



Dirección  
Académica

# Programación estructurada

Ciclo Mayo 2022

---

## CONTENIDO

## **REPASO CLASE ANTERIOR**

# Repaso clase anterior

---

¿Qué es un algoritmo?

- Secuencia de pasos finitos para resolver un problema o realizar una actividad.
- Tiene un inicio y un fin.
- Datos de entrada.
- Procesos.
- Salidas.

# Repaso clase anterior

## Algoritmo Suma:

- Inicio ✓
- Escribir "Dame el valor de a"
- Leer a
- Escribir "Dame el valor de b"
- Leer b
- Suma = a + b
- Escribir "La suma es: " + suma

Fin ✓

Inicio  
Fin  
↳ Datos a, b  
Números  
↳ Proceso  $a + b$   
↳ Salida Suma

# Ejemplo de repaso

---

Actividad:

Piensa y reflexiona la siguiente pregunta:

¿Puedo programar todo lo que imagine?

Todo lo que ves....PENSAMIENTOS, representaciones de cosas externas

¡¡¡Los pensamientos ayudan al hombre a explicar el mundo!!!

La pregunta sobre la que reflexionarás en esta unidad es:

¿De qué manera los elementos básicos de un programa permiten la lectura y escritura de datos, así como las operaciones lógicas y matemáticas?

En esta semana conocerás el entorno de trabajo de C#, así como los elementos básicos que integran un programa de tal manera que construyas tu primera aplicación con instrucciones de lectura y escritura de datos en la consola de interacción. También conocerás los tipos de variables soportadas por el lenguaje, las palabras reservadas que son utilizadas y la estructura de un programa.

# Objetivo

---

- Conocer que es un lenguaje de programación.
- Comprender los conceptos básicos de Programación.
- Expresar un algoritmo en el lenguaje de programación C#.



# Resultados de aprendizaje

---

- Identificar los elementos básicos para el diseño de programas estructurados.
- Utilizar la sintaxis del lenguaje de programación en la implementación de operaciones básicas.

- Estructurar programas mediante la aplicación de las instrucciones de lectura y escritura, así como el uso de los distintos operadores para resolver problemas simples.

## DESARROLLO

# Introducción

---

## *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.

Secuencia de instrucciones.

Instrucciones que entiende la computadora.

Y que persiguen un objetivo: resolver un problema!

# Programadores

Trabajo en equipo

Múltiples roles...

Gestores

Analistas

Diseñadores

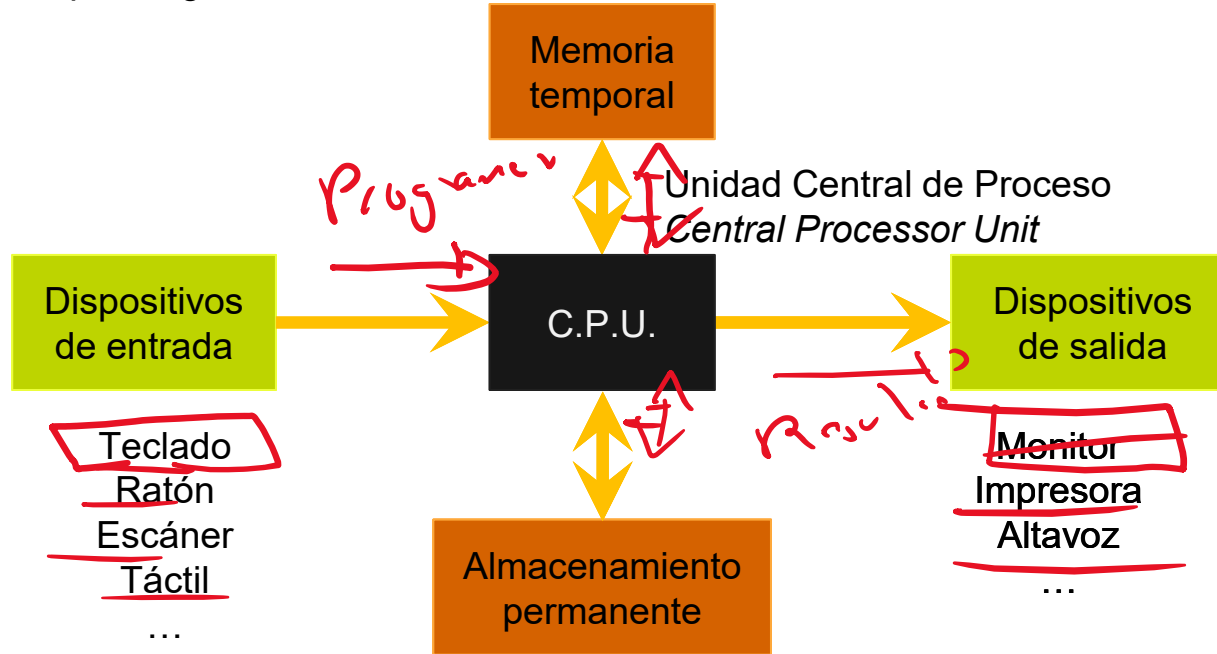
Programadores

Probadores

Administradores de sistemas.

