



# PROGRAMACIÓN I C++

UCM

Grado en Estadística Aplicada. EUE.

## Problemas, algoritmos y programas

T  
E  
M  
A  
1

Introducción

Programación

Algoritmo

Características de los algoritmos

Lenguajes algorítmicos

Lenguajes de programación

Desarrollo de programas



C++

## Introducción

**Informática (Ciencia de la computación) (RAE)**

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

**Computadora (RAE)**

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



C++

## Introducción



C++

## Introducción

5

**Hardware-** Conjunto de componentes que integran la parte material del ordenador

1 Bit = 0 / 1

1 Byte = 8 bits = 1 carácter

1 Kilobyte (KB) = 1024 Bytes

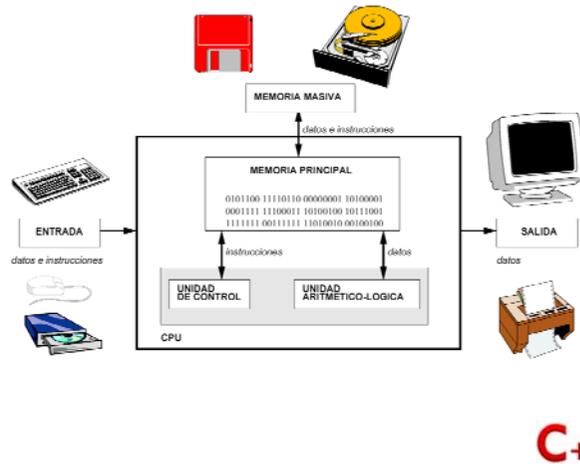
1 Megabyte (MB) = 1024 KB

1 Gigabyte (GB) = 1024 MB

1 Terabyte (TB) = 1024 GB

1 Petabyte (PB) = 1024 TB

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



## Introducción

6

□ **Software-** Conjunto de programas, instrucciones y reglas informáticas para ejecutar tareas en un ordenador



## Todo el mundo debería aprender a programar

7



<https://www.youtube.com/watch?v=OWsyrnOBsJs>



<https://youtu.be/1bDK1-U1edE>

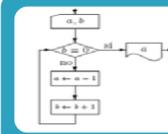
C++

## La programación

8



**Análisis del problema**, estableciendo con precisión lo que se plantea. **(QUÉ Especificación)**



**Solución conceptual** del problema, describiendo un método (algoritmo) que lo resuelva. **(CÓMO)**



**Escritura del algoritmo** en un lenguaje de programación.

C++

Calcular la media

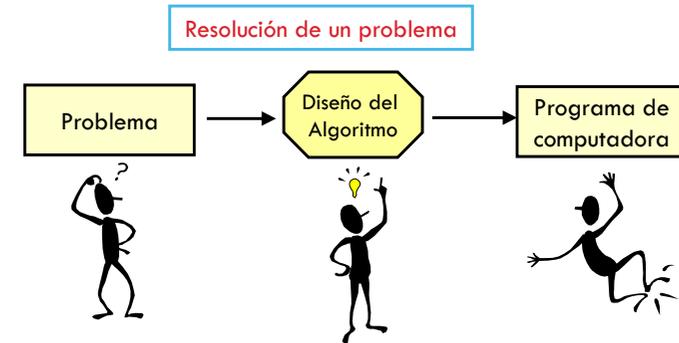
Un número dado es primo?



C++

## La programación

- Para llegar a ser un programador eficaz se necesita aprender a resolver problemas de un modo **riguroso y sistemático**.
- Esto significa que sólo se puede llegar a realizar un buen programa con el **diseño previo de un algoritmo**.



C++

## Algoritmo

- Un algoritmo es la descripción precisa de los pasos que nos llevan a la solución de un problema planteado
- Estos pasos son, en general, acciones u operaciones que se efectúan sobre ciertos objetos
- La descripción de un algoritmo afecta a tres partes: entrada (datos), proceso (instrucciones) y salida (resultados)



C++

## Algoritmo

- Para poder definir correctamente un problema, es conveniente responder a las siguientes preguntas:
  - ¿Qué entradas se requieren?
  - ¿Cuál es la salida deseada?
  - ¿Qué método produce la salida deseada?

