

1. Introducción a la programación

Fundamentos de Informática

Especialidad de Electrónica – 2009-2010

Ismael Etxeberria Agiriano

13/10/2009



Escuela Universitaria
de Ingeniería
Vitoria-Gasteiz

Ingeniaritzako
Unibertsitate Eskola
Vitoria-Gasteiz



Universidad
del País Vasco

Euskal Herriko
Unibertsitatea

Índice

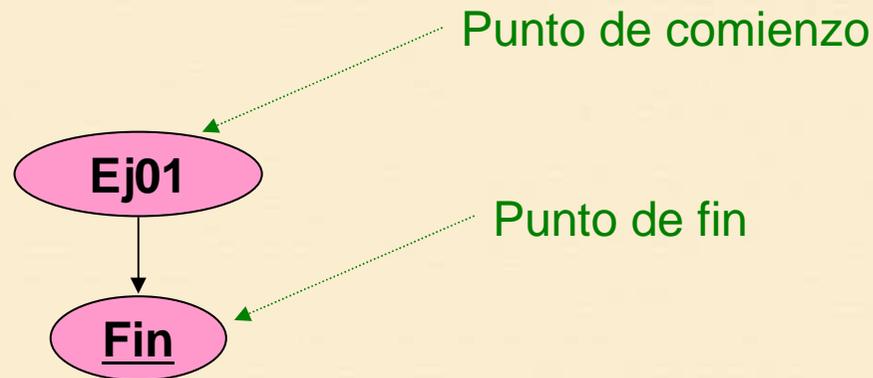
1. Introducción a la programación

1. Ej01: Programa vacío
2. Ej02: Saludo
3. Ej03: Asignación
4. Ej04: Lectura
5. Ej05: Suma
6. Resumen
7. Constantes
8. Formatos stdio
9. Ejercicios

1. Ejemplo 01

- **Título:**
 - Programa vacío
- **Nombre**
 - Ej01
- **Descripción**
 - Programa que no hace nada
- **Observaciones**
 - Comienzo y fin de un programa
 - Codificación teórica y en laboratorio
 - Patrón básico \emptyset

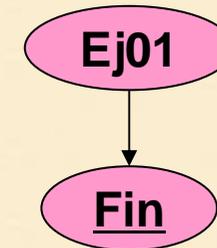
Ej01: Diagrama de Flujo



- **Punto de comienzo**
 - Nombre del programa
 - Nombre del fichero
- **Punto de fin**
 - Único para cada diagrama de flujo
 - Nomenclatura: Fin

Ej01: Codificación C

```
/* Ej01 (papel) */  
void main (void)  
{  
}
```



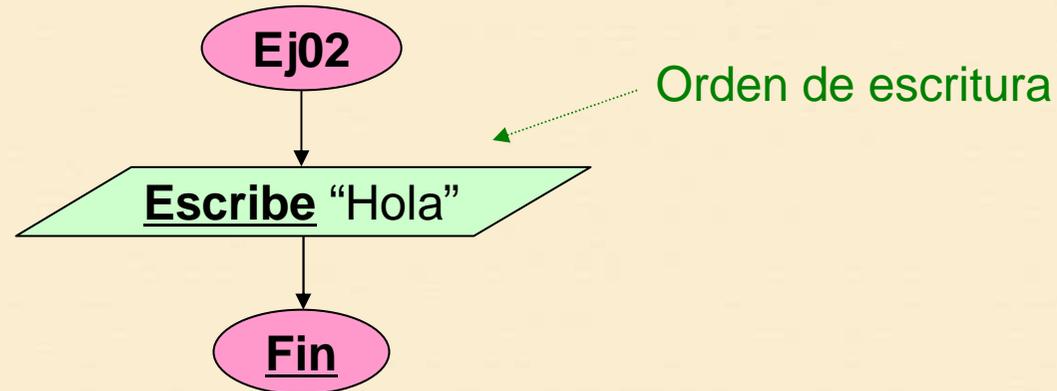
- Para poder probar los programas en las prácticas pondremos una **pausa** y haremos que `main` devuelva un 0.

```
/* Ej01 (laboratorio) */  
#include <stdlib.h>  
  
int main (void)  
{  
    system("pause");  
    return 0;  
}
```