# Modelo de Computación

Esquema general



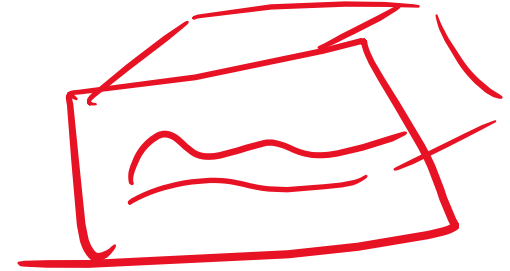
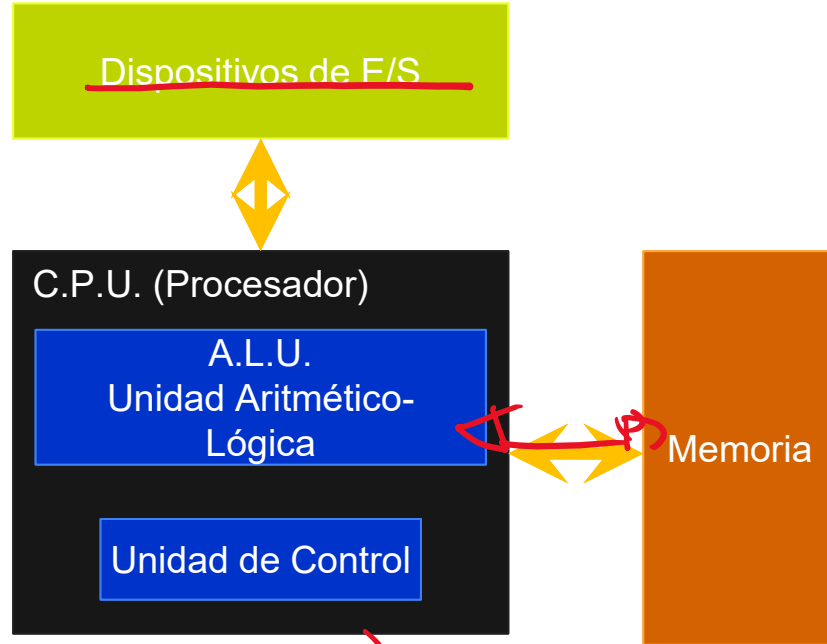
Datos

Procesos

Solides

# Modelo de Computación

La arquitectura de Von Neumann



Handwritten red text: "FS a l. de" (likely "FS a l. de" or "FS a l. de").



