

Unidad 1

Conceptos básicos de programación. Algoritmos y programas

Programación
1º D.A.M.

1

Contenido

1. Representación de la información
2. Definiciones informáticas
3. Algoritmos
4. Programación

2

1. Representación de la información

1. Sistemas de numeración

1. Sistema binario
2. Sistema hexadecimal
3. Conversiones

2. Códigos

3

1.1. Sistemas de numeración

- Definición y objetivo
 - Símbolos y reglas
 - Representar y operar con cantidades
- Base
 - Número de símbolos usados
- Tipos
 - Aditivos
 - Posicionales

4

1.1.1. Sistema binario

■ Unidades de medida de información

- Bit → 0 ó 1
- Byte → 8 bits
- Kbyte → 1024 bytes
- Mbyte → 1024 Kbytes
- Gbyte → 1024 Mbytes
- Tbyte → 1024 Gbytes
- Pbyte → 1024 Tbytes

5

1.1.1. Sistema binario

■ Operaciones

- Aritméticas
 - Suma
 - Resta
- Lógicas
 - AND
 - OR
 - NOT
 - XOR
 - Combinación de las anteriores (NAND, NOR)

6

1.1.1. Sistema binario

OR	0	1
0	0	1
1	1	1

AND	0	1
0	0	0
1	0	1

XOR	0	1
0	0	1
1	1	0

NOT	0	1
	1	0

7

1.1.2. Sistema hexadecimal

■ Sistema hexadecimal

Decimal	Hexadecimal	Binario
0	0x0	0000
1	0x1	0001
2	0x2	0010
3	0x3	0011
4	0x4	0100
5	0x5	0101
6	0x6	0110
7	0x7	0111

8	0x8	1000
9	0x9	1001
10	0xA	1010
11	0xB	1011
12	0xC	1100
13	0xD	1101
14	0xE	1110
15	0xF	1111

8

1.1.3. Conversiones

a... De ...	Decimal	Binario	Hexa
Decimal	-	Divisiones (entre 2)	Divisiones (entre 16)
Binario	Producto (base 2)	-	Cada 4 bits, 1 dígito
Hexa	Producto (base 16)	Cada dígito, 4 bits	-

9

1.2. Códigos

- Definición y objetivo
 - Símbolos y reglas
 - Representar una información
- Algunos códigos
 - Texto
 - ASCII
 - 7 bits → 128 caracteres posibles
 - 8 bits → 256 caracteres posibles
 - UNICODE
 - Otra información
 - Códigos complejos para imágenes, vídeo, etc

10

1.2. Códigos

000	<nul>	016	<dle>	032	sp	048	0	064	@	080	P	096	`	112	p
001	<soh>	017	<dc1>	033	!	049	1	065	A	081	Q	097	a	113	q
002	<stx>	018	<dc2>	034	"	050	2	066	B	082	R	098	b	114	r
003	<etx>	019	<dc3>	035	#	051	3	067	C	083	S	099	c	115	s
004	<eot>	020	<dc4>	036	\$	052	4	068	D	084	T	100	d	116	t
005	<enq>	021	<nak>	037	%	053	5	069	E	085	U	101	e	117	u
006	<ack>	022	<syn>	038	&	054	6	070	F	086	V	102	f	118	v
007	<bel>	023	<etb>	039	'	055	7	071	G	087	W	103	g	119	w
008	<bs>	024	<can>	040	<	056	8	072	H	088	X	104	h	120	x
009	<tab>	025		041	>	057	9	073	I	089	Y	105	i	121	y
010	<lf>	026	<eof>	042	*	058	:	074	J	090	Z	106	j	122	z
011	<vt>	027	<esc>	043	+	059	;	075	K	091	[107	k	123	{
012	<np>	028	<fs>	044	,	060	<	076	L	092	\	108	l	124	
013	<cr>	029	<gs>	045	-	061	=	077	M	093]	109	m	125	}
014	<so>	030	<rs>	046	.	062	>	078	N	094	^	110	n	126	~
015	<si>	031	<us>	047	/	063	?	079	O	095	_	111	o	127	Δ

