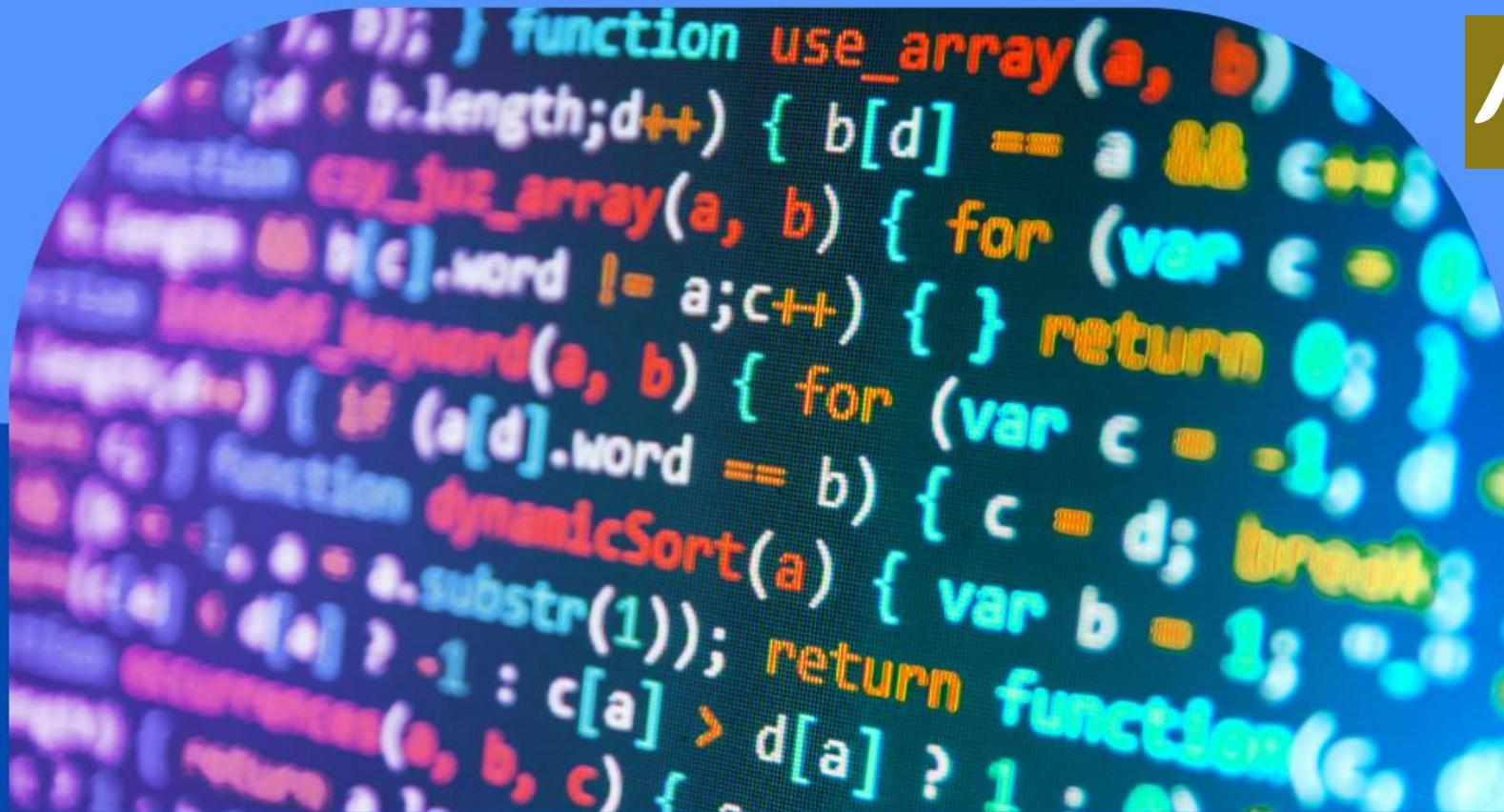


INTRODUCCIÓN AL LENGUAJE C / C++



INDICE

- **Breve Repaso**
- **Hola Mundo**
- **Ejemplos de Aplicación**
- **Material Complementario**

Comandos DOS

comando /? → para pedir ayuda.

dir → para ver contenido del directorio

control → para acceder al Panel de Control.

systemPropertiesAdvanced → Propiedades del Sistema. Para llegar al PATH.

copy archivo_origen archivo_destino → para copiar archivos.

mkdir nombre → para crear un directorio.

cls → para limpiar pantalla

cd directorio → para entrar dentro de un directorio.

cd c: o **cd ** → para ir a la partición C: de Windows

cd .. → para salir de un directorio.

cd ... → para salir de dos directorios, uno dentro de otro.

del archivo → para borrar un archivo.

rd /s → para borrar un directorio con su contenido.

shutdown /s → para apagar el sistema windows.

ping web → para ver si una web esta operativa. Ejm. **ping google.com**

tasklist → para mostrar servicios ejecutándose en ese preciso momento.

ipconfig /all → Para ver los Números IP de tu sistema.

comando --help → para pedir ayuda.

ls -all → para ver contenido del directorio, inclusive archivos ocultos.

cat archivo → para el interior de un archivo de texto.

pwd → para ver la ruta actual.

cp origen destino → para copiar archivos y directorios.

mkdir nombre → para crear un directorio.

clear → para limpiar pantalla

cd directorio → para entrar dentro de un directorio.

cd /c/ → para entrar en la **partición C:** de Windows o cualquier otra.

cd .. → para salir de un directorio.

cd ../../ → para salir de dos directorios, uno dentro de otro.

rm -rf directorio → Elimina archivo y/o directorios.

mv → para mover o renombrar archivos o directorios.

ping web → para ver si una web está operativa. Ejm. **ping google.com**

wc → para contar palabras y el número de bytes de un archivo.

nvim → para activar el editor de texto plano neovim.

nano → para activar el editor de texto plano nano.

df -h → para ver el uso del espacio en disco.

cal → para un calendario en el mes actual.

