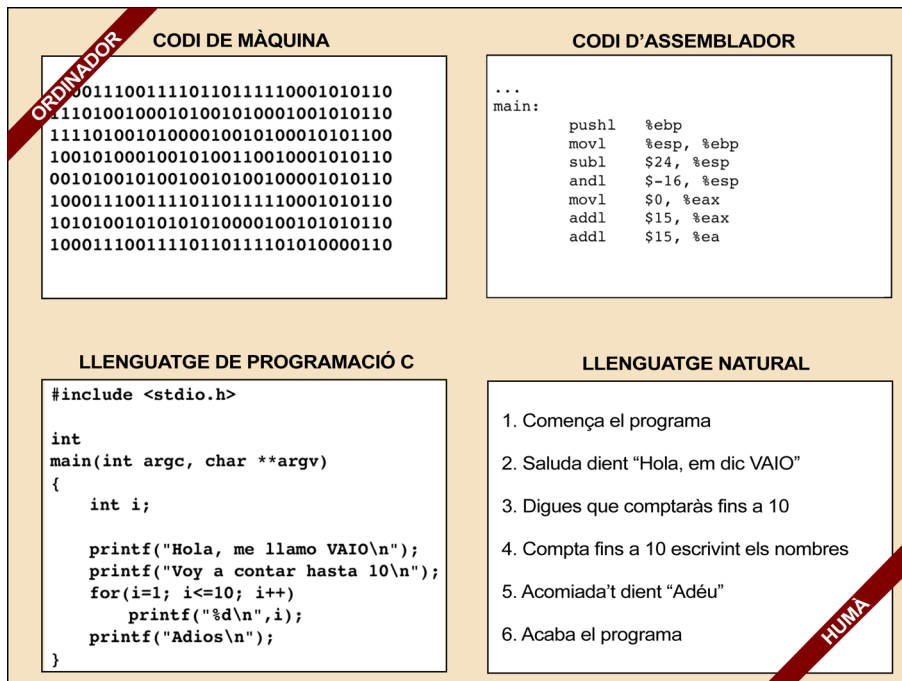
Per a mostrar la conveniència dels llenguatges de programació, a continuació es mostra un programa exemple, obtingut de la Viquipèdia, implementat en el llenguatge de programació Pascal. Aquest programa suma dos nombres enters proporcionats per l'usuari. A més d'instruccions i dades, el programa inclou comentaris (es poden identificar fàcilment perquè, en Pascal, els comentaris

es troben entre claus). Aquests comentaris no són ni instruccions ni dades: simplement ajuden el programador actual (i als futurs) a entendre què fa el programa.

```
program suma;  
  
var x,s,r:integer; {declaració de les variables}  
begin {començament del programa principal}  
    writeln('Introduïu 2 nombres enters'); {imprimeix el text}  
    readln(x,s); {llegeix 2 nombres i els col·loca en les variables x i s}  
    r:= x + s; {suma els 2 nombres i col·loca el resultat en r}  
    writeln('La suma és ',r); {imprimeix el resultat}  
    readln;  
end. {acaba el programa principal}
```

En un llenguatge de programació d'alt nivell, el text que el programador genera es coneix com a **codi font**. Per a poder estudiar un programa és indispensable disposar del codi font. Això és perquè, amb la complexitat dels programes actuals, el codi binari té una extensió tal que faria falta molt temps i esforç per a desxifrar-lo. Tanmateix, si s'ha utilitzat un llenguatge d'alt nivell, això és almenys més senzill.

En la figura següent exposem l'evolució que han seguit els llenguatges de programació en crear-se diversos nivells d'abstracció. Partint del nostre algoritme fins a arribar al codi màquina que entenen els ordinadors, podem veure els nivells intermedis (codi font, en aquest cas en el llenguatge de programació C, i codi assemblador). El procés de transformar l'algoritme en codi font es coneix com a **implementació** i, com hem comentat, el realitza un humà. Tanmateix, la transformació de codi font en codi màquina és un procés que es pot automatitzar, fins i tot obviant el pas intermedi de transformar el codi font en codi assemblador. Així, aquesta tasca és responsabilitat d'un programa especial conegut com a **compilador** i rep el nom de **compilació**. Cada llenguatge de programació té un compilador específic que transforma les instruccions de la sintaxi del llenguatge de programació en el codi màquina final.



Nivells d'abstracció. (c) Enrique Soriano. Llicència: GFDL i Creative Commons Atribució Compartir Igual 3.0

3.3. Tipus de programari

Arribats a aquest punt, es tenen unes nocions bàsiques suficients de com es desenvolupa un programa nou. Suposem que es volgués tenir un joc propi de cercamines. Primer es començaria per aprendre un llenguatge de programació d'alt nivell, s'estudiaria el problema per a obtenir els algorismes del joc i es començaria amb la implementació.

Tanmateix, hi ha una sèrie de detalls que no s'han abordat i que són de gran importància, com la interacció amb l'usuari. Tornant a l'exemple del cercamines, a més dels algorismes del joc per si mateixos, s'haurien d'incloure algorismes per a capturar les decisions de l'usuari, en particular si ha polsat certes tecles, si ha mogut el ratolí, etc.

Això, en realitat, no hauria de ser problema, ja que no deixen de ser més algorismes que es poden implementar amb el mateix llenguatge de programació acabat d'aprendre. Imaginem que es vol que el nostre cercamines funcioni per a molts tipus de ratolí diferents, idealment per a tots. S'haurien d'implementar les instruccions per a tots i cada un dels ratolins existents! O, el que és pitjor, mantenir i modificar el joc per a tots els ratolins que hi haguessin.