2. Ejemplo 02

- **Título:**
 - Saludo
- **Nombre**
 - Ej02
- **Descripción**
 - Programa que saluda
- **Observación**
 - Escritura en pantalla de un texto literal

Ej02: Diagrama de Flujo



- La orden de escritura vendrá expresada en el DdF como **Escribe**
- En general no detallaremos todo lo que se escribe en el diagrama de flujo. Para ello puede haber anotaciones aparte junto al diagrama de flujo.

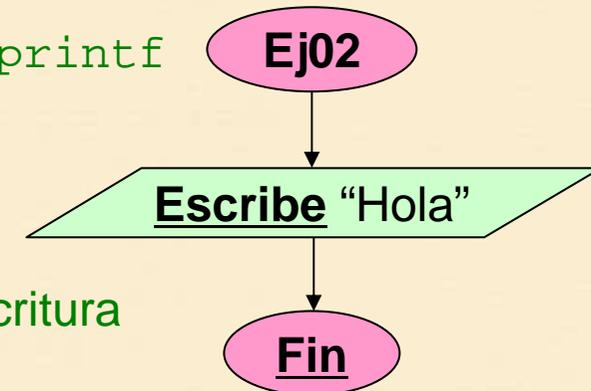
Ej02: Codificación C

```
/* Ej02 */
#include <stdio.h>

void main (void)
{
    printf ("Hola mundo\n");
}
```

Prototipo de `printf`

Orden de escritura

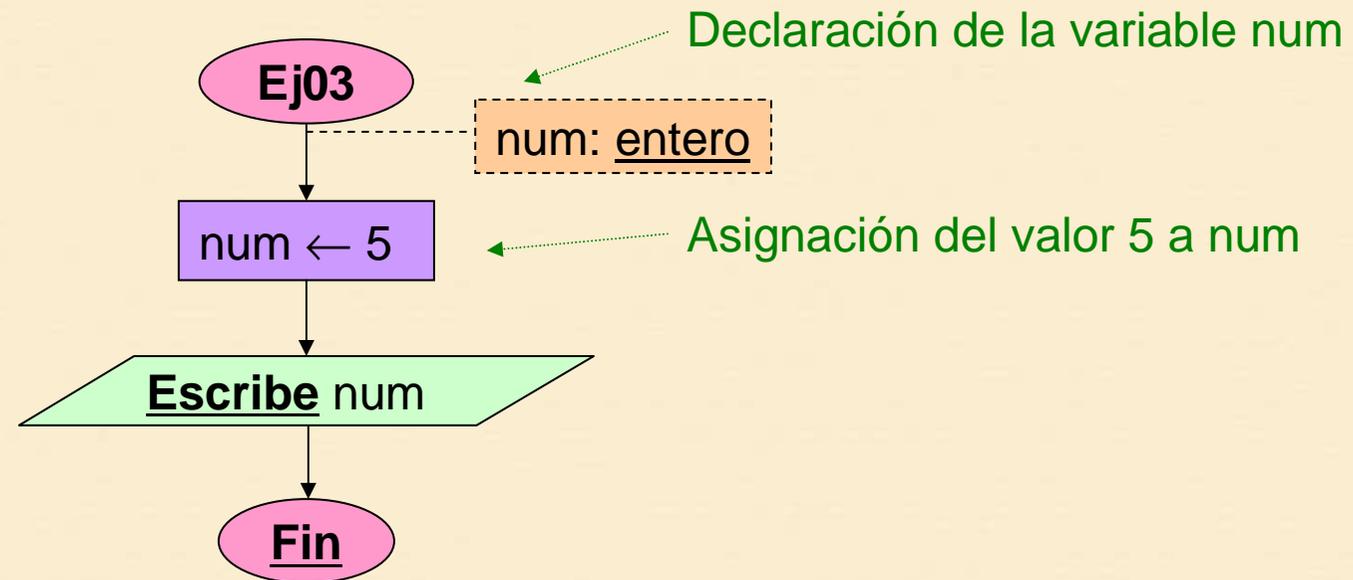


- Para escritura en C utilizaremos la función `printf` cuyo prototipo está definido en el fichero `stdio.h`.
- Veremos más en detalle las múltiples posibilidades de la función `printf`.
- La secuencia `\n` representa un salto de línea (*newline*)

