

# Programación 1

## Tema 1

---

Problemas de tratamiento de información,  
algoritmos y programas



Escuela de  
Ingeniería y Arquitectura  
**Universidad** Zaragoza



# Problemas, algoritmos y programas

---

- Problemas de **tratamiento de información**
  - Objetivo: resolución **automática** del problema
  - ¿Quién? Un computador
  - Necesidad de **programarlo**

# Algoritmo

---

## □ Conjunto de operaciones

- ordenado,
- finito,
- carente de ambigüedades,

que permite hallar la solución de un problema [de tratamiento de información]



# «Deberes» para hoy

---

- Análisis de una receta para hacer tortilla de patata
  - Estructura
  - Modos de las formas verbales



# Índice

---

- ❑ Problemas de tratamiento de información
- ❑ Algoritmos y programas
- ❑ Ejemplos de programas C++
- ❑ Propiedades de un algoritmo



# Ejemplos de problemas de tratamiento de información

---

- ❑ Facilitar la escritura, edición, impresión y preservación digital de un texto
- ❑ Gestionar la información académica de los alumnos de la Universidad de Zaragoza
- ❑ Averiguar el número primo que sigue a 104743
- ❑ Permitir que una o varias personas jueguen en un entorno virtual persiguiendo un determinado objetivo
- ❑ Guiar el rayo láser que realiza queratectomía fotorrefractiva para corregir la miopía en ojos humanos
- ❑ Permitir que varias personas compartan entre sí en Internet información personal como noticias, fotografías, etc.

# Problemas, algoritmos y programas

---

- **Problema** (de tratamiento de información)



- **Método para su resolución**



- **Algoritmo**



- **Programa**

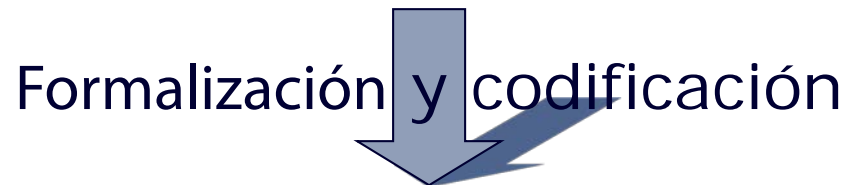
# Problemas, algoritmos y programas

---

- **Problema** (de tratamiento de información)



- **Método para su resolución**



- **Programa**

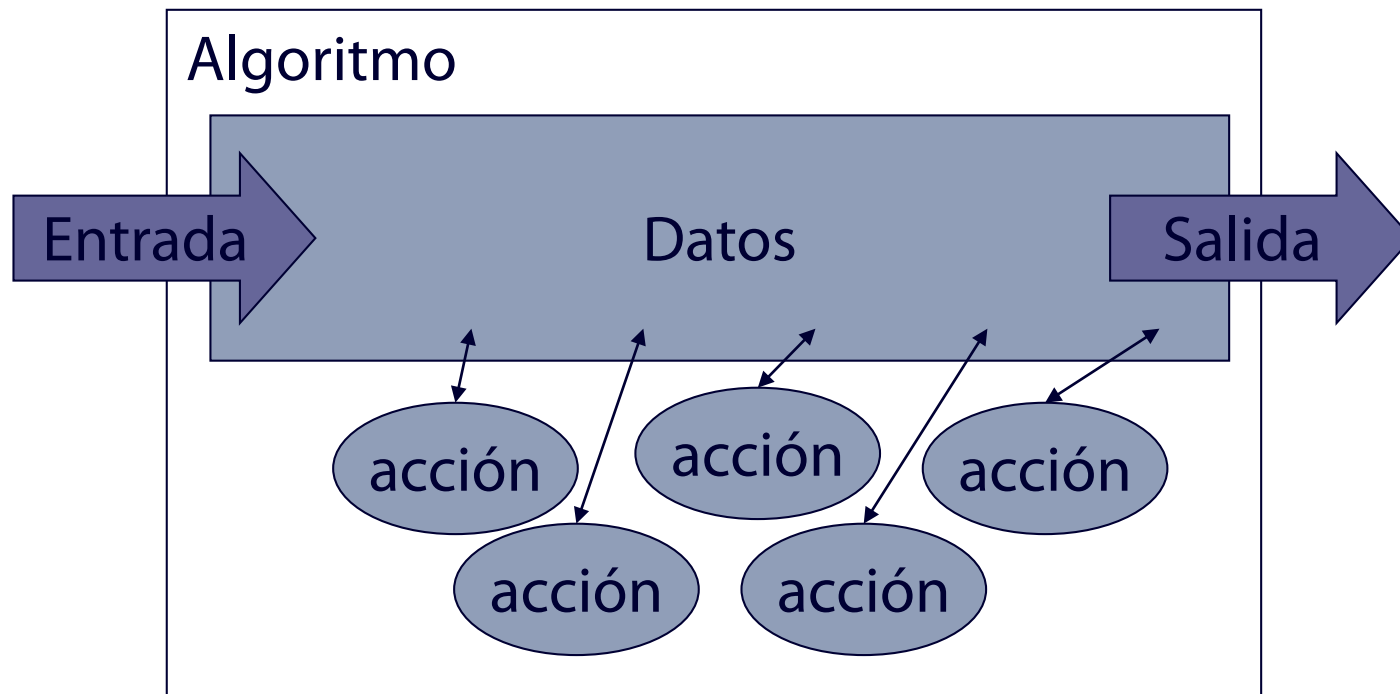


# Algoritmo

---

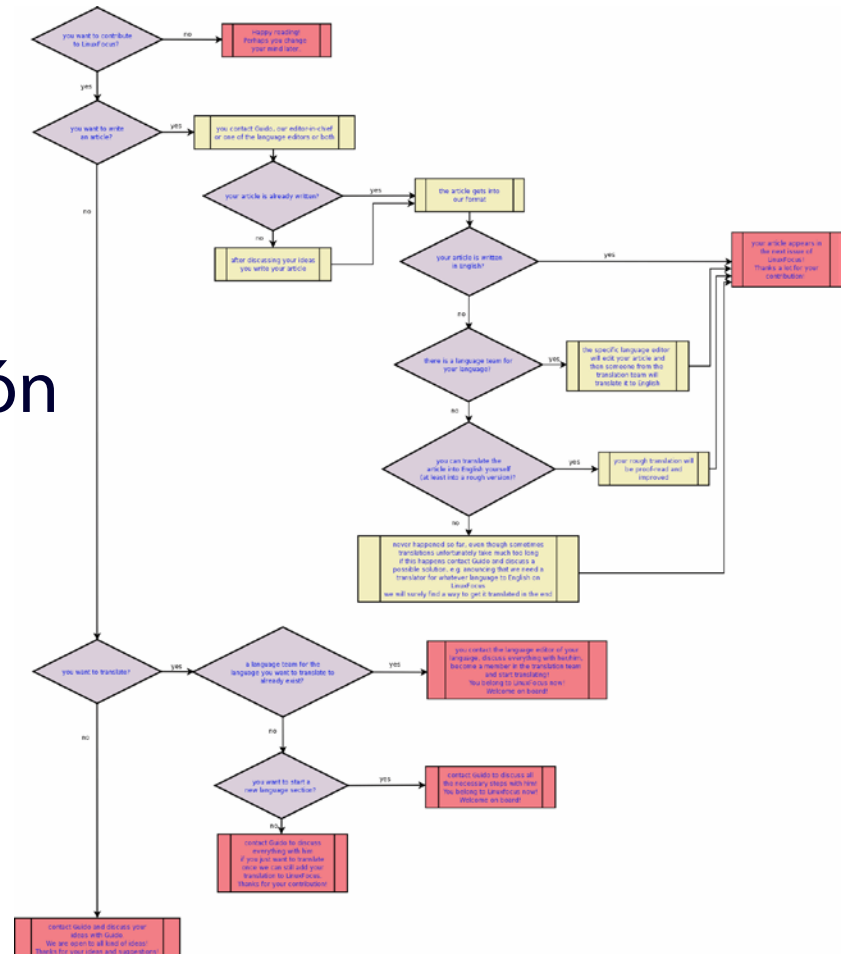
- Conjunto ordenado y finito de operaciones, carente de ambigüedades, que permite hallar la solución de un problema de tratamiento de información
- Consta de
  - **Descripción de la información** asociada al problema
  - **Descripción del modo de tratamiento** de esta información.