Tot això no té gaire sentit, sobretot perquè la idea és crear un bon joc de cercamines i no embolicar-nos amb el maquinari dels usuaris. És per aquesta raó, i per d'altres que veurem a continuació, que el programari no es pot encarregar de tot, des de la funcionalitat que brinda a l'usuari fins a la interacció

amb els dispositius connectats a l'ordinador. També hi ha nivells d'abstracció a l'hora de crear funcionalitats i, per això, el programari es divideix en dues grans famílies: el programari de sistemes i el programari per a l'usuari final.

3.3.1. El programari de sistemes: el sistema operatiu

El **programari de sistemes** té com a objectiu interactuar amb el maquinari i proveir de serveis i funcionalitats altres aplicacions. D'aquesta manera, es tracta d'un programa que està més centrat en l'ordinador que en l'usuari. Encara que hi ha molt programari de sistemes, com els antivirus o els servidors, el programari de sistemes més conegut és el sistema operatiu.

Els **sistemes operatius** són programes que creen un nivell d'abstracció superior sobre el maquinari.

En altres paraules: el sistema operatiu s'encarrega de comunicar els programes per a usuaris i altre programari de sistema amb el maquinari de l'ordinador. Seguint amb l'exemple del joc de cercamines del subapartat anterior, el sistema operatiu s'encarregarà de la interacció amb tots els ratolins que existeixen o poden existir. Els programadors d'aplicacions per a usuari final només han de tenir en compte que s'interactua amb un ratolí genèric (i abstracte). Els programadors sabran la posició del punter i si s'han polsat els botons independentment del ratolí utilitzat.

A més, el sistema operatiu és l'encarregat de gestionar un gran nombre d'altres tasques i elements, com veurem a continuació. Per aquesta raó els sistemes operatius s'han convertit des de fa un parell de dècades en un component essencial de qualsevol ordinador. Entre les seves responsabilitats podem trobar:

- La **comunicació amb els perifèrics**. És el cas que hem vist fins ara. El sistema operatiu té com a objectiu que els programes d'usuari puguin treballar de manera independent de les especificitats del maquinari que es tingui. Per això, quan es compra un nou dispositiu, s'han d'instal·lar els controladors¹⁵.
- La **gestió de la memòria**. És tasca del sistema operatiu gestionar les diferents memòries que podem trobar en un ordinador, des de la memòria principal (RAM) fins a la secundària (el disc dur), passant per la memòria virtual i la memòria cau que hem vist anteriorment. La gestió de la memòria és una tasca molt important si recordem que la memòria principal és molt ràpida, però molt cara i volàtil, mentre que la memòria secundària és lenta, però barata i persistent. És justament per això, per la velocitat, que ens interessa que tant les aplicacions que s'executen com les dades es trobin en la memòria principal. Tanmateix, a causa del seu cost, generalment

El cas de les impressores

Els creadors dels processadors de textos com Microsoft Office o OpenOffice.org Writer no es preocupen de la impressora instal·lada i com s'envien els treballs. Aquestes aplicacions simplement indiquen al sistema operatiu el fitxer que s'ha d'imprimir, i serà tasca del sistema operatiu que això passi.

⁽¹⁵⁾En anglès, *drivers*.

Els controladors

Els **controladors** són els programes que permeten la comunicació del sistema operatiu amb els dispositius.

no es té memòria principal suficient per a emmagatzemar-ho tot i cal que el sistema operatiu també utilitzi la memòria secundària.

- La **concurrència de processos**. Encara que el processador només pot executar una instrucció cada vegada, es pot aconseguir concurrència virtual mitjançant el sistema operatiu. En altres paraules, el sistema operatiu pot aconseguir que l'usuari tingui la impressió que s'executen diversos programes alhora. Imaginem que s'està escrivint una carta amb el processador de textos, mentre es descarrega un document amb el navegador web i es calcula quant espai lliure queda en el disc dur mitjançant un explorador d'arxius. En la pantalla es veu la finestra del processador de textos, per la qual cosa aparentment es podria pensar que aquesta és la tasca que té entre mans el sistema operatiu, deixant les altres per a després. I, tanmateix, es té la impressió que totes passen al mateix temps. En realitat, aquesta impressió és falsa: el processador només pot executar una instrucció alhora, així que no pot realitzar múltiples tasques de manera simultània. Tanmateix, la velocitat a què el processador executa les instruccions és tan gran (ho fa a la velocitat de milers de milions d'instruccions per segon), que ens sembla que les realitza totes alhora. Quan les tasques deixen ocios el processador, aquest realitza altres tasques. En el nostre exemple, mentre no s'escriu res en el processador de textos, el processador pot calcular l'espai lliure en el disc dur, seguir descarregant un document amb el navegador web i probablement fer altres tasques de les quals ni tan sols s'és conscient, com veure quanta bateria ens queda al portàtil. Fixeu-vos que encara que actualment la majoria dels ordinadors tenen més d'un processador, es continua utilitzant la mateixa tècnica explicada. El sistema operatiu expulsa les tasques ocioses dels processadors per executar altres tasques pendents. En aquest segon cas hi ha concurrència real entre processos, ja que s'executen múltiples tasques al mateix temps en processadors diferents.
- El **monitoratge i la seguretat**. Els sistemes operatius moderns són multiusuari. Això introdueix un problema de seguretat. Hi ha documents personals en els quals s'ha de restringir l'accés a altres usuaris de l'ordinador. El sistema operatiu s'encarrega d'aquesta funció afegint metadades als fitxers com, per exemple, l'usuari propietari. D'altra banda, per a saber què ha passat en l'execució d'alguna tasca en especial, hi ha fitxers en què es registren els esdeveniments per al seu estudi posterior. Són el que es coneix com a *registres*¹⁶ i permeten saber què ha passat i les tasques executades en un ordinador. Això és útil, per exemple, per a detectar si una màquina ha estat compromesa (algú hi ha entrat sense tenir permís d'accés).
- La **gestió d'emmagatzemament**. Els ordinadors contenen programes i dades. Tots dos es desen (emmagatzemen) com a fitxers en el disc dur. És tasca del sistema operatiu fer això de manera ordenada (per exemple, en directoris o carpetes).