3. Ejemplo 03

- **Título**
 - Asignación
- **Nombre**
 - Ej03
- **Descripción**
 - Definir una variable entera, asignarle un valor cualquiera y mostrarlo en pantalla
- **Observaciones**
 - Declaración de variables
 - Asignación de un valor a una variable

Ej03: Diagrama de Flujo



- Hay que declarar todas las variables del programa, indicando de qué *tipo* son
- Subrayamos el tipo para diferenciarlo de los nombres
- Mediante la *asignación* una variable recibe un valor

Ej03: Codificación C

```

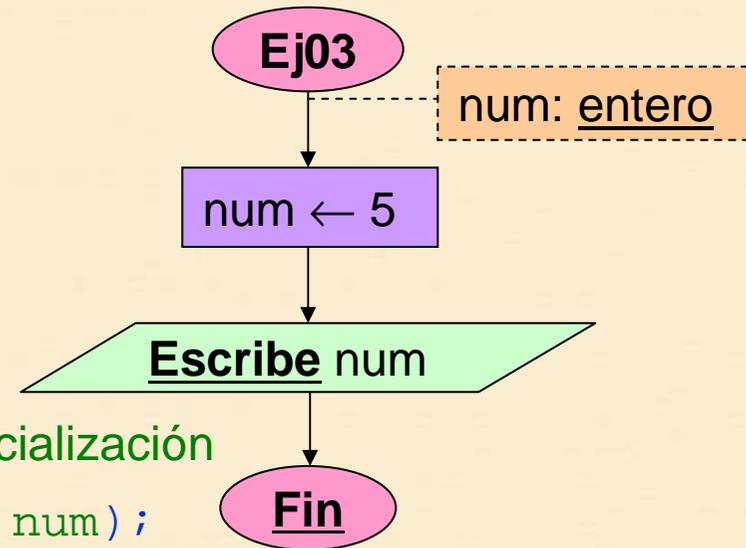
/* Ej03 */
#include <stdio.h>
void main (void)
{
    int num;
    num = 5;
    printf ("Número: %d\n", num);
}