### Ejemplo:

*Se desea obtener el coste final de un automóvil sabiendo que el importe inicial es de 12.000 € y los descuentos a aplicar son del 10% si la venta se produce antes de Junio de 2016 y del 20% si se produce con posterioridad.*

**Entradas:** Coste original y los descuentos según el mes.

**Salidas:** Coste del automóvil hasta Junio, coste del automóvil a partir de Junio.

**Proceso:** Cálculo del descuento aplicado hasta Junio y a partir de Junio.



C++

## Características de un algoritmo

13

### 1. Precisión

Un algoritmo debe expresarse de forma no ambigua. La precisión afecta por igual a dos aspectos:

- Al **orden** (encadenamiento o concatenación) de los pasos que han de llevarse a cabo.
- Al **contenido** de las mismas, pues cada paso debe "saberse realizar" con toda precisión, de forma automática.

Por ejemplo, **una receta de cocina**, es un método, pero carece de precisión para ser un algoritmo, al tener expresiones de la forma "**una pizca de sal**"



C++

## Características de un algoritmo

14

### 2. Determinismo

Todo algoritmo debe responder del mismo modo ante las mismas condiciones.

La **acción de barajar un mazo de cartas** no es un algoritmo, ya que es y debe ser un proceso no determinista.

### 3. Finitud

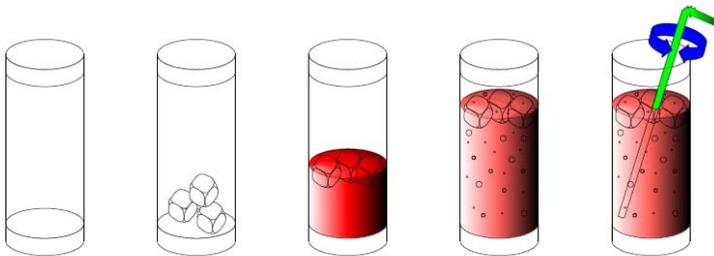
La descripción de un algoritmo debe ser finita.



C++

## Preparación de un tinto de verano

15



- Tomar un vaso.
- Colocar algunos cubitos de hielo en el vaso.
- Echar vino tinto en el vaso.
- Añadir gasosa al contenido del vaso.
- Agitar el contenido.

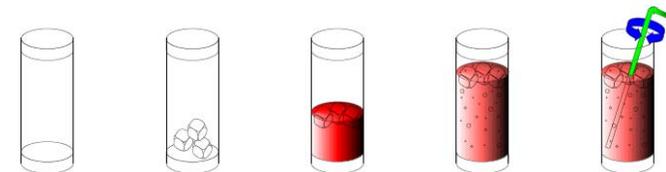


C++

## Preparación de un tinto de verano

16

- Tomar un vaso vacío.
- Colocar tres cubitos de hielo en el vaso.
- Echar vino tinto hasta la mitad del vaso.
- Añadir gasosa hasta llenar el vaso.
- Agitar tres segundos el contenido.



C++

## Preparación de un tinto de verano

17

1. Tomar un vaso vacío.
2. Colocar tres cubitos de hielo en el vaso.
  - a) Sacar la cubitera del congelador.
  - b) Rociar la parte inferior con agua.
  - c) REPETIR
    - d) Extraer un cubito.
    - e) Echarlo al vaso.
  - f) HASTA QUE el no de cubitos sea 3.
  - g) Rellenar los huecos de la cubitera con agua.
  - h) Meter de nuevo la cubitera en el congelador.
3. Echar vino tinto hasta la mitad del vaso.
4. Añadir gaseosa hasta llenar el vaso.
5. Agitar tres segundos el contenido.



C++

## Calcular la media de tres números con Calculadora

18

1. Pulsar la tecla "ON"
2. Teclear el primer numero
3. Pulsar la tecla "+"
4. Teclear el segundo numero
5. Pulsar la tecla "+"
6. Teclear el tercer numero
7. Pulsar la tecla "/"
8. Pulsar la tecla "3"
9. Pulsar la tecla "="
10. La media de los tres números aparece en la pantalla
11. Pulsar la tecla "OFF"



C++

## Cualidades deseables de un algoritmo

19

Un algoritmo ha de ser suficientemente general y que se ejecute eficientemente.