Cheat Sheet

Lenguaje C

Cheatography

Lenguaje C Cheat Sheet

by Briana via cheatography.com/66326/cs/16552/

Operadores

|| Or
&&/& And
== Igual que
! Not
!= Diferente de
+ - / Operadores aritméticos
* %
+/- Incremento/Decremento
-
< <= Comparaciones
> >=

? : condición? Expresión 1: Expresión 2: Si se cumple la condición se ejecuta la expresión 1 y si no se cumple se ejecuta la expresión 2

Comentarios y salto de línea

// Comentario de una línea
/* */ Comentario en bloque
\n Salto de línea

Tipos de datos

char	'a'	Rango: 0...255
short	-15	Rango: -32..-127
int	1012	Rango: -32768...32767
unsigned int	128	Rango: 0...65535
long	123456	%lli
float	15.4	%f
double	123123.123123	%lf
long double	Número muy grande con decimales	%Lf
char	string X[50]	%s

En el string, 'X' es la variable y [50] es la cantidad de espacios

Directivas del preprocesador

include: Para la inclusión de archivos
define: Para la definición de macros
if: Se usa para inclusiones condicionales
undef numero: Indefino anumero, es decir, anulo su definición previa

Estructura general de un programa

```
#include <stdio.h> // Cabecera estándar de entrada/salida de datos
Prototipos // Aquí se definen los prototipos de las funciones se usarán dentro de la función main()

int main() // Función principal
{
    instrucciones;
    return 0; // Buena praxis de programación
}
```

Entradas y salidas de datos

printf() Muestra en pantalla los datos
scanf() Guarda un dato que el usuario %n.&va; digitó
readable()
gets() Se usa para cadenas de strings, dado que el scanf solo lee hasta que haya un espacio
puts() Muestra datos en pantalla pero solo si está dentro de un condicional
%i: Define el tipo de dato que se está guardando.
&variable: Define en qué posición de memoria se guardará el dato, seguido del nombre de la variable donde se guardará.

Sentencia if-else

```
if(condición){
    instrucción;
}
else{
    instrucción;
}
```

Sentencia switch

```
switch(selector){
    case etiqueta1: sentencia1;break;
    case etiqueta2: sentencia2;break;
    case etiqueta3: sentencia3;break;
    default: sentencia;
}
```

-El selector de un switch solo puede ser de tipo entero o char.
-Si no se cumple ninguno de los casos, entra al default.

Funciones matemáticas

ceil(x): Redondea a techo siempre
fabs(x): Devuelve el valor absoluto de x
floor(x): Redondea a piso
fmod(x,y): Calcula el residuo de la división de x/y
pow(x,p): Eleva x a la potencia p
sqrt(x): Devuelve la raíz cuadrada de x

Otros

getch(): Para hacer que el ejecutable no se cierre inmediatamente, si no hasta que se presione una tecla
flush(stdin): Para limpiar el buffer
Para usar getch() ocupo incluir la biblioteca <conio.h>.

Bucles

While
while(){
 instrucciones;
}
For
for(inicialización;condición;incremento){
 instrucciones;
}
Do while
do{
 instrucciones;
} while();

Guía de Lenguaje C++

GUÍA DE SINTAXIS DEL LENGUAJE C++ (ESTÁNDAR C++ ANSI)

(Fundamentos de Programación de Luis Aguilar)

GUÍA DE SINTAXIS DEL LENGUAJE C++ 1

B.1. ELEMENTOS DEL LENGUAJE.....	3
B.1.1. CARACTERES	3
B.1.2. COMENTARIOS	3
B.1.3. IDENTIFICADORES	3
B.1.4. PALABRAS RESERVADAS	4
B.2. TIPOS DE DATOS	5
B.2.1. VERIFICACIÓN DE TIPOS	5
B.3. CONSTANTES	5
B.3.1. DECLARACIÓN DE CONSTANTES.....	6
B.4. CONVERSIÓN DE TIPOS.....	6
B.5. DECLARACIÓN DE VARIABLES	6
B.6. OPERADORES.....	6
B.6.1. OPERADORES ARITMÉTICOS.....	7
B.6.2. OPERADORES DE ASIGNACIÓN	8
B.6.3. OPERADORES LÓGICOS Y RELACIONALES.....	8
B.6.4. OPERACIONES DE MANIPULACIÓN DE BITS.....	9
B.6.5. EL OPERADOR sizeof	10
B.6.6. PRIORIDAD Y ASOCIATIVIDAD DE OPERADORES.....	10
B.6.7. SOBRECARGA DE OPERADORES.....	10
B.7. ENTRADAS Y SALIDAS BÁSICAS.....	10
B.7.1. SALIDA	11
B.7.2. ENTRADA.....	11
B.7.3. MANIPULADORES	12
B.8. SENTENCIAS	12
B.8.1. SENTENCIAS DE DECLARACIÓN.....	12
B.8.2. SENTENCIAS EXPRESIÓN	13
B.8.3. SENTENCIAS COMBINADAS	13
B.9. SENTENCIAS CONDICIONALES: IF	13
B.9.1. SENTENCIAS IF_ELSIF ANIDADAS	14
B.9.2. SENTENCIAS DE ALTERNATIVA MÚLTIPLE: SWITCH	15
B.10. BUCLES: SENTENCIAS REPETITIVAS.....	15
B.10.1. SENTENCIA WHILE	15
B.10.2. SENTENCIA DO.....	16
B.10.3. LA SENTENCIA FOR.....	16
B.10.4. SENTENCIAS BREAK Y CONTINUE	17
B.10.5. SENTENCIA NULL	17
B.10.6. SENTENCIA RETURN	17
B.11. PUNTEROS (APUNTADORES)²	17
B.11.1. DECLARACIÓN DE PUNTEROS	18
B.11.2. PUNTEROS A ARRAYS	18
B.11.3. PUNTEROS A ESTRUCTURAS	19
B.11.4. PUNTEROS A OBJETOS CONSTANTES	19
B.11.5. PUNTEROS A VOID	19
B.11.6. PUNTEROS Y CADENAS	19

