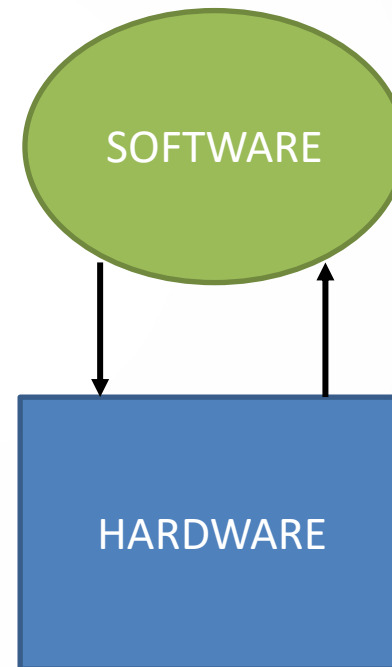


Conceptos generales

Qué es un sistema computacional?

- Hardware
- Software



Hardware

- Dispositivos (entrada/salida)

- pantalla
- teclado
- micrófono
- etc.

- “La caja” o “la torre”

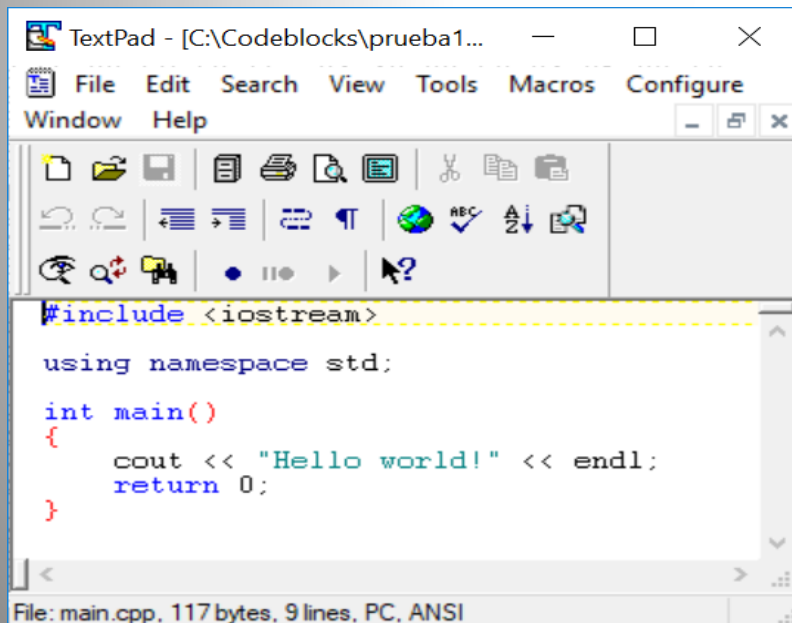
- fuente de alimentación
- CPU
- memoria
- etc.

Software

- Sistema operativo
 - LINUX, Windows, MacOS, Android, iOS, FreeDOS, etc.
- Programas/Aplicaciones
 - Juegos, procesadores de texto, navegadores web, entornos de desarrollo, etc.
- Compiladores
 - C++, Java, Pascal, FORTRAN, LISP, etc.

¿Cómo nos comunicamos?

- C++ es un lenguaje de alto nivel (basado en texto)
- Un programa compilador traduce el código C++ a código máquina (binario)