```

Declaración

Asignación/Inicialización

Mostrar valor

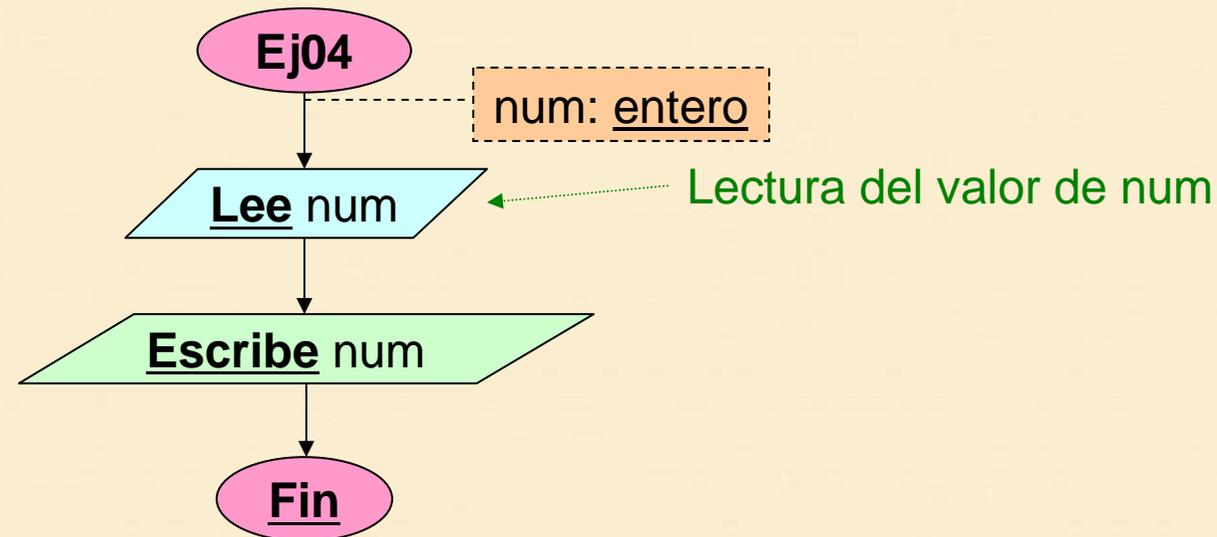


- Para declarar una variable en C especificaremos el tipo (`int` para un dato entero) y a continuación el nombre
- La asignación en C se expresa mediante el símbolo `=`
- En la **parte izquierda** de la asignación siempre irá una variable y en la **derecha** una expresión que se evaluará
- No hay que confundir con la igualdad matemática

4. Ejemplo 04

- **Título**
 - Lectura
- **Nombre**
 - Ej04
- **Descripción**
 - Leer un número entero y mostrarlo en pantalla
- **Observaciones**
 - Lectura de un valor

Ej04: Diagrama de Flujo



- En vez de asignarle un valor fijo como en el programa Ej03 ahora leemos el valor del teclado
- La orden de lectura vendrá expresada como **Lee**
- Luego escribimos el valor leído en pantalla
- Nótese que no detallamos cómo lo escribimos

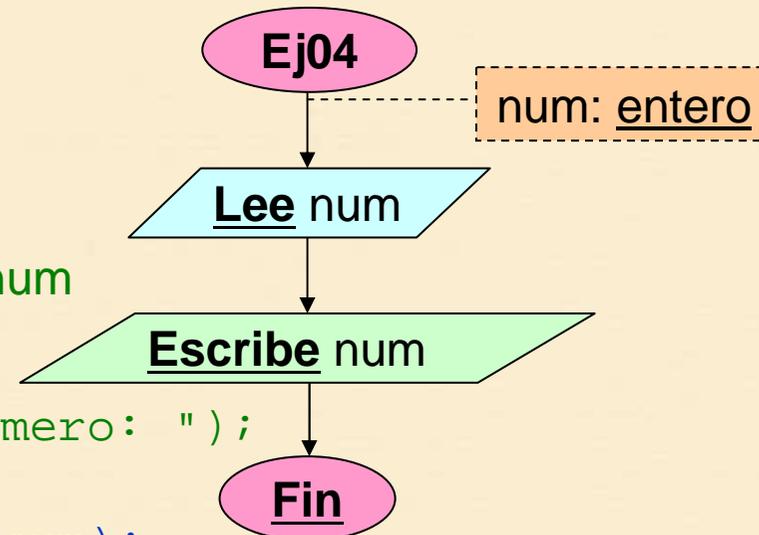
Ej04: Codificación C

```

/* Ej04 */
#include <stdio.h>
void main (void)
{
    int num;
    printf ("Introduce un número: ");
    scanf ("%d", &num);
    printf ("Número: %d\n", num);
}