Ejemplo 01: hola_mundo.c

The screenshot shows the Notepad++ interface with the file `hola_mundo.c` open. The code is a simple C program that prints "Hola Mundo" to the console. The code is color-coded: `#include`, `int`, `main`, `printf`, `return`, and the string "Hola Mundo" are in blue; the brace characters {}, the line separator \n, and the comments // are in black.

```
1
2     #include <stdio.h>
3
4     int main() {
5         printf("Hola Mundo\n");
6         return 0;
7     }
8
9     // https://chatgpt.com/share/67f6b4cb-0b00-800a-a1e4-4c80ddc8453d
10
11    // gcc hola_mundo.c -o hola_mundo.exe
```

The screenshot shows a Command Prompt window with the path `C:\code\c\hola_mundo>`. The command `hola_mundo.exe` is run, and the output "Hola Mundo" is displayed in green text. The prompt then returns to the command line.

```
C:\code\c\hola_mundo>hola_mundo.exe
Hola Mundo
C:\code\c\hola_mundo>
```

Ejemplo 01: hola_mundo.cc

C:\code\cc\hola_mundo\hola_mundo.cc - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

hola_mundo.cc

```
1
2     #include <iostream>
3
4     int main() {
5         std::cout << "Hola Mundo" << std::endl;
6         return 0;
7     }
8
9     // Referencia
10    //
11    // OpenAI. (2025, abril 9). Ejemplos en Lenguaje C y C++. ChatGPT.
12    // https://www.openai.com/chatgpt
13
14    // g++ hola_mundo.cc -o hola_mundo.exe
15    //
```

Command Prompt

C:\code\cc\hola_mundo>hola_mundo.exe
Hola Mundo

C:\code\cc\hola_mundo>

Ejemplo 02: calcular_area.c

```
calculator_area.c ✘ x
1  #include <stdio.h>
2  #define PI 3.14159
3
4  // Declaración de la función
5  float calcular_area(float radio);
6
7  int main() {
8      float radio;
9      printf("Introduce el radio del círculo: ");
10     scanf("%f", &radio);
11
12     // Llamada a la función
13     float area = calcular_area(radio);
14
15     printf("El área del círculo con radio %.2f es: %.2f\n", radio, area);
16
17     return 0;
18 }
19
20 // Definición de la función
21 float calcular_area(float radio) {
22     return PI * radio * radio;
23 }
24
```

Command Prompt

```
C:\code\c\calcular_area>calcular_area.exe
Introduce el radio del círculo: 10
El área del círculo con radio 10.00 es: 314.16
```

Ejemplo 02: calcular_area.cc

```
calculator_area.cc ✘ x
1
2 #include <iostream>
3 #define PI 3.14159
4
5 // Declaración de la función
6 float calcular_area(float radio);
7
8 int main() {
9     float radio;
10    std::cout << "Introduce el radio del círculo: ";
11    std::cin >> radio;
12
13    // Llamada a la función
14    float area = calcular_area(radio);
15
16    std::cout << "El área del círculo con radio " << radio << " es: " << area << std::endl;
17
18    return 0;
19}
20
21 // Definición de la función
22 float calcular_area(float radio) {
23     return PI * radio * radio;
24}
```

Command Prompt

```
C:\code\cc\calcular_area>calcular_area.exe
Introduce el radio del círculo: 10
El área del círculo con radio 10 es: 314.159
```

Ejemplo 03: leer_archivo.c

```
lee_archivo.c ✘ x
4 int main() {
5     FILE *file;
6     char line[256];
7     int line_count = 0;
8
9     // Abrir archivo para lectura
10    file = fopen("archivo.txt", "r");
11    if (file == NULL) {
12        perror("No se puede abrir el archivo");
13        return 1;
14    }
15
16    // Leer cada línea del archivo y contar las líneas
17    while (fgets(line, sizeof(line), file) != NULL) {
18        printf("Línea %d: %s", ++line_count, line);
19    }
20
21    printf("\nTotal de líneas: %d\n", line_count);
22
23    // Cerrar archivo
24    fclose(file);
25    return 0;
26}
27
28 // REFERENCIA
29 // https://chatgpt.com/share/67f6b4cb-0b00-800
30 //
31 // gcc lee_archivo.c -o lee_archivo.exe
```

Command Prompt
C:\code\c\lee_archivo>lee_archivo.exe archivo.txt
Línea 1: hola esta es una prueba
Línea 2: de lectura.
Línea 3: Espero que te guste.
Total de líneas: 3

Ejemplo 03: leer_archivo.cc

```
lee_archivo.cc ✘ x
1
2 #include <iostream>
3 #include <fstream>
4 #include <string>
5
6 int main() {
7     std::ifstream file("archivo.txt");
8     std::string line;
9     int line_count = 0;
10
11    if (!file.is_open()) {
12        std::cerr << "No se puede abrir el archivo" << std::endl;
13        return 1;
14    }
15
16    // Leer cada linea del archivo y contar las líneas
17    while (std::getline(file, line)) {
18        std::cout << "Línea " << ++line_count << ": " << line << std::endl;
19    }
20
21    std::cout << "\nTotal de líneas: " << line_count << std::endl;
22
23    // Cerrar archivo
24    file.close();
25    return 0;
26}
27
```

```
C:\code\cc\lee_archivo>lee_archivo.exe archivo.txt
Línea 1:
Línea 2: hola esta es una prueba
Línea 3: de lectura.
Línea 4: Espero que te guste.
Línea 5: Nos vemos pronto
Línea 6: en otro ejemplo.
Línea 7:
```

