

# Tema 1: Introducción a la Programación

Oscar Perpiñán Lamigueiro

# ¿Por qué programar?

Tema 1:  
Introducción a la  
Programación

Oscar Perpiñán  
Lamigueiro

“Everybody in this country should  
learn how to program a computer...  
because it teaches you how to think.”

- Steve Jobs

What most schools don't teach

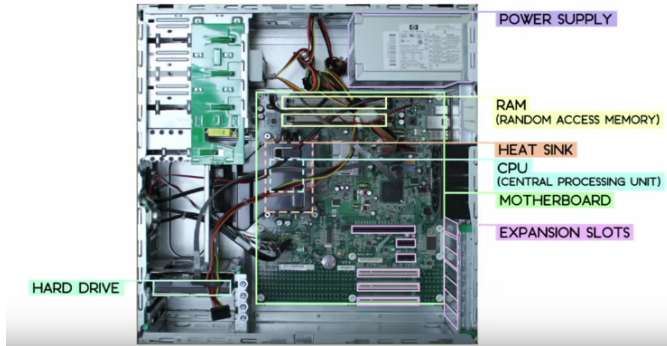
¿Por qué todos tienen que aprender a programar?

Code Stars

# ¿Qué hay dentro de un ordenador?

Tema 1:  
Introducción a la  
Programación

Oscar Perpiñán  
Lamigueiro



Inside a Computer

- ▶ Un **programa informático** es una colección de instrucciones expresadas de forma que un ordenador pueda resolver un determinado problema.
- ▶ Un **algoritmo** es un método para resolver un problema. Un programa informático implementa el algoritmo para un determinado sistema informático.
- ▶ Las instrucciones deben estar codificadas en **lenguaje binario** (sucesiones de 0s y 1s).

# Cómo saluda un ordenador

Tema 1:  
Introducción a la  
Programación

Oscar Perpiñán  
Lamigueiro

01001000 01101111 01101100 01100001

# Cómo saluda un ordenador

Tema 1:  
Introducción a la  
Programación

Oscar Perpiñán  
Lamigueiro

01001000 01101111 01101100 01100001

01001000	01101111	01101100	01100001
72	111	108	97
H	o	l	a

- ▶ Lenguaje máquina: los ordenadores utilizan el sistema de numeración binario (dos dígitos, 0 y 1) para almacenar información.
- ▶ Un dígito binario (0 ó 1) se denomina *bit* (*binary digit*).
- ▶ Con N bits pueden representarse  $2^N$  símbolos o  $2^N$  números
  - ▶ Ejemplo: con  $N = 8$  bits se pueden representar los números positivos desde el 0 al 255 ( $2^8 - 1$ ).

# Representación de la información: binario y decimal

Ejemplo en decimal: 3452

$10^3$	$10^2$	$10^1$	$10^0$
1000	100	10	1
<hr/>			
3	4	5	2

$$3452 = 3 \cdot 1000 + 4 \cdot 100 + 5 \cdot 10 + 2 \cdot 1$$

Ejemplo en binario: 10001111

$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
<b>128</b>	64	32	16	<b>8</b>	<b>4</b>	<b>2</b>	<b>1</b>
<hr/>							
<b>1</b>	0	0	0	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>

$$128 + 8 + 4 + 2 + 1 = 143$$



- ▶ Byte: 8 bits ( $2^8 = 256$ )
- ▶ Kilobyte (KB): 1024 bytes ( $2^{10} = 1024$ )
- ▶ Megabyte (MB): 1024 KB ( $2^{20}$  bytes =  $2^{10}$  KB)
- ▶ Gigabyte (GB): 1024 MB ( $2^{30}$  bytes =  $2^{10}$  MB =  $2^{20}$  KB)
- ▶ ...

# No sólo números

- ▶ Cualquier información puede representarse con un conjunto de bits.
- ▶ ASCII (American Standard Code for Information Interchange): Estándar de 7 bits (128 caracteres), 95 caracteres imprimibles (del 32 al 126).

