

# Fundamentos de la programación

---

## 1

# Computadoras y programación

Grado en Ingeniería Informática  
Grado en Ingeniería del Software  
Grado en Ingeniería de Computadores

Luis Hernández Yáñez  
Facultad de Informática  
Universidad Complutense



# Índice

---

Informática, computadoras y programación	3
Lenguaje máquina y ensamblador	12
Lenguajes de programación de alto nivel	15
Un poco de historia	19
Programación e Ingeniería del Software	24
El lenguaje de programación C++	27
Sintaxis de los lenguajes de programación	30
Un primer programa en C++	35
Herramientas de desarrollo	39
C++: Un mejor C	45



## Informática, computadoras y programación

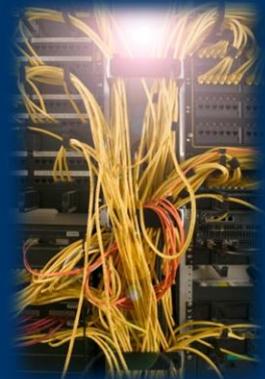


## *Informática (Ciencia de la computación)*

Conjunto de conocimientos científicos y técnicas que hacen posible el tratamiento automático de la información por medio de ordenadores

## *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 **ejecución** de **programas** informáticos



# Computadoras

*En todas partes y con muchas formas*



# Hardware y software

## *Hardware*

Componentes que integran la parte material de una computadora



## *Software*

Programas, instrucciones y reglas informáticas para ejecutar tareas en una computadora



# Programación de computadoras

---

## *Programar*

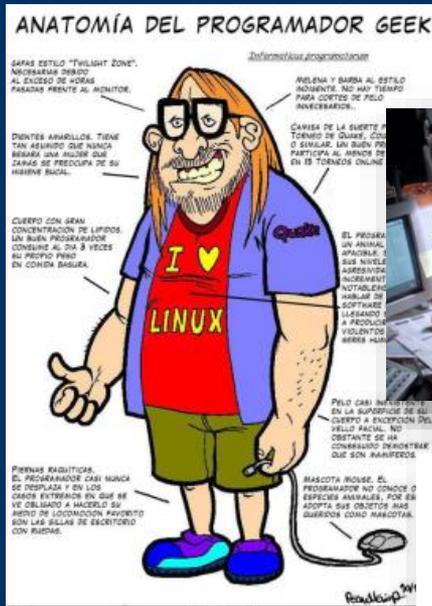
Indicar a la computadora qué es lo que tiene que hacer

## *Programa*

- ✓ Secuencia de instrucciones
- ✓ Instrucciones que entiende la computadora
- ✓ Y que persiguen un objetivo: *¡resolver un problema!*



# Programadores



Parque Jurásico



## Trabajo en equipo Múltiples roles...

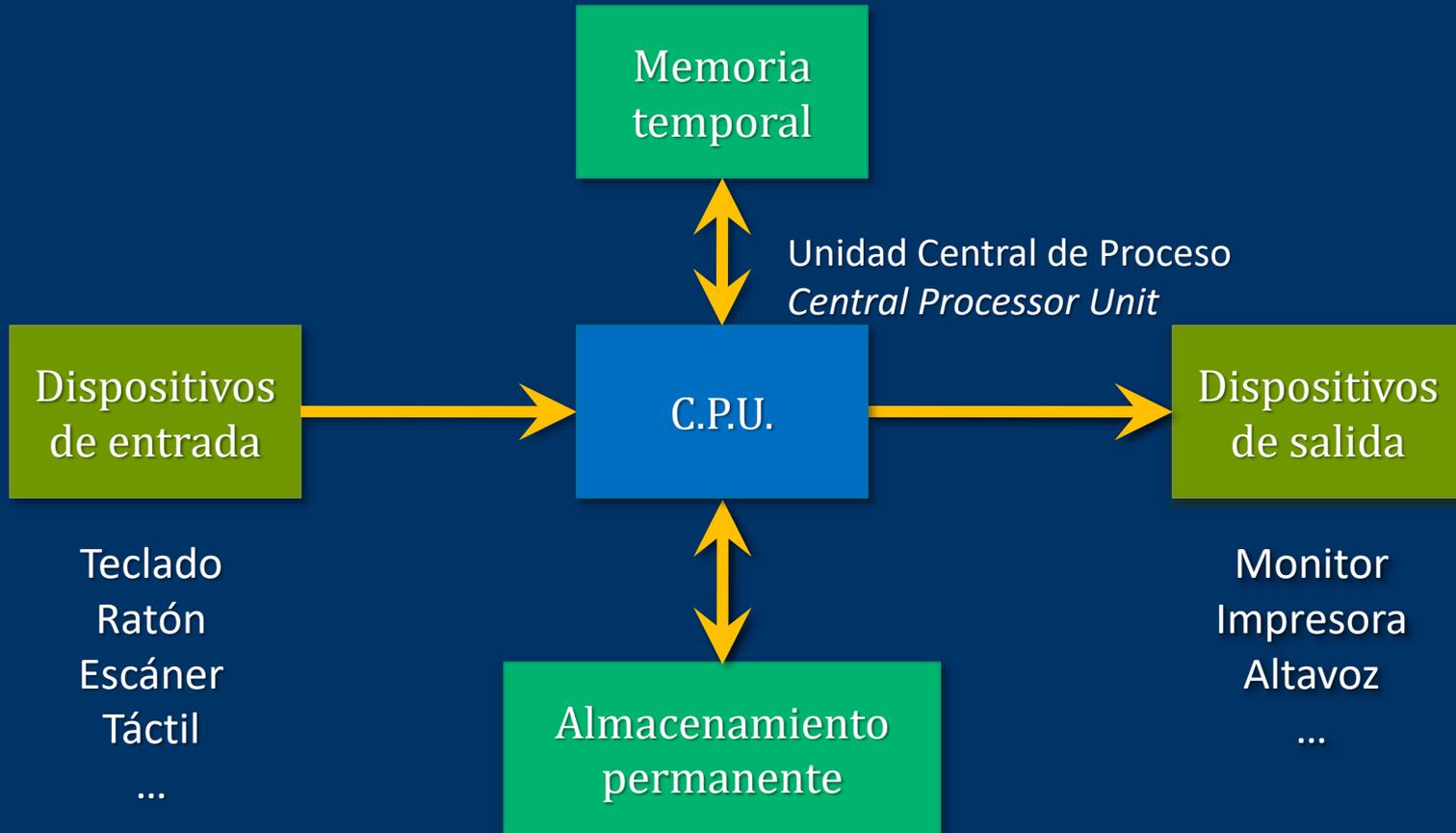
- ✓ Gestores
- ✓ Analistas
- ✓ Diseñadores
- ✓ Programadores
- ✓ Probadores
- ✓ Administradores de sistemas

...