Ejm 04

estudiante.c

```
estudiante.c > x |  
1 #include <stdio.h>  
2 #include <string.h>  
3 struct Estudiante {  
4     char nombre[50];  
5     int edad;  
6     float calificacion;  
7 };  
8 void mostrarInfo(struct Estudiante est) {  
9     printf("Nombre: %s\nEdad: %d\nCalificación: %.2f\n", est.nombre, est.edad, est.calificacion);  
10 }  
11 void actualizarCalificacion(struct Estudiante *est, float nuevaCalificacion) {  
12     est->calificacion = nuevaCalificacion;  
13 }  
14 int main() {  
15     struct Estudiante est1 = {"Juan", 20, 85.5};  
16     struct Estudiante est2 = {"Maria", 22, 90.0};  
17  
18     printf("Estudiante 1:\n");  
19     mostrarInfo(est1);  
20  
21     printf("\nEstudiante 2:\n");  
22     mostrarInfo(est2);  
23  
24     // Actualizar calificación  
25     actualizarCalificacion(&est1, 88.0);  
26     printf("\nActualización de calificación para Juan:\n");  
27     printf("Nueva calificación: %.2f\n", est1.calificacion);  
28  
29     return 0;  
30 }
```

C:\code\c\estudiante>estudiante.exe
Estudiante 1:
Nombre: Juan
Edad: 20
Calificación: 85.50

Estudiante 2:
Nombre: Maria
Edad: 22
Calificación: 90.00

Actualización de calificación para Juan:
Nueva calificación: 88.00

```

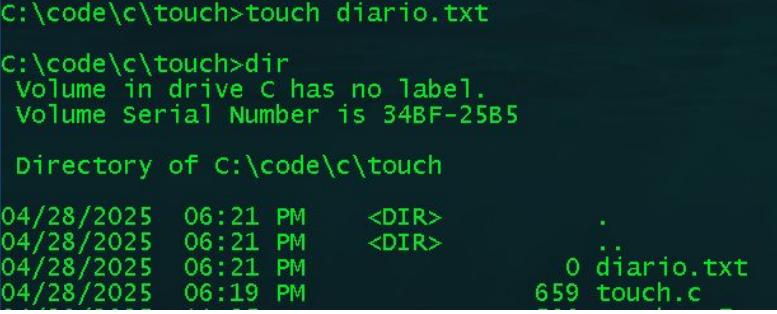
1 #include <iostream>
2 #include <string>
3 class Estudiante {
4 private:
5     std::string nombre;
6     int edad;
7     float calificacion;
8 public:
9     // Constructor
10    Estudiante(std::string n, int e, float c) : nombre(n), edad(e), calificacion(c) {}
11
12    // Métodos
13    void mostrarInfo() const {
14        std::cout << "Nombre: " << nombre << "\nEdad: " << edad << "\nCalificación: " << calificacion << std::endl;
15    }
16    void actualizarCalificacion(float nuevaCalificacion) {
17        calificacion = nuevaCalificacion;
18    }
19    float obtenerCalificacion() const {
20        return calificacion;
21    }
22};
23 int main() {
24    Estudiante est1("Juan", 20, 85.5);
25    Estudiante est2("Maria", 22, 90.0);
26    std::cout << "Estudiante 1:" << std::endl;
27    est1.mostrarInfo();
28    std::cout << "\nEstudiante 2:" << std::endl;
29    est2.mostrarInfo();
30    // Actualizar calificación
31    est1.actualizarCalificacion(88.0);
32    std::cout << "\nActualización de calificación para Juan:" << std::endl;
33    std::cout << "Nueva calificación: " << est1.obtenerCalificacion() << std::endl;
34    return 0;
35}

```

C:\code\cc\estudiante>estudiante.exe
 Estudiante 1:
 Nombre: Juan
 Edad: 20
 Calificación: 85.5
 Estudiante 2:
 Nombre: Maria
 Edad: 22
 Calificación: 90
 Actualización de calificación para Juan:
 Nueva calificación: 88

Ejemplo 05

touch
permite
crear un
archivo
vacío.



```
C:\code\c\touch>touch diario.txt
C:\code\c\touch>dir
Volume in drive C has no label.
Volume Serial Number is 34BF-25B5

Directory of C:\code\c\touch

04/28/2025  06:21 PM    <DIR>
04/28/2025  06:21 PM    <DIR>
04/28/2025  06:21 PM
04/28/2025  06:19 PM          0 diario.txt
                           659 touch.c

touch.c
```

```
touch.c
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(int argc, char *argv[]) {
5     FILE *fp;
6
7     if (argc != 2) {
8         printf("Uso: %s nombre_archivo\n", argv[0]);
9         return 1;
10    }
11
12    fp = fopen(argv[1], "ab"); // abrir en modo append binario
13    if (fp == NULL) {
14        perror("Error al abrir/crear el archivo");
15        return 1;
16    }
17
18    fclose(fp);
19    return 0;
20}
21 // OpenAI. (2025, abril 28). Creación de un programa tipo "touch"
22 // para consola CMD de Windows en C y C++ compilado con MingW64 (UCRT64).
23 // ChatGPT. https://chatgpt.com/share/680f753d-df38-8013-9f6a-e662a9b9cda6
24
25 // gcc touch.c -o touch.exe
```