```

Lectura de num

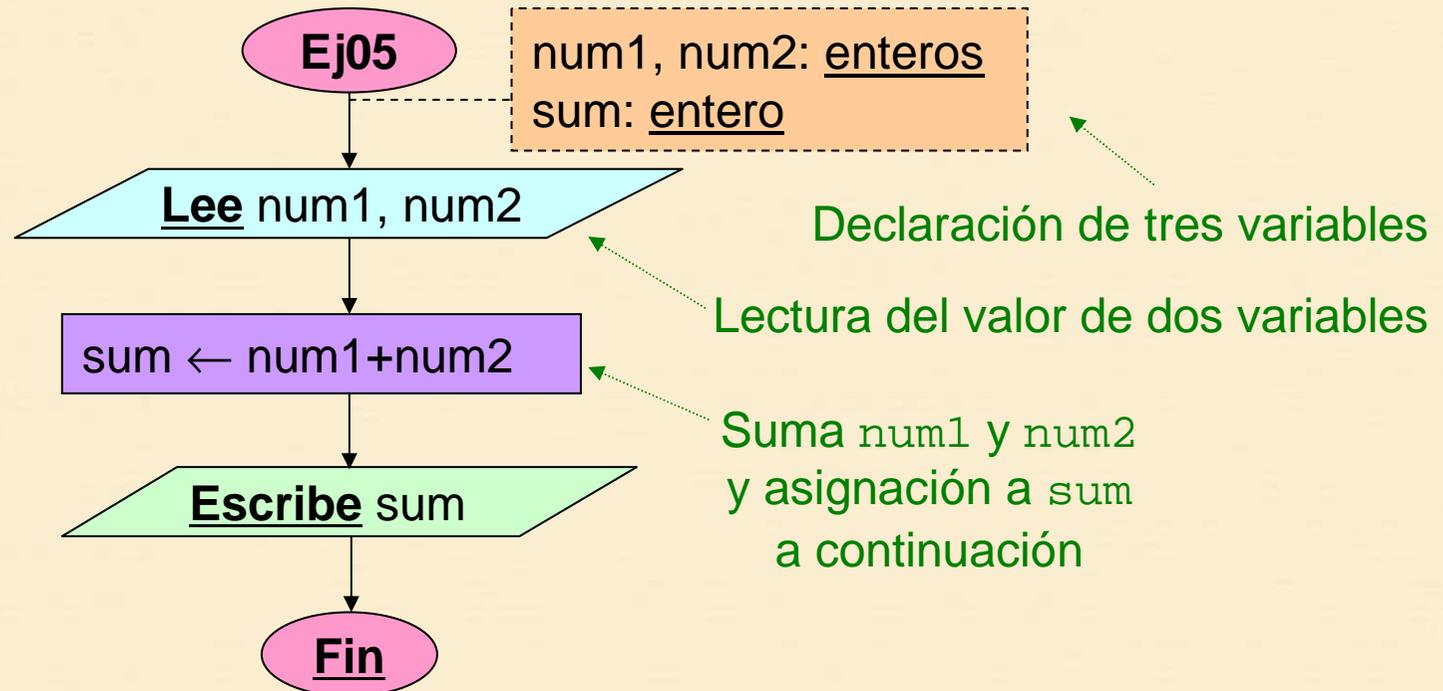


- Para la lectura en C utilizaremos la función `scanf` cuyo prototipo está definido en el fichero `stdio.h`.
- En C la orden **Lee** se corresponde con una escritura de petición del dato (primer `printf`) y una lectura del dato (función `scanf`), dándole la dirección del dato (`&num`)
- El olvido del `&` en `scanf` es un error típico

5. Ejemplo 05

- **Título**
 - Suma
- **Nombre**
 - Ej05
- **Descripción**
 - Leer dos números enteros, obtener su suma y mostrarla en pantalla
- **Observaciones**
 - Declaración múltiple de variables
 - Lectura de dos valores
 - Expresión: suma de dos variables

