



UNIVERSIDAD
POLITECNICA
DE VALENCIA

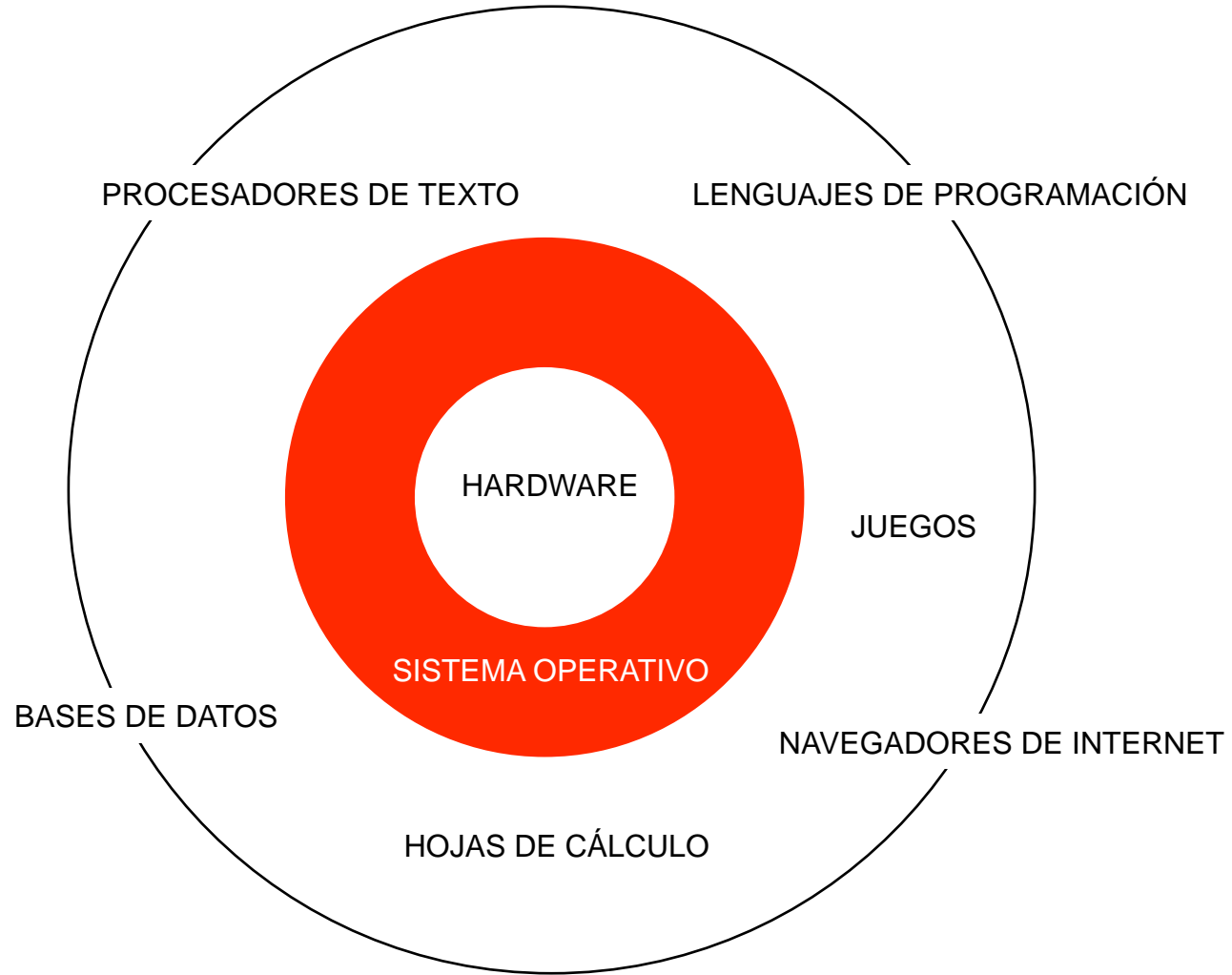
INTRODUCCIÓN A LA **PROGRAMACIÓN**

Dpto. Escultura. Facultad de BB.AA. Valencia. UPV



Prof. Moisés Mañas Carbonell
Dpto. Escultura. UPV
moimacar@esc.upv.es

**¿DÓNDE ESTAMOS/
DÓNDE VAMOS A