637 #2  
✓  
Tessie



Celdas de 8 / 16 / 32 / 64 bits

## Información volátil

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

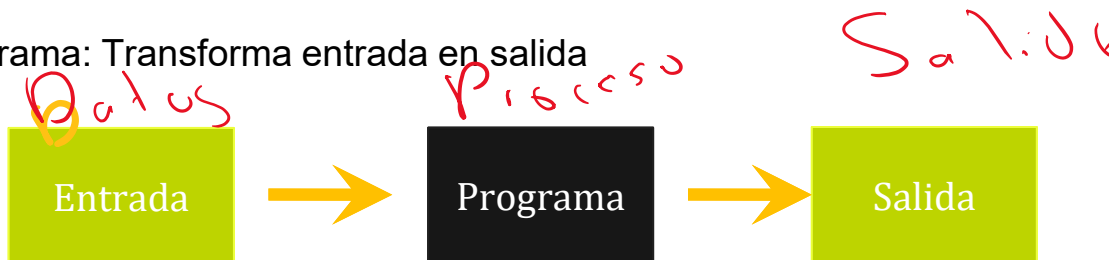
$$2^{10} = 1024 \approx 1000$$

# Proceso programación

¿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



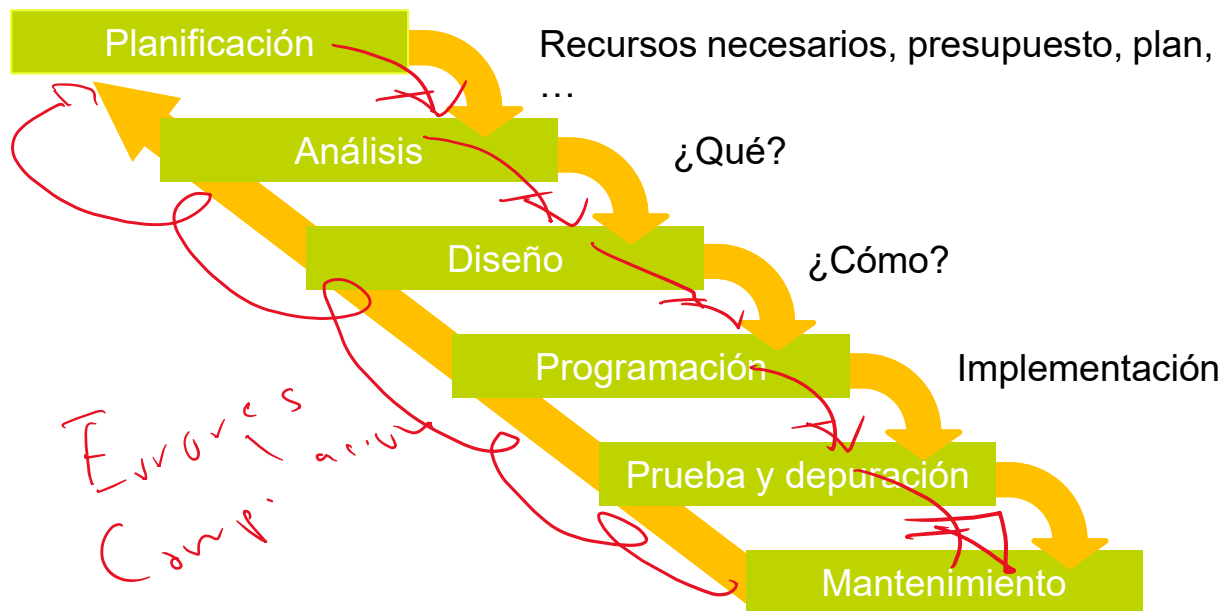
Algoritmo: Secuencia de pasos y operaciones que debe realizar  
el programa para resolver el problema

El programa implementa el algoritmo en un lenguaje concreto

# Proceso desarrollo de Software

La programación es sólo una etapa del proceso de desarrollo

Modelo de desarrollo “en cascada”:



# 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~~

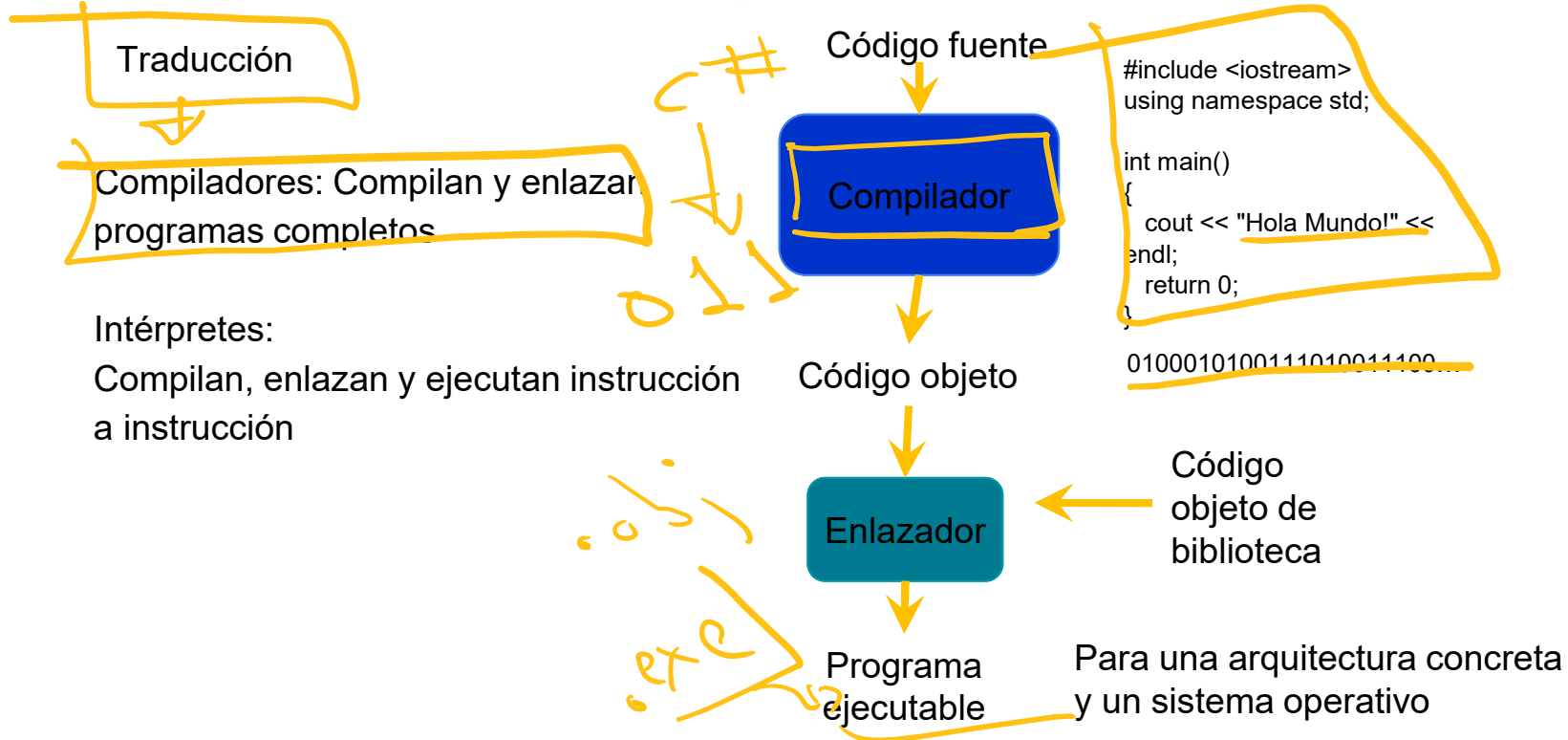
Java

Eiffel

C++

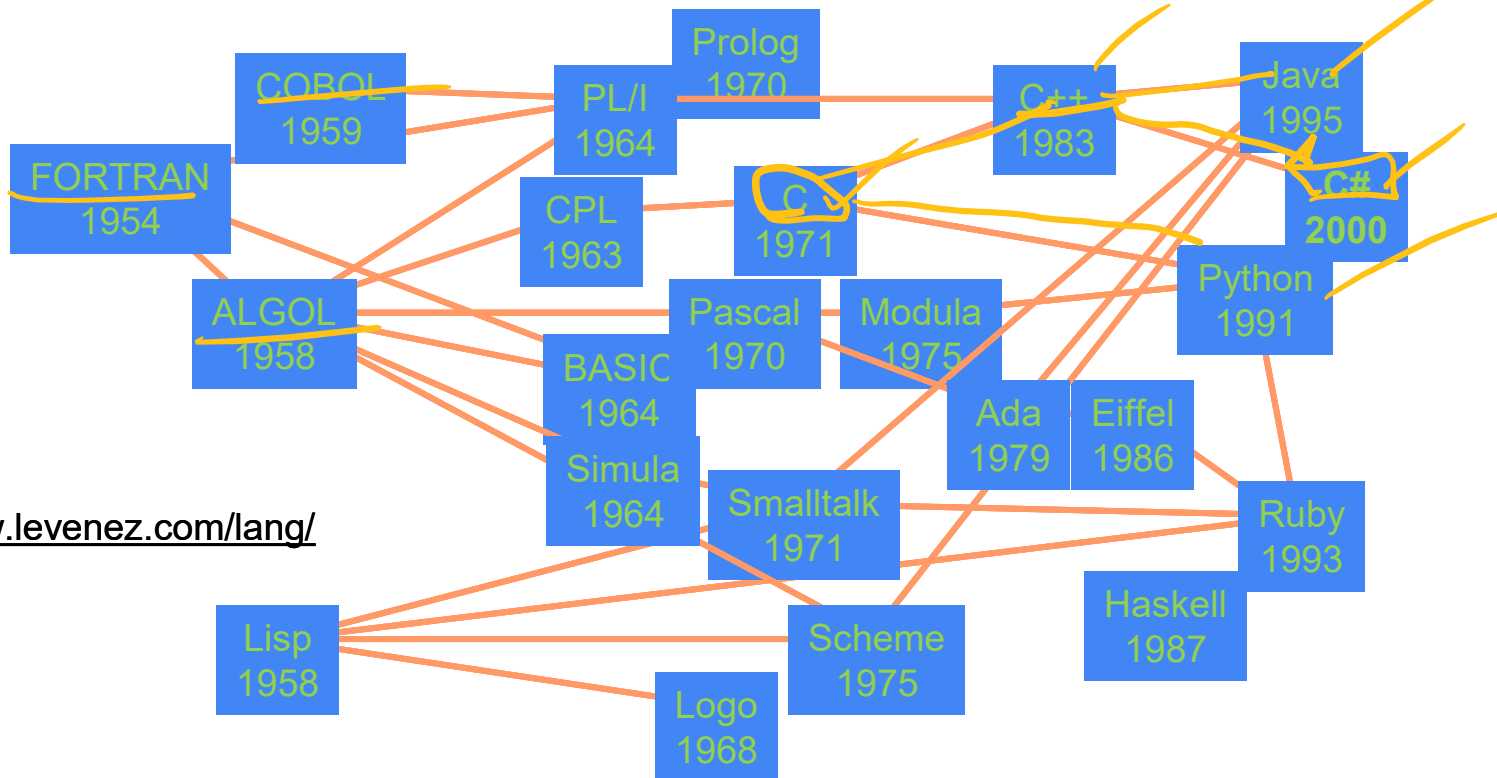
...