### 1. Generalidad

Es deseable que un algoritmo sirva para una clase de problemas lo más amplia posible.

la clase de problemas para **resolver una ecuación de segundo grado**,  $ax^2 + bx + c = 0$  es **más general** que la consistente en **resolver ecuaciones de primer grado**,  $a + bx = 0$

### 2. Eficiencia

En términos muy generales, se considera que un algoritmo es tanto más eficiente cuantos menos pasos emplea en llevar a cabo su cometido.

La **suma de dos números naturales**, la regla tradicional que se aprende en enseñanza primaria es más eficiente que el rudimentario procedimiento de contar con los dedos, de uno en uno.



C++

## Lenguajes algorítmicos

20

- Una vez ideado **el algoritmo**, el modo más natural e inmediato (y también el menos formal) de expresar esa organización es redactándolo con palabras y frases del **lenguaje cotidiano**
- Todo lenguaje algorítmico debe poseer mecanismos con los que expresar las acciones así como el orden en que han de llevarse a cabo

Los lenguajes naturales tienen características como **libertad, flexibilidad y ambigüedad**

Los lenguajes de programación se caracterizan por la **precisión, rigidez y limitaciones de expresividad**



C++

## Características de los lenguajes algorítmicos

21

- Tienden un puente entre la forma humana de resolver problemas y su resolución mediante programas de ordenador.
- Cierta independencia de los lenguajes de programación particulares, de modo que están libres de sus limitaciones y así los algoritmos escritos en ellos se pueden traducir indistintamente a un lenguaje de programación u otro.

Las únicas restricciones que deberán imponerse a estos lenguajes proceden de las características que tienen los algoritmos:

- expresar sus acciones (qué deben realizar y cuándo) con la precisión necesaria
- que estas acciones sean deterministas.



C++

## Lenguajes algorítmicos

22

- **Las acciones**, se expresan mediante instrucciones (también llamadas órdenes o sentencias) que son comparables a **verbos en infinitivo**: asignar. . . , leer. . . , escribir. . . y otras.
- La concatenación de las instrucciones expresa en qué orden deben sucederse las acciones; esto es, cómo se ensamblan unas tras otras.
- Los modos más usados para ensamblar órdenes son la secuencia, la selección y la repetición

Los diagramas de flujo  
Pseudocódigo



C++

## Lenguajes algorítmicos

23

### Pseudocódigo

- Es un lenguaje utilizado para definir algoritmos con una sintaxis muy parecida a la de un lenguaje de programación.
- Las instrucciones se escriben en palabras similares al inglés o al español, facilitando así la comprensión del algoritmo.
- La ventaja es que es muy fácil pasar de pseudocódigo a un lenguaje de programación.

### Diagramas de flujo

- Es la representación gráfica de un algoritmo.
- Permite representar la secuencia de operaciones que se deben realizar para la resolución de un problema, es decir, permite representar el flujo de información desde su entrada hasta su salida.
- Dispone de un conjunto de **símbolos gráficos** con significado referente al tipo de instrucción que se va a realizar.
- Dichos símbolos van unidos con **flechas** que indican el orden de secuencia a seguir.

**DFD-** Edita e interpreta diagramas de flujo  
**DIA-** edita diagramas de flujo



C++

## Pseudocódigo

24

### Secuenciales

```
instrucción1  
instrucción2  
...  
instrucciónn
```

### Selectivas

```
si condición entonces  
instrucciones  
fin si
```

```
si condición entonces  
instrucciones1  
si no entonces  
instrucciones2  
fin si
```

```
si condición1 entonces  
instrucciones1  
si no si condición2 entonces  
instrucciones2  
si no si condición3 entonces  
instrucciones3  
...  
si no entonces  
instruccionesn  
fin si
```

### Iterativas o Repetitivas

```
mientras condición hacer  
instrucciones  
fin mientras
```

```
hacer  
instrucciones  
mientras condición
```

```
para i ← x hasta n hacer  
instrucciones  
fin para
```



C++

## Ejercicios

25

- Sumar dos números
- Valor absoluto
- Decidir si un número es par o impar
- Dado un n° real escribir la nota a la que corresponde: suspenso, aprobado, notable, sobresaliente, matrícula de honor
- Sumar los pares desde 2 hasta un número dado
- Resolver una ecuación de segundo grado
- Dado un número decir si es o no primo