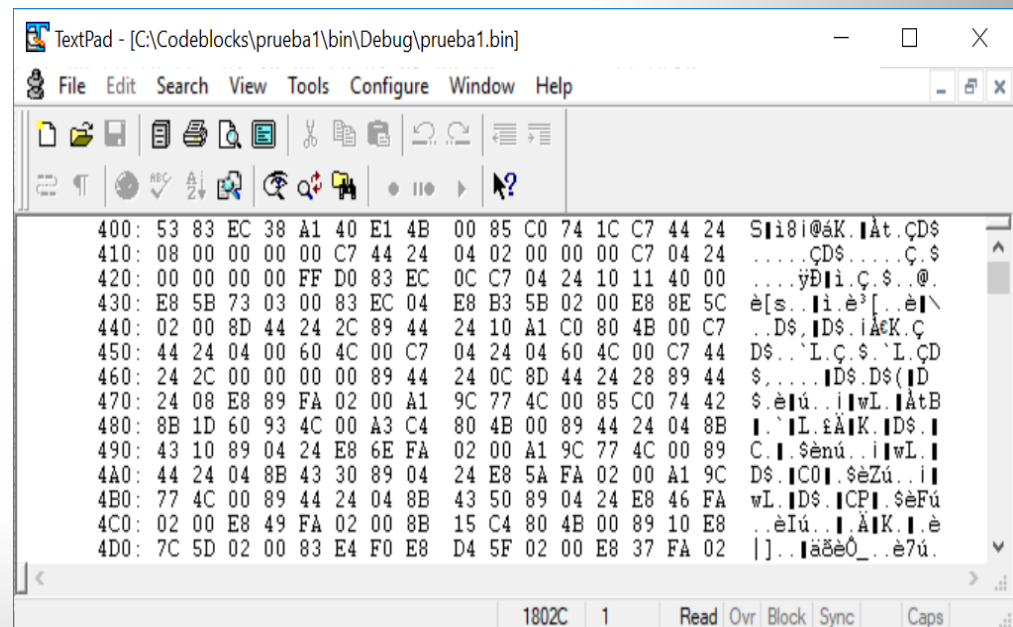
TextPad - [C:\Codeblocks\prueba1...

```
#include <iostream>

using namespace std;

int main()
{
    cout << "Hello world!" << endl;
    return 0;
}
```

File: main.cpp, 117 bytes, 9 lines, PC, ANSI



TextPad - [C:\Codeblocks\prueba1\bin\Debug\prueba1.bin]

400:	53 83 EC 38 A1 40 E1 4B 00 85 C0 74 1C C7 44 24	S1i8i@aK.1At.CD\$
410:	08 00 00 00 00 C7 44 24 04 02 00 00 00 C7 04 24CD\$.....Ç.\$
420:	00 00 00 00 FF D0 83 EC 0C C7 04 24 10 11 40 00ÿDi.Ç.\$..@.
430:	E8 5B 73 03 00 83 EC 04 E8 B3 5B 02 00 E8 8E 5C	è[s...i.è³[...èI\
440:	02 00 8D 44 24 2C 89 44 24 10 A1 C0 80 4B 00 C7	..D\$.ID\$.1AeK.Ç
450:	44 24 04 00 60 4C 00 C7 04 24 04 60 4C 00 C7 44	D\$.1L.Ç.\$..1.CD
460:	24 2C 00 00 00 00 89 44 24 0C 8D 44 24 28 89 44	\$.ID\$.D\$(ID
470:	24 08 E8 89 FA 02 00 A1 9C 77 4C 00 85 C0 74 42	\$.èIú...i wL.1AtB
480:	8B 1D 60 93 4C 00 A3 C4 80 4B 00 89 44 24 04 8B	1..1L.éA1K.ID\$.1
490:	43 10 89 04 24 E8 6E FA 02 00 A1 9C 77 4C 00 89	C.1.\$ènú...i wL.1
4A0:	44 24 04 8B 43 30 89 04 24 E8 5A FA 02 00 A1 9C	D\$.1C01.\$èZú...i
4B0:	77 4C 00 89 44 24 04 8B 43 50 89 04 24 E8 46 FA	wL.ID\$.1CPI.\$èFú
4C0:	02 00 E8 49 FA 02 00 8B 15 C4 80 4B 00 89 10 E8	..èIú...1A1K.1.è
4D0:	7C 5D 02 00 83 E4 F0 E8 D4 5F 02 00 E8 37 FA 02]...1aèè0...è7ú.

1802C 1 Read Ovr Block Sync Caps

Funcionamiento básico del compilador

1. Comprueba la sintaxis
2. Enlaza con código externo
3. Genera un fichero ejecutable



El proceso completo

1. Tenemos un problema que resolver
2. Escribimos un programa en C++
3. El compilador traduce de C++ a código máquina
4. El sistema operativo ejecuta las instrucciones del código máquina enviando órdenes a los componentes hardware cuando sea necesario
5. La CPU, RAM, discos, etc. trabajan en equipo para ejecutar las órdenes recibidas

¿Qué es un algoritmo?