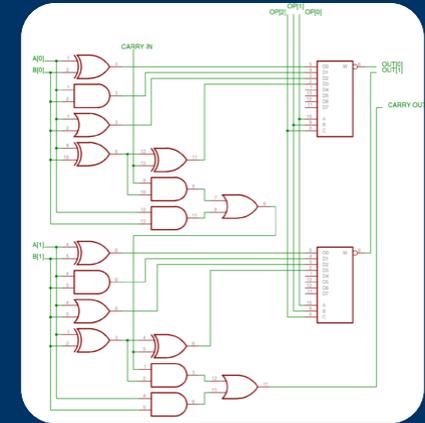
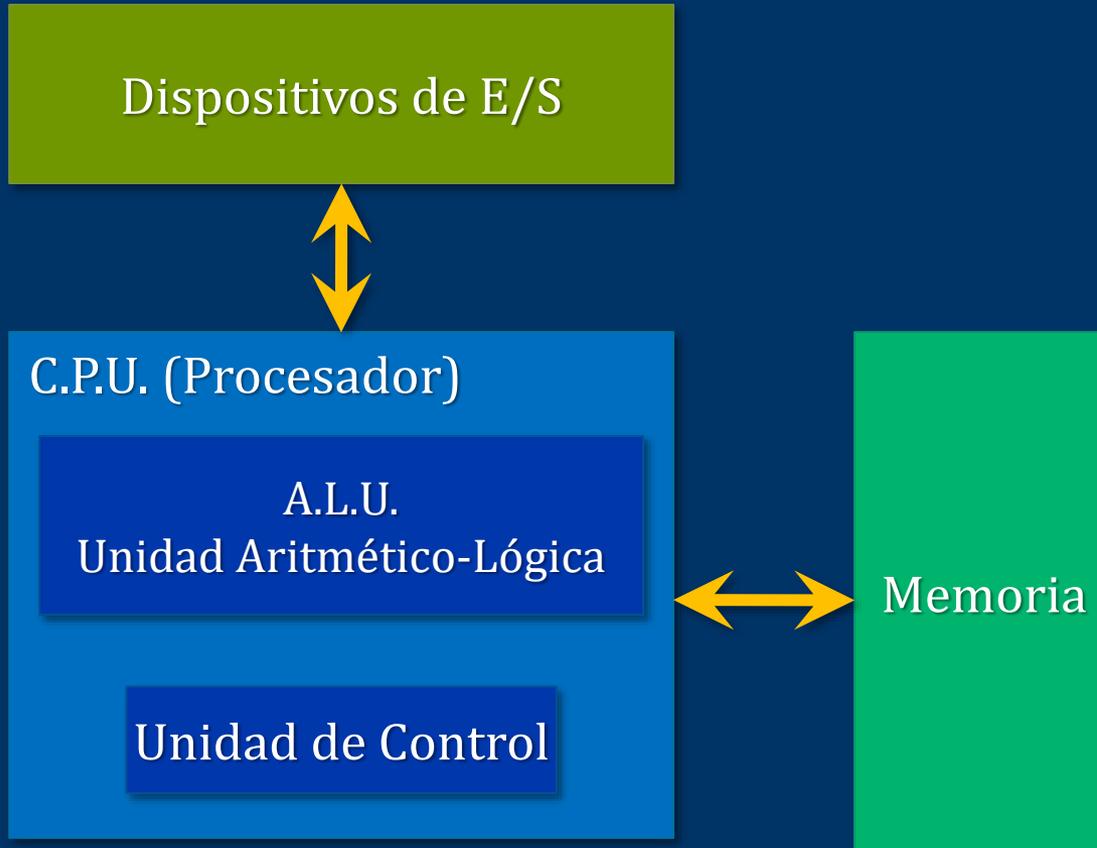
# Computadoras

## Esquema general



# Computadoras

## La arquitectura de Von Neumann

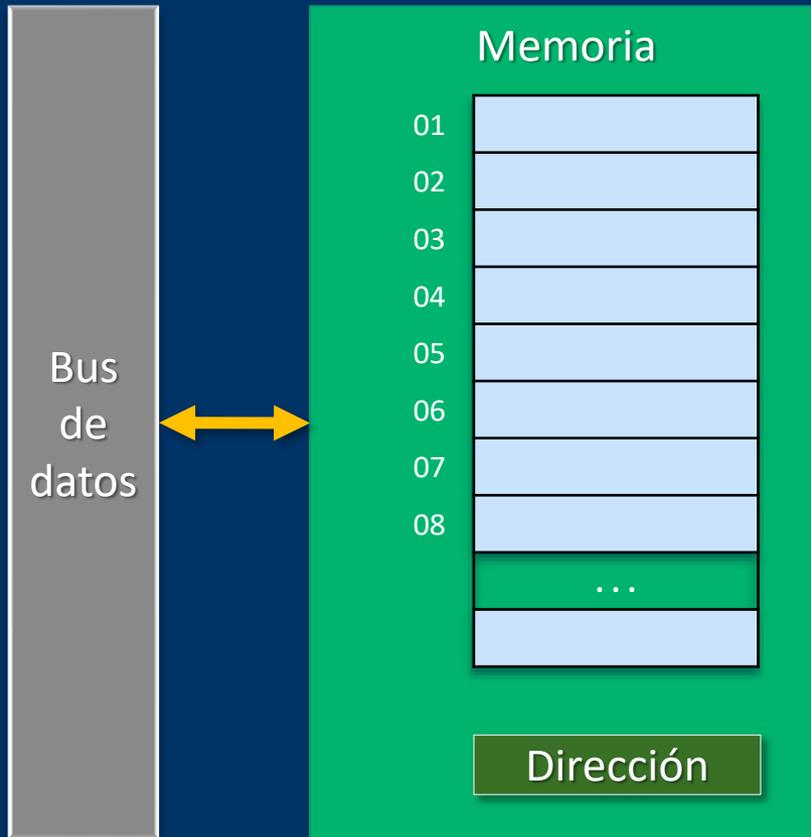


Una ALU de 2 bits (Wikipedia)