C++

## Diagramas de flujo

26

	Indica el <b>inicio</b> y <b>final</b> del diagrama de flujo
	Indica la entrada y salida de <b>datos</b>
	Símbolo de <b>proceso</b> . Indica la realización de una operación.
	Símbolo de <b>decisión</b> . Indica operaciones de comparación entre datos. En función del resultado se sigue por distintos caminos.
	Llamada a otro proceso complejo. Llamada a subrutina.
	Indica la salida de información por impresora.
	Conector. Representa la continuidad del diagrama.
	Línea de flujo. Indica el sentido de ejecución de las operaciones

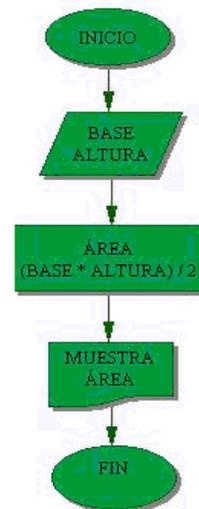


C++

## Calcular el área de un triángulo

27

- 1.- Entrada de datos: base y altura.  
Leer base y altura
- 2.- Calcular el área, base por altura dividido por 2
- 3.- Sacar en pantalla o impresora el área



C++

## Realizar la suma de todos los números pares entre 2 y 100

28

Inicialmente SUMA vale 0.  
el valor de NUMERO es 2.

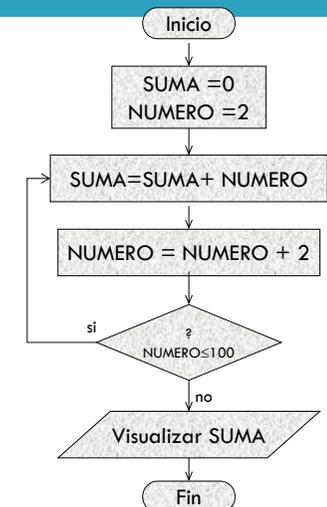
**hacer**

**acumular** NUMERO a SUMA.

aumentar el valor de NUMERO en dos unidades.

**mientras** el valor de NUMERO ≤ 100

Escribir SUMA



C++

## Ejercicios

29

- Mcd, algoritmo de Euclides
  - 1 Se divide el número mayor entre el menor.
  - 2 Si:
    - 1 La división es exacta, el divisor es el m.c.d.
    - 2 La división no es exacta, dividimos el divisor entre el resto obtenido y se continúa de esta forma hasta obtener una división exacta, siendo el último divisor el m.c.d.
- Sumar e imprimir los números desde el 3 al 99, de 3 en 3  
3,6,9,12, ..., 99
- Encontrar los números que cumplen la siguiente propiedad:  
*"la suma de tres números enteros es igual al producto de esos tres números enteros"*  
$$x+y+z=x*y*z$$



C++

## Calcular el mcd por el algoritmo de Euclides

30

```
Inicio
Leer los número a y b
Mientras el resto de la división entera de a y b sea distinto de cero
  Calcular resto
  a= b
  b= resto
mcd= b
escribir el mcd
fin
```



C++

## Sumar e imprimir los números 3,6,9,12,... 99

31

```
Inicio
suma = 0
numero=0
Mientras numero <= 99 hacer
  numero = numero + 3
  escribir numero
  suma=suma + numero
escribir suma
fin.
```