# Esquema de algoritmo



# Expresión de un algoritmo

- ❑ Lenguaje natural
- ❑ Notación algorítmica
- ❑ Notación gráfica
  - Diagramas de flujo
- ❑ Lenguaje de programación
  - Ada, Pascal, Módulo-2, C
  - **C++**, Java
  - Lisp, Prolog
  - Fortran, Cobol



# Algoritmo en lenguaje natural

## □ Ingredientes para 4 comensales

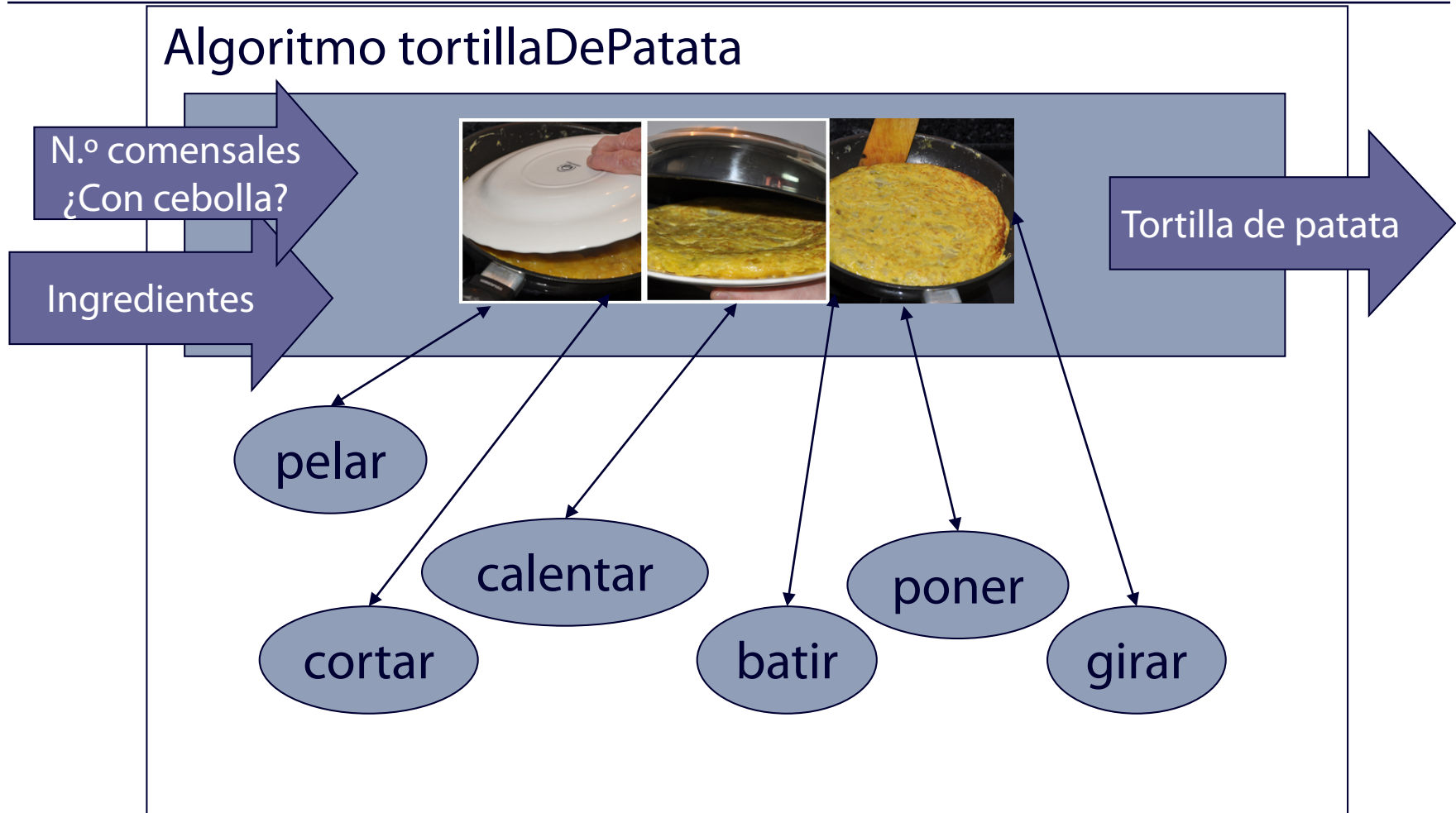
- 4 huevos
- Medio kilo de patatas
- Media cebolla
- Aceite de oliva
- Sal

## □ Elaboración:

- Corte las patatas en trocitos bien finos. Ponga a calentar abundante aceite de oliva en la sartén. Ponga las patatas en la sartén cuando el aceite esté bien caliente (nunca debe humear). Añada un poco de sal. Si la quiere con cebolla, añada la cebolla picada. Cuando las patatas estén bien doraditas, sáquelas y escúrralas. Bata bien los huevos, con una pizca de sal. Añada las patatas ya fritas y mezcle bien. Retire el aceite sobrante de la sartén y vuelva a ponerla al fuego. Cuando la sartén esté bien caliente, eche la mezcla de huevo y patatas. Cuando ya está hecha o cuajada por debajo, darle la vuelta con un plato plano o una tapadera.



# Algoritmo para cocinar una tortilla de patata



(1)  $\frac{1}{2}$  Kg patatas en tiritas

1 cebolla picadita

sal

mezclar

RELLENO

aceite en sartén (  $\frac{1}{2}$  cm de espesor )

calentar a  $\frac{1}{2}$  fuego

(2) 20 min

rehogar a fuego medio, tapando la sartén. Mezclar relleno

### INGREDIENTES

$\frac{1}{2}$  kg. patatas

4 huevos

1 cebolla

250 cc aceite

sal

patatas cocidas

sacar con espumadera

(3) PATATA Y CEBOLLA REHOGADA

aceite en la sartén

quitar aceite volcando sartén

(4)

aceite cubriendo justamente fondo sartén

MEZCLA REHOGADA

extender mezcla en fondo sartén

(5) 10 min

freir a medio fuego hasta cuajar tortilla por abajo

dar  $\frac{1}{2}$  vuelta a tortilla ( ayudarse de una tapadera )

4 huevos en bol

sal

batir bien

mezclar bien

# Algoritmo en notación algorítmica

```
algoritmo tortillaDePatata(  
    personas: datoDeEntrada entero;  
    seQuiereConCebolla:  
        datoDeEntrada booleano);  
{ Versión en una notación algorítmica  
  de la receta de tortilla de patata }
```

## ingredientes

1 huevo por persona  
125 g de patatas por persona  
1/8 de cebolla por persona  
Aceite de oliva  
Sal

## menaje

sartén  
tenedor  
plato

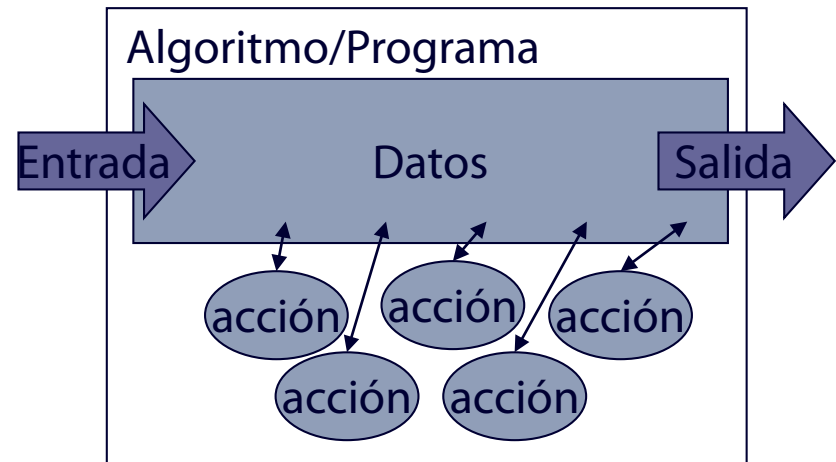
## principio

```
    pelar(patatas);  
    cortar(patatas);  
    calentar(aceite, sartén);  
    mientrasQue no estéBienCaliente(aceite) hacer  
        esperar;  
    finMQ;  
    poner(patatas, sartén);  
    poner(sal, sartén);  
  
    si seQuiereConCebolla entonces  
        picar(cebolla);  
        poner(cebolla, sartén);  
    finSi;  
    mientrasQue no esténDoradas(patatas) hacer  
        esperar;  
    finMQ;  
    batir(huevos);  
    poner(sal, huevos);  
    ...
```