TRABAJAR ?**



Los **datos** se constituyen de hechos y **cifras en
bruto**

Los **datos** se constituyen de hechos y **cifras en bruto**

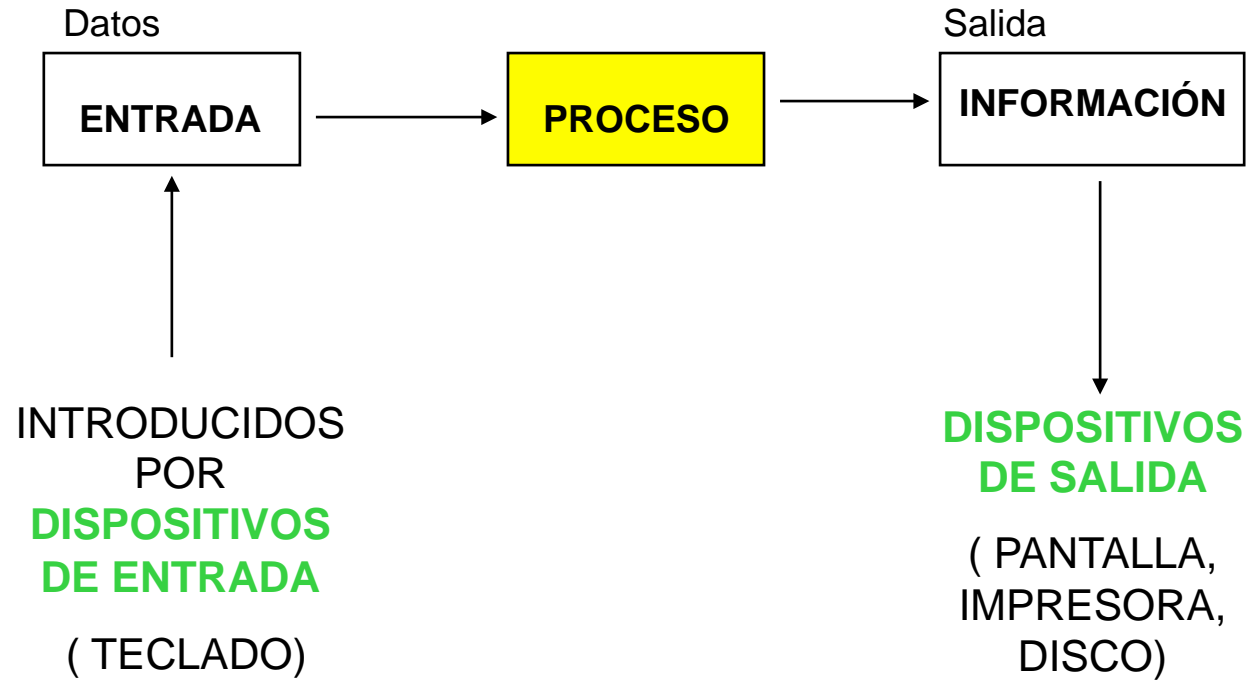


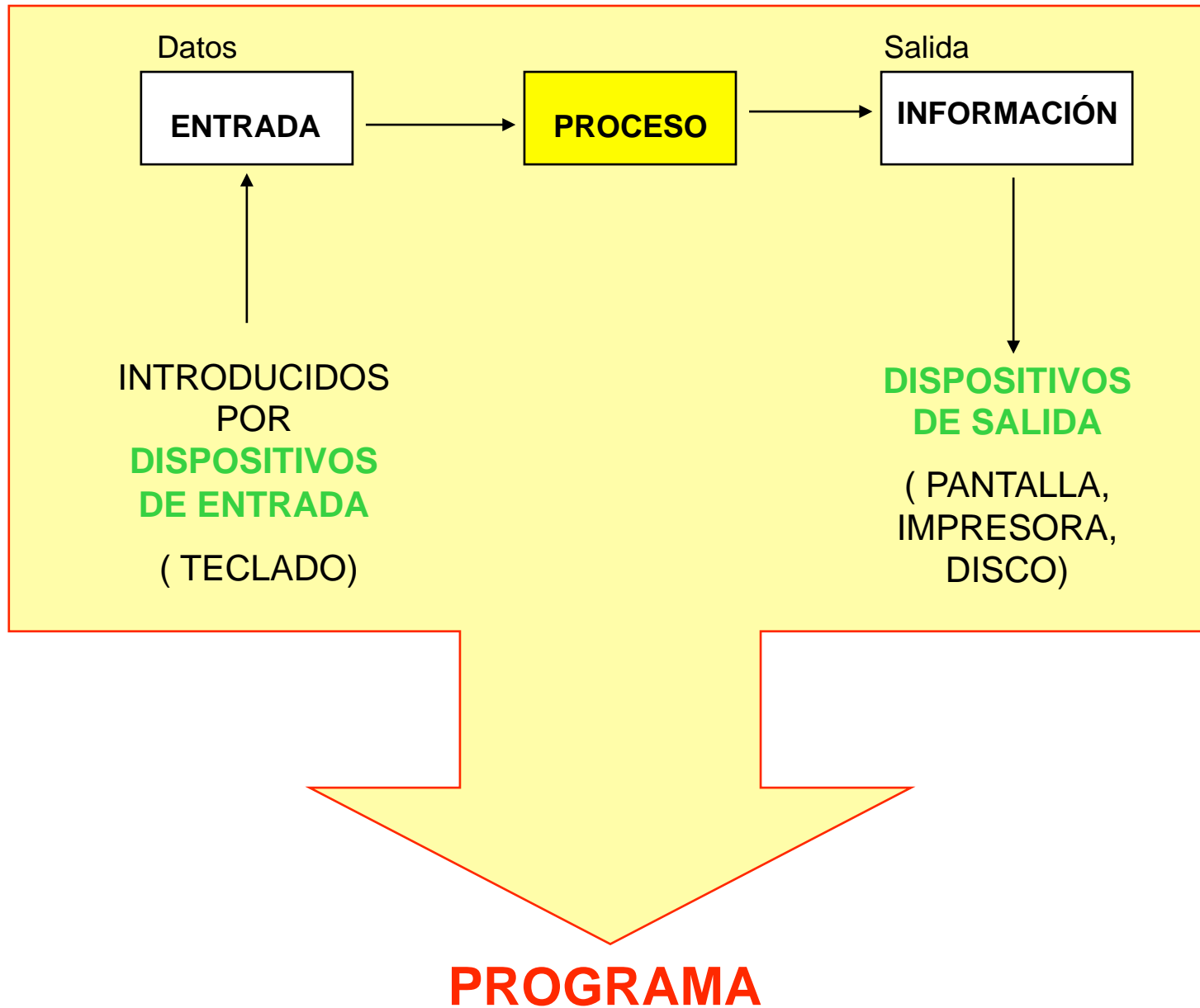
La **información** tiene un **significado**; los datos no