128	Ç	143	8	158	È	172	¼	186		200	Ù	214		228	Σ	242	¿
129	ü	144	É	159	Ê	173	½	187]]	201	Ú	215		229	σ	243	≤
130	é	145	Ê	160	Á	174	¾	188]]	202	Û	216		230	μ	244	∫
131	à	146	Ë	161	Â	175	»	189]]	203	Ü	217		231	ν	245	∫
132	á	147	Ì	162	Ã	176	»	190]]	204	Ý	218		232	ξ	246	÷
133	â	148	Í	163	Ä	177	»	191]]	205	ÿ	219		233	θ	247	≈
134	ã	149	Î	164	Å	178	»	192]]	206	ÿ	220		234	Ω	248	°
135	ä	150	Ï	165	Æ	179	»	193]]	207	ÿ	221		235	δ	249	·
136	å	151	Û	166	Ç	180	»	194]]	208	ÿ	222		236	ω	250	∞
137	æ	152	Ü	167	È	181	»	195]]	209	ÿ	223		237	ø	251	√
138	è	153	Ý	168	É	182	»	196]]	210	ÿ	224		238	€	252	¤
139	é	154	ÿ	169	Ê	183	»	197]]	211	ÿ	225		239	£	253	£
140	ê	155	ÿ	170	Ë	184	»	198]]	212	ÿ	226		240	≡	254	■
141	ë	156	ÿ	171	Ì	185	»	199]]	213	ÿ	227		241	±	255	■
142	ä	157	ÿ														

Tabla ASCII

11

2. Definiciones informáticas

1. Informática
2. Computadora
3. Hardware y software
4. Sistema operativo

12

2.1. Informática

- Ciencia del tratamiento automático de la información mediante un computador.
- INFORmación autoMÁTICA

13

2.2. Computadora

- Máquina electrónica, analógica o digital, dotada de una memoria de gran capacidad y de métodos de tratamiento de la información, capaz de resolver problemas matemáticos y lógicos mediante la utilización automática de programas informáticos

14

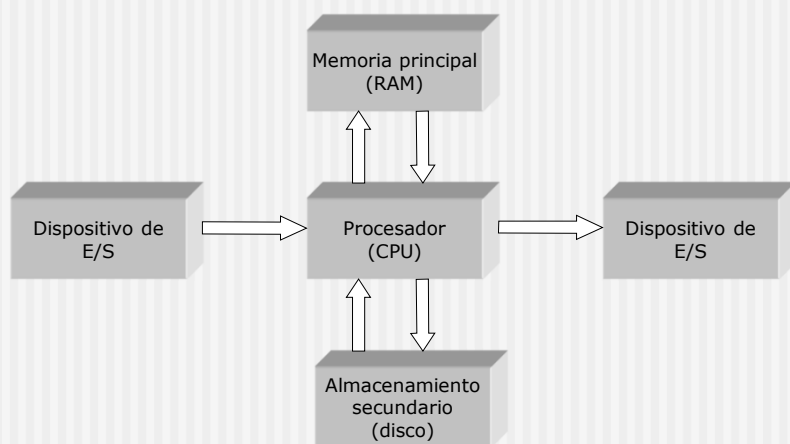
2.2. Computadora

■ Elementos básicos

- Procesador
- Placa base
- Memoria RAM
- Memoria caché
- Periféricos
- Unidades de almacenamiento

15

2.2. Computadora



16

2.2. Computadora

■ Funcionamiento básico

■ Memoria

- Alberga los programas en ejecución
 - Conjunto de instrucciones
 - Datos con los que operan esas instrucciones
- Disco → Memoria

■ Procesador (CPU)

- Busca instrucciones y datos en la memoria
- Opera con las instrucciones y los datos
- Graba los resultados en la memoria
 - Si se desea que sean permanentes, grabar en disco

■ Dispositivos de E/S

- Comunicación con el exterior

17

2.3. Hardware y software

■ Hardware

- Componentes físicos del ordenador
- Procesador, RAM, periféricos...

■ Software

- Componentes lógicos del ordenador
- Programas y datos

18

2.4. Sistema operativo

■ Definición

- Conjunto de programas o software destinado a permitir la comunicación del usuario con un ordenador, y gestionar sus recursos de manera cómoda y eficiente.
- Comienza a trabajar cuando se enciende el ordenador, y gestiona el hardware de la máquina desde los niveles más básicos.