C++

## Lenguajes de programación

32

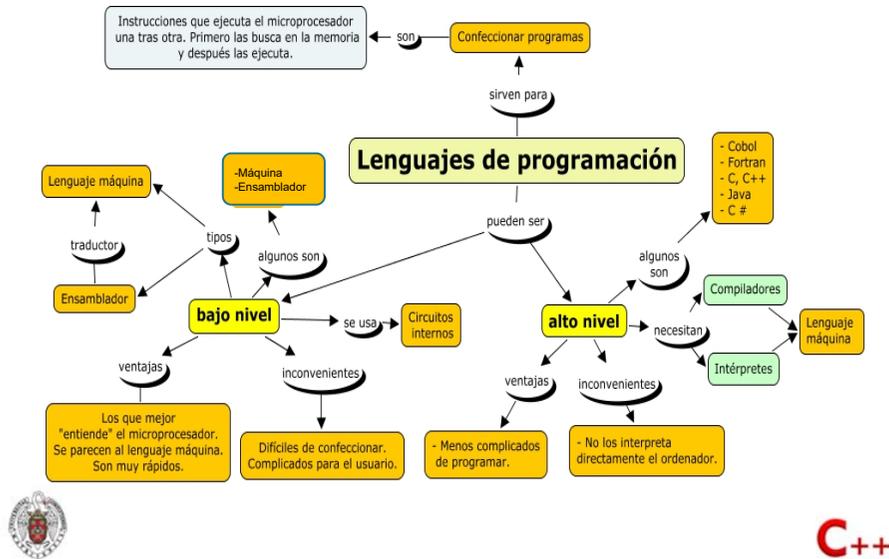
- Conjunto de signos y reglas que permite la comunicación con un ordenador.
- Los lenguajes de programación son un medio de expresar un algoritmo o solucionar el problema de forma que la máquina lo entienda
- Sintaxis y semántica



C++

## Lenguajes de programación

33



C++

## Lenguajes de programación

34

### Sintaxis

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

### Semántica

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



C++

## Sintaxis de los lenguajes de programación

35

### Lenguajes (BNF)

### Diagramas

BNF

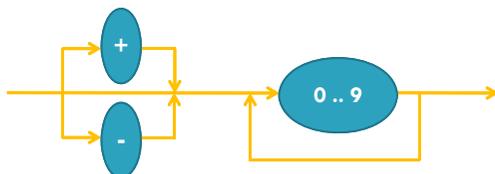
$\langle \text{numero entero} \rangle ::= \langle \text{signo opcional} \rangle \langle \text{secuencia de dígitos} \rangle$   
 $\langle \text{signo opcional} \rangle ::= + | - | \langle \text{nada} \rangle$   
 $\langle \text{secuencia de dígitos} \rangle ::= \langle \text{dígito} \rangle | \langle \text{dígito} \rangle \langle \text{secuencia de dígitos} \rangle$   
 $\langle \text{dígito} \rangle ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9$   
 $\langle \text{nada} \rangle ::=$

(Backus–Naur Form)

Notación formal para definir la sintaxis de un lenguaje. Usada para especificar la mayoría de los lenguajes de programación

| significa ó

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



C++

## Sintaxis de los lenguajes de programación

36

1. Sean las siguientes reglas BNF de los identificadores de un lenguaje:

$\langle \text{identificador} \rangle ::= \langle c \rangle \langle m \rangle \langle \text{resto} \rangle$

$\langle \text{resto} \rangle ::= \langle c \rangle | \langle c \rangle \langle \text{resto} \rangle$

$\langle c \rangle ::= \langle n \rangle | \langle m \rangle$

$\langle n \rangle ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9$

$\langle m \rangle ::= A | B | C$

Indica si las siguientes ocurrencias de símbolos corresponden a identificadores válidos del lenguaje:

1.1) **1AB** 1.2) **ABAc** 1.3) **4278** 1.4) **1B**



Ejercicios 1 y 2 de los apuntes de Luis Hernández

C++

## Desarrollo de programas

37

**Planificación**- Recursos humanos y técnicos que se precisan

**Análisis**-

- Cuáles deben ser las funciones que deben cumplir la aplicación
- Cómo debe realizarse el trabajo

**Diseño**- Conjunto de bloques, se dividen en partes y se asignan a equipos de programadores. Algoritmos.

**Codificación**- Escribir el algoritmo en un lenguaje de programación

**Validación**- Pruebas

**Mantenimiento**- documentación tanto para el programador como para el usuario.



C++

## Desarrollo de programas

38



C++

## Videos interesantes

39

<https://www.youtube.com/watch?v=6sTPEtzNlsA>

Historia de la Informática

<https://www.youtube.com/watch?v=CZBoCT8dozY>

Mala calidad, explica binario expresiones lógicas mediante circuitos

<https://www.youtube.com/watch?v=6K3JRwNeFEM>

Programa Redes, el ordenador del futuro

<https://www.youtube.com/watch?v=bxPmnwzwxAA>

Dibujos animados, historia de la Informática



C++