# Lenguajes de programación de alto nivel



- Genealogía de lenguajes

## Versiones / Estándares



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

# Lenguajes de programación C#

```
using System;

namespace HolaMundo {

class Program {

static void Main(string[] args) {

    Console.WriteLine("Hola Mundo!");
}

}

}
```

Hola Mundo!

# Elementos de programación C#

Instrucciones

Datos: literales, variables, tipos

Subprogramas (funciones)

Comentarios

Directivas

...

```
using System;  
  
namespace HolaMundo {  
  
    class Program {  
  
        static void Main(string[] args) {  
  
            Console.WriteLine("Hola Mundo!");  
            //Muestra Hola Mundo!  
        }  
    }  
}
```

Diagram illustrating C# code elements:

- Directiva**: `using System;`
- Subprograma**: `class Program {`
- Instrucción**: `static void Main(string[] args) {`
- Comentario**: `//Muestra Hola Mundo!`
- Dato**: `"Hola Mundo!"`



# Sintaxis de programación C#

---

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 programación C#

## Especificación

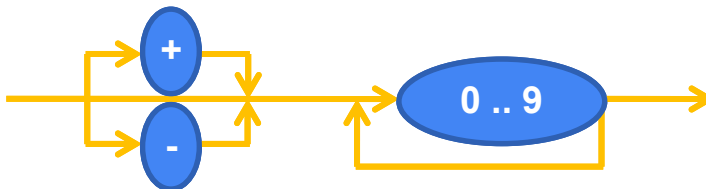
- Lenguajes (BNF)
- Diagramas

Ejemplo: Números enteros (sin decimales)

BNF

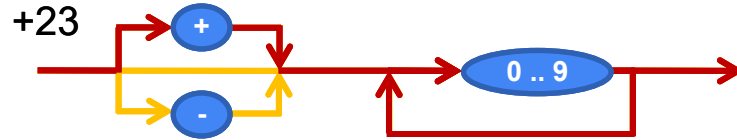
```
<numero entero> ::= <signo opcional><secuencia de dígitos>
<signo opcional> ::= +|-|<nada>
<secuencia de dígitos> ::= <dígito>|<dígito><secuencia de dígitos>
<dígito> ::= 0|1|2|3|4|5|6|7|8|9
<nada> ::=
```