19

2.4. Sistema operativo

■ Funciones

- Interfaz usuario – ordenador
 - Gráfica
 - Textual (comando)
- Gestión del hardware
- Organización lógica de datos (archivos, directorios, ...)
- Organización de aplicaciones
- Gestión del acceso a la red
- Gestión de errores
- Base de la creación de programas
- Gestión de usuarios
- Gestión de la seguridad

20

2.4. Sistema operativo

■ Algunos sistemas operativos

■ De Microsoft

- MS-DOS
- Familia Windows
 - Windows 3.1 y 3.11
 - Windows 95
 - Windows 98
 - Windows Millenium
 - Windows NT
 - Windows XP
 - Windows Vista
 - Windows 7
 - Windows 8 y 8.1
 - Windows 10



- Windows 2000
- Windows Server 2003
- Windows Server 2008
- Windows Server 2012
- Windows Server 2016
- Windows Server 2019

21

2.4. Sistema operativo

■ Algunos sistemas operativos

■ UNIX

■ Solaris (para Sun)

■ Linux

- Distribuciones
 - Ubuntu, openSUSE, Red Hat, Debian, Fedora, ...

■ MacOS (para Macintosh)



22

3. Algoritmos

1. Definición
2. Representación de algoritmos
 1. Diagramas de flujo
 2. Pseudocódigo

23

3.1. Definición

- Secuencia de pasos a seguir para solucionar un problema específico.
- Elementos
 - Entrada
 - Proceso
 - Salida
- Programa = Algoritmo + Estructura de datos

24

3.1. Definición

- Características necesarias
 - Preciso
 - Definido
 - Finito
 - Válido
- Características deseables
 - Eficiente
 - Óptimo

25

3.2. Representación de algoritmos

- Objetivo de representación
 - Pasos para resolver el problema
 - Orden de ejecución de dichos pasos
- Formas más usadas
 - Diagramas de flujo
 - Pseudocódigo

26

3.2. Representación de algoritmos

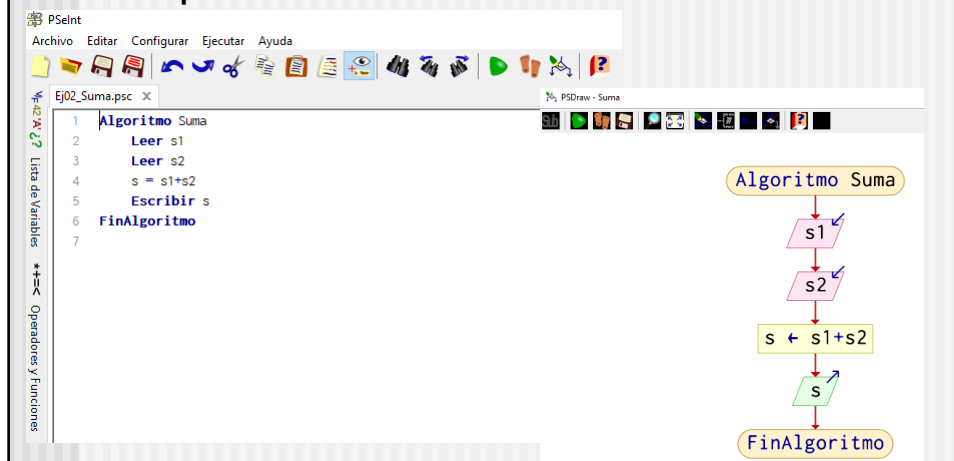
■ Aplicación Pseint

- Permite diseñar algoritmos
 - Diagramas de flujo
 - Pseudocódigo
- Permite cambiar de representación
- Permite probar el algoritmo
 - Ejecución
 - Ejecución paso a paso

27

3.2. Representación de algoritmos

■ Aplicación Pseint



28

3.2.1. Diagramas de flujo

- Elementos

- Símbolos → Sentencias (pasos)
- Flechas → Secuencia (orden)

- Tipos

- Diagrama de sistema o de configuración
- Diagrama de macroprocesos o bloques
- Diagrama de detalle u **ordinograma**

29

3.2.1. Diagramas de flujo

- Deben mostrar

- Inicio del programa
- Operaciones a realizar
- Secuencia de dichas operaciones
- Final del programa