**fin.**

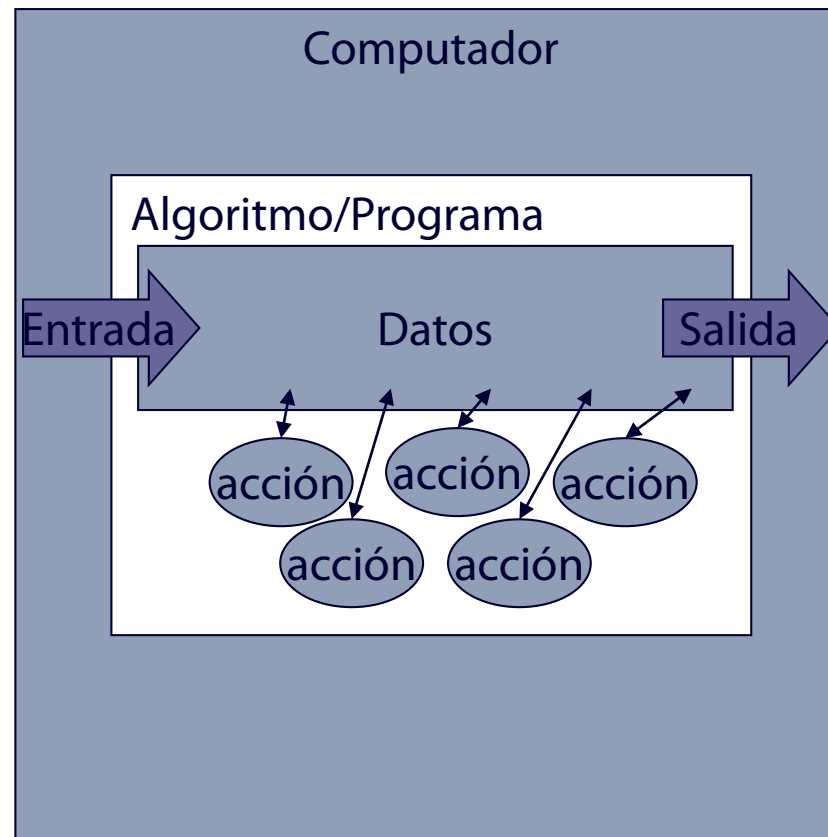
# Nuestro modelo de computador

Computador

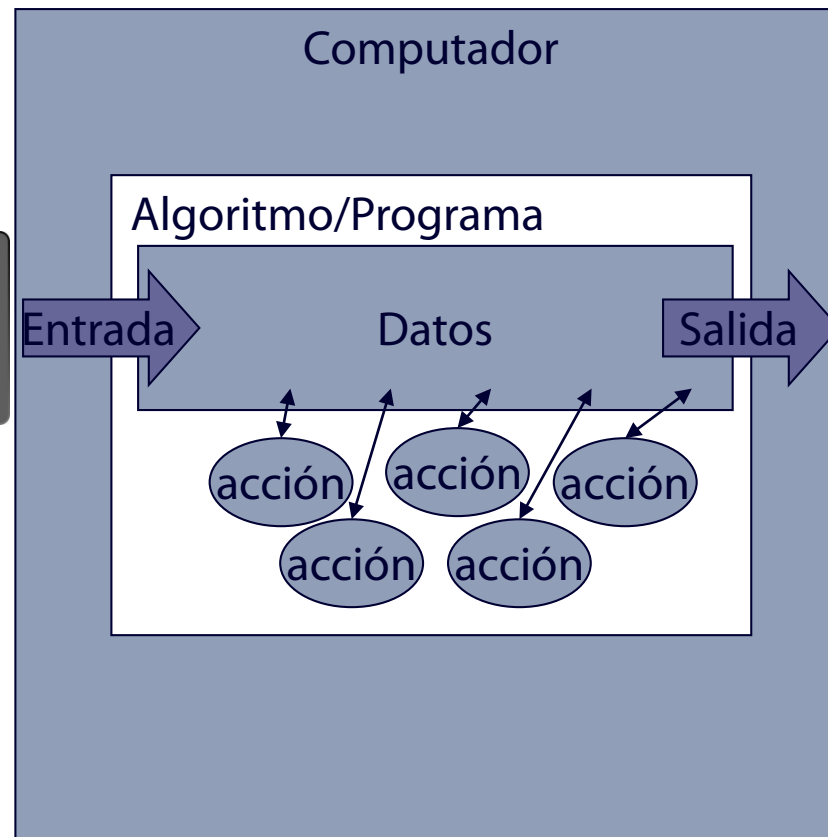




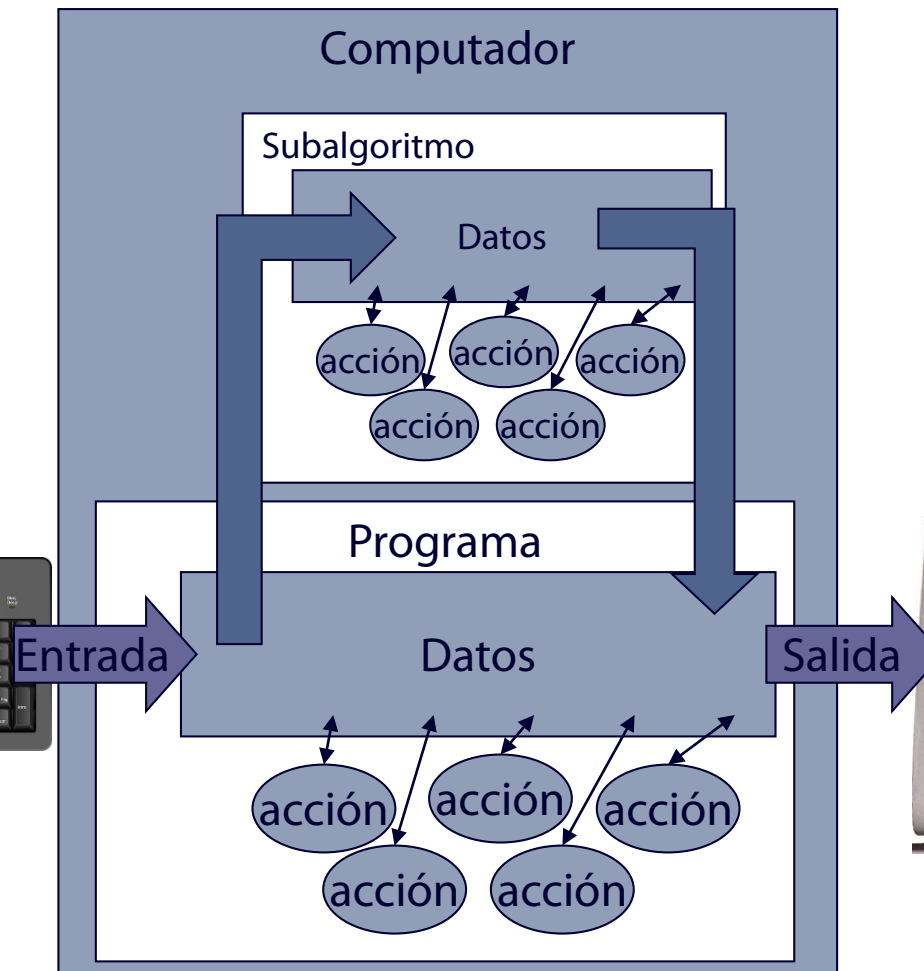
# Nuestro modelo de computador



# Nuestro modelo de computador



# Nuestro modelo de computador



# Un algoritmo en una notación algorítmica

**algoritmo** bienvenida;

*{ Algoritmo que escribe una línea de texto en  
la pantalla con un mensaje de bienvenida }*