(Procesamiento de datos a nivel elemental)

INTRODUCCIÓN A LA PROGRAMACIÓN







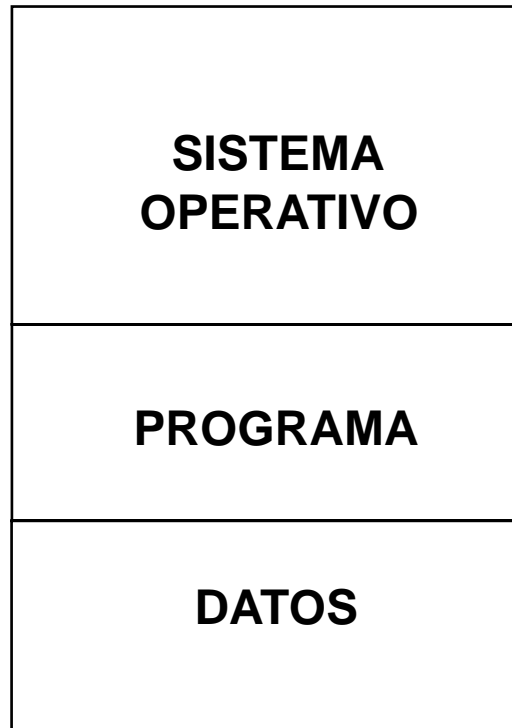
UNA LISTA DE INSTRUCCIONES DETALLADAS QUE REALIZA LA COMPUTADORA

PROGRAMA

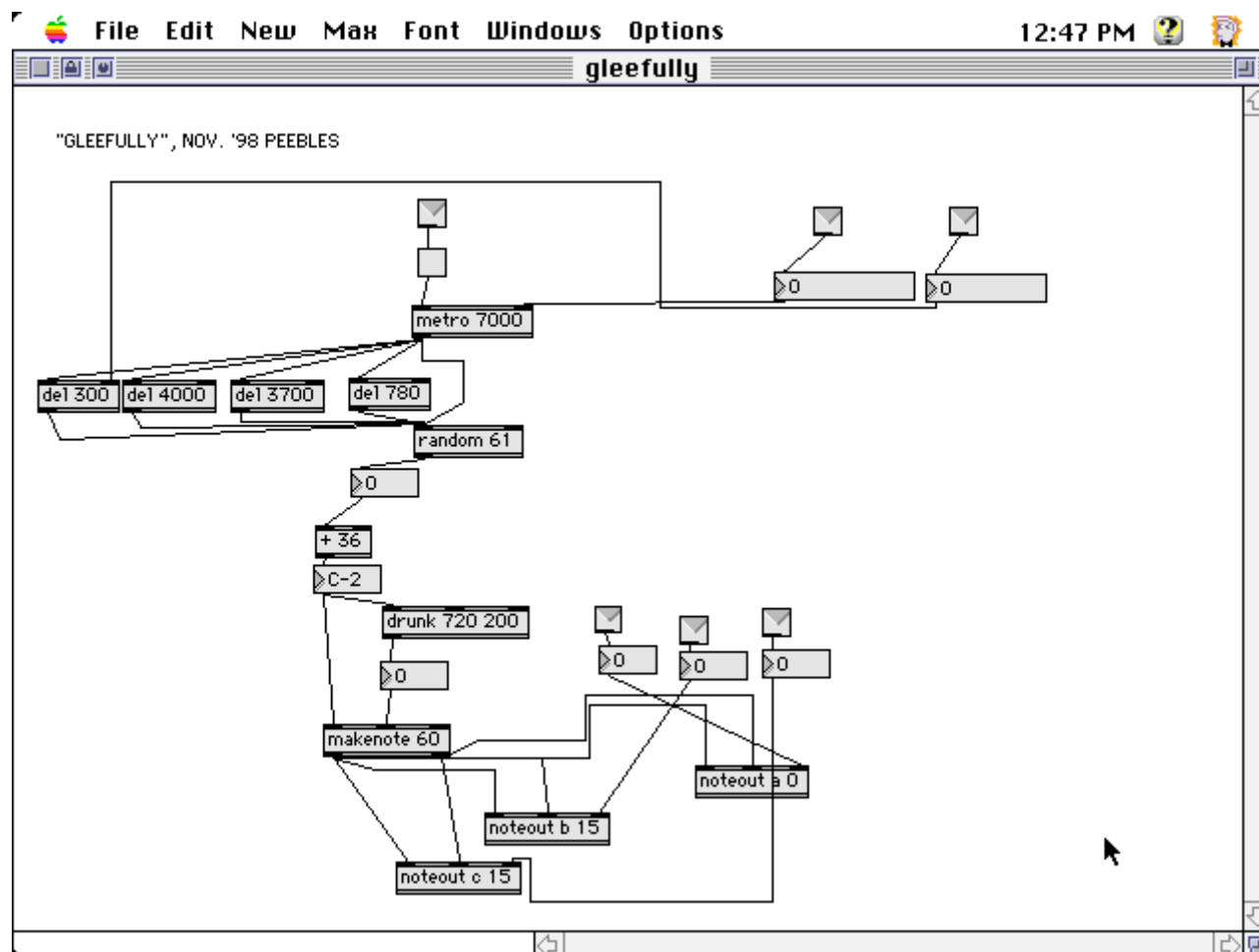
```
IF (.NOT. ga_create(MT_INT, 6, 4, 'A', 6, 1, g_a))
$   CALL GA_ERROR(' GA_CREATE FAILED ',0)
call ga_fill_patch(g_a, 1, 6, 1, 6, ME)
call ga_print(g_a)
call ga_get(g_a, 1, 6, ME+1, ME+1, B, 6)
call ga_list_nodeid(list, NPROC)
MYRANK = LIST(ME+1)
UP = MOD(MYRANK + 1, NPROC)
DOWN = MYRANK - 1
if (DOWN.LT.0)DOWN = NPROC - 1
call ga_sync()
    call MPI_Irecv(T, 6, MPI_INTEGER, DOWN, 0, MPI_COMM_WORLD, IREQ, IE
    call MPI_Send(B, 6, MPI_INTEGER, UP, 0, MPI_COMM_WORLD, IERR)
    call MPI_Wait(IREQ, STATUS, IERR)
call ga_sync()
call ga_put(g_a, 1, 6, ME+1, ME+1, T, 6)
call ga_print(g_a)
```