Ejemplo 05: touch.cc

```
touch.cc ✘ x
1 #include <iostream>
2 #include <fstream>
3
4 using namespace std;
5
6 int main(int argc, char *argv[]) {
7     if (argc != 2) {
8         cout << "Uso: " << argv[0] << " nombre_archivo" << endl;
9         return 1;
10    }
11
12    ofstream archivo(argv[1], ios::app | ios::binary);
13    if (!archivo) {
14        cerr << "Error al abrir/crear el archivo" << endl;
15        return 1;
16    }
17
18    archivo.close();
19    return 0;
20}
21
22 // OpenAI. (2025, abril 28). Creación de un programa tipo "touch" para consola CMD
23 // de Windows en C y C++ compilado con MinGW64 (UCRT64). ChatGPT.
24 // https://chatgpt.com/share/680f753d-df38-8013-9f6a-e662a9b9cda6
25
26 // g++ touch.cc -o touch.exe
```

```
C:\code\cc\touch>touch thor.txt
C:\code\cc\touch>dir
Volume in drive C has no label.
Volume Serial Number is 34BF-25B5

Directory of C:\code\cc\touch

04/28/2025  06:32 PM    <DIR>   .
04/28/2025  06:32 PM    <DIR>   ..
04/28/2025  06:32 PM            0  thor.txt
04/28/2025  11:47 AM       677  touch.cc
```

Ejemplo 06: mtouch.c

```
mtouch.c ✘ x
1  #include <stdio.h>
2
3  int main(int argc, char *argv[]) {
4      if (argc < 2) {
5          printf("Uso: %s archivo1 [archivo2 ... archivoN]\n", argv[0]);
6          return 1;
7      }
8
9      for (int i = 1; i < argc; ++i) {
10         FILE *fp = fopen(argv[i], "ab"); // modo append binario
11         if (fp == NULL) {
12             perror(argv[i]);
13             continue;
14         }
15         fclose(fp);
16     }
17
18     return 0;
19 }
20 // OpenAI. (2025, abril 28). Creación de un programa tipo "touch"
21 // para consola CMD de Windows en C y C++ compilado con MinGW64 (UCRT64). ChatGPT.
22 // https://chatgpt.com/share/680f753d-df38-8013-9f6a-e662a9b9cda6
23
24 // gcc mtouch.c -o mtouch.exe
25
26 // mtouch archivo1 archivo2 archivo3
```

```
C:\code\c\mtouch>mtouch dana.txt diana.txt dina.txt
C:\code\c\mtouch>dir
  Volume in drive C has no label.
  Volume Serial Number is 34BF-25B5

  Directory of C:\code\c\mtouch

04/28/2025  06:45 PM    <DIR>        .
04/28/2025  06:45 PM    <DIR>        ..
04/28/2025  06:45 PM            0 dana.txt
04/28/2025  06:45 PM            0 diana.txt
04/28/2025  06:45 PM            0 dina.txt
```

Ejemplo 06: mtouch.cc

```
mtouch.cc ✘ x
1 #include <iostream>
2 #include <fstream>
3
4 using namespace std;
5
6 int main(int argc, char *argv[]) {
7     if (argc < 2) {
8         cout << "Uso: " << argv[0] << " archivo1 [archivo2 ... archivoN]" << endl;
9         return 1;
10    }
11
12    for (int i = 1; i < argc; ++i) {
13        ofstream archivo(argv[i], ios::app | ios::binary);
14        if (!archivo) {
15            cerr << "Error al abrir/crear " << argv[i] << endl;
16            continue;
17        }
18        archivo.close();
19    }
20
21    return 0;
22}
23
24 // g++ mtouch.cc -o mtouch.exe
```

```
C:\code\cc\mtouch>mtouch alicia.txt alan.txt alberto.txt
C:\code\cc\mtouch>dir
Volume in drive C has no label.
Volume Serial Number is 34BF-25B5

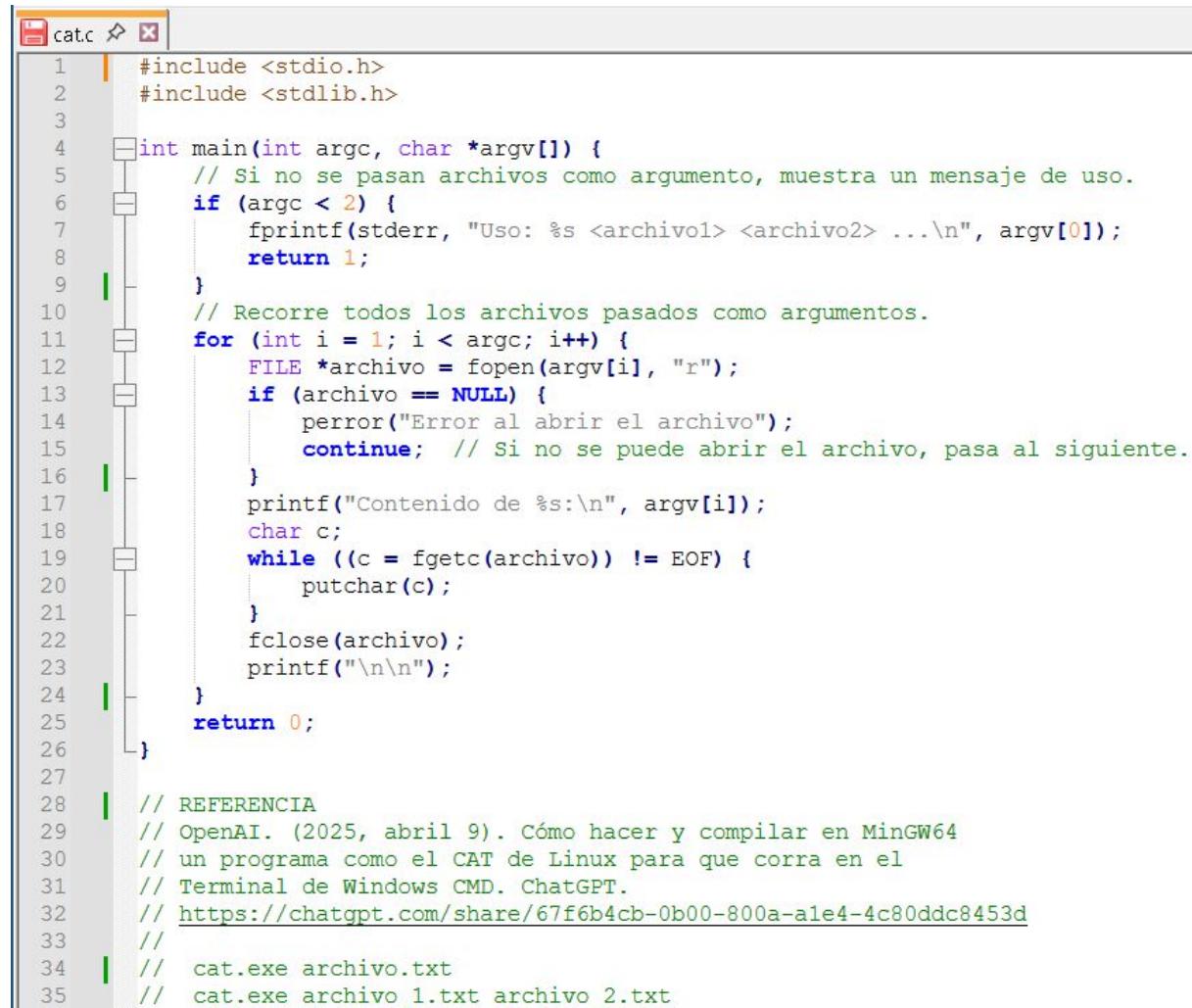
Directory of C:\code\cc\mtouch

04/28/2025  06:52 PM    <DIR>      .
04/28/2025  06:52 PM    <DIR>      ..
04/28/2025  06:52 PM          0  alicia.txt
04/28/2025  06:52 PM          0  alberto.txt
04/28/2025  06:52 PM          0  alan.txt
04/28/2025  06:49 PM      774  mtouch.cc
```