**principio**

*{ Acciones a ejecutar cuando sea invocado }*

escribir(pantalla, "Bienvenidos a UNIZAR");

**fin.**

# Un programa en Ada

```
with ada.text_IO;  
  
procedure bienvenida is  
  -- Programa que escribe en la pantalla una línea  
  -- de texto con un mensaje de bienvenida  
  
begin  
  -- Acciones que ejecutará el programa cada vez  
  -- que sea invocado  
  ada.text_IO.put("Bienvenidos a UNIZAR");  
  ada.text_IO.new_line;  
end bienvenida;
```

# Un programa en Java

```
package es.unizar.eina.prog1.cap1;

/**
 * Al construir un programa Java alrededor de esta clase se
 * ejecuta su método «main» que escribe un mensaje de
 * bienvenida a la Universidad.
 */
public class Bienvenida {

    /**
     * Escribe en la pantalla una línea de texto con el mensaje
     * “Bienvenidos a la Universidad”
     */
    public static void main(String[] argumentos) {
        // El código a ejecutar se limita a una sola instrucción
        System.out.println("Bienvenidos a la Universidad");
    }
}
```



# Un programa en Python

```
print ('Bienvenidos a UNIZAR')
```

# Un primer programa en C++

```
#include <iostream>

/*
 * Programa que escribe en la pantalla el mensaje
 * "Bienvenidos a la Universidad"
 */
int main() {
    // primera instrucción
    std::cout << "Bienvenidos a la Universidad" << std::endl;

    // segunda instrucción
    return 0;
}
```



# ¿Cómo se ejecuta el código C++?

## □ Edición del código fuente

```
#include <iostream>

/*
 * Pre: ---
 * Post: Escribe por pantalla el mensaje
 */
int main() {
    // primera instrucción
    std::cout << "Bienvenidos a la Uni

    // segunda instrucción
    return 0;
}
```

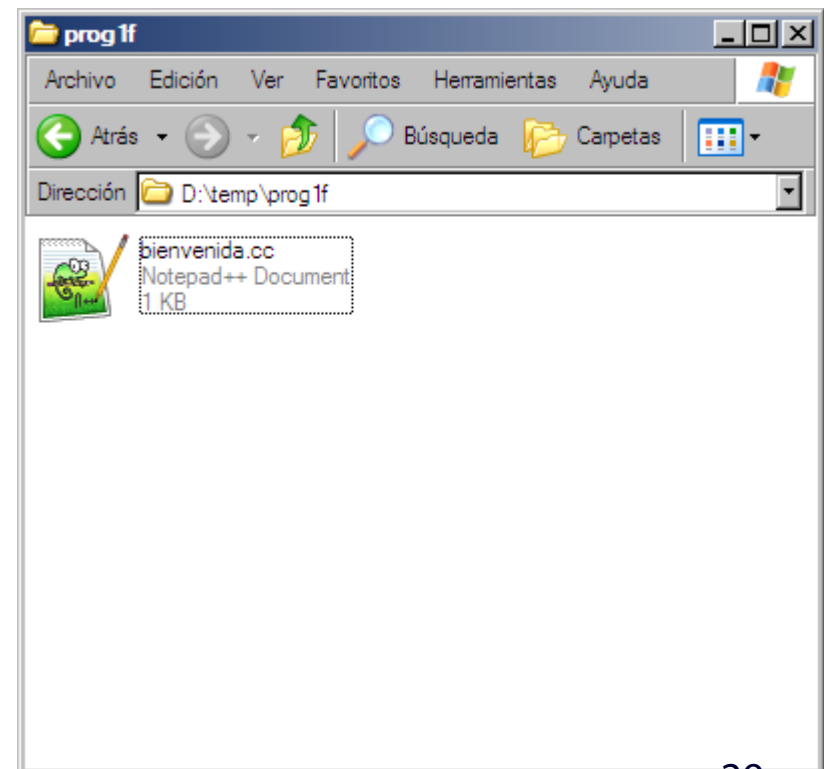
```
D:\bienvenida.cpp - Notepad++
Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Macro  Ejecutar  TextFX  Plugins  Ventana  ?
bienvenida.cpp x

1  #include <iostream>
2
3  /*
4   * Pre: ---
5   * Post: Escribe por pantalla el mensaje
6   *      "Bienvenidos a la Universidad"
7   */
8  int main() {
9      // primera instrucción
10     std::cout << "Bienvenidos a la Universidad" << std::endl;
11
12     // segunda instrucción
13     return 0;
14 }
```

length : 277 lines : 14    Ln : 1 Col : 1 Sel : 0 | 0    Dos\Windows    UTF-8 w/o BOM    INS

# ¿Cómo se ejecuta el código C++?

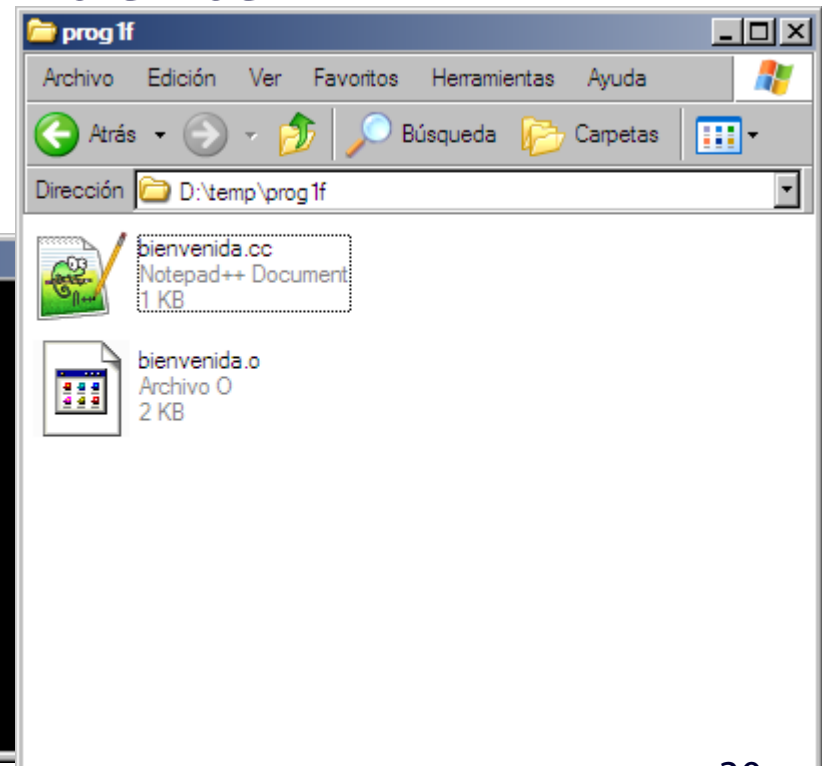
## □ Edición del código fuente



# ¿Cómo se ejecuta el código C++?

- ❑ Edición del código fuente
- ❑ Compilación del código fuente

```
C:\ Símbolo del sistema  
D:\temp\prog1f>g++ -c bienvenida.cc  
D:\temp\prog1f>_
```