| significa ó

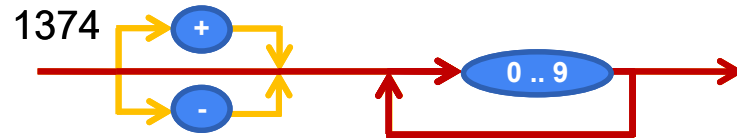


+23	✓
-159	✓
1374	✓
1-34	✗
3.4	✗
002	✓

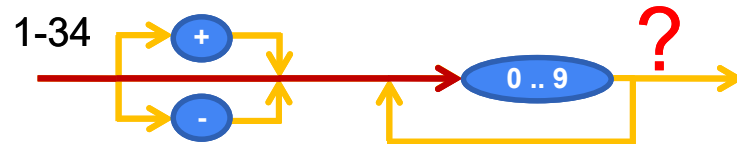
# Sintaxis de programación C#



+23 ✓



1374 ✓



1- ✗

## Ejemplo de programación C#

```
using System;

namespace Programacion {

    public class Ejemplo{

        public static void Main() {

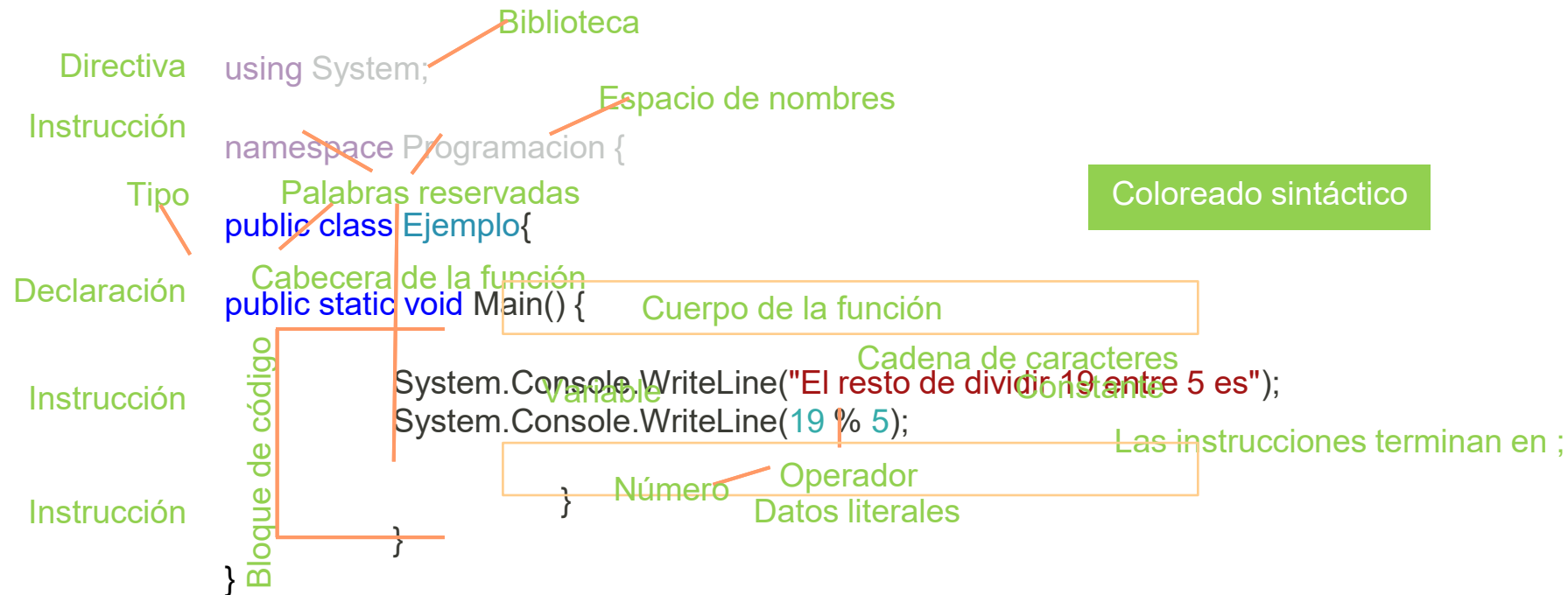
            System.Console.WriteLine("El resto de dividir 19 entre 5 es");
            System.Console.WriteLine(19 % 5);

        }

    }

}
```

# Ejemplo de programación C#



Coloreado sintáctico

## Ejemplo de programación C#

Hola Mundo!

Casi todo es infraestructura

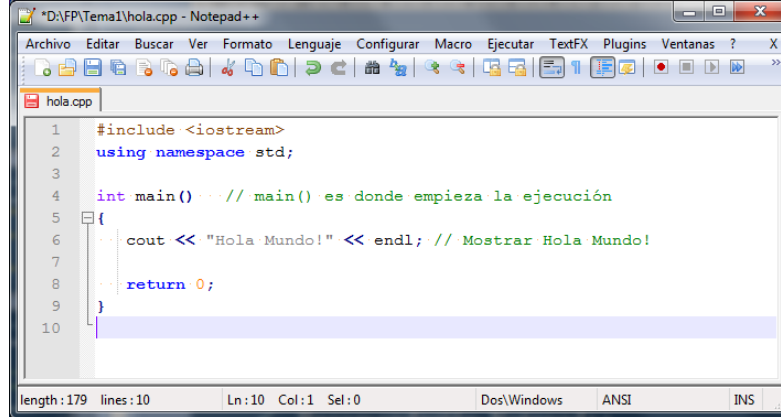
La infraestructura (notación, bibliotecas y otro soporte) hace nuestro código simple, completo, confiable y eficiente

¡El estilo importa!

# Herramientas de desarrollo en C#

## Editor

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



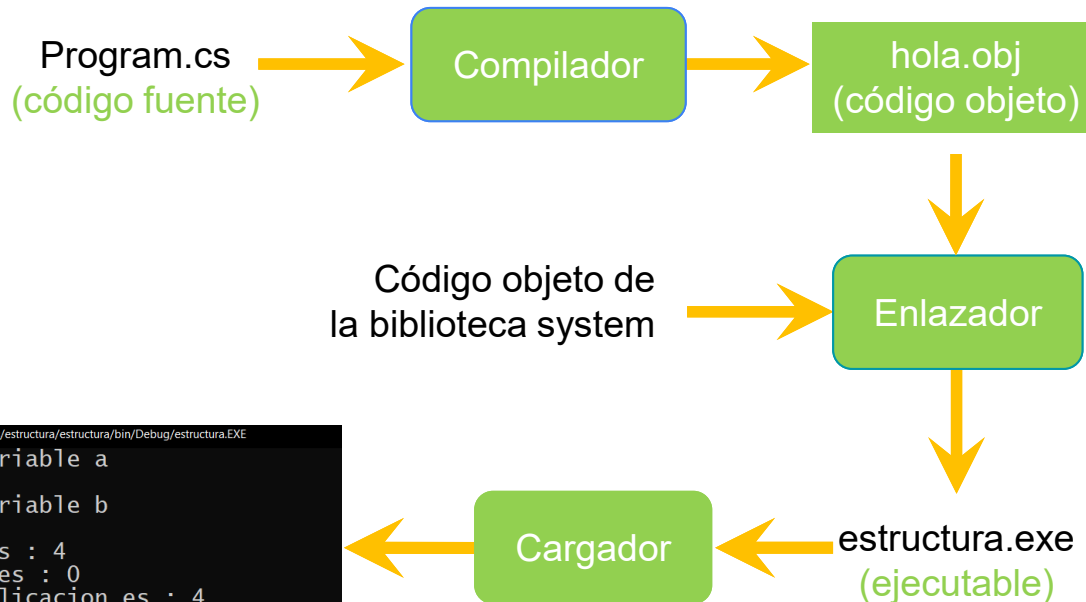
The screenshot shows the Notepad++ application window with the file 'hola.cpp' open. The code is written in C++ and includes syntax highlighting. The code is as follows:

```
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', and the encoding is 'ANSI'.

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

# Compilación, enlace y ejecución en C#