30

3.2.1. Diagramas de flujo

■ Símbolos

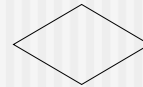
■ Terminal



■ Entrada / Salida



■ Decisión



■ Proceso



31

3.2.1. Diagramas de flujo

■ Símbolos

■ Subprograma



■ Conector a la misma página



■ Conector a distinta página



■ Líneas de flujo



32

3.2.1. Diagramas de flujo

■ Reglas

- Usar líneas rectas
- Comienzo en la parte superior
- Un único símbolo de comienzo y otro de fin
- Flujo de arriba abajo y de izqda. a dcha.
- Simetría
- Evitar cruces
- Número mínimo de instrucciones
- Restringir el uso de comentarios

33

3.2.2. Pseudocódigo

- Aproximación a la codificación final de la solución a un problema dado, sin usar un lenguaje de programación concreto.
- Paso intermedio entre la solución propuesta para el problema, y su codificación en un lenguaje de programación concreto.

34

3.2.2. Pseudocódigo

■ Elementos

- Instrucciones de entrada / salida
- Instrucciones de proceso
- Sentencias de control del flujo
- Módulos en que se ha dividido, si hay

35

3.2.2. Pseudocódigo

■ Instrucciones

- Declarativas
- De asignación
- De entrada
- De salida
- De control
 - Condicionales
 - Condicional simple
 - Condicional doble
 - Condicional múltiple
 - De iteración o repetición
 - Bucle Mientras
 - Bucle Hasta
 - Bucle Para

36

3.2.2. Pseudocódigo

- Instrucciones declarativas
 - VALOR1 Real
 - VALOR2 Entero
- Instrucciones de asignación
 - VALOR1 \leftarrow 2
 - I \leftarrow I + 1
- Instrucciones de entrada
 - Leer VALOR1
- Instrucciones de salida
 - Imprimir VALOR1
 - Grabar VALOR2

37

3.2.2. Pseudocódigo

- Instrucción condicional simple

```
Si (condición) Entonces
    Instrucción1
    Instrucción2
Fin Si
```

38

3.2.2. Pseudocódigo

■ Instrucción condicional doble

```
Si (condición) Entonces
    Instrucción1
    Instrucción2
Si no
    InstrucciónA
    InstrucciónB
Fin Si
```

39

3.2.2. Pseudocódigo

■ Instrucción condicional múltiple

```
Caso (Expresión)
    Valor1:
        InstrucciónA
    Valor2:
        InstrucciónB
    Valor3:
        InstrucciónC
Fin Caso
```

40

3.2.2. Pseudocódigo

■ Bucle Mientras

```
Mientras (condición) Hacer
    Instrucciones
Fin Mientras
```

41

3.2.2. Pseudocódigo

■ Bucle Hasta

```
Hacer
    Instrucciones
Hasta (condición)
```

42

3.2.2. Pseudocódigo

■ Bucle Para

```
Hacer Para VC desde VI hasta VF paso (IN)
    Instrucciones
Fin Hacer
```

■ Donde

- VC → Variable de control
- VI → Valor inicial
- VF → Valor final
- IN → Incremento

43

4. Programación

1. Definición
2. Paradigmas de programación
3. Lenguajes de programación
4. Entornos integrados de desarrollo
5. Ciclo de vida del software

44

4.1. Definición

- Paso de algoritmos a un lenguaje de programación, de modo que lo entienda el ordenador.

45

4.2. Paradigmas de programación

- Programación desordenada
- Programación estructurada
- Programación modular
- Programación orientada a objetos


46

4.2. Paradigmas de programación

■ Programación desordenada

- Ausencia de un método
- Programación *spagetti*
- Corrección dudosa
- Entendimiento difícil

47

- 1.- Comprobar si hay patatas
 - 2.- Si no hay patatas, saltar a 8
 - 3.- Comprobar si hay huevos
 - 4.- Si no hay huevos, saltar a 10
 - 5.- Comprobar si hay aceite
 - 6.- Si no hay aceite, saltar a 12
 - 7.- Saltar a 13
 - 8.- Comprar patatas
 - 9.- Saltar a 3
 - 10.- Comprar huevos
 - 11.- Saltar a 5
 - 12.- Comprar aceite
 - 13.- Batir los huevos
 - 14.- Si no están bien batidos, saltar a 13
 - 15.- Freir la patata
 - 16.- Si no está bien frita, saltar a 15
 - 17.- Mezclar patata y huevo
 - 18.- Cuajar
 - 19.- Voltear
 - 20.- Cuajar
- 