# ¿Cómo se ejecuta el código C++?

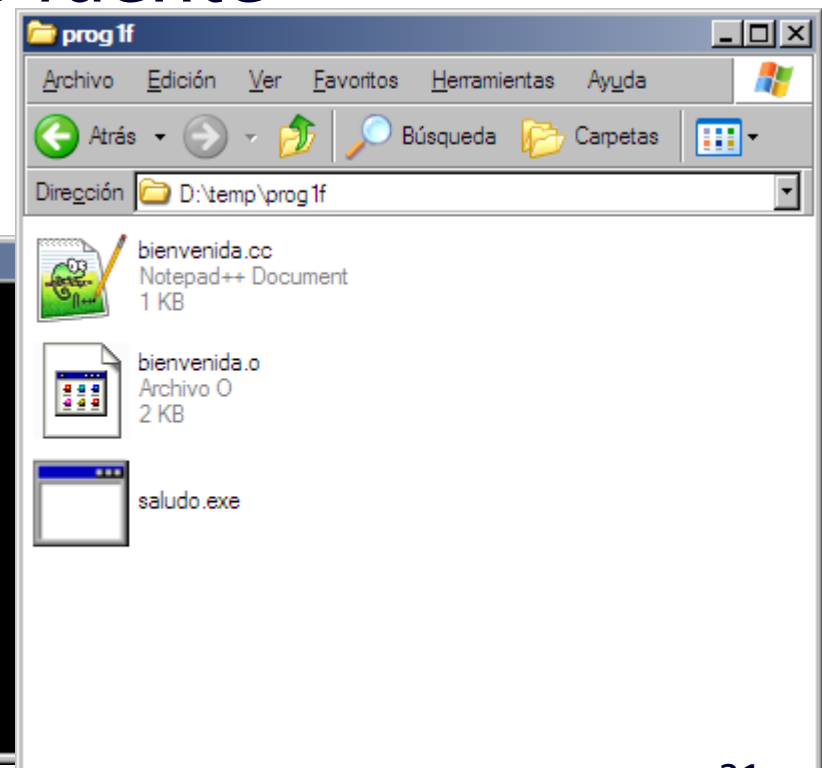
- ❑ Edición del código fuente
- ❑ Compilación del código fuente

```
C:\ Símbolo del sistema

D:\temp\prog1f>g++ -c bienvenida.cc

D:\temp\prog1f>g++ -o saludo bienvenida.o

D:\temp\prog1f>_
```



# ¿Cómo se ejecuta el código C++?

- ❑ Edición del código fuente
- ❑ Compilación del código fuente
- ❑ Ejecución del código ejecutable

```
C:\ Símbolo del sistema

D:\temp\prog1f>g++ -c bienvenida.cc

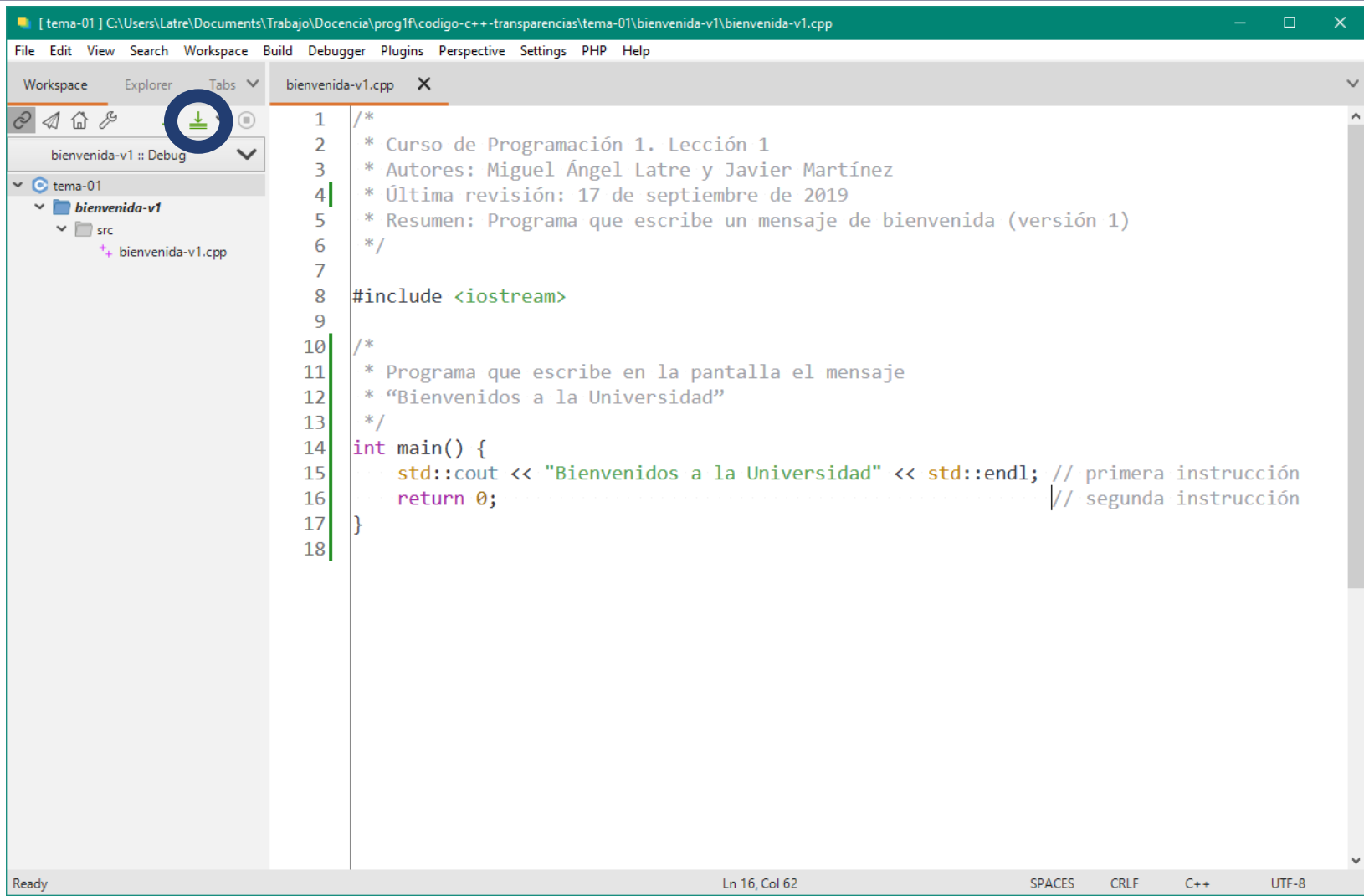
D:\temp\prog1f>g++ -o saludo bienvenida.o

D:\temp\prog1f>saludo
Bienvenidos a la Universidad

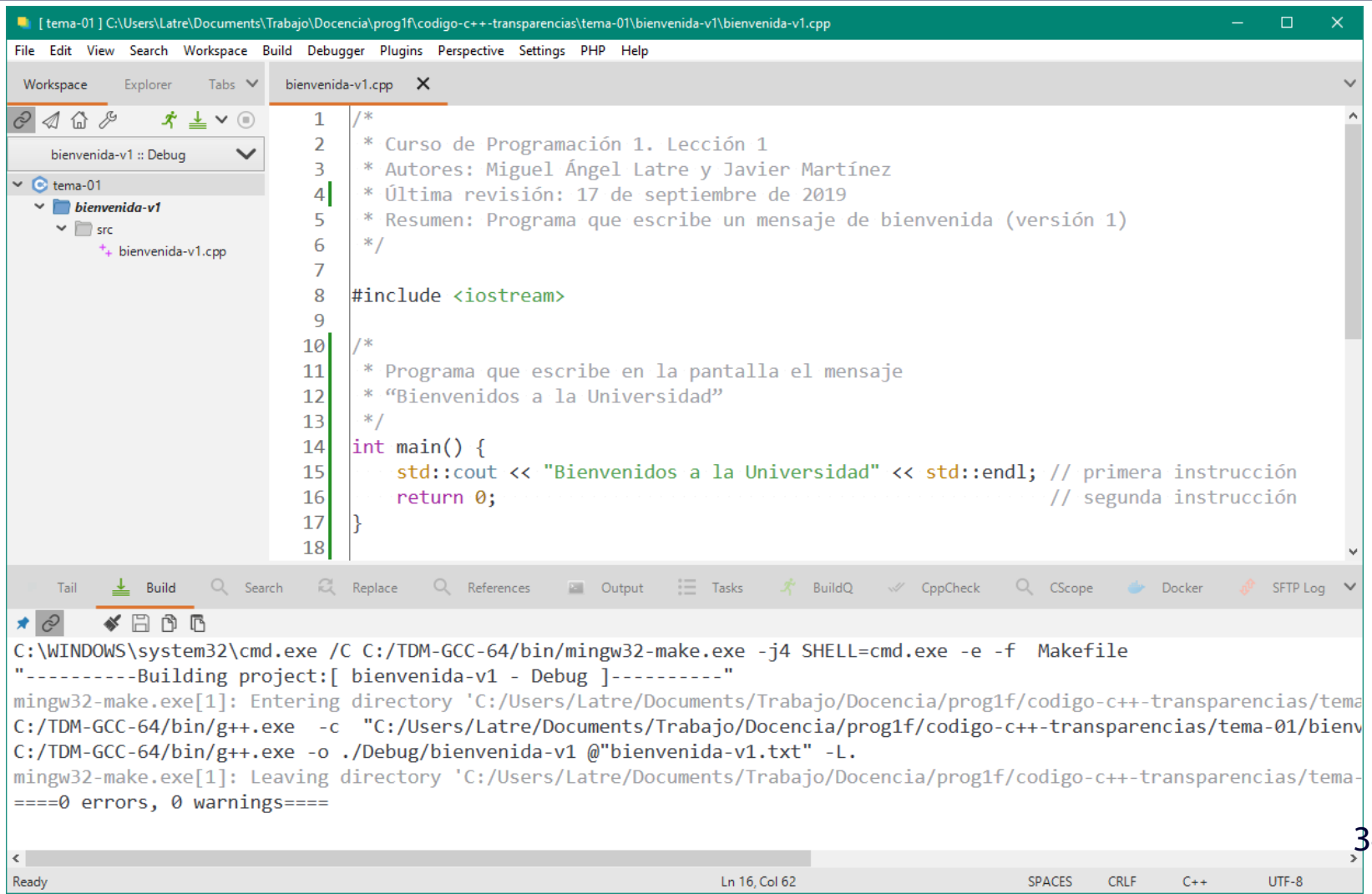
D:\temp\prog1f>_
```