Ejemplo 07: cat.c

Con el comando **cat** podemos ver el contenido de un texto plano o código de programa.

```
C:\code\c\cat>cat notas.txt
Contenido de notas.txt:
hola esta son
las notas de mi vida
...
```



The screenshot shows a code editor window titled "cat.c" containing the C source code for a program named "cat". The code is color-coded for syntax highlighting. It includes comments explaining its functionality, such as handling command-line arguments and reading files from standard input. The code is annotated with line numbers from 1 to 35. A vertical green bar on the left indicates the current line of execution.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(int argc, char *argv[]) {
5      // Si no se pasan archivos como argumento, muestra un mensaje de uso.
6      if (argc < 2) {
7          fprintf(stderr, "Uso: %s <archivo1> <archivo2> ...\\n", argv[0]);
8          return 1;
9      }
10     // Recorre todos los archivos pasados como argumentos.
11     for (int i = 1; i < argc; i++) {
12         FILE *archivo = fopen(argv[i], "r");
13         if (archivo == NULL) {
14             perror("Error al abrir el archivo");
15             continue; // Si no se puede abrir el archivo, pasa al siguiente.
16         }
17         printf("Contenido de %s:\\n", argv[i]);
18         char c;
19         while ((c = fgetc(archivo)) != EOF) {
20             putchar(c);
21         }
22         fclose(archivo);
23         printf("\\n\\n");
24     }
25     return 0;
26 }
27
28 // REFERENCIA
29 // OpenAI. (2025, abril 9). Cómo hacer y compilar en MinGW64
30 // un programa como el CAT de Linux para que corra en el
31 // Terminal de Windows CMD. ChatGPT.
32 // https://chatgpt.com/share/67f6b4cb-0b00-800a-a1e4-4c80ddc8453d
33 //
34 // cat.exe archivo.txt
35 // cat.exe archivo_1.txt archivo_2.txt
```

Ejemplo 07: cat.cc

Con el comando **ccat** podemos ver el contenido de un texto plano o código de programa.

```
ccat.cpp <input>
1 // catcc.cpp
2 #include <iostream>
3 #include <fstream>
4 #include <string>
5
6 int main(int argc, char* argv[]) {
7     if (argc < 2) {
8         std::cerr << "Uso: " << argv[0] << " <archivo1> <archivo2> ..." << std::endl;
9         return 1;
10    }
11
12    for (int i = 1; i < argc; ++i) {
13        std::ifstream archivo(argv[i]);
14        if (!archivo) {
15            std::cerr << "Error al abrir el archivo: " << argv[i] << std::endl;
16            continue;
17        }
18
19        std::cout << "Contenido de " << argv[i] << ":\n";
20
21        std::string linea;
22        while (std::getline(archivo, linea)) {
23            std::cout << linea << '\n';
24        }
25        std::cout << "\n";
26    }
27    return 0;
28}
29
30 // REFERENCIA
31 // OpenAI. (2025, abril 9). Cómo hacer y compilar en MinGW64
32 // un programa como el CAT de Linux para que corra en el
33 // Terminal de Windows CMD. ChatGPT.
34 // https://chatgpt.com/share/67f6b4cb-0b00-800a-a1e4-4c80ddc8453d
35
36 // g++ ccat.cpp -o ccat.exe
```

```
C:\code\cc\ccat>ccat thor.txt
Contenido de thor.txt:
me llamo thor
me me dicen
DON GATO
```

Ejemplo 08: add.c