# Computadoras

## La memoria



Cada celda en una dirección  
Celdas de 8 / 16 / 32 / 64 bits  
Información volátil

- 1 Bit = 0 / 1
- 1 Byte = 8 bits = 1 carácter
- 1 Kilobyte (KB) = 1024 Bytes
- 1 Megabyte (MB) = 1024 KB
- 1 Gigabyte (GB) = 1024 MB
- 1 Terabyte (TB) = 1024 GB
- 1 Petabyte (PB) = 1024 TB

$$2^{10} = 1024 \approx 1000$$



# Fundamentos de la programación

---

## Lenguaje máquina y ensamblador



# Programación de computadoras

*Los procesadores trabajan con ceros y unos (bits)*

Unidad de memoria básica: *Byte* (8 bits)

(2 dígitos hexadecimales: 01011011 → 0101 1011 → 5B)

## *Lenguaje máquina*

Códigos hexadecimales que representan instrucciones, registros de la CPU, direcciones de memoria o datos

Instrucción *Significado*

A0 2F *Acceder a la celda de memoria 2F*

3E 01 *Copiarlo el registro 1 de la ALU*

A0 30 *Acceder a la celda de memoria 30*

3E 02 *Copiarlo en el registro 2 de la ALU*

1D *Sumar*

B3 31 *Guardar el resultado en la celda de memoria 31*

Lenguaje de bajo nivel

Dependiente de la máquina

Programación difícil



# Lenguaje ensamblador

Nemotécnicos para los códigos hexadecimales:

A0 → READ    3E → REG    1D → ADD    ...

Mayor legibilidad:

```
READ 2F
REG 01
READ 30
REG 02
ADD
WRITE 31
```

Lenguaje de nivel medio



# Fundamentos de la programación

---

## Lenguajes de programación de alto nivel



# Lenguajes de programación de alto nivel

---

- ✓ Más cercanos a los lenguajes natural y matemático  
`resultado = dato1 + dato2;`
- ✓ Mayor legibilidad, mayor facilidad de codificación
- ✓ Estructuración de datos / abstracción procedimental

**FORTRAN Python Prolog C#**  
**C Pascal Cobol Lisp Ruby**  
**BASIC Smalltalk Haskell Ada**  
**Simula Java Eiffel C++**  
...



# Lenguajes de programación de alto nivel

## Traducción

**Compiladores:**  
Compilan y enlazan programas completos

**Intérpretes:**  
Compilan, enlazan y ejecutan instrucción a instrucción

Código fuente



```
#include <iostream>
using namespace std;
```

```
int main()
{
    cout << "Hola Mundo!" << endl;
    return 0;
}
```

Código objeto

0100010100111010011100...



Código objeto de biblioteca

Programa ejecutable

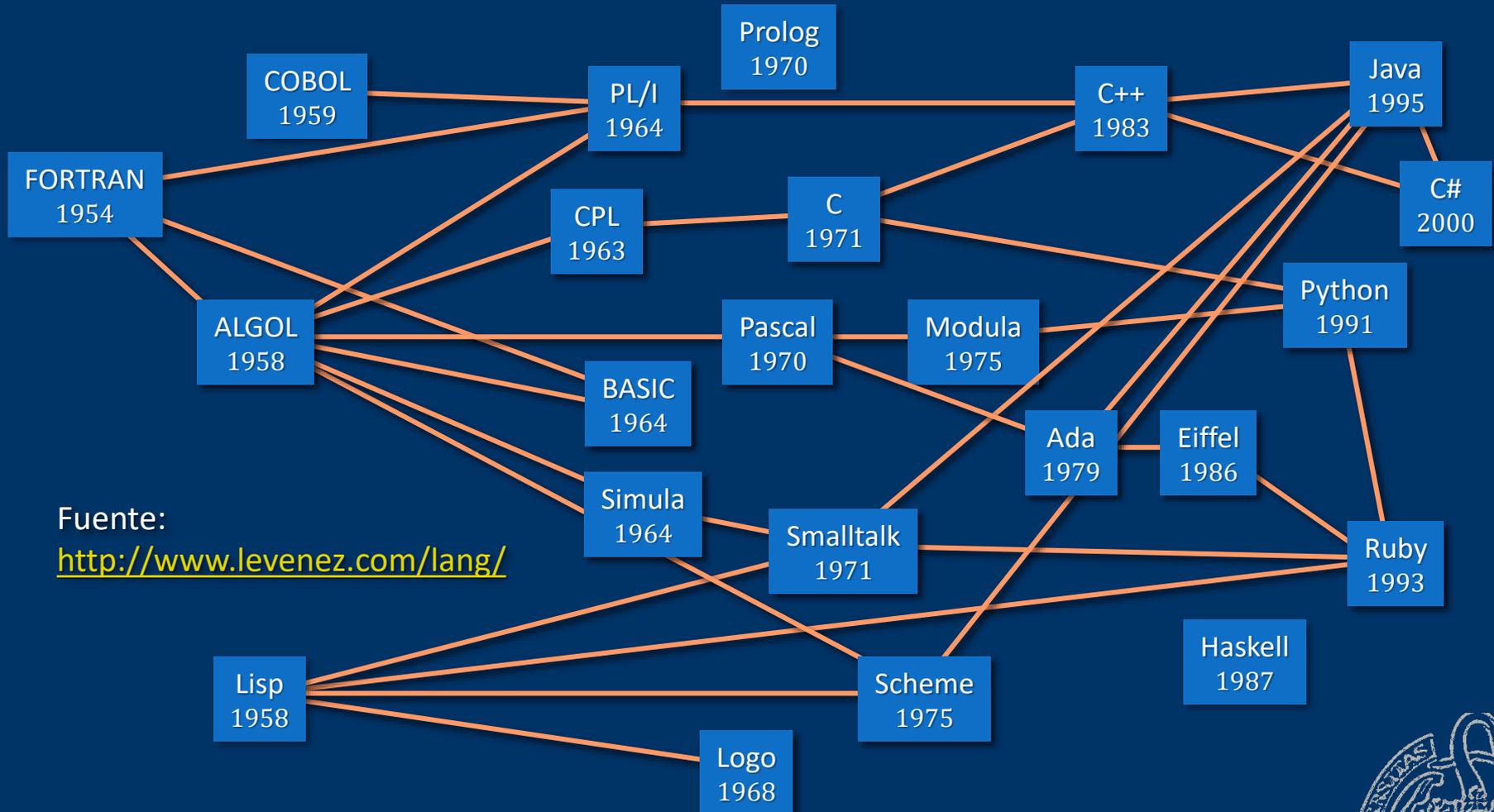
Para una arquitectura concreta y un sistema operativo