⁽¹⁶⁾En anglès, *logs*.

- **La gestió de comunicacions.** Avui en dia, els ordinadors poques vegades es fan servir de manera aïllada; generalment es connecten a una xarxa, com podria ser Internet. Per a això, utilitzen una sèrie de regles establertes, normalment conegudes públicament que reben el nom de *protocols*. El sistema operatiu és el que s'encarrega de la tramesa i recepció d'informació utilitzant protocols de xarxa, permetent que les aplicacions (en realitat, els que les programen) no s'hagin d'ocupar dels detalls de la xarxa.

A causa de totes aquestes capacitats que tenen els sistemes operatius moderns, es pot entendre ràpidament que són de gran profit per a crear noves aplicacions de manera més ràpida i senzilla.

Tot això té un preu, i és el de la compatibilitat. Hem comentat que els programadors fan la seva feina creant codi font perquè posteriorment un programa conegut com a *compilador* el transformi en codi màquina. Bé, doncs aquesta etapa de compilació depèn del sistema operatiu, ja que el codi màquina que "entén" un sistema operatiu no ha de ser "compès" necessàriament per un altre. Com s'aconsegueix això? Doncs, principalment, gràcies a dues qüestions:

- D'una banda, el llenguatge de programació utilitzat disposa de diversos compiladors diferents, un per a cada sistema operatiu suportat. Per això, a l'hora de compilar es tria el compilador per al nostre sistema operatiu, de manera que obté el codi màquina corresponent.
- D'altra banda, els programadors han anat prou amb compte per a, a més d'utilitzar un llenguatge multiplataforma (és a dir, que es pugui compilar en diferents sistemes operatius), no utilitzar seqüències de programació o mètodes que són exclusius d'un sistema operatiu. Fixeu-vos que hi ha algunes funcionalitats o mètodes que són específics d'un sistema operatiu. La inclusió d'aquests mètodes en el programa produiria errors en la compilació en altres sistemes operatius.

Exemples de dependència del sistema operatiu

El programa PADRE per a fer la declaració de l'impost sobre la renda a Espanya, el paquet ofimàtic lliure OpenOffice.org o del navegador Mozilla Firefox, tenen diferents versions depenent del sistema operatiu utilitzat.

Anteriorment havíem vist que el compilador depèn de l'arquitectura de l'ordinador. Així, la transformació de codi font en codi màquina seria diferent per a una màquina Intel o per a una Mac. Ara, a més, veiem que el sistema operatiu també és una qüestió a tenir en compte en aquesta transformació.

En definitiva, els compiladors han de ser específics per a una arquitectura i per a un sistema operatiu.

Exemple de compiladors

Perquè un programa creat en el llenguatge de programació Pascal es pugui executar en un ordinador de l'arquitectura Intel i amb el sistema operatiu GNU/Linux, es necessita el compilador per a Pascal per a l'arquitectura Intel i el sistema operatiu GNU/Linux. El codi màquina que generarà aquest compilador no funcionarà si només coincideixen l'arquitectura (per exemple, en una màquina Intel amb sistema operatiu Microsoft Windows) o el sistema operatiu (per exemple, en una màquina amb arquitectura PowerPC amb sistema operatiu GNU/Linux).

3.3.2. Programari per a usuari final

El **programari per a usuari final** té com a finalitat resoldre problemes de l'usuari, donant suport o millorant el treball que vol realitzar. La seva funcionalitat té més a veure amb la lògica de la tasca a realitzar per l'usuari que amb el maquinari del sistema o dels dispositius amb què l'usuari està interactuant, que com hem vist és gestionat pel sistema operatiu.

Exemple de programari per a usuari final

Prenem una analogia del món urbà per a comprendre la diferència entre el programari de sistemes i el d'ús final. A les ciutats tenim el servei de neteja i manteniment, que fa les tasques més properes a l'infraestructura, mentre que les botigues i els supermercats ofereixen als ciutadans serveis finals. Així, el programari per a l'usuari final és com les botigues o els supermercats, ja que ofereixen serveis i productes directament al ciutadà, mentre que el programari de sistemes s'assembla als serveis de neteja i manteniment, que s'encarreguen d'altres tasques amb les que el ciutadà no sol interactuar directament, però sense les quals seria difícil viure (almenys còmodament) en una ciutat.

Generalment els programes d'usuari final realitzen un objectiu específic, com pot ser la manipulació d'imatges, l'enregistrament de CD-ROM, la navegació pel web... La llista és llarga i hi ha hagut múltiples intents de crear una taxonomia de programes per a usuari final. A continuació s'ofereix una possible classificació d'aquest tipus de programari:

- **Programari d'infraestructura empresarial.** Es tracta de programari com ara bases de dades per a emmagatzemar grans volums de dades, programari de fluxos del procés de negoci que ajuden en la gestió del producte o servei que ofereix l'empresa, o programari de sistemes d'informació geogràfica que permet enllaçar dades amb cartografia, per exemple, per a conèixer la localització exacta de cada camió que compon la flota de l'empresa.
- **Programari d'accés a continguts.** Mitjançant aquest programari es pot accedir a informació textual o multimèdia. És el cas de navegadors web per a navegar per Internet o de programes de reproducció multimèdia. El programari d'entreteniment, com jocs o estalvis de pantalla, també se sol incloure en aquesta categoria.
- **Programari educatiu.** És el programari amb finalitat educativa. En aquesta categoria trobem programari de gestió d'aules, programari d'aprenentatge o d'entrenament, o programari de consulta (com les antigues enciclopèdies

Edutainment

L'*edutainment* és un programari d'entreteniment dissenyat per a educar.

que venien en CD-ROM, avui en dia més en desús gràcies a Internet i la Viquipèdia).