# Compilación en CodeLite



# Compilación en CodeLite



The screenshot shows the CodeLite IDE interface. The top menu bar includes File, Edit, View, Search, Workspace, Build, Debugger, Plugins, Perspective, Settings, PHP, and Help. The left sidebar shows the project structure: 'tema-01' containing a folder 'bienvenida-v1' with a subfolder 'src' and a file 'bienvenida-v1.cpp'. The main editor displays the content of 'bienvenida-v1.cpp' with the following code:

```
1  /*
2  * Curso de Programación 1. Lección 1
3  * Autores: Miguel Ángel Latre y Javier Martínez
4  * Última revisión: 17 de septiembre de 2019
5  * Resumen: Programa que escribe un mensaje de bienvenida (versión 1)
6  */
7
8  #include <iostream>
9
10 /*
11 * Programa que escribe en la pantalla el mensaje
12 * "Bienvenidos a la Universidad"
13 */
14 int main() {
15     std::cout << "Bienvenidos a la Universidad" << std::endl; // primera instrucción
16     return 0; // segunda instrucción
17 }
18
```

The bottom panel shows the output of the compilation process:

```
C:\WINDOWS\system32\cmd.exe /C C:/TDM-GCC-64/bin/mingw32-make.exe -j4 SHELL=cmd.exe -e -f Makefile
"-----Building project:[ bienvenida-v1 - Debug ]-----"
mingw32-make.exe[1]: Entering directory 'C:/Users/Latre/Documents/Trabajo/Docencia/prog1f/codigo-c++-transparencias/tema-01/bienvenida-v1'
C:/TDM-GCC-64/bin/g++.exe -c "C:/Users/Latre/Documents/Trabajo/Docencia/prog1f/codigo-c++-transparencias/tema-01/bienvenida-v1.cpp" -o ./Debug/bienvenida-v1.o
C:/TDM-GCC-64/bin/g++.exe -o ./Debug/bienvenida-v1 @"bienvenida-v1.txt" -L.
mingw32-make.exe[1]: Leaving directory 'C:/Users/Latre/Documents/Trabajo/Docencia/prog1f/codigo-c++-transparencias/tema-01/bienvenida-v1'
====0 errors, 0 warnings====
```

The status bar at the bottom indicates 'Ready', 'Ln 16, Col 62', 'SPACES', 'CRLF', 'C++', and 'UTF-8'.



# Compilación en CodeLite

The screenshot displays the CodeLite IDE interface. The main editor window shows a C++ file named `bienvenida-v1.cpp` with the following content:

```
1 /*
2  * Curso de Programación 1. Lección 1
3  * Autores: Miguel Ángel Latre y Javier Martínez
4  * Última revisión: 17 de septiembre de 2019
5  * Resumen: Programa que escribe un mensaje de bienvenida (versión 1)
6  */
```

The Explorer panel on the left shows the project structure, with the `Run` button (a green play icon) circled in blue. Below the IDE, a `codelite-terminal` window is open, showing the output of the program:

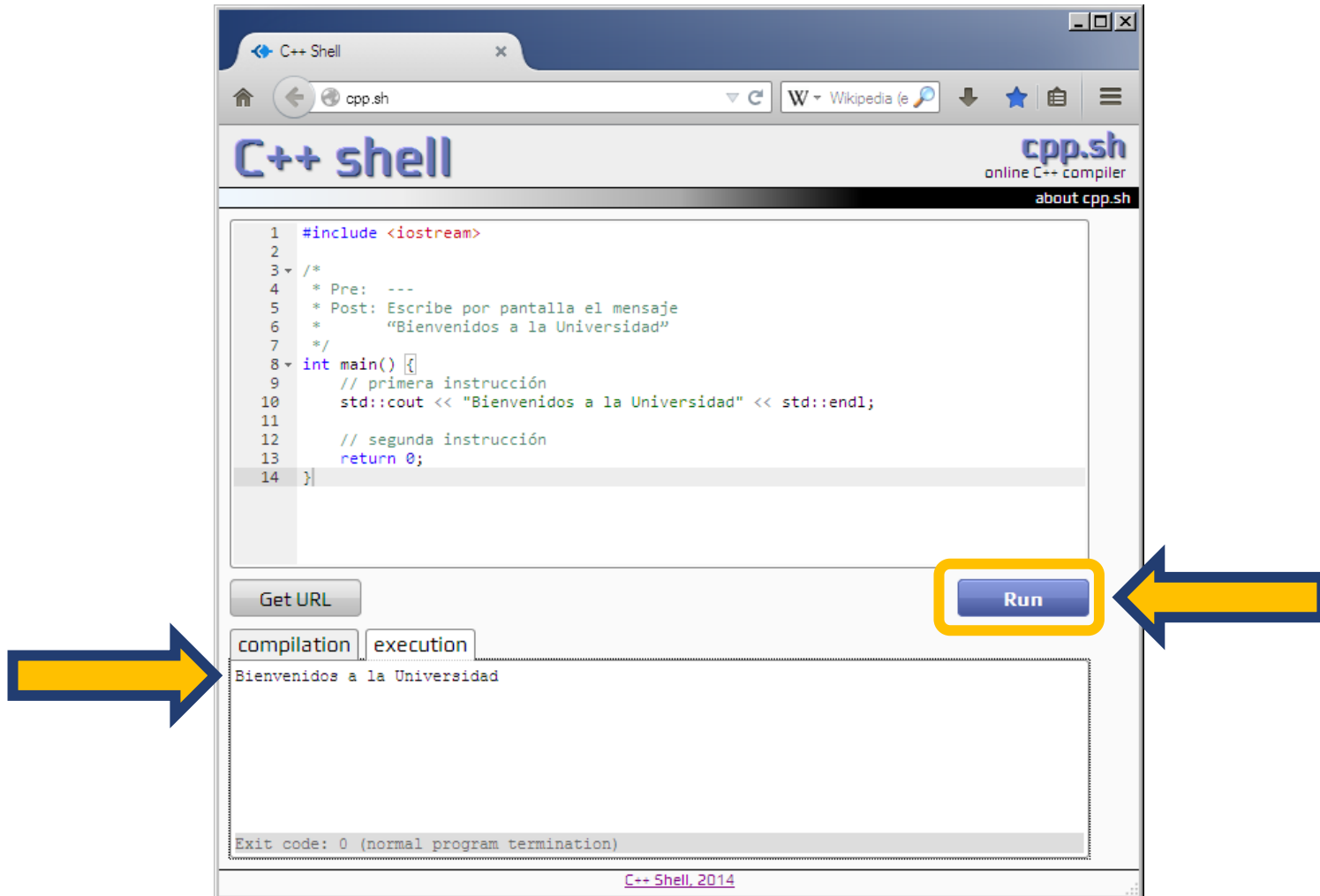
```
Bienvenidos a la Universidad
Hit any key to continue...
```

The terminal window also shows the command prompt and the execution of the program, including the path to the executable and the output file.