# Los lenguajes de programación de alto nivel

## Genealogía de lenguajes

Versiones / Estándares



Fuente:

<http://www.levenez.com/lang/>



# Fundamentos de la programación

---

## Un poco de historia



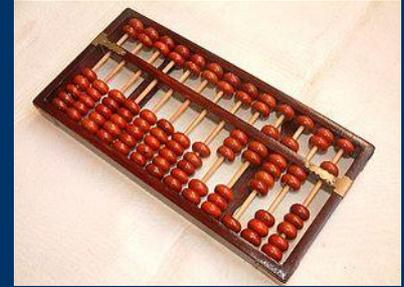
# Un poco de historia

## *La prehistoria*

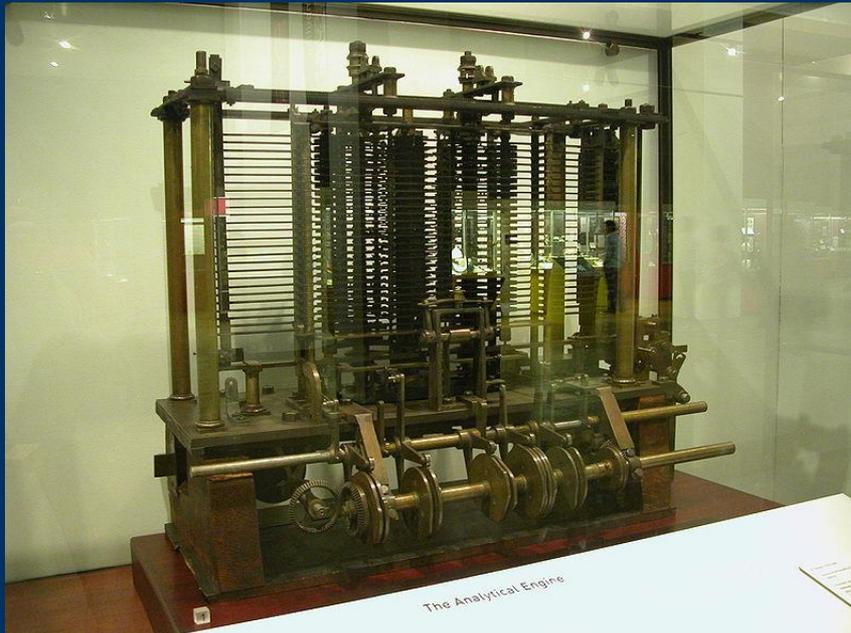
El ábaco

Siglo XIX

**Máquina analítica** de Charles Babbage



(Wikipedia)



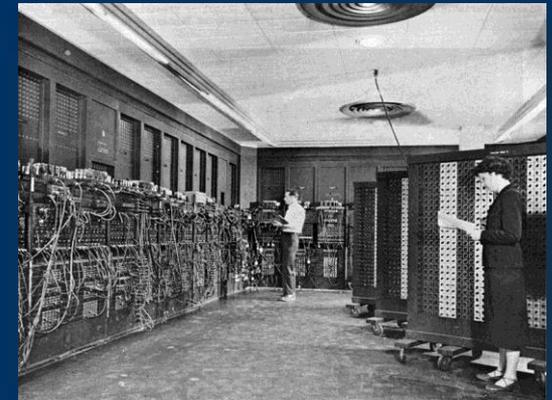
Lady Ada Lovelace  
es considerada  
la primera  
programadora



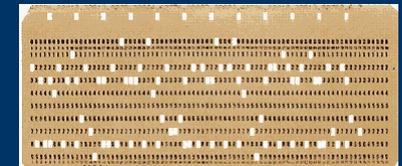
# Un poco de historia

## Siglo XX

- 1936 Máquina de **Turing**
- 1946 **ENIAC**: Primera computadora digital de propósito general
- 1947 El **transistor**
- 1953 **IBM 650**: Primera computadora a gran escala
- 1966 **ARPANET**: Origen de Internet
- 1967 El *disquete*
- 1970 Sistema operativo **UNIX**
- 1972 Primer **virus** informático (*Creeper*)  
Lenguaje de programación **C**
- 1974 Protocolo **TCP**. Primera red local



ENIAC (Wikipedia)



# Un poco de historia

1975 Se funda **Microsoft**

**Microsoft**

1976 Se funda **Apple**



Apple II (Wikipedia)

1979 Juego **Pacman**



1981 **IBM PC**

Sistema operativo **MS-DOS**

1983 Lenguaje de programación **C++**

1984 **CD-ROM**

1985 **Windows 1.0**

1990 Lenguaje **HTML**

**World Wide Web**

1991 Sistema operativo **Linux**



Linux



IBM PC (Wikipedia)



# Un poco de historia

1992 Windows 3.1

1995 Lenguaje de programación **Java**  
**DVD**

1998 Se funda **Google**

1999 MSN **Messenger**



## Siglo XXI

2001 Windows XP  
**Mac OS X**



2002 Mozilla **Firefox**



2007 **iPhone**



2008 **Android ...**



## Programación e Ingeniería del Software



# Programa informático

## ¿Qué es programar?

Decirle a un tonto *muy rápido* **exactamente** lo que tiene que hacer

Especificar la **estructura** y el **comportamiento** de un **programa**, así como **probar** que el programa realiza su tarea **adecuadamente** y con un **rendimiento** aceptable