48

4.2. Paradigmas de programación

■ Programación estructurada

- Permitidas tres estructuras de control
 - Secuencias
 - Alternativas
 - Iteraciones
- Código mucho más legible

49

```
graph TD; A[Si no hay patatas Entonces  
Añadir patatas a la Lista] --> B[Fin Si]; B --> C[Si no hay huevos Entonces  
Añadir huevos a la Lista]; C --> D[Fin Si]; D --> E[Si no hay aceite Entonces  
Añadir aceite a la Lista]; E --> F[Fin Si]; F --> G[Si Lista no está vacía Entonces  
Comprar ingredientes de Lista]; G --> H[Fin Si]; H --> I[Mientras huevos no bien batidos Hacer  
Batir huevos]; I --> J[Fin Mientras]; J --> K[Mientras patata no bien frita Hacer  
Freir la patata]; K --> L[Fin Mientras]; L --> M[Mezclar patata y huevo]; M --> N[Cuajar]; N --> O[Voltear]; O --> P[Cuajar];
```

Si no hay patatas Entonces
 Añadir patatas a la Lista
Fin Si
Si no hay huevos Entonces
 Añadir huevos a la Lista
Fin Si
Si no hay aceite Entonces
 Añadir aceite a la Lista
Fin Si
Si Lista no está vacía Entonces
 Comprar ingredientes de Lista
Fin Si
Mientras huevos no bien batidos Hacer
 Batir huevos
Fin Mientras
Mientras patata no bien frita Hacer
 Freir la patata
Fin Mientras
Mezclar patata y huevo
Cuajar
Voltear
Cuajar

50

4.2. Paradigmas de programación

■ Programación modular

■ Módulo

- Subprograma
- Independiente del resto del programa

■ Programas = módulos integrados

- Programación estructurada de los módulos
- Integración de los módulos

■ Ventajas

- Facilita localización de errores
- Facilita mantenimiento del programa

51



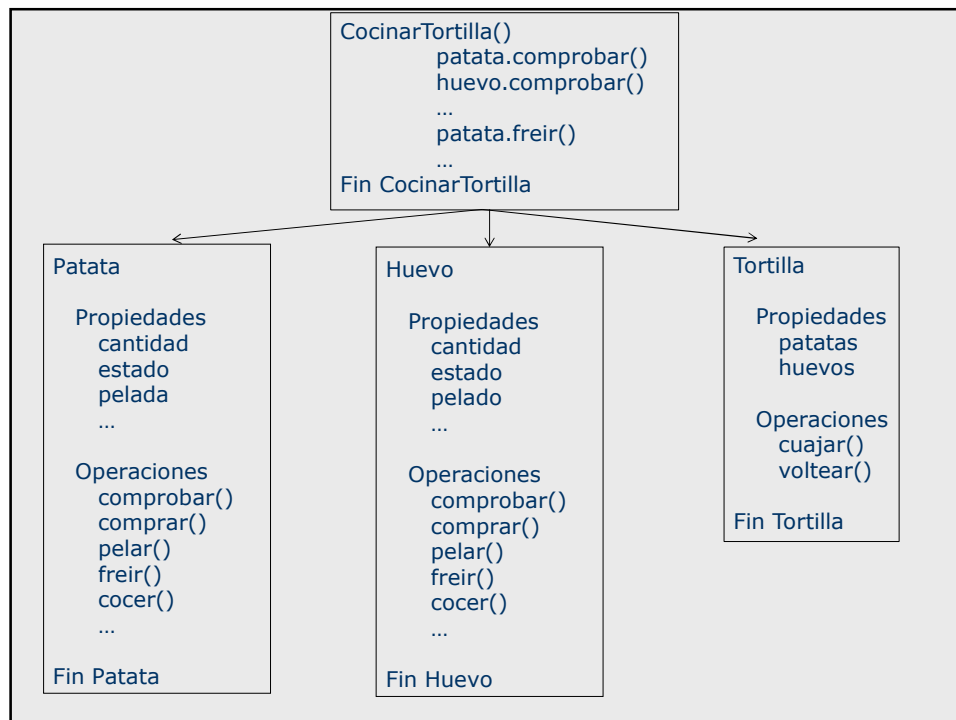
52

4.2. Paradigmas de programación

■ Programación orientada a objetos

- Acerca programa y pensamiento humano
- Composición
 - Objetos
 - Propiedades (atributos)
 - Operaciones (métodos)
 - Comunicación entre objetos
- Ventajas
 - Facilita localización de fallos
 - Facilita la reutilización de código
 - Facilita la documentación del código