```

file:///c:/users/ia-games/documents/visual studio 2012/Projects/estructura/estructura/bin/Debug/estructura.EXE
Ingresa el valor de la variable a
2
Ingresa el valor de la variable b
2
El resultado de la suma es : 4
El resultado de la resta es : 0
El resultado de la multiplicacion es : 4
El resultado del cociente de la division es : 1
El resultado del modulo de la division es : 0
  
```



# Herramientas de desarrollo en C#

Compilador

:Net SDK

Net Runtime

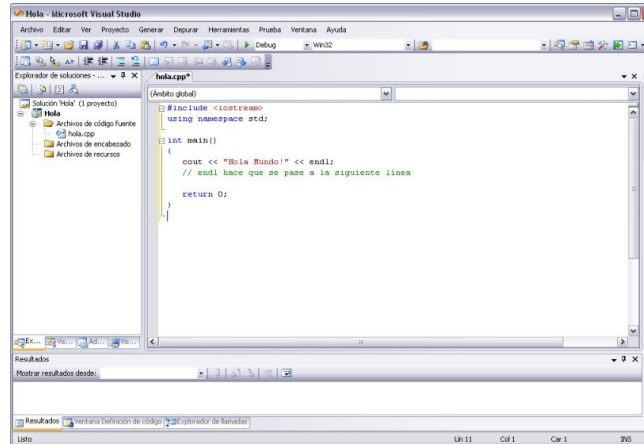
```
file:///c:/users/ia-games/documents/visual studio 2012/Projects/estructura/estructura/bin/Debug/estructura.EXE
Ingresa el valor de la variable a
2
Ingresa el valor de la variable b
2
El resultado de la suma es : 4
El resultado de la resta es : 0
El resultado de la multiplicacion es : 4
El resultado del cociente de la division es : 1
El resultado del modulo de la division es : 0
```

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

# Herramientas de desarrollo en C#

## Entornos de desarrollo

- Para editar, compilar y probar el código del programa
- Recomendaciones:
  - Windows: MS Visual Studio Express
  - Linux: Mono Develop

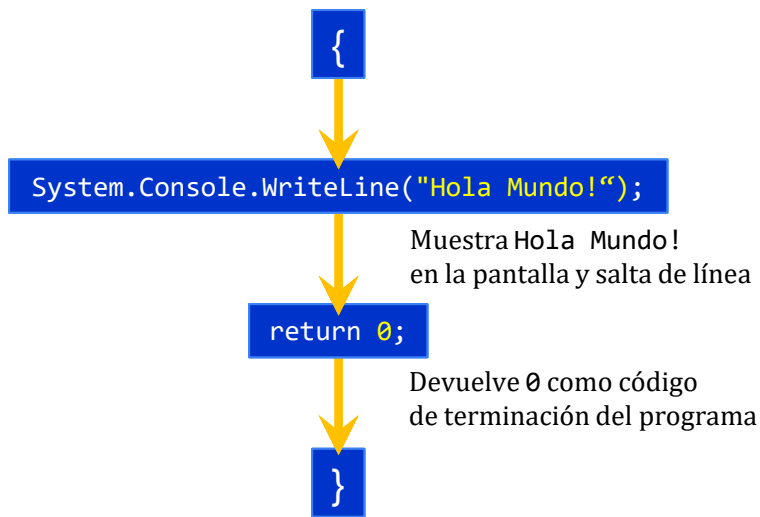


Instalación y uso:  
Sección  
Herramientas de desarrollo  
en el Campus Virtual

## Ejercicio de refuerzo en C#

¿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!

---

## **ACTIVIDADES DE REFORZAMIENTO**

# Actividades de reforzamiento

---

## Ejercicio 1. Operaciones básicas

A partir de los recursos revisados, programa las operaciones básicas de matemáticas con dos valores:

# Actividades de reforzamiento