**Programa:** Transforma entrada en salida



**Algoritmo:** Secuencia de pasos y operaciones que debe realizar el programa para resolver el problema

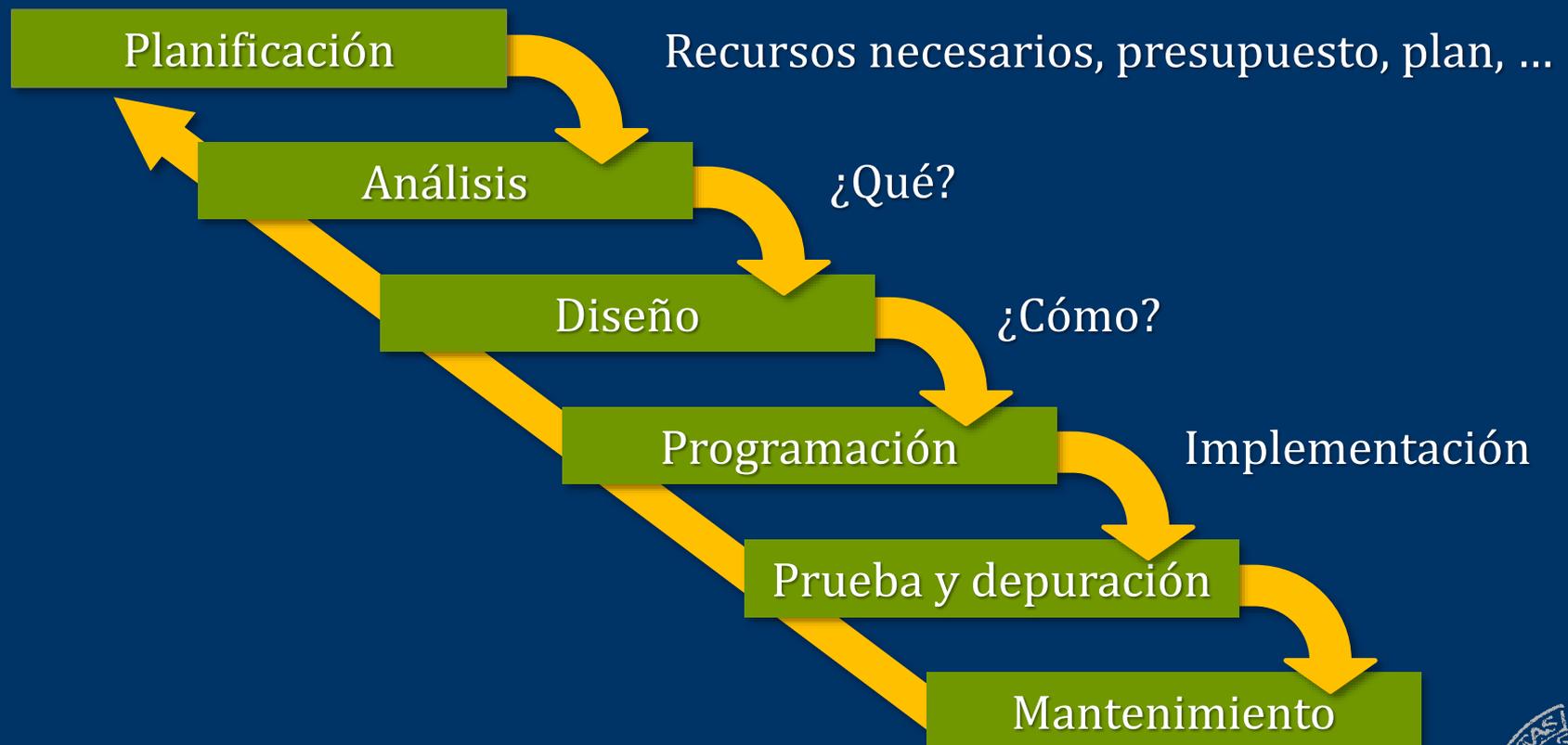
El programa implementa el algoritmo en un lenguaje concreto



# La Ingeniería del Software

*La programación es sólo una etapa del proceso de desarrollo*

Modelo de desarrollo “en cascada”:



# Fundamentos de la programación

---

## El lenguaje de programación C++



# El lenguaje de programación C++

---

*Bjarne Stroustrup (1983)*

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Hola Mundo!" << endl;
    // Muestra Hola Mundo!

    return 0;
}
```

A terminal window with a black background and a white border. The text "Hola Mundo!" is displayed in white at the top left of the window.



# Elementos del lenguaje

Instrucciones

Datos: literales, variables, tipos

Subprogramas (funciones)

Comentarios

Directivas

...

```
#include <iostream>  
using namespace std;
```

Directiva

Subprograma

```
int main()
```

```
{
```

Instrucción

```
cout << "Hola Mundo!" << endl;  
// Muestra Hola Mundo!
```

Dato

Comentario

Instrucción

```
return 0;
```

Dato

```
}
```



## Sintaxis de los lenguajes de programación



# Los lenguajes de programación

---

## *Sintaxis y semántica de los lenguajes*

### Sintaxis

- Reglas que determinan cómo se pueden construir y secuenciar los elementos del lenguaje

### Semántica

- Significado de cada elemento del lenguaje  
¿Para qué sirve?



# Sintaxis de los lenguajes de programación

## Especificación

- ✓ Lenguajes (BNF)
- ✓ Diagramas

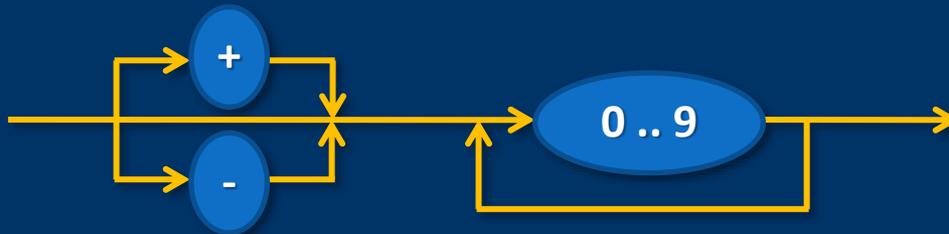
Ejemplo: Números enteros (sin decimales)

BNF

```
<numero entero> ::= <signo opcional><secuencia de dígitos>  
<signo opcional> ::= + | - | <nada>  
<secuencia de dígitos> ::= <dígito> | <dígito><secuencia de dígitos>  
<dígito> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9  
<nada> ::=
```

| significa ó

+23	✓
-159	✓
1374	✓
1-34	✗
3.4	✗
002	✓



# Backus-Naur Form (BNF)