At the bottom of the IDE, the status bar indicates the current line and column: `Ln 16, Col 62`, and the file encoding: `UTF-8`.



# Compilación y ejecución en cpp.sh



The screenshot shows the cpp.sh online C++ compiler interface. The browser window has a single tab titled "C++ Shell" and the address bar shows "cpp.sh". The page header includes the "C++ shell" logo and the text "online C++ compiler" and "about cpp.sh". The main area contains a code editor with the following C++ code:

```
1 #include <iostream>
2
3 /*
4  * Pre: ---
5  * Post: Escribe por pantalla el mensaje
6  *       "Bienvenidos a la Universidad"
7  */
8 int main() {
9     // primera instrucción
10    std::cout << "Bienvenidos a la Universidad" << std::endl;
11
12    // segunda instrucción
13    return 0;
14 }
```

Below the code editor, there is a "Get URL" button and two tabs: "compilation" and "execution". The "execution" tab is selected, and the output area displays "Bienvenidos a la Universidad". At the bottom of the output area, it says "Exit code: 0 (normal program termination)". A yellow arrow points to the "Run" button, and another yellow arrow points to the output area.

# Un primer programa en C++

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

/*
 * Programa que escribe en la pantalla el mensaje
 * "Bienvenidos a la Universidad"
 */
int main() {
    // primera instrucción
    cout << "Bienvenidos a la Universidad" << endl;

    // segunda instrucción
    return 0;
}
```

# Un programa que lee del teclado

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

/*
 * Programa que solicita un número entero al usuario y lo
 * escribe en pantalla.
 */
int main() {
    cout << "Escriba un número entero: ";

    int numero;
    cin >> numero;

    cout << "El número escrito es el " << numero << endl;

    return 0;
}
```



# Un programa que lee del teclado

## Ejecución

Escriba un número entero:



# Un programa que lee del teclado

## Ejecución

Escriba un número entero: 2019



# Un programa que lee del teclado

## Ejecución

Escriba un número entero: 2019↵



# Un programa que lee del teclado

## Ejecución

Escriba un número entero: 2019

El número escrito es el 2019



# Programa interactivo que hace un cálculo

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

const double PI = 3.14159265358979323846;

/*
 * Programa que solicita al usuario la longitud de un radio y escribe en la
 * pantalla el área del círculo correspondiente.
 */
int main() {
    cout << "Escriba el radio de un círculo: ";

    double r;
    cin >> r;

    cout << "El área de un círculo de radio " << r << " es "
         << PI * r * r << endl;

    return 0;
}
```





# Programa interactivo que hace un cálculo. Ejecución

Escriba el radio de un círculo: 2.5

El área de un círculo de radio 2.5 es 19.635

# Un programa que utiliza una función auxiliar

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

/*
 * Pre:  agnoNacimiento <= agnoActual
 * Post: Ha devuelto el valor entero que representa el número de años
 *       cumplidos en el año «agnoActual» por una persona nacida en el
 *       año «agnoNacimiento».
 */
int calcularEdad(int agnoNacimiento, int agnoActual) {
    return agnoActual - agnoNacimiento;
}
```

# Un programa que utiliza una función auxiliar

```
const int AGNO_ACTUAL = 2019;

/*
 * Programa que ha preguntado al usuario por el año de su
 * nacimiento y le ha informado de la edad que ha cumplido o
 * cumplirá este año.
 */
int main() {
    cout << "Escribe el año de tu nacimiento: ";
    int agnoNacimiento;
    cin >> agnoNacimiento;

    cout << "En el " << AGNO_ACTUAL
         << " has cumplido o cumplirás "
         << edad(agnoNacimiento, AGNO_ACTUAL) << " años."
         << endl;
    return 0;
}
```



# Un programa que utiliza una función auxiliar. Ejecución

Escribe el año de tu nacimiento: 2001

En el 2019 has cumplido o cumplirás 18 años.

# Funciones y procedimientos

---

- ❑ **Algoritmos** que resuelven un problema concreto de tratamiento de información
- ❑ Pueden invocarse unos a otros
- ❑ En C++, siempre los denominaremos funciones

# Funciones y procedimientos

---

- Trabajan con datos
  - **Datos de entrada**
    - **Parámetros**
    - Datos leídos de teclado
  - **Datos de salida**
    - **Valor devuelto**
    - Datos escritos en la pantalla
- Tienen un **cuerpo**
  - Acciones algorítmicas que manipulan los datos para resolver el problema concreto
- Se describen con una **especificación**
  - Documentación sobre lo que realiza el procedimiento o función
  - Puede ser formal, a través de **precondición** y **postcondición**

# Propiedades de un algoritmo

---

## □ **Imprescindibles**

- Corrección
- Legibilidad

## □ **Deseables**

- Generalidad
- Reusabilidad
- Eficiencia
- Independencia de la máquina y del lenguaje
- Simplicidad
- Fiabilidad

# Programa sintácticamente incorrecto

```
#include <iostream>

/*
 * Programa que escribe en la pantalla el mensaje
 * "Bienvenidos a La Universidad"
 */
{
    // primera instrucción
    cout << "Bienvenidos a La Universidad" << endl;

    // segunda instrucción
    return 0;
}
```



# Programa formalmente incorrecto

```
#include <iostream>

using namespace std;

/*
 * Programa que escribe en la pantalla la suma de los
 * números del 1 al 5.
 */
int main() {
    cout << 1 + 2 + 3 + 4 << endl;
    return 0;
}
```

# Propiedades de un algoritmo

---

## □ **Imprescindibles**

- Corrección
- **Legibilidad**

## □ **Deseables**

- Generalidad
- Reusabilidad
- Eficiencia
- Independencia de la máquina y del lenguaje
- Simplicidad
- Fiabilidad

# Programa C++ ilegible

```
#include <iostream>
#include <iomanip>
using namespace std; void o(double oo) {
const double ooo=3.14159265358979323846;
cout<<setw(7)<<oo<<setw(16)<<2.0*ooo*oo<<
endl;}int main(){cout<<setprecision(2); cout
<<fixed; cout<<setw(7)<<"Radi o"<<setw(20)<<
"Circunferenci a"<<endl; cout<<setw(7)<<
"====="<<setw(20)<<"===== " <<endl; o
(1.234); o(5.012); o(11.5178); cout<<endl;
return 0; }
```

# Propiedades de un algoritmo

---

## □ **Imprescindibles**

- Corrección
- Legibilidad

## □ **Deseables**

- **Generalidad**
- **Reusabilidad**
- Eficiencia
- Independencia de la máquina y del lenguaje
- Simplicidad
- Fiabilidad

# Generalidad

```
/*  
 * Pre: ---  
 * Post: Devuelve la suma de los enteros  
 * comprendidos en el intervalo [1, 100]  
 */  
int sumaDe1A100() {  
    int resultado = 0;  
    for (int i = 1; i <= 100; i++) {  
        resultado = resultado + i;  
    }  
    return resultado;  
}
```