- **Programari de simulació.** En aquesta categoria s'enquadrarien els coneguts jocs de simulació (com els simuladors aeris o els de motor), però també hi ha una gran gamma de simuladors científics que s'utilitzen en investigació.
- **Programari de desenvolupament multimèdia.** Aquí tenim el programari de gestió d'imatges, i de creació i edició de continguts multimèdia (vídeo, imatges, so, etc.). Aquest camp està en gran apogeu en els últims temps i està aconseguint introduir grans canvis en la indústria cinematogràfica.
- **Programari d'enginyeria.** Ajuda en la creació en l'enginyeria, incloent-hi el disseny (els famosos programes CAD, de *computer aided design*, o DAO, de disseny assistit per ordinador, utilitzats per enginyers mecànics, enginyers aeronàutics, enginyers industrials i arquitectes). També cal incloure dins d'aquesta categoria el programari de creació de programes programari, alguns dels quals hem comentat anteriorment en aquest mòdul com, per exemple, el compilador.
- **Programari de sistemes d'informació.**

Un subconjunt del programari per a usuari final és el programari integrat com, per exemple, els paquets ofimàtics. Tenen una infraestructura pròpia i una manera d'interactuar amb l'usuari homogènia. Això permet que, una vegada que es coneix un programa del paquet, puguem utilitzar els altres de manera més senzilla, ja que els menús, la manera d'utilitzar-lo, etc. és semblant. Un exemple senzill és l'ús d'icones similars, mentre que entre les possibilitats més avançades podem trobar la d'integrar el que fem amb una aplicació en una altra. Així, podríem incloure una taula del nostre full de càlcul en el nostre document en el processador de textos.

Finalment, hi ha un tipus de programari per a usuari final que també podem trobar amb força freqüència: es tracta de les aplicacions a mida. Aquests programes els solem trobar en el món empresarial, com, per exemple, en programes de facturació o en l'aplicació que manegen els empleats de banca, fins i tot el que s'utilitza en un hospital. Aquest programari és diferent de l'anterior, ja que se sol dissenyar i crear amb els ulls posats en un usuari específic, i no per al gran públic. Podríem dir que són programes *ad hoc* creats específicament per a l'usuari.

Vegeu també

El programari de sistemes d'informació es veurà amb més detall en l'apartat 5.

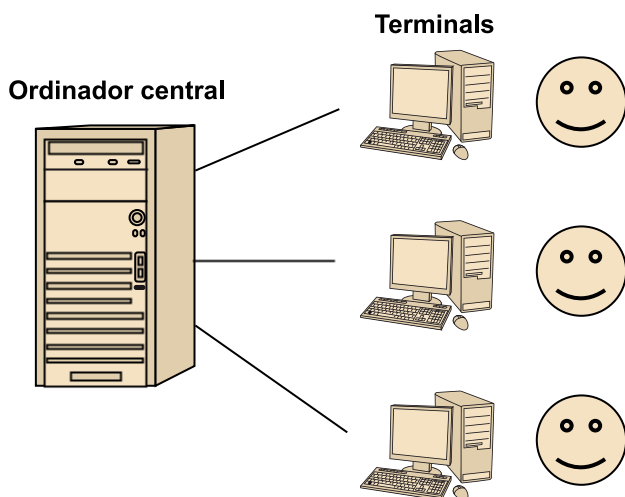
4. Tipus d'ordinador

Els ordinadors personals no són estranys per a nosaltres. Quan parlem d'ordinador personal ens ve al cap la pantalla, el teclat, el ratolí i una torre, gran o petita, que conté el lector de DVD, les connexions de dispositius externs, els altaveus, etc. Tanmateix, durant la història de la informàtica els tipus d'ordinador s'han diversificat. Alguns es consideren històrics i altres són ben populars en l'actualitat. En aquest apartat descriurem alguns dels sistemes informàtics més importants.

4.1. Els ordinadors centrals

Els primers ordinadors tenien com a objectiu la realització de càlculs sobre grans volums de dades. Durant dècades aquesta va ser l'activitat principal dels computadors. Va arribar un moment en què la tecnologia va permetre no tan sols la miniaturització sinó executar diverses tasques a la vegada gràcies als sistemes operatius. Gràcies a això, un mateix microprocessador era capaç d'executar diversos programes alhora, senzillament repartint el temps de procés entre els diferents programes que s'estaven executant. De la mateixa manera, els sistemes multiusuari van permetre que diferents treballadors accedissin a un mateix ordinador central o *mainframe* per poder-lo utilitzar. Per treballar amb l'ordinador, calia utilitzar un terminal que ens connectaria remotament a l'equip, tal com mostra la figura següent.

Un ordinador central o *mainframe* amb tres terminals connectats



Actualment aquest model continua essent vàlid en certa manera: molts dels sistemes d'informació que utilitzem són grans servidors accessibles per Internet. Per accedir-hi, sols cal fer servir el navegador web (que fa de terminal).

4.2. Els microcontroladors

Un microcontrolador és un xip que conté totes les funcionalitats bàsiques descrites en l'arquitectura Von Neumann, és a dir, té una CPU, una memòria i una unitat capaç de gestionar l'entrada i la sortida d'informació. Aquests xips poden executar programes. Els microcontroladors són ordinadors tan petits que és possible integrar-los en una gran varietat d'aparells: rentadores, mòbils, màquines industrials, forns microones digitals, càmeres de vídeo, etc.

Hi ha microcontroladors genèrics, que es poden programar i es poden fer servir en diferents àmbits. D'altra banda, hi ha microcontroladors específicament dissenyats per a un entorn determinat.