```
<numero entero> ::= <signo opcional><secuencia de dígitos>  
<signo opcional> ::= +|-|<nada>  
<secuencia de dígitos> ::= <dígito>|<dígito><secuencia de dígitos>  
<dígito> ::= 0|1|2|3|4|5|6|7|8|9  
<nada> ::=
```

**+23**

```
<numero entero> ::= <signo opcional><secuencia de dígitos>  
::= +<secuencia de dígitos> ::= +<dígito><secuencia de dígitos>  
::= +2<secuencia de dígitos> ::= +2<dígito> ::= +23
```



**1374**

```
<numero entero> ::= <signo opcional><secuencia de dígitos>  
::= <secuencia de dígitos> ::= <dígito><secuencia de dígitos>  
::= 1<secuencia de dígitos> ::= 1<dígito><secuencia de dígitos>  
::= 13<secuencia de dígitos> ::= 13<dígito><secuencia de dígitos>  
::= 137<secuencia de dígitos> ::= 137<dígito> ::= 1374
```



**1-34**

```
<numero entero> ::= <signo opcional><secuencia de dígitos>  
::= <secuencia de dígitos> ::= <dígito><secuencia de dígitos>  
::= 1<secuencia de dígitos> ::= ERROR (- no es <dígito>)
```



# Diagramas de sintaxis



# Fundamentos de la programación

---

## Un primer programa en C++



# Un primer programa en C++

---

## *Hola Mundo!*

Un programa que muestra un saludo en la pantalla:

```
#include <iostream>
using namespace std;

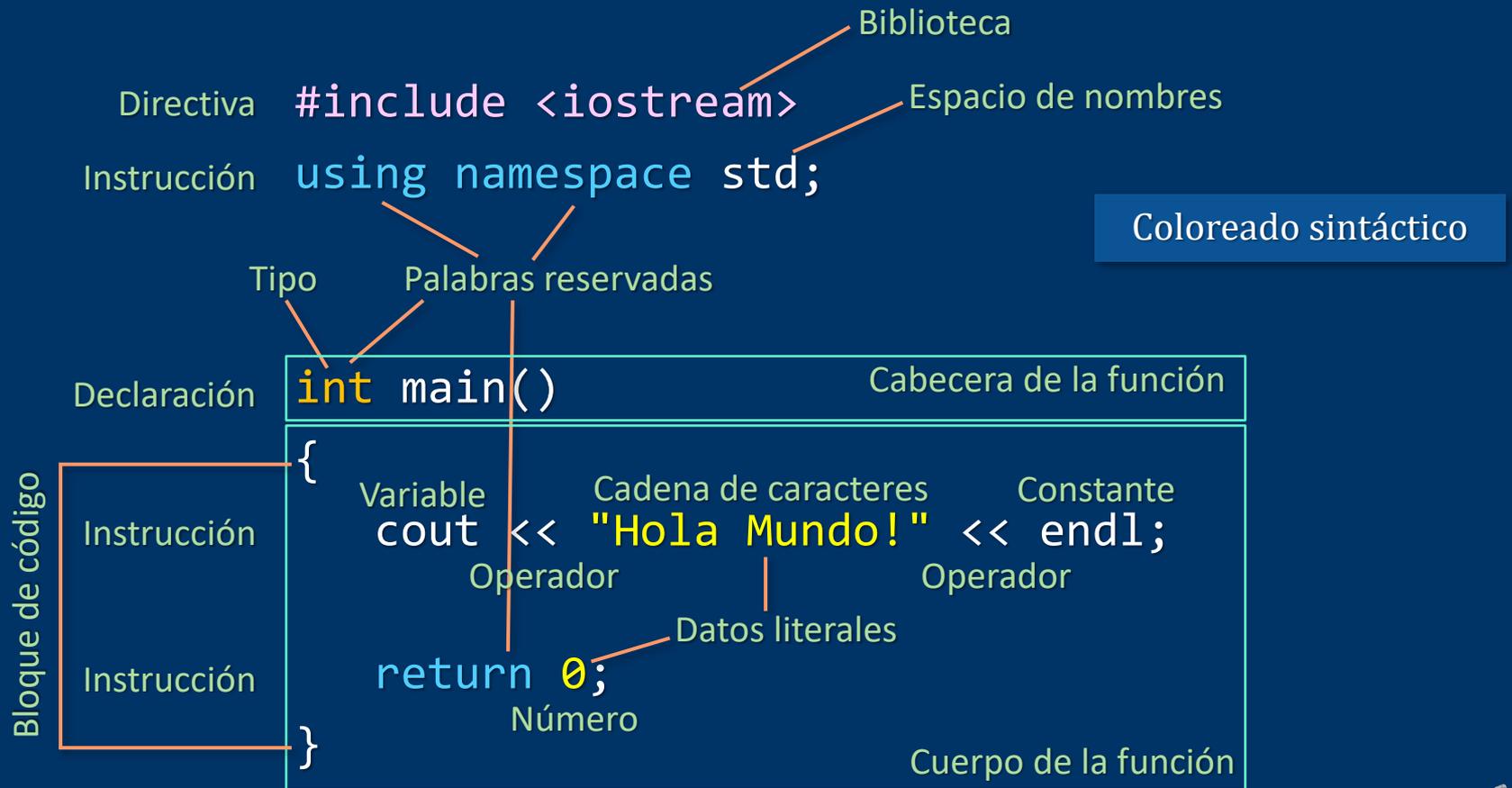
int main()
// main() es donde empieza la ejecución
{
    cout << "Hola Mundo!" << endl; // Muestra Hola Mundo!

    return 0;
}
```



# Un primer programa en C++

## Análisis del programa



# Un primer programa en C++

---

*Hola Mundo!*

Casi todo es *infraestructura*

Sólo