# Generalidad

```
/*  
 * Pre:  inicial <= final  
 * Post: Ha devuelto la suma de los enteros  
 *       comprendidos en el intervalo [inicial, final].  
 */  
int suma(int inicial, int final) {  
    int resultado = 0;  
    for (int i = inicial; i <= final; i++) {  
        resultado = resultado + i;  
    }  
    return resultado;  
}
```

# Propiedades de un algoritmo

---

## □ **Imprescindibles**

- Corrección
- Legibilidad

## □ **Deseables**

- Generalidad
- Reusabilidad
- **Eficiencia**
- Independencia de la máquina y del lenguaje
- Simplicidad
- Fiabilidad

# Eficiencia

```
/*  
 * Pre: inicial <= final  
 * Post: Ha devuelto la suma de los enteros  
 * comprendidos en el intervalo  
 * [inicial, final].  
 */  
int sumaEficiente(int inicial, int final) {  
    return (inicial + final)  
        * (final - inicial + 1) / 2;  
}
```



# Propiedades de un algoritmo

---

## □ **Imprescindibles**

- Corrección
- Legibilidad

## □ **Deseables**

- Generalidad
- Reusabilidad
- Eficiencia
- **Independencia** de la máquina y del lenguaje
- **Simplicidad**
- **Fiabilidad**

# Resumen

---

- ❑ Problemas de tratamiento de información
- ❑ Algoritmos y programas
- ❑ Ejemplos de programas C++
- ❑ Propiedades de un algoritmo

# ¿Cómo se estudia este tema?

---

- Repasando las transparencias
- Leyendo las secciones 1.1, 1.2 y 1.4 del capítulo 1 de los apuntes del profesor Martínez
  - Disponibles en Moodle
- Ejecutando los programas presentados en un entorno de ejecución en línea, como <http://cpp.sh/>
- Ejecutando los programas en el entorno CodeLite:
  - Instalándolo según las instrucciones de «[Instalación de CodeLite](#)» en la web de la asignatura
  - Leyendo la descripción del entorno CodeLite en la sección 1.3 de la 1.ª práctica de la asignatura.
    - Disponible en Moodle a lo largo de la semana

# ¿Cómo se estudia este tema?

---

- Código fuente de este tema
  - Se puede copiar y pegar
  - También está disponible en el repositorio <https://github.com/prog1-eina/tema-01-introduccion>



# ¿Cómo se estudia este tema?

GitHub - prog1-eina / tema-01-introduccion

Why GitHub? Enterprise Explore Marketplace Pricing Search Sign in Sign up

prog1-eina / tema-01-introduccion Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Security Insights

Código del tema 1 (Problemas de tratamiento de información, algoritmos y programas) de la asignatura Programación 1. Escuela de Ingeniería y Arquitectura de la Universidad de Zaragoza

2 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Find File Clone or download

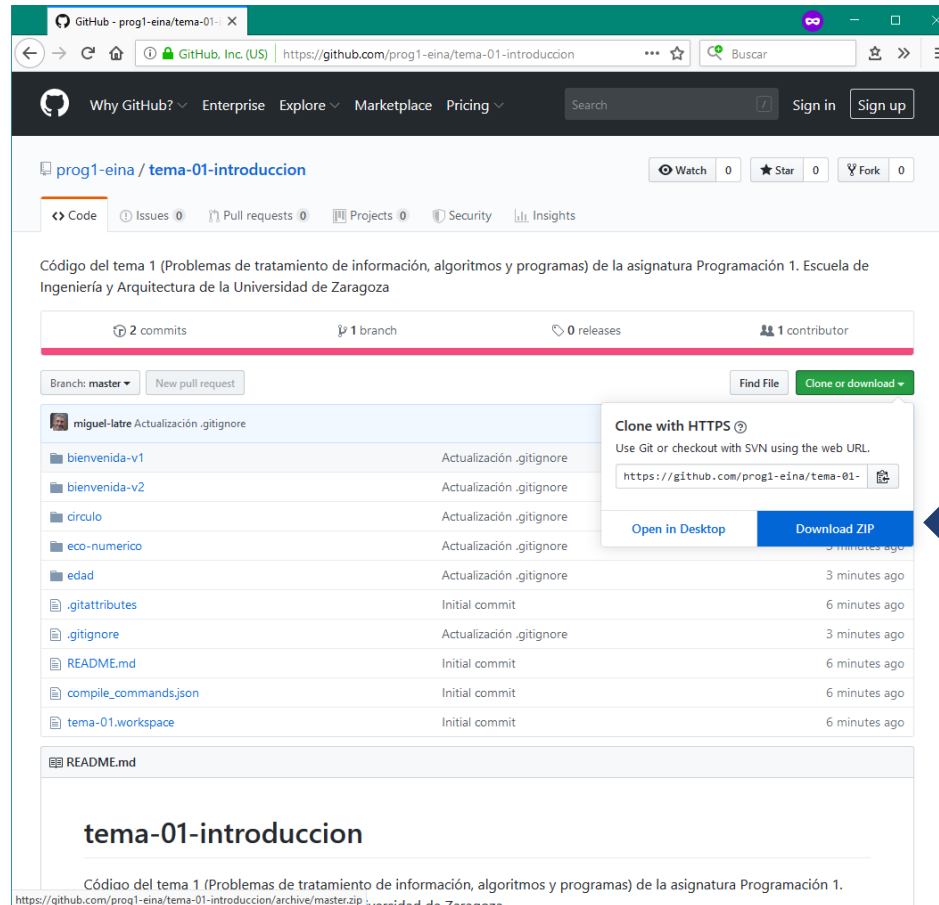
File	Commit	Time
bienvenida-v1	Actualización .gitignore	2 minutes ago
bienvenida-v2	Actualización .gitignore	2 minutes ago
circulo	Actualización .gitignore	2 minutes ago
eco-numerico	Actualización .gitignore	2 minutes ago
edad	Actualización .gitignore	2 minutes ago
.gitattributes	Initial commit	5 minutes ago
.gitignore	Actualización .gitignore	2 minutes ago
README.md	Initial commit	5 minutes ago
compile_commands.json	Initial commit	5 minutes ago
tema-01.workspace	Initial commit	5 minutes ago

README.md

## tema-01-introduccion

Código del tema 1 (Problemas de tratamiento de información, algoritmos y programas) de la asignatura Programación 1. Escuela de Ingeniería y Arquitectura de la Universidad de Zaragoza.

# ¿Cómo se estudia este tema?



GitHub - prog1-eina/tema-01-introduccion

Why GitHub? Enterprise Explore Marketplace Pricing Search Sign in Sign up

prog1-eina / tema-01-introduccion Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Security Insights

Código del tema 1 (Problemas de tratamiento de información, algoritmos y programas) de la asignatura Programación 1. Escuela de Ingeniería y Arquitectura de la Universidad de Zaragoza

2 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Find File Clone or download

miguel-latre Actualización .gitignore

bienvenida-v1	Actualización .gitignore	3 minutes ago
bienvenida-v2	Actualización .gitignore	3 minutes ago
circulo	Actualización .gitignore	3 minutes ago
eco-numerico	Actualización .gitignore	3 minutes ago
edad	Actualización .gitignore	3 minutes ago
.gitattributes	Initial commit	6 minutes ago
.gitignore	Actualización .gitignore	3 minutes ago
README.md	Initial commit	6 minutes ago
compile_commands.json	Initial commit	6 minutes ago
tema-01.workspace	Initial commit	6 minutes ago

README.md

## tema-01-introduccion

Código del tema 1 (Problemas de tratamiento de información, algoritmos y programas) de la asignatura Programación 1.

<https://github.com/prog1-eina/tema-01-introduccion/archive/master.zip>

Escuela de Ingeniería y Arquitectura Universidad Zaragoza

# ¿Cómo se estudia este tema?

- ❑ Descomprimir
- ❑ Abrir el fichero tema-01.workspace desde CodeLite
- ❑ Compilar y ejecutar cada proyecto
- ❑ Configurar una copia del compilador «C++ (Prog1)» cuando lo solicite