```
add.c < X
1 // add.c
2 #include <stdio.h>
3
4 int main(int argc, char *argv[]) {
5     if (argc < 3) {
6         printf("Uso: %s <nombre_archivo> <texto_a_agregar>\n", argv[0]);
7         return 1;
8     }
9
10    FILE *archivo = fopen(argv[1], "a"); // "a" = abrir para añadir (crear si no existe)
11    if (archivo == NULL) {
12        perror("Error al abrir o crear el archivo");
13        return 1;
14    }
15
16    // Agregar todos los argumentos después del nombre del archivo
17    for (int i = 2; i < argc; ++i) {
18        fprintf(archivo, "%s", argv[i]);
19        if (i < argc - 1) {
20            fputc(' ', archivo); // Añadir espacio entre palabras
21        }
22    }
23    fputc('\n', archivo); // Agregar salto de línea al final
24
25    fclose(archivo);
26    printf("Texto agregado correctamente.\n");
27    return 0;
28}
29 // https://chatgpt.com/share/68101692-58b4-8013-a8e9-9a3f6595a086
30
31 // gcc -o add add.c
32
33 // add.exe nombre_del_archivo.txt Texto que quiero agregar
```

```
C:\code\c\add>add thor.txt me llamo Thor, pero me dicen DON PERRO.
Texto agregado correctamente.

C:\code\c\add>cat thor.txt
Contenido de thor.txt:
me llamo Thor, pero me dicen DON PERRO.
```

Ejemplo 08: add.cc

```
add.cc ✘ x
1 #include <iostream>
2 #include <fstream>
3
4 int main(int argc, char *argv[]) {
5     if (argc < 3) {
6         std::cout << "Uso: " << argv[0] << " <nombre_archivo> <texto_a_agregar>\n";
7         return 1;
8     }
9
10    std::ofstream archivo(argv[1], std::ios::app); // modo append (añadir al final)
11    if (!archivo) {
12        std::cerr << "Error al abrir o crear el archivo.\n";
13        return 1;
14    }
15
16    // Agregar todos los argumentos después del nombre del archivo
17    for (int i = 2; i < argc; ++i) {
18        archivo << argv[i];
19        if (i < argc - 1) {
20            archivo << ' ';
21        }
22    }
23    archivo << '\n'; // Salto de línea al final
24
25    archivo.close();
26    std::cout << "Texto agregado correctamente.\n";
27    return 0;
28}
29
30 // https://chatgpt.com/share/68101692-58b4-8013-a8e9-9a3f6595a086
31
32 // g++ -o add.exe add.cc
33
34 // add.exe miarchivo.txt Este es el texto que quiero agregar
```

```
C:\code\cc\add>add.exe milky.txt Me llamo MILKY SHAKE. Pero me dicen SHEIK
Texto agregado correctamente.

C:\code\cc\add>cat milky.txt
Contenido de milky.txt:
Me llamo MILKY SHAKE. Pero me dicen SHEIK
```

Ejemplo 09: fibonacci.c

```
fibonacci.c ✘ x
1 #include <stdio.h>
2
3 int main() {
4     int n, i;
5     unsigned long long int a = 0, b = 1, siguiente;
6
7     printf("Ingrese el número de términos de la serie de Fibonacci: ");
8     scanf("%d", &n);
9
10    if (n <= 0) {
11        printf("Por favor, ingrese un número positivo.\n");
12        return 1;
13    }
14
15    printf("Serie de Fibonacci:\n");
16
17    for (i = 0; i < n; ++i) {
18        printf("%llu ", a);
19        siguiente = a + b;
20        a = b;
21        b = siguiente;
22    }
23
24    printf("\n");
25    return 0;
26}
27
28 // https://chatgpt.com/share/680fc0b7-c494-8013-b24e-232079c26345
29
30 // gcc fibonacci.c -o fibonacci.exe
```

```
C:\code\c\fibonacci>fibonacci.exe
Ingrese el número de términos de la serie de Fibonacci: 50
Serie de Fibonacci:
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 41
81 6765 10946 17711 28657 46368 75025 121393 196418 317811 51
4229 832040 1346269 2178309 3524578 5702887 9227465 14930352
24157817 39088169 63245986 102334155 165580141 267914296 4334
94437 701408733 1134903170 1836311903 2971215073 4807526976 7
778742049
```

Ejemplo 09: fibonacci.cc

```
fibonacci.cc ✘ x
1
2     #include <iostream>
3
4     int main() {
5         int n;
6         unsigned long long int a = 0, b = 1, siguiente;
7
8         std::cout << "Ingrese el número de términos de la serie de Fibonacci: ";
9         std::cin >> n;
10
11        if (n <= 0) {
12            std::cout << "Por favor, ingrese un número positivo.\n";
13            return 1;
14        }
15
16        std::cout << "Serie de Fibonacci:\n";
17
18        for (int i = 0; i < n; ++i) {
19            std::cout << a << " ";
20            siguiente = a + b;
21            a = b;
22            b = siguiente;
23        }
24
25        std::cout << std::endl;
26        return 0;
27    }
28
29 // https://chatgpt.com/share/680fc0b7-c494-8013-b24e-232079c26345
30
31 // g++ fibonacci.cc -o fibonacci.exe
```

```
C:\code\cc\fibonacci>fibonacci.exe
Ingrese el número de términos de la serie de Fibonacci: 60
Serie de Fibonacci:
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 41
81 6765 10946 17711 28657 46368 75025 121393 196418 317811 51
4229 832040 1346269 2178309 3524578 5702887 9227465 14930352
24157817 39088169 63245986 102334155 165580141 267914296 4334
94437 701408733 1134903170 1836311903 2971215073 4807526976 7
778742049 12586269025 20365011074 32951280099 53316291173 862
67571272 139583862445 225851433717 365435296162 591286729879
956722026041
```