```
cout << "Hola Mundo!" << endl
```

hace algo palpable

La infraestructura (notación, bibliotecas y otro soporte)  
hace nuestro código simple, completo, confiable y eficiente

*¡El estilo importa!*



# Fundamentos de la programación

---

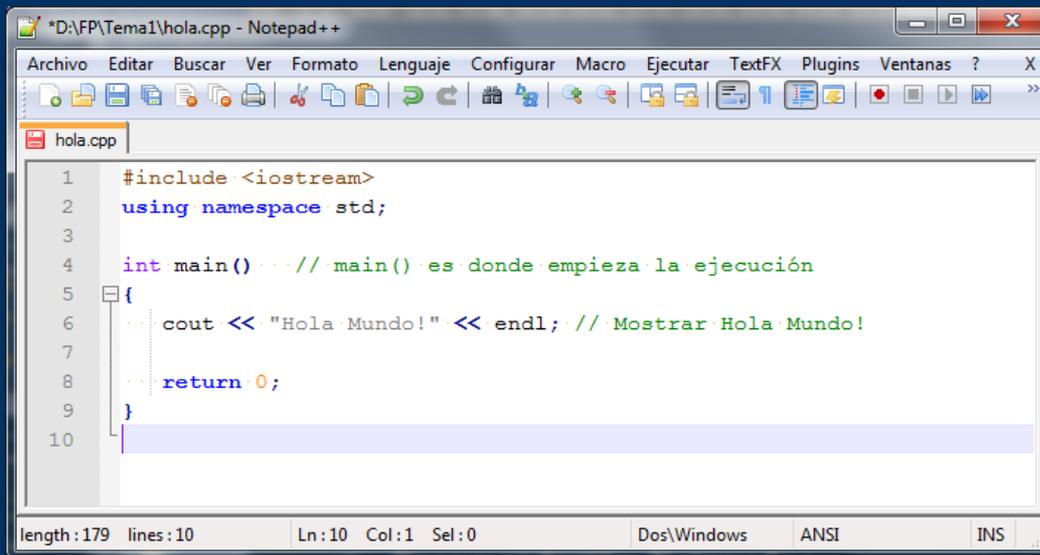
## Herramientas de desarrollo



# Herramientas de desarrollo

## Editor

- ✓ Bloc de notas, Wordpad, Word, Writer, Gedit, Kwrite, ... (texto simple, sin formatos)
- ✓ Editores específicos: coloreado sintáctico
- ✓ Recomendación: **Notepad++**



The screenshot shows the Notepad++ application window with the file 'hola.cpp' open. The code is as follows:

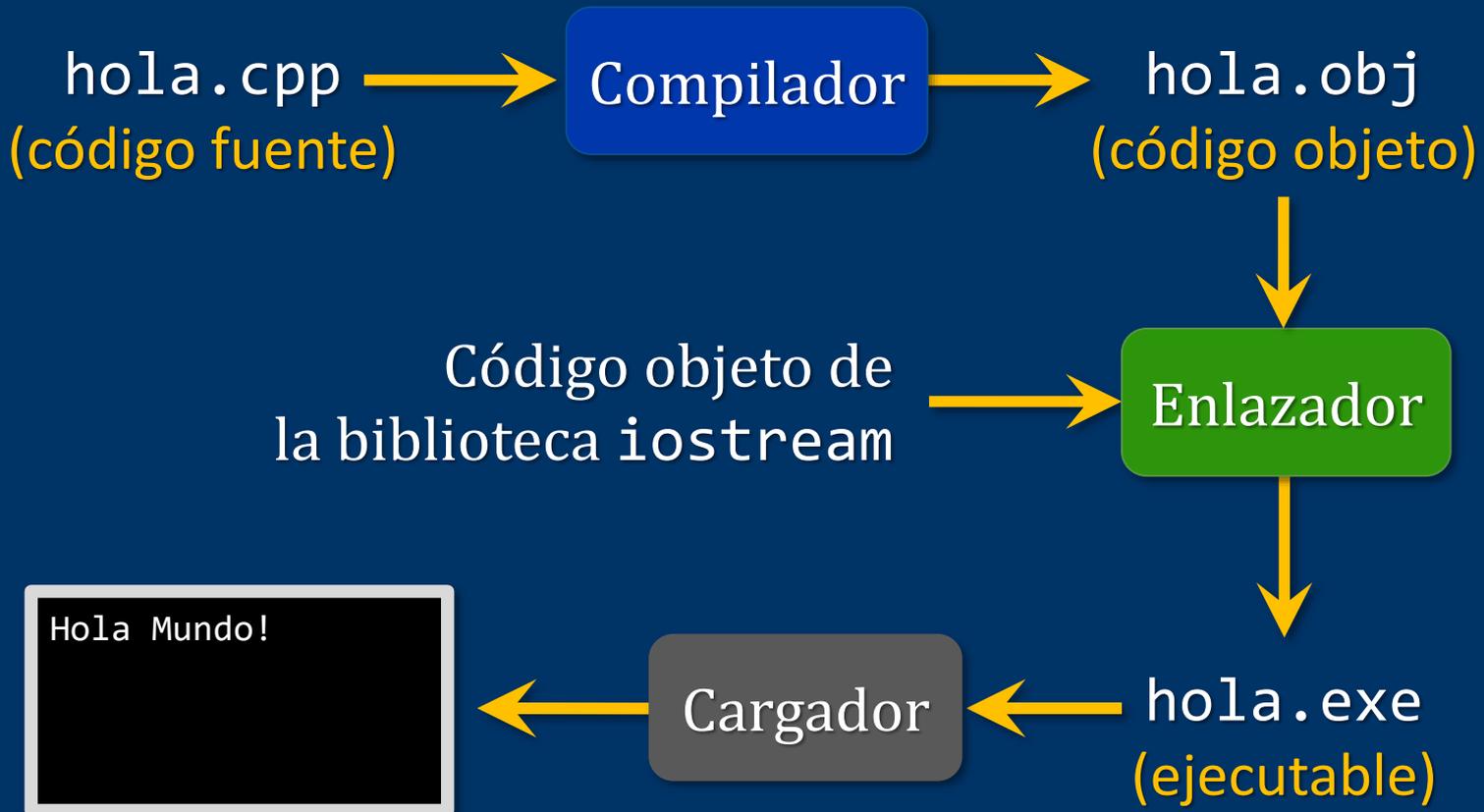
```
1 #include <iostream>
2 using namespace std;
3
4 int main() // main() es donde empieza la ejecución
5 {
6     cout << "Hola Mundo!" << endl; // Mostrar Hola Mundo!
7
8     return 0;
9 }
10
```

The status bar at the bottom indicates: length: 179 lines: 10 Ln: 10 Col: 1 Sel: 0 Dos\Windows ANSI INS

Instalación y uso:  
Sección  
Herramientas de desarrollo  
en el Campus Virtual



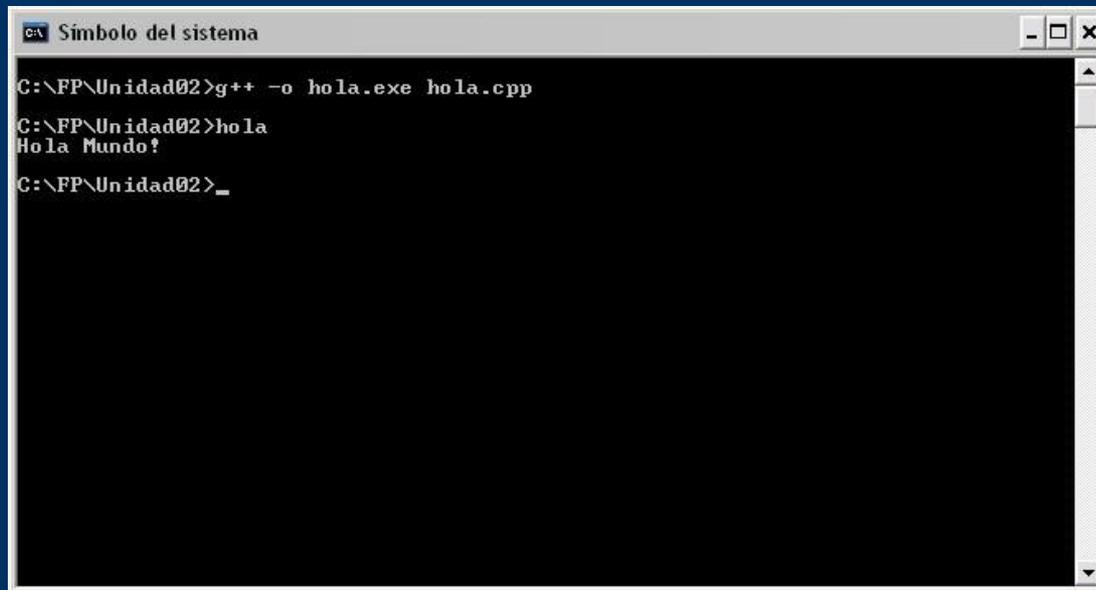
# Compilación, enlace y ejecución



# Más herramientas de desarrollo

## Compilador

- ✓ Importante: C++ estándar
- ✓ Recomendación: GNU G++ (*MinGW* en Windows)