DISEÑO DE MEMORIA TÍPICA: MUESTRA QUE EL SISTEMA OPERATIVO
COMPARTE LA MEMORIA CON LOS PROGRAMAS EN EJECUCIÓN

MEMORIA



PROGRAMA



PROGRAMA



LENGUAJE
QUE ENTIENDA LA COMPUTADORA





LENGUAJES DE BAJO NIVEL

LENGUAJES MÁQUINA (0, 1)
UTILIZAN
ENSAMBLADORES
TRADUCEN EN CEROS Y
UNOS A
MNEMÓNICO (ADD A,6)



DIVERSIDAD DE LENGUAJES Y DIALECTOS



**C, C++, Basic, Cobol, SMALLTALK,
PASCAL, JAVA, PROLOG, ASP, PHP,
ACTIONSCRIPT, LINGO, ETC**

LENGUAJES DE ALTO NIVEL

LENGUAJES DE MÁQUINA

ALGOL

PROLOG

COBOL

RPG

PASCAL

SNOBOL

C

OBJECTIVE C

SMALTALK

BASIC →

APL

EJ: DIALECTOS

QUICK BASIC
QBASIC
VISUAL BASIC

ENSAMBLADOR

PL/I

LISP

FORTH

RPG II

OBJETC PASCAL

ADA

C++ →

FORTRAN

IEFFEL

VISUAL BASIC

APL

Ej: DIALECTOS

ACTION SCRIPT
LINGO
ETC...

ASCII

- Las diferentes combinaciones de ocho ceros y unos. 00000000 11111111
- Se asigna a cada una de las combinaciones un carácter diferente.
- Con los resultado se crea la tabla **ASCII**

carácter	Cod. ASCII	decimal
Espacio	00100000	32
0	00110000	48
1	00110001	49
2	00110010	50
A	01000001	65
B	01000010	66
A	01100001	97

8 BITS = 1 CARÁCTER

Ejemplo tabal ASCII

Aritmética binaria:

LAS COMPUTADORAS SOLO SUMAN
NO PUEDEN RESTAR



Pero es capaz de **negar números**. Es decir, puede tomar el negativo del número

$$42 + (-6)$$

Aritmética binaria:

LAS COMPUTADORAS SOLO SUMAN
NO PUEDEN RESTAR



Pero es capaz de **negar números**. Es decir, puede tomar el negativo del número

$$42 + (-6)$$



También puede simular la **multiplicación** y la **división**

$$6 \times 7 \quad \rightarrow \quad 6 + 6 + 6 + 6 + 6 + 6 + 6$$

Para **dividir** 42 entre 7, resta 7 de 42 (suma el negativo de 7 a 42) hasta llegar a 0 y cuenta la cantidad de veces **(6)** que lo hizo hasta llegar a 0.

$$42 + (-7) + (-7) + (-7) + (-7) + (-7) + (-7)$$

Aritmética binaria:

Relación entre el número de bits y el número de combinaciones binarias

Nº	1bits
0	0
1	1

$$2^1 = 2$$

Nº	2bits
0	00
1	11

$$2^2 = 4$$

Nº	3bits
0	000
1	001
2	010
3	011
4	100
5	101
7	111

$$2^3 = 8$$

$$2^1 = 2$$

$$2^2 = 4$$

$$2^3 = 8$$

$$2^4 = 16$$

$$2^5 = 32$$

$$2^6 = 64$$

$$2^7 = 128$$

$$2^8 = 256$$

$$2^9 = 512$$

$$2^{10} = 1024 = 1k$$

$$2^{20} = 1048,576 = 1M$$

$$2^{30} = 1G$$

Con n bits podemos obtener 2^n combinaciones o número diferentes

Tabla de conversión entre decimal, binario, hexadecimal y octal

4bits

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

Octal: Sistema numerico en base 8 se llama **octal** y utiliza los digitos 0 a 7.