53



54

4.3. Lenguajes de programación

■ Definición

- Símbolos e instrucciones con una sintaxis determinada
- Permiten codificación de algoritmos en el ordenador

■ Tipos

- Lenguajes de bajo nivel
 - Código máquina (primera generación)
 - Ensamblador (segunda generación)
- Lenguajes de alto nivel (tercera generación)
- Lenguajes de cuarta generación (4GL)

55

4.4. Entornos integrados de desarrollo

■ IDE's

■ Elementos mínimos

- Editor de texto o código
- Compilador y/o intérprete
- Depurador de errores

■ Otros elementos

- Herramienta de control de versiones
- Herramientas para GUI's

56

4.4. Entornos integrados de desarrollo

■ IDEs para Microsoft Windows

■ De código abierto

- Geany
- Visual-MinGW
- DevC++
- SharpDevelop
- V IDE
- BVRDE

■ Propietarios

- Borland C++
- Visual Studio
- MinGW Developer Studio
- Delphi
- C++ Builder
- Wind River Workbench

57

4.4. Entornos integrados de desarrollo

■ IDEs para Linux

■ De código abierto

- Geany
- Emacs
- Vim
- Anjuta
- KDevelop
- Gambas
- GPS
- Kinterdev
- OpenLDev
- Maguma Open Studio

■ Propietarios

- Kylix
- Code Forge
- Maguma Workbench

58

4.4. Entornos integrados de desarrollo

■ IDEs basados en Java

■ De código abierto

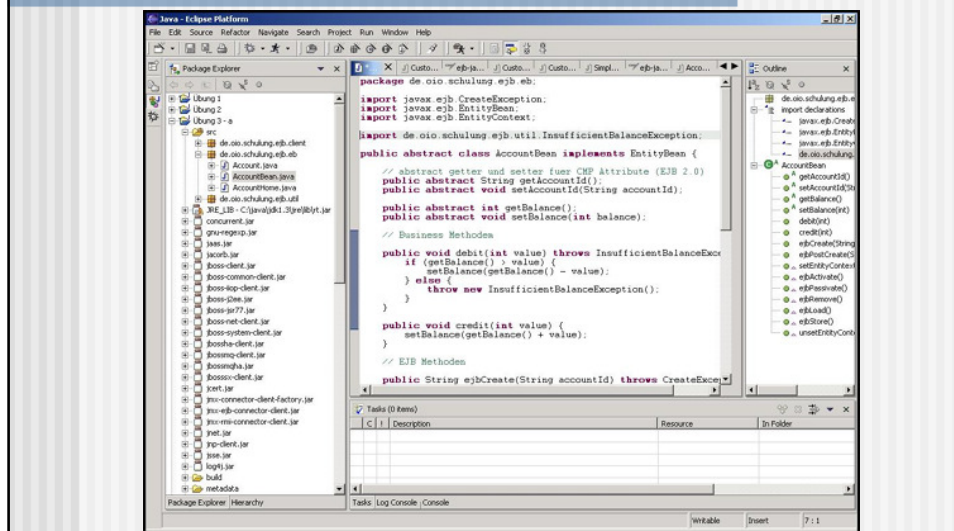
- Eclipse
- NetBeans

■ Proprietarios

- JBuilder
- IntelliJ IDEA
- JCreator

59

4.4. Entornos integrados de desarrollo



60

4.5. Ciclo de vida del software

- Análisis
- Diseño
- Implementación
- Pruebas
- Implantación
- Explotación
- Mantenimiento
- Retirada

61

Unidad 1

Conceptos básicos de programación. Algoritmos y programas

Programación
1º D.A.M.

62