Ej05: Diagrama de Flujo



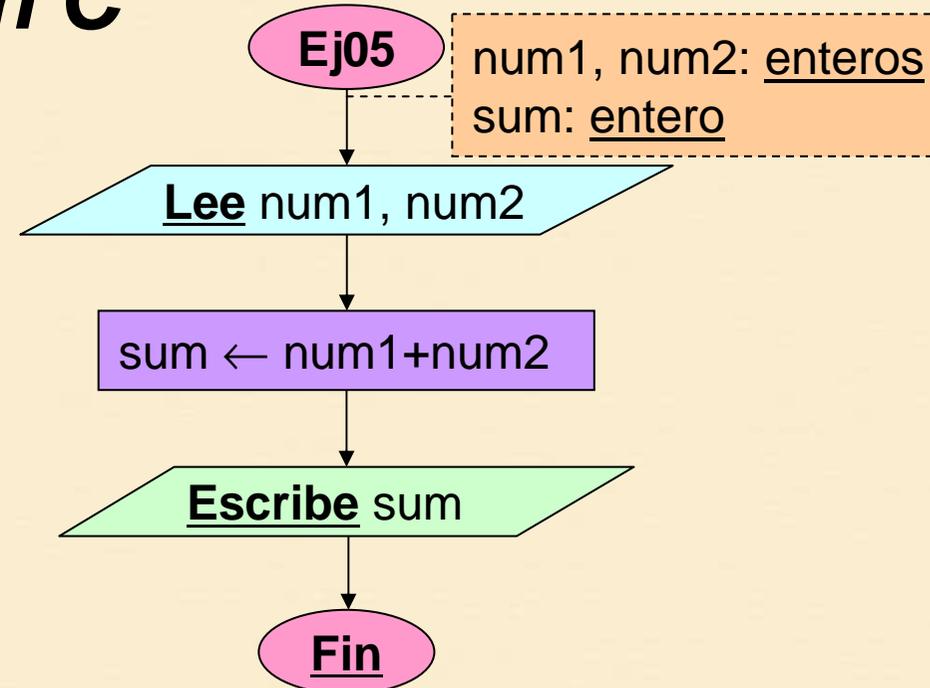
- La variable `sum` recibe el resultado de calcular la expresión `num1+num2`
- Escribimos el resultado, `sum`

Ej05: Codificación C

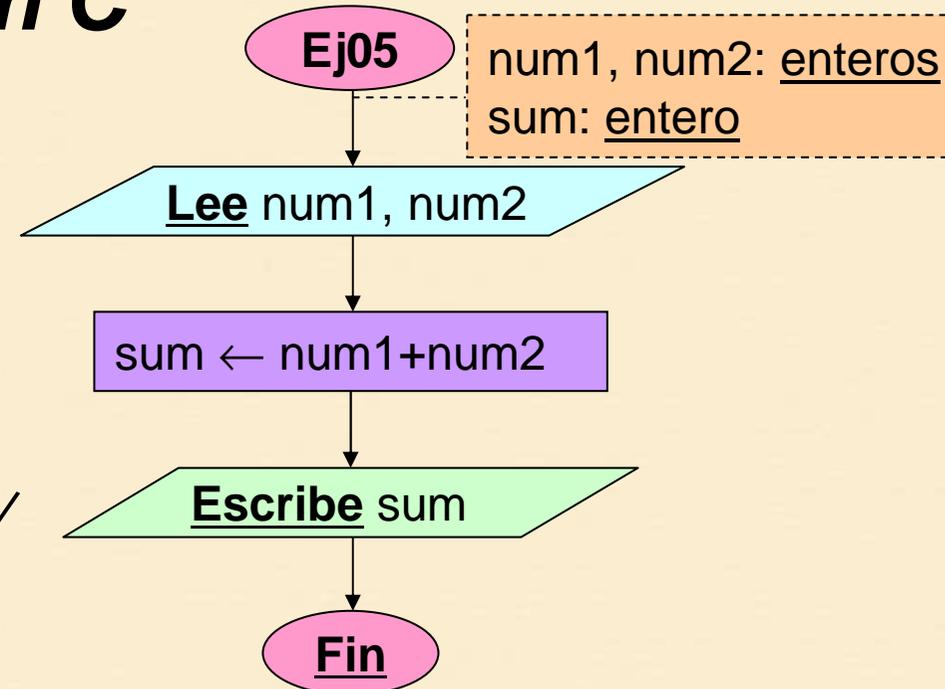
```

/* Ej05 */
#include <stdio.h>
void main (void)
{
    int num1, num2;
    int sum;
    printf ("Introduce un número: ");
    scanf ("%d", &num1);
    printf ("Introduce otro número: ");
    scanf ("%d", &num2);
    sum = num1 + num2;
    printf ("Suma %d+%d: %d\n", num1, num2, sum);
}