Aritmética binaria:

Paso de binario a decimal el número de **8 bits** : 10110111

128	64	32	16	8	4	2	1
1	0	1	1	0	1	1	1

$$128 + 0 + 32 + 16 + 0 + 4 + 2 + 1$$



183

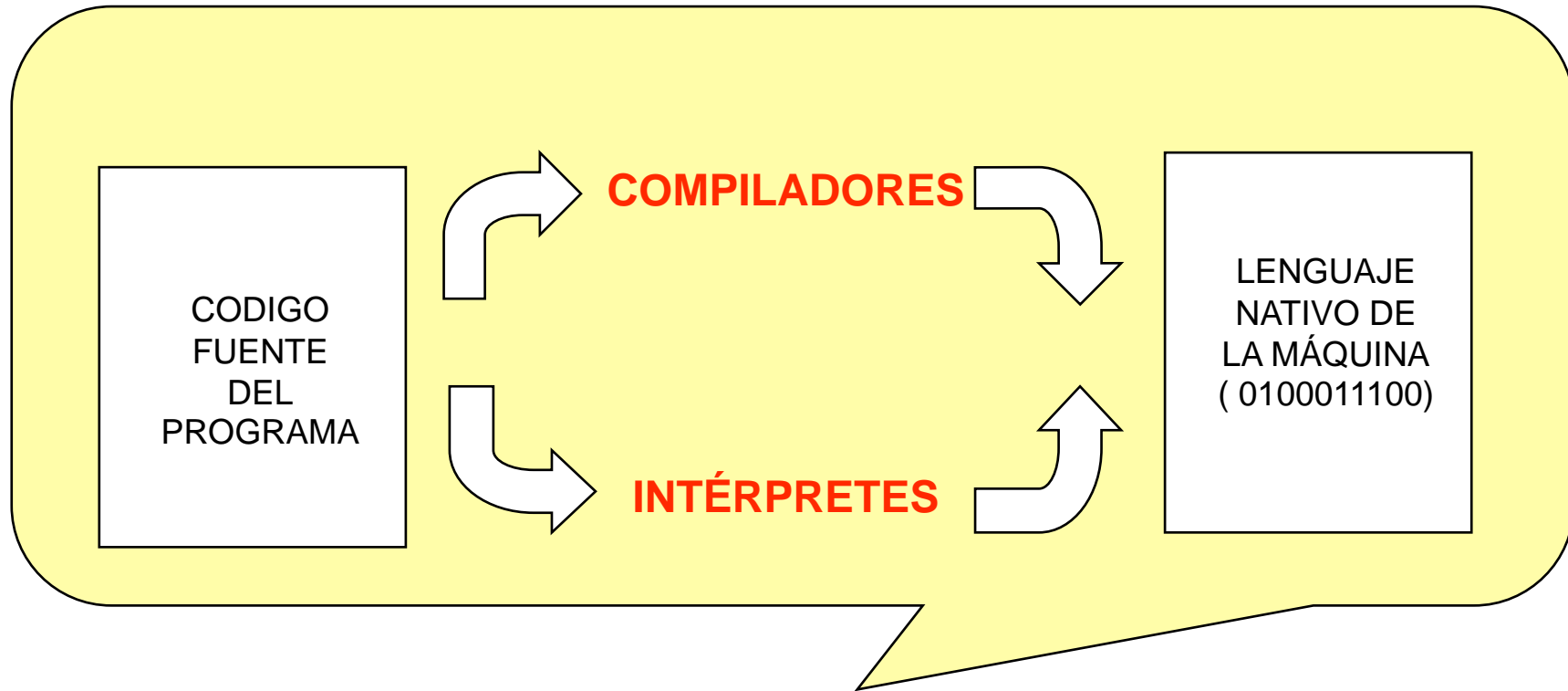


LA COMPUTADORA **NO** PUEDE ENTENDER , EL LENGUAJE



LA COMPUTADORA **NO** PUEDE ENTENDER , EL LENGUAJE

NECESITA DE TRADUCTORES



NECESITA DE TRADUCTORES

Libro en otro idioma

UNA LINEA CADA VEZ



INTERPRETE

TRADUCCIÓN SIMULTANEA



USUARIO

Libro en otro idioma

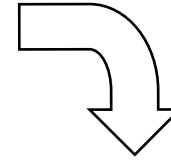
UNA LINEA CADA VEZ



INTERPRETE

TRADUCCIÓN SIMULTANEA

USUARIO



COMPILADOR



LIBRO TOTALMENTE TRADUCIDO

USUARIO

C

```
#include <stdio.h>
main()
{
printf("hola mundo");
}
```



**NECESITA COMPILARLO
FICHERO FINAL = NOMBREFICHERO.EXE**

HTML

```
<head>
<title>Página de hola
mundo</title>
</head>
<body>
hola mundo
</body>
```



NECESITA INTERPRETE

VELOCIDAD + MEMORIA IMPORTANTAN



VELOCIDAD + MEMORIA IMPORTAN

VELOCIDAD

LA VELOCIDAD DE LA CPU SE MIDE EN MEGAHERTZ (MHZ).

LA VELOCIDAD DETERMINA LA CANTIDAD DE CICLOS DE INSTRUCCIONES QUE PUEDE EFECTUAR LA CPU EN UN SEGUNDO.

MEGA = 1 MILLÓN -> UNA CPU DE 400 MHZ = 400 MILLONES DE INSTRUCCIONES X SEGUNDO

MEMORIA

RAM (Memoria de Acceso Aleatorio)

DURANTE LA EJECUCIÓN DE LOS PROGRAMAS, LA COMPUTADORA LOS ALMACENA EN LA RAM.