4.3. Els superordinadors

Durant els anys setanta i vuitanta del segle *xx*, diverses companyies van treure a la llum ordinadors molt potents si es comparaven amb els ordinadors centrals habituals en les grans empreses i, evidentment, molt més potents que els microordinadors personals. Aquests ordinadors, anomenats *supercomputadors*, es feien servir per a aplicacions específiques, que requerien gran potència de càlcul: des de simulacions atòmiques fins a competicions d'escacs contra humans. Tanmateix, avui dia aquests superordinadors serien poc potents si els comparéssim amb els ordinadors portàtils actuals.

D'aquesta manera, actualment la idea de superordinador s'assoleix no amb un únic ordinador molt potent, sinó amb el treball col·laboratiu de centenars o milers d'ordinadors "normals" connectats amb xarxes de gran velocitat i rendiment. Aquest concepte l'expliquem tot seguit.

4.4. Multicomputadors

Hem vist anteriorment que s'ha arribat al límit en la velocitat de rellotge dels processadors. Així, doncs, els nous ordinadors ja no augmenten el rendiment a base d'augmentar la velocitat de rellotge: es fa servir la tècnica de la "multicomputació" per a produir equips més potents.

Els **multicomputadors** pròpiament dits són els sistemes formats a partir d'una **granja d'ordinadors**, habitualment amb les mateixes capacitats i característiques de rendiment. D'aquesta manera, les tasques es poden distribuir entre els diferents ordinadors que integren la granja i fer la feina més ràpidament. Per exemple, en una productora de cinema d'animació per ordinador, una escena es podria dividir en petits trossos, cadascun dels quals passaria a ser tractat per un ordinador. D'altra banda, amb sistemes multicomputador es poden arribar a atendre peticions de milers d'usuaris. Per exemple, els grans serveis de cerca d'Internet disposen de desenes de milers d'ordinadors per a atendre els milions de peticions que reben cada minut.

Vegeu també

Recordeu que hem parlat dels microordinadors personals en l'apartat 1 d'aquest mòdul, dedicat a la història dels ordinadors.

Deep Blue

Hi ha hagut diversos ordinadors dissenyats específicament per a jugar a escacs. Un dels més famosos és el Deep Blue que l'any 1997 va guanyar al campió d'escacs Garry Kasparov.

Exemple de multicomputació

Els ordinadors personals amb processadors de diversos nuclis són un exemple de multicomputació. Un microprocessador de doble nucli conté duplicats molts dels elements que formen un processador, és a dir, és com tenir dos microprocessadors en un. D'aquesta manera, si es fan dues tasques en paral·lel (per exemple, treballar amb el campus virtual de la UOC alhora que comencen a passar fotografies per a enregistrar en un DVD) cada processador es dedica a una de les tasques. En un sistema amb un únic processador, i a l'igual dels primers ordinadors centrals, es trigaria molt més que amb els dos nuclis.

Finalment, cal parlar dels models de **graella de càlcul**. L'objectiu també és el processament en un temps factible de tasques costoses per mitjà de la divisió de la feina en tasques que realitzaran els diferents computadors que integren el grup de treball. A diferència del cas anterior, en la graella de càlcul els equips no són iguals, ni estan connectats per una xarxa d'alta velocitat, ni s'usen en exclusiva per la tasca: els equips que es fan servir solen ser ordinadors personals d'usuaris particulars que reben les petites tasques a fer per mitjà de la connexió a Internet. Aquestes petites tasques es van fent en períodes d'inactivitat de l'equip.

Exemple de graella de càlcul

Un exemple de graella de càlcul o computació *grid* és el *Search for extra-terrestrial intelligence (SETI)*, el qual es basa en l'anàlisi de senyals procedents de radiotelescopis per a descobrir si algun d'aquests pot haver estat generat per éssers intel·ligents.

4.5. Els ordinadors portàtils

La miniaturització dels equips va permetre la fabricació d'equips portàtils. Històricament, els **ordinadors portàtils** o dispositius electrònics, com agendes digitals o de **telèfons mòbils avançats**, han estat equips poc potents si se'ls compara amb els de sobretaula. Tanmateix, la tecnologia ha permès fer ordinadors portàtils prou potents, o bé prou petits, per a ser considerats ultraportàtils. Aquests productes són habituals, per la qual cosa convé esmentar-los en aquest material.

Un ordinador portàtil és actualment tan potent com un ordinador de sobretaula. De totes maneres, aquesta mena d'ordinadors no són gaire còmodes de portar sempre a sobre a causa del pes. Els ordinadors **ultraportàtils** són equips més còmodes de portar. Fins i tot hi ha una variant que permet utilitzar-los com si es tractés d'una llibreta per a prendre notes (els anomenats **tablet PC**), i tenen una pantalla tàctil per a poder-hi escriure directament a sobre. Els ordinadors **netPC** són ordinadors encara més portàtils (normalment tenen dimensions més reduïdes), a canvi de tenir un rendiment inferior als portàtils i els ultraportàtils. Tanmateix són prou potents per a permetre l'accés a Internet, la visualització de fotografies i vídeos i les tasques ofimàtiques més habituals.

5. Sistemes d'informació

En termes generals, un sistema d'informació està constituït per una sèrie d'elements que es relacionen entre ells per a donar suport a les activitats d'una institució o empresa. Els sistemes d'informació es troben dins del programari per a usuari final. Els elements de què consta un sistema d'informació són:

- Les persones que interactuen amb el sistema.
- Les dades.
- Les activitats per a processar la informació i les dades d'una organització.
- El component informàtic, que inclou tant maquinari com programari.

En un sentit més restringit del terme, de vegades molts es refereixen al sistema d'informació com l'aplicació de programari que realitza les tasques d'emmagatzemament, i algunes de manipulació i processament de dades i informació.