```
using System;

namespace Programacion {

public class Ejemplo 2{

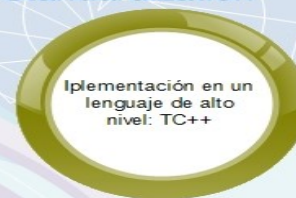
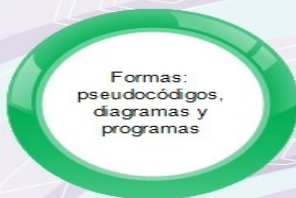
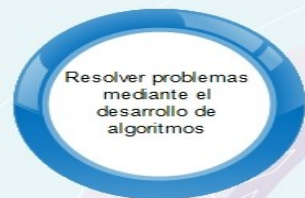
public static void Main() {
    double Num1, Num2;
    Num1 = Convert.ToDouble(TxtNum1.Text);
    Num2 = Convert.ToDouble(TxtNum2.Text);
    //Suma de dos números en C#
    double Suma;
    Suma = Num1 + Num2;
    Console.WriteLine("El Resultado de la suma es: " + Suma);
    //Resta de dos números en C#
    double Resta;
    Resta = Num1 - Num2;
    Console.WriteLine("El Resultado de la Resta es: " + Resta);
    //Multiplicación de dos números en C#
    double Multiplicacion;
    Multiplicacion = Num1 * Num2;
    Console.WriteLine("El Resultado de la Multiplicación es: " + Multiplicacion);
    //División de dos números en C#
    double Division;
    Division = Num1 / Num2;
    Console.WriteLine("El Resultado de la División es: " + Division);
}
}
}
```

# Conclusión

## Representación de algoritmos



International  
Organization for  
Standardization



Elementos y sintaxis de los diagramas de flujo. Símbolos para E/S, asignación, selección simple y múltiple, Ciclos "para", "mientras" y "repite". Elementos y sintaxis para pseudo-código. Elementos de codificación en diferentes lenguajes de programación de alto

### Diagrama de flujo:

método mas extendido y popular para realizar el diseño grafico de procesos . Su simplicidad y versatilidad han contribuido notablemente a su difusión.



pseudocódigo es como un falso lenguaje que está formado por una serie de palabras con un formalismo muy sencillo, que permite describir el funcionamiento de un programa.

- Se usa tanto en la fase de diseño como en la fase de análisis.

Inicio  
Leer(A,B)  
C=A+B  
Escribir(C)  
Fin

### Lenguajes de alto nivel.

- Son aquellos que se encuentran más cercanos al lenguaje natural que al lenguaje máquina.
- Están dirigidos a solucionar problemas mediante el uso de EDD (Estructuras de Datos Dinámicas).



Problema

Solución

Técnica de  
algoritmos

Pseudocódigo  
Diagramas de  
flujo  
Codificación

## **PREGUNTA DE INVESTIGACION**



# Pregunta de investigación

---

¿En el lenguaje de programación C# que tipo de aplicaciones se pueden desarrollar?

---

**CIERRE**

---

## ¿QUÉ ES C# ?

Lanzado en 2001 por Microsoft, **inspirado en C++, Java y Pascal para funcionar con la plataforma .NET.**



**Anders Hejlsberg**  
Creador de C#



### CARACTERÍSTICAS PRINCIPALES

Multiplataforma



Sintaxis similar a C++ y Java para **atraer a esos desarrolladores.**



**Multiparadigma:** tiene diferentes estilos para programar. Aunque está **principalmente orientado a objetos**, también se puede dirigir a **eventos y funciones.**



### SE UTILIZA PARA:

Aplicaciones del estándar **web assembly (Blazor).**



Construcción de **videojuegos (Unity).**



Aplicaciones de escritorio. **mono**



Aplicaciones móviles nativas para Android y iOS (Xamarin).



# BIBLIOGRAFÍA COMPLEMENTARIA

<http://www.msdn.microsoft.com/net/ecma>.

“A programmer’s introduction to C#” escrito por Eric Gunnerson y publicado por Apress en 2000.

C# and the .NET Framework”, escrito por Andrew Troelsen y publicado por Apress en 2001

“C# Essentials”, escrito por Beb Albahari, Peter Drayton y Brand Merril y publicado por O’Reilly en 2000.

“C# Programming with the Public Beta”, escrito por Burton Harvey, Simon Robinson, Julian Templeman y Karli Watson y publicado por Wrox Press en 2000.

“Inside C#”, escrito por Tom Archer y publicado por Microsoft en 2000 • “Presenting C#”, escrito por Christoph Wille y publicado por Sams Publishing en 2000.

[ÍNDICE](#)



---

**DUDAS**