LA RAM ES VOLATIL ES DECIR, CUANDO APAGA LA COMPUTADORA, SU CONTENIDO SE BORRA AUTOMATICAMENTE

+ RAM = MEJOR GESTIÓN DEL SOFTWARE

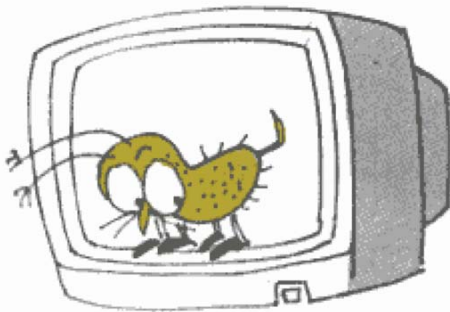
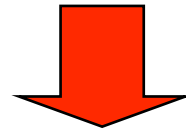
DISEÑO DE UN PROGRAMA

DISEÑO DE UN PROGRAMA

UN PROGRAMA ES **UNA LISTA** DE
INSTRUCCIONES **DETALLADAS** QUE SIGUE
LA COMPUTADORA

DISEÑO DE UN PROGRAMA

UN PROGRAMA ES UNA LISTA DE INSTRUCCIONES **DETALLADAS** QUE SIGUE LA COMPUTADORA



AL FALLO DE ESTA LISTA SE LE DENOMINA

“**BUG**” O FALLA (ERROR DEL PROGRAMA)

DISEÑO DE UN PROGRAMA



Es interesante construir un tipo de diseño descendente:

**Es el proceso de descomposición de un problema en detalles
Más específicos, hasta completar todos los detalles.**

DISEÑO DE UN PROGRAMA

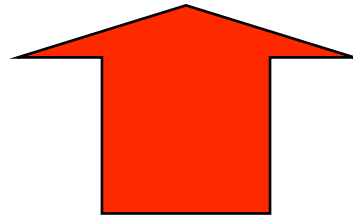


Es interesante construir un tipo de diseño descendente:

**Es el proceso de descomposición de un problema en detalles
Más específicos, hasta completar todos los detalles.**

PASOS DE DISEÑO:

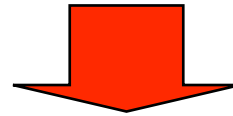
- Determinar la meta general
- Ir determinando los pasos siguientes intentando no olvidar nada.



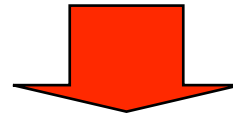
DIAGRAMAS DE FLUJOS
DISEÑO DE LA INTERACCIÓN

DIAGRAMAS DE FLUJOS

EL DIAGRAMA DE FLUJOS **NO INCLUYE TODOS LOS DETALLES DEL PROGRAMA**, PERO REPRESENTA EL FLUJO LÓGICO GENERAL DE L PROGRAMA.

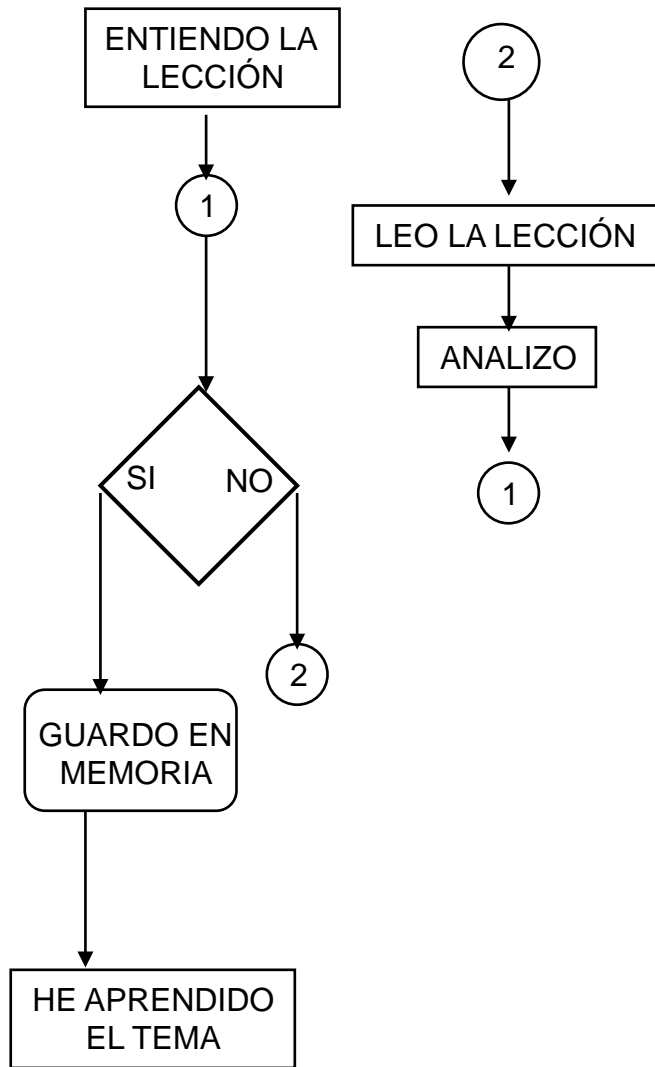


PROVEE LA LÓGICA DEL PROGRAMA FINAL



SI EL DIAGRAMA DE FLUJOS ESTÁ CORRECTAMENTE DIBUJADO, LA ESCRITURA DEL PROGRAMA (ES LO ÚLTIMO QUE HAREMOS), SE CONVERTIRÁ EN UN ASUNTO DE RUTINA

DIAGRAMAS DE FLUJOS



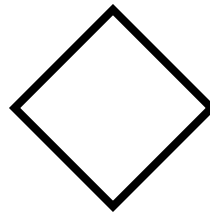
Flecha. Indica el sentido y trayectoria del proceso de información o tarea.