```
C:\FP\Unidad02>g++ -o hola.exe hola.cpp
C:\FP\Unidad02>hola
Hola Mundo!
C:\FP\Unidad02>_
```

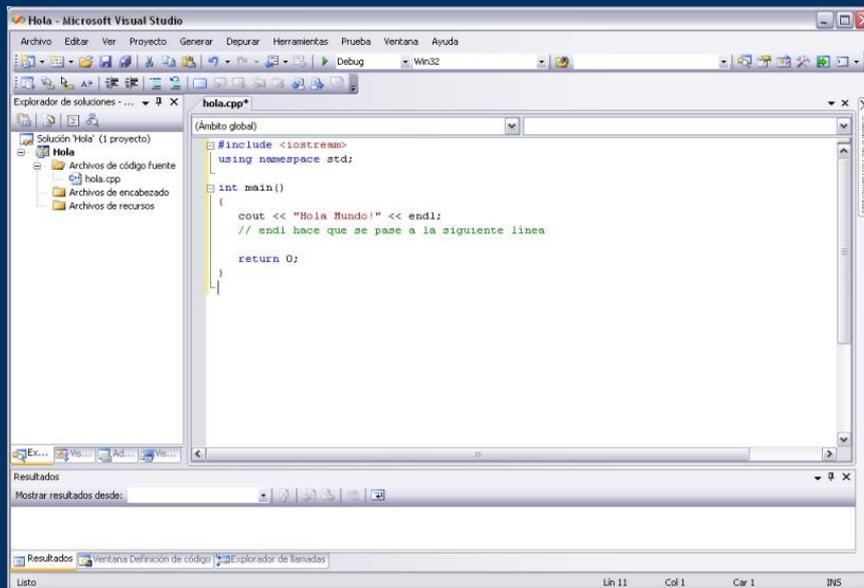
Instalación y uso:  
Sección  
Herramientas de desarrollo  
en el Campus Virtual



# Más herramientas de desarrollo

## *Entornos de desarrollo*

- ✓ Para editar, compilar y probar el código del programa
- ✓ Recomendaciones:
  - Windows: MS Visual Studio / C++ Express o Eclipse
  - Linux: Netbeans o Eclipse



Instalación y uso:  
Sección  
Herramientas de desarrollo  
en el Campus Virtual



# Un primer programa en C++: ejecución

*¿Qué hace el programa?*

- ✓ La ejecución del programa siempre empieza en `main()`
- ✓ Se ejecutan las instrucciones en secuencia de principio a fin



Pantalla (cout)

```
Hola Mundo!
```



# Fundamentos de la programación

---

## C++: Un mejor C



# C++: Un mejor C

---

## *El lenguaje C*

- ✓ Lenguaje creado por Dennis M. Ritchie en 1972
- ✓ Lenguaje de nivel medio:
  - Estructuras típicas de los lenguajes de alto nivel
  - Construcciones para control a nivel de máquina
- ✓ Lenguaje sencillo (pocas palabras reservadas)
- ✓ Lenguaje estructurado (no estrictamente estructurado en bloques)
- ✓ Compartimentalización de código (funciones) y datos (ámbitos)
- ✓ Componente estructural básico: la función (subprograma)
- ✓ Programación modular
- ✓ Distingue entre mayúsculas y minúsculas
- ✓ Palabras reservadas (o clave): en minúsculas



# Acerca de *Creative Commons*



## Licencia CC (*Creative Commons*)

Este tipo de licencias ofrecen algunos derechos a terceras personas bajo ciertas condiciones.

Este documento tiene establecidas las siguientes:

-  Reconocimiento (*Attribution*):  
En cualquier explotación de la obra autorizada por la licencia hará falta reconocer la autoría.
-  No comercial (*Non commercial*):  
La explotación de la obra queda limitada a usos no comerciales.
-  Compartir igual (*Share alike*):  
La explotación autorizada incluye la creación de obras derivadas siempre que mantengan la misma licencia al ser divulgadas.

Pulsa en la imagen de arriba a la derecha para saber más.