01001000	01101111	01101100	01100001
72	111	108	97
H	o	l	a

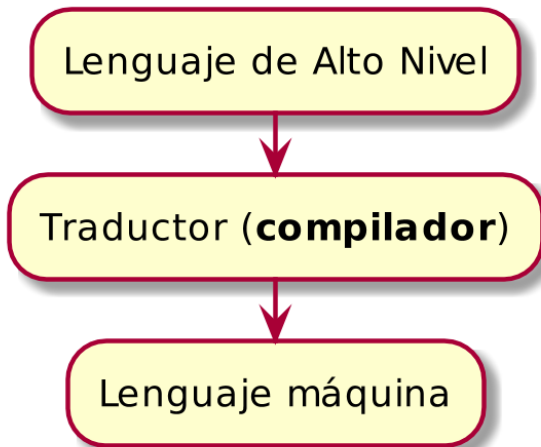
# ¿Qué es un lenguaje de programación?

Un lenguaje artificial que emplea **expresiones similares al lenguaje humano** y un **traductor** para convertir a código binario.

- ▶ Lenguaje de **alto nivel**: utiliza signos convencionales cercanos a los de un lenguaje natural.
- ▶ Lenguaje de **bajo nivel**: similar al lenguaje máquina.

## Ejemplo de lenguaje de bajo nivel: ensamblador

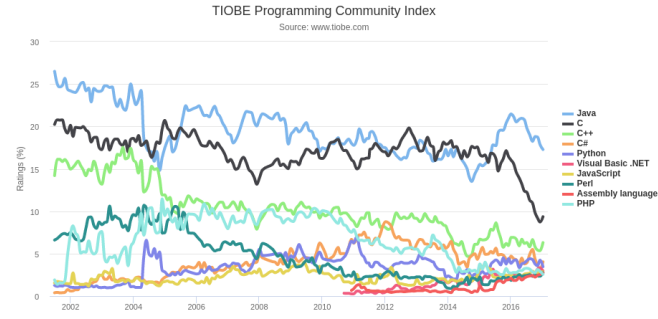
```
STACK          SEGMENT STACK
                DW 64 DUP (?)
STACK          ENDS
DATA           SEGMENT
SALUDO         DB "Hola Mundo",13,10,"$" ; Cadena
DATA           ENDS
CODE           SEGMENT
                ASSUME CS:CODE, DS:DATA, SS:STACK
INICIO:
                MOV AX,DATA
                MOV DS,AX
                MOV DX,OFFSET SALUDO
                MOV AH,09H
                INT 21H
                MOV AH,4CH
                INT 21H
CODE           ENDS
                END INICIO
```



# Lenguajes de alto nivel

Tema 1:  
Introducción a la  
Programación

Oscar Perpiñán  
Lamigueiro



<http://www.tiobe.com/tiobe-index/>

## Características

- ▶ Lenguaje de nivel *medio*.
- ▶ De propósito general
- ▶ Compacto (sólo 32 palabras)
- ▶ Estructurado. Permite reutilizar el código.
- ▶ Funciona en plataformas diferentes.

## Historia

- ▶ C (Ritchie, 1972. Laboratorios Bell).
- ▶ ANSI C American National Standards Institute C (1989).
- ▶ C99 (ISO/IEC 9899, 1999).

# Ejemplo: programa escrito en lenguaje C

Tema 1:  
Introducción a la  
Programación

Oscar Perpiñán  
Lamigueiro

```
#include <stdio.h>

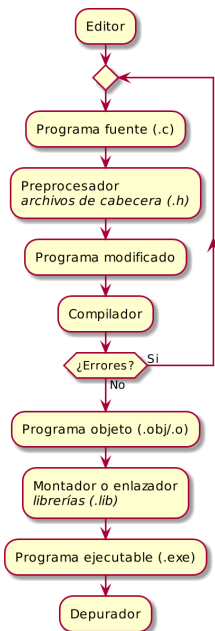
void main()
{
    printf("Hola Mundo\n");
}
```

[https://es.wikipedia.org/wiki/Anexo:Ejemplos\\_de\\_implementaci%C3%B3n\\_del\\_%C2%ABHola\\_mundo%C2%BB](https://es.wikipedia.org/wiki/Anexo:Ejemplos_de_implementaci%C3%B3n_del_%C2%ABHola_mundo%C2%BB)

# Desarrollo de programas en C

Tema 1:  
Introducción a la  
Programación

Oscar Perpiñán  
Lamigueiro





## Extraído de [Best Practices for Scientific Computing](#)

- ▶ Write programs for people, not computers.
- ▶ Automate repetitive tasks
- ▶ Use the computer to record history
- ▶ Make incremental changes
- ▶ Use version control
- ▶ Don't repeat yourself (or others)
- ▶ Plan for mistakes
- ▶ Optimize software only after it works correctly
- ▶ Document design and purpose, not mechanics
- ▶ Collaborate