Exemple de sistema d'informació

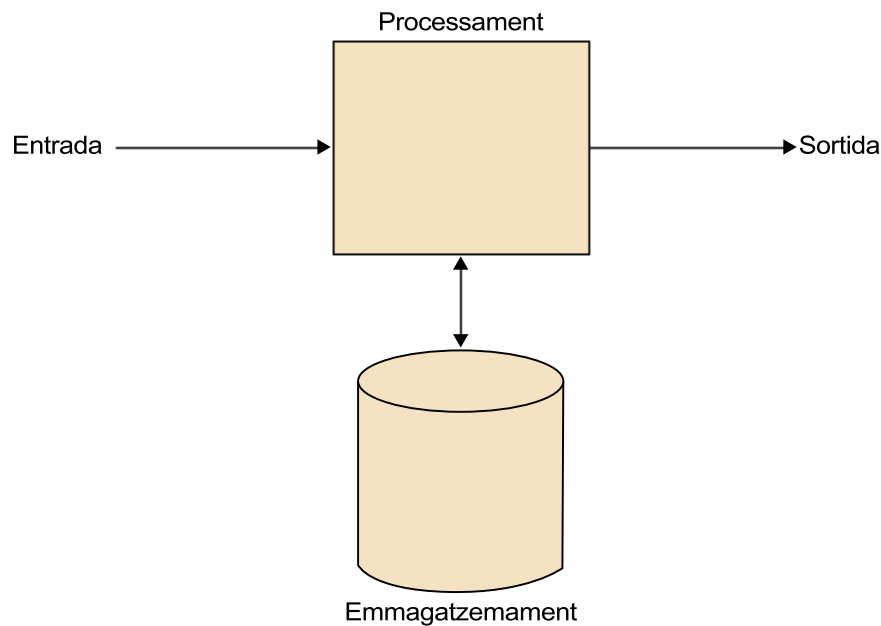
L'exemple més paradigmàtic de sistema d'informació és el cens d'una població. Herman Hollerith va crear el 1890 un sistema d'informació per gestionar de manera automàtica una gran quantitat de dades que s'havien de tractar. La diferència principal d'aquest primer sistema d'informació complex respecte als actuals és que era completament mecànic, en comptes de basar-se en l'electrònica. Però el sistema de Hollerith va servir d'inspiració per a la computació moderna, per la qual cosa no deixa de ser lògic que el primer gran sistema d'informació s'instal·lés a l'Oficina del Cens nord-americana el 1951, l'UNIVAC I, un dels primers ordinadors.

En l'actualitat, els sistemes d'informació s'utilitzen bàsicament amb tres objectius:

- automatitzen processos,
- tracten informació per ajudar en decisions, i
- busquen avantatges competitius.

Independentment de la tecnologia que s'utilitzi, els sistemes d'informació consten de quatre activitats: entrada, processament, emmagatzemament i sortida d'informació, com es pot veure en la figura següent.

Activitats bàsiques d'un sistema d'informació



Atenint-nos a l'aplicació de sistemes d'informació, tenim un gran nombre de programes dins d'aquest àmbit com, per exemple:

- **Programes de gestió de temps i recursos.** Aquí és on podem trobar els famosos ERP¹⁷, utilitzats per a la gestió empresarial. Tanmateix, hi ha altres programes, com els de comptabilitat o els de gestió de tasques i agenda, que també es poden englobar en aquesta categoria.
- **Gestió de dades.** Es tracta de programari que permet gestionar quantitats de dades importants, encara que no tan grans ni tan diverses per a tenir una base de dades específica. Així, tenim aplicacions de gestió de contactes, els fulls de càlcul o les petites bases de dades personals.
- **Documentació.** Els més coneguts dins d'aquest apartat són els processadors de textos, encara que també hi ha programari per a publicacions, programari per a la creació de diagrames i programari per a fer presentacions.
- **Programari analític.** Són programes que permeten el càlcul o la manipulació de dades numèriques per a la seva anàlisi, amb caràcter estadístic, financer o d'un altre tipus.
- **Programari cooperatiu.** Programari utilitzat per a comunicar-se amb altres persones, tant de manera asíncrona (correu electrònic, blocs o wikis) com síncrona (sistemes de missatgeria instantània o programari de telefonia IP).

⁽¹⁷⁾De l'anglès, *enterprise resource planning*, planificació de recursos de l'empresa.

La importància que ha cobrat Internet en els últims temps ha fet que els sistemes d'informació s'estiguin adaptant a un entorn distribuït, en què la informació i els processos no es troben localment.

Vegeu també

En el mòdul "El World Wide Web" tractarem de la importància d'Internet.

Resum

En aquest mòdul hem vist les bases de funcionament dels sistemes informàtics. Hem començat amb una visió històrica de les fites més importants de la història de la computació. Hem vist que els sistemes informàtics tenen una història relativament recent, i que en els darrers anys han entrat en gairebé tots els àmbits de les activitats humanes.

També hem estudiat com la informàtica tracta la informació. Hem après que els nombres i els caràcters es codifiquen en una sèrie de bits. D'aquesta manera, la informació es pot desar fent servir diferents tecnologies, alhora que se'n permet el processament per mitjà de circuits lògics.

Un cop hem vist els fonaments del tractament d'informació, hem tractat les diferents parts que formen l'ordinador, alhora que hem fet èmfasi en els aspectes del maquinari que caracteritzen el rendiment i la potència de la màquina. Finalment, hem classificat els ordinadors en funció de la grandària i complexitat.

La segona part del mòdul s'ha dedicat a presentar el programari. Hem vist que el programari té com a objectiu resoldre problemes. Així, per a crear un programa, primer s'especifica un algoritme en llenguatge natural, que posteriorment es traduirà. El pas següent consta del procés de compilació, mitjançant el qual es transforma el codi font en codi màquina (o binari), que és el que executa un ordinador.

Finalment, hem presentat els diferents tipus de programari que hi ha. El que s'ha vist amb més deteniment és el sistema operatiu, el programari que gestiona una gran quantitat dels serveis de l'ordinador. S'ha mostrat com el sistema operatiu ofereix suport a altres programes per als usuaris finals, que com el seu nom indica són aquells amb què habitualment interactuen les persones. Per acabar, s'ha presentat amb una mica més de detall un tipus de programari d'usuari final com són els sistemes d'informació. Aquests es troben a tot arreu en entorns corporatius i domèstics, i amb l'apogeu d'Internet estan guanyant una nova dimensió.