- Es una secuencia de acciones utilizada paso por paso para resolver un problema
- Debe ser claro, conciso y correcto

Ejemplo

- **Problema:** ordenar una baraja de cartas de arriba a abajo como sigue:
 - Del as de bastos al rey de bastos
 - Del as de copas al rey de copas
 - Del as de espadas al rey de espadas
 - Del as de oros al rey de oros

Algoritmo solución

1. Ir cogiendo una a una cada carta de la baraja
2. Hacer 4 montones nuevos de cartas, uno para cada palo (bastos, copas, espadas, oros)
3. Para cada montón, ordenar las cartas del as al rey
4. Combinar los 4 montones en este orden: bastos, copas, espadas, oros

Ejemplo (II)

- **(Sub)Problema:** ordenar una pila de papeles numerados, de menor a mayor

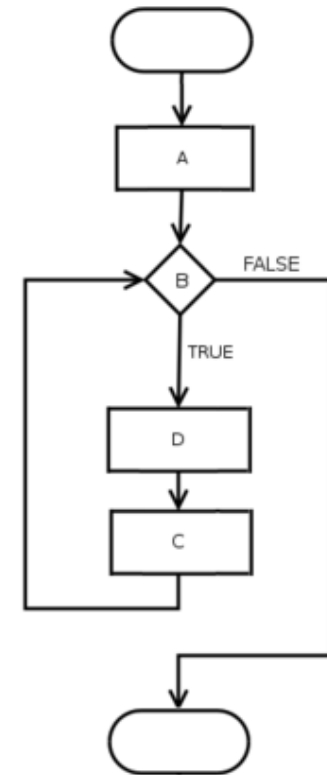
Algoritmo solución (II)

1. Comparar los 2 primeros papeles de la pila; colocar el menor de ellos en el fondo de la pila y guardar el mayor en la mano
2. Sacar el siguiente papel y compararlo con el de la mano; colocar el menor de ellos en el fondo de la pila y guardar el mayor en la mano
3. Repetir el paso anterior hasta haber examinado todos los papeles; en este punto tenemos el papel de valor máximo en la mano: colocarlo en una nueva pila (ordenada)
4. Repetir los 3 pasos anteriores, colocando siempre en la nueva pila el papel que nos quede en la mano en el paso 3, hasta que no quede ningún papel en la pila original

Aplicar algoritmos a la programación

- 1. Diagramas de control de flujo:**
Reflejan de forma gráfica cómo la información pasa de estado en estado

```
for(A;B;C)  
D;
```



Aplicar algoritmos a la programación

2. Pseudocódigo: Podemos usar nuestro propio lenguaje que contenga las construcciones usuales de un lenguaje de programación de alto nivel para escribir el algoritmo que después codificaremos

```
EDF(Target, t)
{
    // Target = peticiones a planificar
    // t = slot más cercano en el que se puede planificar
    // Scheduled = peticiones en Target que han sido planificadas
    // Di = información demandada por la petición i
    // Ti = plazo asociado a la petición i

    // inicializamos
    Scheduled = 0;

    // ordenamos Target en orden de plazos crecientes
    Sort(Target);

    while ( Target != Empty ) {
        // Elegimos petición con plazo más cercano
        i = GetFirstRequest(Target);

        // Comprobamos si i es planificable
        if ( t + Tamaño(Di) <= Ti ) {
            // Reservamos Slots para transmitir i
            ReserveSlotInterval(t, t + Tamaño(Di));
            t = t + Tamaño(Di);

            // Añadimos i a Scheduled
            AddToScheduled(i);
        }
    }
}
```

Ejemplo de programa en C++

```
1  // Programmer: John Doe          Date: 9/4/19          File: fahr2cels.cpp
2  // Purpose: this file contains the main function of the program which will input
3  //          Fahrenheit temps from the user, then convert and output Celsius to same.
4  #include <iostream>
5  using namespace std;
6
7  int main()
8  {
9      /* ----- DECLARATIONS ----- */
10     float cels;          // output variable
11     float fahr;          // input variable
12
13     /* ----- GREETINGS AND INPUT ----- */
14     cout << "\t\tWelcome to Temperature Conversion Program" << endl << endl;
15     cout << "Please enter a temperature in Fahrenheit:          ";
16     cin >> fahr;
17
18     /* ----- COMPUTATIONS AND OUTPUT ----- */
19     cels = (5.0 / 9.0) * (fahr - 32);
20     cout << "\n\nYour temperature input of " << fahr << " degrees Fahrenheit is "
21          << cels << " degrees Celsius" << endl;
22     cout << "Have a nice day" << endl;
23
24     return 0;
25 }
26
```