```



Ej05: Codificación C



```

/* Ej05_01 (variante) */
#include <stdio.h>
void main (void)
{
    int num1, num2, sum;
    printf ("Introduce dos números: ");
    scanf ("%d%d", &num1, &num2);
    sum = num1 + num2;
    printf ("Suma %d+%d: %d\n", num1, num2, sum);
}
  
```



6. Resumen

```

graph TD
    Ej0x([Ej0x]) --> Fin([Fin])
    
```

```

void main (void)
{
}
    
```

```

#include <stdlib.h>
int main (void)
{
    system("pause");
    return 0;
}
    
```

n1, n2: enteros
sum: entero

```

int n1, n2;
int sum;
    
```

sum ← n1+n2

```

sum = n1 + n2;
    
```

Escribe "Hola"

```

#include <stdio.h>
...
printf ("Hola\n");
    
```

Lee n1, n2

```

#include <stdio.h>
...
printf ("Introduce: ");
scanf ("%d%d", &n1, &n2);
    
```

7. Constantes

```
ptas1Eur=166,386  
retencion=18
```

7.1 Macrodefiniciones (modelo tradicional)

```
#define ptas1Eur 166.386  
#define retencion 18
```

7.2 Datos constantes (modelo ANSI)

```
const double ptas1Eur= 166.386;  
const double retencion= 18;
```

8. Notación DdF vs notación C

Diagrama de flujo	Lenguaje C	Comentario
←	=	Asignación
+	+	Suma
-	-	Resta y cambio de signo
.	*	Producto
<u>Div</u>	/	Cociente división entera
<u>Mod</u>	%	Resto división entera
/	/	División real
<u>entero</u>	int	Entero
<u>real</u>	double	Real (doble precisión)
,	.	Coma (punto) decimal

8.1 Formatos de lectura (stdio)

Especifican qué se desea leer sobre qué tipo de variable, cuya dirección vendrá a continuación:

```
int n1, n2; /* Declaración de dos enteros */
double d1; /* Declaración de un real doble */
scanf ("%d%o%lf", &n1, &n2, &d1);
```

Formato scanf	Significado
%d	Entero decimal
%o	Entero octal
%x	Entero hexadecimal
%f	Real (precisión simple)
%lf	Real (largo o precisión doble)

8.2 Formatos de escritura (stdio)

Especifican cómo mostrar un dato proporcionado como argumento a continuación:

```
int n1, n2; double d1;  
printf ("%02d %o %lf", n1, n2, d1);
```

Formato printf	Significado
%4d	Entero de anchura 4 (rellenar de espacios)
%04d	Entero de anchura 4 (rellenar de 0s)
%8lf	Doble anchura 8 (con 6 decimales)
%.2lf	Real doble con dos decimales
%8.2lf	Real doble de anchura 8 y 2 decimales

9. Ejercicios

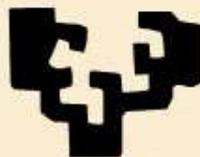
9.1. Diseña y codifica un programa en lenguaje C para pedir una **cantidad en euros**, pasarla a **pesetas** y mostrar el resultado.

9.2. Diseña y codifica un programa en lenguaje C para calcular el interés de una determinada **cantidad** a un **tipo anual** dado para un cierto **número de días** suponiendo que el año tiene 365 días exactos. Mostrar tanto el interés como lo que finalmente cobraremos si nos retienen un 18%.



Escuela Universitaria de Ingeniería Vitoria-Gasteiz Ingeniaritzako Unibertsitate Eskola Vitoria-Gasteiz

eman ta zabal zazu



Universidad del País Vasco Euskal Herriko Unibertsitatea