Activitats

1. Un amic nostre ens ha comentat que vol aprendre a programar. El primer que se li ha ocorregut és obrir amb l'editor de textos un arxiu executable d'una aplicació (per exemple, word.exe). Què és el que veurà? Podrà programar així? Per què no?

Solució

Solució de l'activitat 1

En obrir un arxiu executable, veurà el contingut en binari de l'arxiu. D'aquesta manera és gairebé impossible programar, ja que és intel·ligible per als éssers humans.

2. Què li aconsellariem al nostre amic si volgués programar?

Solució

Solució de l'activitat 2

Per a programar, el millor és que utilitzi un llenguatge de programació en el qual pugui realitzar les instruccions pertinents de manera més senzilla. Els llenguatges de programació permeten utilitzar un llenguatge semiestructurat que és fàcil d'entendre per als humans.

3. Una vegada que l'hem convençut, el nostre amic ha escrit unes quantes línies de codi font. Ara vol executar-les, què ha de fer?

Solució

Solució de l'activitat 3

Compilar-lo, de manera que les instruccions en llenguatge de programació del codi font siguin transformades en instruccions en binari comprensibles per a la computadora.

4. Estudieu la vida d'Ada Lovelace, la primera programadora de la història. Han donat el seu nom a un llenguatge de programació. Investigueu en quins àmbits se sol utilitzar aquest llenguatge.

5. Volem incloure en binari la informació del nostre nom (en ASCII), la nostra data de naixement (una xifra per al dia, una altra per al mes i una tercera per a l'any) i el nostre lloc de naixement (en ASCII). Quants bits ocuparia tota aquesta informació?

6. Busqueu a Internet una foto del primer ratolí.

7. Tenim dues portes lògiques AND i NOT, encadenades una darrere d'una altra. Primer es troba la porta AND i, a continuació, la seva sortida és l'entrada de la porta NOT. Què obtindríem a la sortida de la segona porta si en l'entrada de la primera porta tenim els valors Verdader i Fals?

8. Anomeneu diferents sistemes operatius que conegueu. Investigueu en la Viquipèdia les famílies que hi ha.

9. Enumereu una dotzena de programes que funcionin tant en sistemes operatius Windows com en Linux.

10. Creeu un algoritme en llenguatge natural per arribar des del vostre lloc de treball fins a la cafeteria més propera.

11. Cerqueu en el diari una oferta actual d'un ordinador de sobretaula i un portàtil. Compareu en tots dos casos els diferents tipus de memòries que inclouen i el preu.

Exercicis d'autoavaluació

Indiqueu quines de les afirmacions següents són correctes i quines són falses.

1. Els antecessors dels ordinadors, les màquines analítiques, es van idear molt abans del segle XX, però per problemes d'enginyeria no es van poder construir.

- a) Verdader
- b) Fals

2. La primera generació d'ordinadors moderns va arribar sota el guiatge dels transistors.

- a) Verdader
- b) Fals

3. Els entorns de finestres van ser ideats en la dècada de 1970 per Xerox, encara que van trigar força anys a arribar al gran públic.

- a) Verdader
- b) Fals

4. El byte 00000111 significa exclusivament el número 7 en binari.

- a) Verdader
- b) Fals

5. L'any 2009 ocupa en ASCII un total de 4 bytes.

- a) Verdader
- b) Fals

6. Quan apaguem l'ordinador, s'esborra la memòria principal (RAM).

- a) Verdader
- b) Fals

7. Els programadors creen els programes majoritàriament en codi binari.

- a) Verdader
- b) Fals

8. El sistema operatiu s'encarrega de gestionar els recursos d'un ordinador.

- a) Verdader
- b) Fals

9. Un algoritme ha d'estar escrit en un llenguatge de programació d'alt nivell.

- a) Verdader
- b) Fals

10. Un *mainframe* és un ordinador central al qual es poden connectar múltiples ordinadors anomenats *terminals*.

- a) Verdader
- b) Fals

Solucionari

Exercicis d'autoavaluació

1. a

2. b

3. a

4. b

5. a

6. a

7. b

8. a

9. b

10. a

Glossari

algoritme *m* Conjunt finit de passos que serveixen per a resoldre un problema o dur a terme una tasca.

analista de programari *m* Persona, generalment de perfil tècnic, que analitza els requisits que ha de complir un futur programa d'ordinador.

arquitectura d'ordinadors *f* Concepte que engloba el disseny i funcionament dels ordinadors que especifiquen, entre altres aspectes, el format i el conjunt d'instruccions. Hi ha moltes arquitectures. La més popular és la i386 perquè és la que s'utilitza en els PC.

byte *m* Grup de 8 bits.

COBOL *m* Llenguatge de programació. Acrònim de *common organization business oriented language* (llenguatge comú orientat a l'organització de negocis), encara que també es pot trobar com a *common business oriented language* (llenguatge comú orientat a negocis). Es tracta d'un llenguatge de programació del final de la dècada dels seixanta el propòsit del qual era poder-se utilitzar en qualsevol ordinador. Actualment, el seu ús és força limitat, encara que es troba amb freqüència en aplicacions bancàries i programes antics.

codi binari *m* Sistema de numeració basat en la utilització de dos valors o bits (típicament 0 i 1) en què, al seu torn, es basa el processament automàtic de la informació. Tot el que es processa amb un ordinador acaba essent transformat a codi binari. Vegeu **codi màquina**.