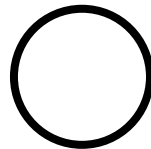
Rectángulo. Se usa para representar un evento o proceso determinado. Éste es controlado dentro del diagrama de flujo en que se encuentra. Es el símbolo más comúnmente utilizado.



Rectángulo redondeado. Se usa para representar un evento que ocurre de forma automática y del cuál generalmente se sigue una secuencia determinada.

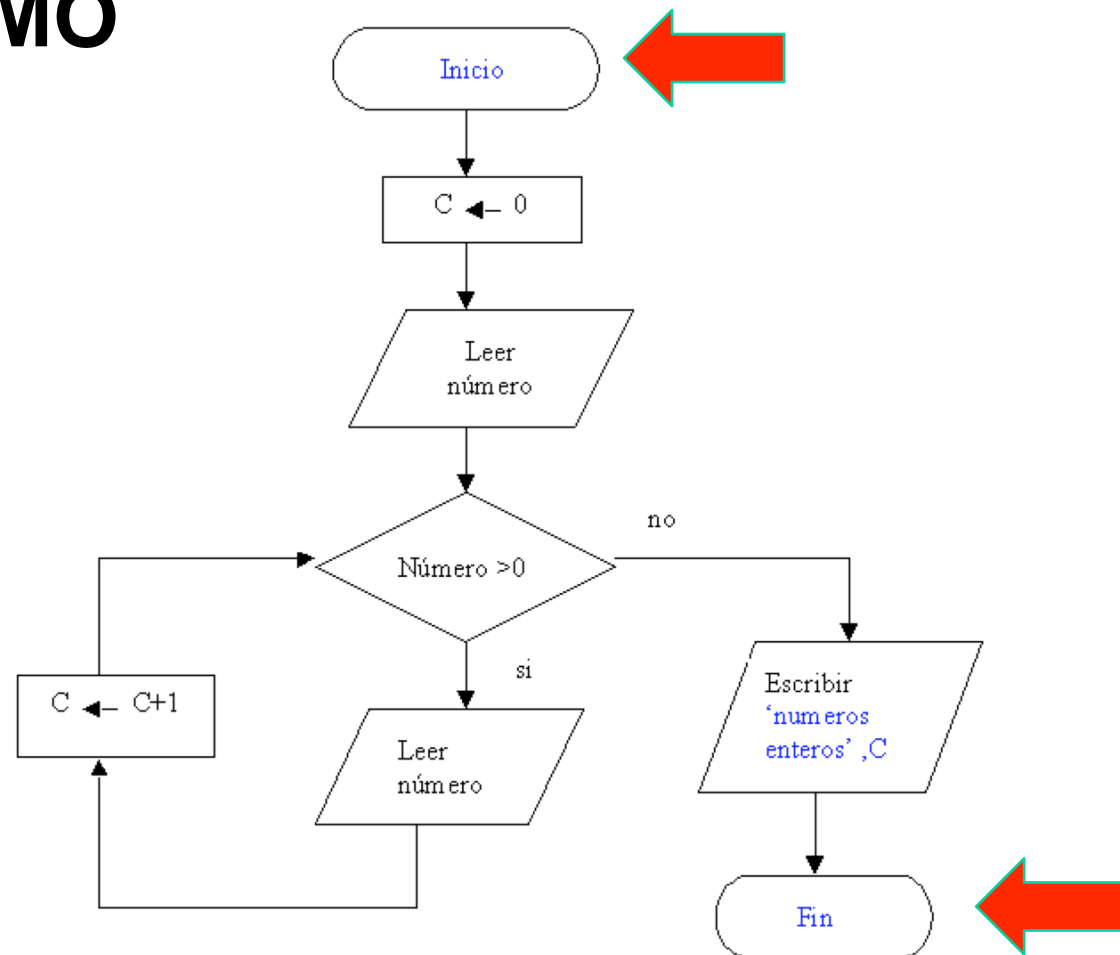


Rombo. Se utiliza para representar una condición. Normalmente el flujo de información entra por arriba y sale por un lado si la condición se cumple o sale por el lado opuesto si la condición no se cumple. Lo anterior hace que a partir de éste el proceso tenga dos caminos posibles.



Círculo. Representa un punto de conexión entre procesos. Se utiliza cuando es necesario dividir un diagrama de flujo en varias partes, por ejemplo por razones de espacio o simplicidad. Una referencia debe de darse dentro para distinguirlo de otros. La mayoría de las veces se utilizan números en los mismos.

ALGORITMO



Un algoritmo es una sucesión finita de pasos no ambiguos que se pueden llevar a cabo en un tiempo finito

PROGRAMACIÓN **ESTRUCTURADA**

**PARA CONSTRUIR UN PROGRAMA NECESITAMOS UN INICIO Y UN FIN
+ ASIGNAR VALORES A ELEMENTOS VARIABLES QUE FORMARÁN PARTE DE NUESTRO PROGRAMA**

ASIGNACIÓN:

Asignaciones Particulares :

VAR PEPE = 30

VAR JUANA = 26

VAR TONI = 28

VAR ANA = 31

VAR PEPE = HOMBRE

VAR JUANA = MUJER

VAR TONI = HOMBRE

VAR ANA = MUJER

VAR PEPE = SOLTERO

VAR JUANA = CASADA

VAR TONI = CASADO

VAR ANA = SOLTERA

Asignaciones Generales y globales:

VAR GLOBAL ESTADO CIVIL = CASADO Ó SOLTERO

VAR GLOBAL GÉNERO = HOMBRE O MUJER

VAR GLOBAL EDAD DEL CLUB = 30

PROGRAMACIÓN **ESTRUCTURADA**

**PARA CONSTRUIR UN PROGRAMA NECESITAMOS UN INICIO Y UN FIN
+ ASIGNAR VALORES A ELEMENTOS VARIABLES QUE FORMARÁN PARTE DE NUESTRO PROGRAMA**

ASIGNACIÓN:

Asignaciones Particulares :

VAR PEPE = 30

VAR JUANA = 26

VAR TONI = 28

VAR ANA = 31

VAR PEPE = HOMBRE

VAR JUANA = MUJER

VAR TONI = HOMBRE

VAR ANA = MUJER

VAR PEPE = SOLTERO

VAR JUANA = CASADA

VAR TONI = CASADO

VAR ANA = SOLTERA

Asignaciones Generales y globales:

VAR GLOBAL ESTADO CIVIL = CASADO Ó SOLTERO

VAR GLOBAL GÉNERO = HOMBRE O MUJER

VAR GLOBAL EDAD DEL CLUB = 30

Con estos valores podríamos UN PROGRAMA / ALGORITMO que nos permita saber si puede entrar nuestras VARIABLES a una club específico de TREINTAÑEROS, pertenecer a un CLUB DE CASADOS o SOLTEROS, BUSCAR PAREJA (BAJO LAS REGLAS QUE MARQUEMOS), ETC....

PROGRAMACIÓN ESTRUCTURADA

Asignaciones Particulares :

VAR PEPE = 30	VAR JUANA = 26	VAR TONI = 28	VAR ANA = 31
VAR PEPE = HOMBRE	VAR JUANA = MUJER	VAR TONI = HOMBRE	VAR ANA = MUJER
VAR PEPE = SOLTERO	VAR JUANA = CASADA	VAR TONI = CASADO	VAR ANA = SOLTERA



Asignaciones Generales y globales:

VAR GLOBAL GÉNERO = TIPO DE GÉNERO
VAR GLOBAL EDAD DEL CLUB = 30
VAR GLOBAL ESTADO CIVIL = CASADO Ó SOLTERO

RESULTADO ASIGNACIÓN

PROGRAMACIÓN **ESTRUCTURADA**

PARA CONSTRUIR UN PROGRAMA NECESITAMOS ELEMENTOS DENOMINADOS CONSTRUCTORES

CONSTRUCTOR:

Un constructor es un bloque de construcción de un lenguaje y una de las operaciones fundamentales de éste.

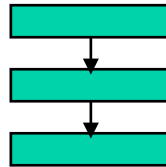
Los principales constructores de un programa son:

SECUENCIA + SELECCIÓN + CICLO

PROGRAMACIÓN **ESTRUCTURADA**

INCLUYE LOS SIGUIENTES CONSTRUCTORES:

SECUENCIA



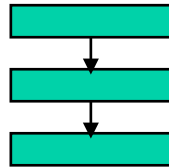
Dos o más instrucciones, una tras otra.

OBTENER LAS HORAS DEL PROYECTO
SUMARLE LAS HORAS DEL DIA
RESTAR LAS QUE DUERMO
DECIRSELO AL PROFESOR

PROGRAMACIÓN **ESTRUCTURADA**

INCLUYE LOS SIGUIENTES CONSTRUCTORES:

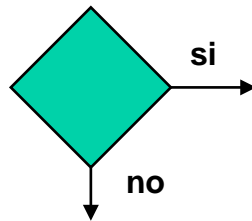
SECUENCIA



Dos o más instrucciones, una tras otra.

OBTENER LAS HORAS DEL PROYECTO
SUMARLE LAS HORAS DEL DIA
RESTAR LAS QUE DUERMO
DECIRSELO AL PROFESOR

SELECCIÓN



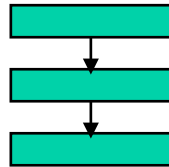
Punto donde el programa debe de tomar una decisión , controlada.

Si soy mayor de 18 años
Puedo entrar
Si no los soy
No puedo entrar

PROGRAMACIÓN **ESTRUCTURADA**

INCLUYE LOS SIGUIENTES CONSTRUCTORES:

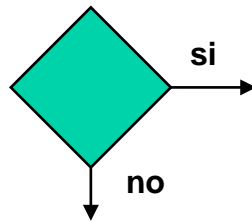
SECUENCIA



Dos o más instrucciones, una tras otra.

OBTENER LAS HORAS DEL PROYECTO
SUMARLE LAS HORAS DEL DIA
RESTAR LAS QUE DUERMO
DECIRSELO AL PROFESOR

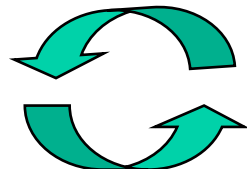
SELECCIÓN



Punto donde el programa debe de tomar una decisión , controlada.

Si soy mayor de 18 años
Voy de fiesta
Si no los soy
Voy a casa

CICLOS



Reiteraciones, acciones que se repiten constantemente hasta que se rompen
Por una variación en los datos

Mientras haya gente
Poner musica
De otro modo
Apaga el equipo
y vamos



PRÁCTICA CONSTRUIR EL PROGRAMA SPACE INVADERS MEDIANTE DIAGRAMA DE FLUJOS



UNIVERSIDAD
POLITECNICA
DE VALENCIA



Prof. Moisés Mañas Carbonell
Dpto. Escultura. UPV
moimacar@esc.upv.es