codi font *m* Instruccions d'ordinador escrites en un llenguatge de programació. En la fase de compilació es transforma en codi màquina. Perquè el programari sigui lliure, el codi font ha de ser accessible, ja que si no la possibilitat de realitzar modificacions, encara que no sigui impossible, es dificulta sobre manera. També conegut com a *font*.

codi màquina *m* Codi que els ordinadors poden executar. Consta d'uns i zeros, encara que hi ha altres formes de representació com l'octal o hexadecimal. El codi màquina és difícilment comprensible per als humans –i la creació d'aplicacions complexes, gairebé impossible–, per la qual cosa es van crear els llenguatges de programació d'alt nivell. També conegut com a *codi binari*.

compilador *m* Element que s'encarrega principalment de traduir els fitxers escrits en llenguatges de programació (comprensibles per als humans) a codi màquina (uns i zeros, generalment només comprensibles per als ordinadors).

computador *m* Artefacte capaç de fer càlculs i de mostrar resultats sobre les dades numèriques que s'introdueixen. La seva complexitat ha anat augmentant, i la finalitat s'ha anat diversificant, i s'ha convertit en un aparell com les calculadores de butxaca o els ordinadors personals d'avui en dia.

controlador *m* Vegeu **driver**.

C/C++ *m* Llenguatge de programació. C es va crear al començament dels setanta per al desenvolupament de l'UNIX. Es tracta d'un dels primers llenguatges de programació que permeten la programació d'alt nivell. C++ és una versió posterior de C que afegeix tècniques modernes de programació. La majoria del codi lliure està escrit en C o C++.

depuració *f* Procés mitjançant el qual es busquen (i corregeixen) errors en el programari. Vegeu també *depurador*.

depurador *m* Programa d'ordinador utilitzat a l'hora de crear programari per a trobar i corregir errors de programació.

driver *m* Programari que s'encarrega d'interactuar entre el sistema operatiu i els dispositius (maquinari).

i386 Arquitectura d'ordinador típica dels ordinadors personals (PC).

Java *m* Llenguatge de programació creat en els anys noranta amb moltes similituds sintàctiques amb C i C++, però amb característiques de programació de nivell d'abstracció més alt.

kernel *m* Vegeu **nucli**.

llenguatge assemblador *m* Llenguatge de baix nivell d'abstracció que permet crear programes d'ordinador. Es tracta del llenguatge de programació més proper al codi màquina.

llenguatge de programació *m* Conjunt de regles semàntiques i sintàctiques utilitzades per a donar instruccions a un ordinador. Els llenguatges de programació permeten treballar en un nivell d'abstracció superior que amb codi màquina, la qual cosa facilita la creació i manteniment de programes informàtics. N'hi ha cents, sinó milers, de llenguatges de programació. Alguns exemples són C, C++, ADA, Java, Pascal i COBOL.

llenguatge natural *m* Llenguatge escrit o parlat pels humans amb finalitats de comunicació.

maquinari *m* Conjunt de dispositius físics que componen l'ordinador: la pantalla, el teclat, el ratolí, etc.

microprocessador *m* Element del maquinari del sistema informàtic encarregat de processar la informació, com també de gestionar l'execució de les ordres que conformen els programes i el traspàs de dades entre els diferents components del computador.

model de desenvolupament en cascada *m* Procediment per a crear programari, basat en la divisió del procés de desenvolupament en una seqüència de tasques, que es realitzen una darrere de l'altra. Les tasques es realitzen en l'ordre següent: definició dels requisits, anàlisi i disseny de programari, implementació i prova d'unitats, integració i proves del sistema, i operació i manteniment.

multicomputador *m* Nom genèric que rep un sistema informàtic basat en l'ús de més d'un computador o processador. sin. **multiprocessador**

nucli *m* Part del sistema operatiu que s'encarrega de les tasques de nivell més baix (el nivell més proper al maquinari) tals com gestió de memòria, d'entrada/sortida de dispositius, etc. El *kernel* més popular en el món del programari lliure és el Linux, encara que n'hi ha molts més (per exemple, els sistemes BSD en tenen un de propi).

Pascal *m* Llenguatge de programació de la dècada dels setanta escassament utilitzat avui en dia, encara que alguns dels seus successors sí que tenen un ampli ressò en la indústria del programari.

programari *m* Component intangible en la informàtica. Generalment es tracta d'una sèrie d'instruccions elaborades per humans en llenguatges de programació d'alt nivell (codi font) que després són traduïdes per un compilador a codi màquina (uns i zeros entesos per les màquines). El programari es divideix en programari de sistema, informe que correspon als sistemes operatius, o d'aplicació, que agrupa els programes que l'usuari sol utilitzar. Estrictament el programari també inclou la documentació del programa, encara que aquesta es trobi en un manual.

sistema operatiu *m* Programa responsable de la gestió i coordinació d'activitats i del repartiment de recursos d'un ordinador.

taula de veritat *f* Taula que especifica, per a les diferents combinacions dels bits d'entrada, quina és la sortida d'una funció lògica o un circuit de càlcul.

temps d'execució *m* Espai temporal en què l'ordinador executa les instruccions corresponents a un programa de programari.

Bibliografia

Angulo, J. M i altres (2003). *Fundamentos y estructura de computadores*. Thomson Paraninfo.

Barceló, M. (2008). *Una historia de la Informática*. Barcelona: Editorial UOC.

Beekman, G. (2004). *Computer Confluence. Standard* (6a. ed.). Addison-Wesley.

Beekman, G. (2004). *Introducción a la informática*. Pearson Prentice Hall.

Leiva Olivencia, J. L. (2009). *Informática para Estudiantes*. Editorial Abecedario. ISBN 9788492669011.

Pareja, C.; Andreyo, Á.; Ojeda, M. (1993). *Introducción a la Informática*. ISBN: 84-7491-489-2.

Reina, P. *Curso de Informática*.

Reynolds, G. W. i altres (2000). *Principios de sistemas de información* (4a. ed.). Thomson Paraninfo.