```
tres_en_raya.c ✘ X
1 #include <stdio.h>
2 // Declaración de funciones
3 void mostrar_tablero(char tablero[3][3]);
4 int verificar_ganador(char tablero[3][3]);
5 int movimiento_valido(char tablero[3][3], int fila, int col);
6 void jugar();
7 int main() {
8     jugar();
9     return 0;
10 }
11 void jugar() {
12     char tablero[3][3] = {{'1', '2', '3'}, {'4', '5', '6'}, {'7', '8', '9'}};
13     int turno = 0, fila, col, ganador = 0;
14     char jugador = 'X';
15     while (1) {
16         mostrar_tablero(tablero);
17         // Turno de jugador
18         printf("Jugador %c, ingresa fila (1-3) y columna (1-3): ", jugador);
19         scanf("%d %d", &fila, &col);
20         fila--; col--; // Ajuste de índice para 0 basado
21         if (movimiento_valido(tablero, fila, col)) {
22             tablero[fila][col] = jugador;
23             ganador = verificar_ganador(tablero);
24             if (ganador != 0) {
25                 mostrar_tablero(tablero);
26                 printf("\nEl Jugador %c ha ganado!\n", jugador);
27                 break;
28             }
29             // Cambiar de jugador
30             jugador = (jugador == 'X') ? 'O' : 'X';
31             turno++;
32             if (turno == 9) {
33                 mostrar_tablero(tablero);
34                 printf("Es un empate\n");
35                 break;
36             }
37         }
38     }
39 }
```

Ejemplo 10: tres_en_raya.c

Ejemplo 10: tres_en_raya.c

```
37 } else {
38     printf("Movimiento inválido, intenta de nuevo.\n");
39 }
40 }
41 }
42 void mostrar_tablero(char tablero[3][3]) {
43     printf("\nTablero:\n");
44     for (int i = 0; i < 3; i++) {
45         for (int j = 0; j < 3; j++) {
46             printf("%c ", tablero[i][j]);
47         }
48         printf("\n");
49     }
50 }
51 int verificar_ganador(char tablero[3][3]) {
52     // Verificar filas y columnas
53     for (int i = 0; i < 3; i++) {
54         if (tablero[i][0] == tablero[i][1] && tablero[i][1] == tablero[i][2])
55             return 1;
56         if (tablero[0][i] == tablero[1][i] && tablero[1][i] == tablero[2][i])
57             return 1;
58     }
59     // Verificar diagonales
60     if (tablero[0][0] == tablero[1][1] && tablero[1][1] == tablero[2][2])
61         return 1;
62     if (tablero[0][2] == tablero[1][1] && tablero[1][1] == tablero[2][0])
63         return 1;
64     return 0; // No hay ganador aún
65 }
66 int movimiento_valido(char tablero[3][3], int fila, int col) {
67     if (fila < 0 || fila >= 3 || col < 0 || col >= 3)
68         return 0; // Fuera de rango
69     if (tablero[fila][col] == 'X' || tablero[fila][col] == 'O')
70         return 0; // Casilla ocupada
71     return 1;
72 }
```

Ejemplo 10: tres_en_raya.c

```
73
74  /*
75   * EXPLICACION
76
77   Tablero: El tablero se representa como una matriz de 3x3
78   que contiene los números del 1 al 9 inicialmente.
79
80   Verificación de Ganador: Después de cada movimiento, se revisa
81   si hay un ganador, verificando las filas, columnas y las dos diagonales.
82
83   Entrada del Usuario: Los jugadores ingresan las filas y columnas
84   para colocar sus marcas (X u O).
85
86   Cambio de Turno: Después de cada movimiento válido, el programa
87   cambia el turno entre los jugadores.
88
89 */
90
91 // Jugador X, ingresa fila (1-3) y columna (1-3): 1 1
92 // Jugador O, ingresa fila (1-3) y columna (1-3): 2 2
93 // Jugador X, ingresa fila (1-3) y columna (1-3): 1 2
94 // Jugador O, ingresa fila (1-3) y columna (1-3): 3 3
95
96 // REFERENCIA
97 // https://chatgpt.com/share/67f6b4cb-0b00-800a-a1e4-4c80ddc8453d
98
99 // gcc tres_en_raya.c -o tres_en_raya.exe
100
```

```
C:\code\c\tres_en_raya>tres_en_raya.exe
Tablero:
1 2 3
4 5 6
7 8 9
Jugador X, ingresa fila (1-3) y columna (1-3): 1 1

Tablero:
X 2 3
4 5 6
7 8 9
Jugador O, ingresa fila (1-3) y columna (1-3): 2 1

Tablero:
X 2 3
O 5 6
7 8 9
Jugador X, ingresa fila (1-3) y columna (1-3): 2 2

Tablero:
X 2 3
O X 6
7 8 9
Jugador O, ingresa fila (1-3) y columna (1-3): 3 1

Tablero:
X 2 3
O X 6
O 8 9
Jugador X, ingresa fila (1-3) y columna (1-3): 3 3

Tablero:
X 2 3
O X 6
O 8 X
¡El Jugador X ha ganado!
C:\code\c\tres_en_raya>
```

Ejemplo 10: tres_en_raya.cc

```
tres_en_raya.cc ✘ x
1 #include <iostream>
2 using namespace std;
3 // Declaración de funciones
4 void mostrar_tablero(char tablero[3][3]);
5 int verificar_ganador(char tablero[3][3]);
6 int movimiento_valido(char tablero[3][3], int fila, int col);
7 void jugar();
8 int main() {
9     jugar();
10    return 0;
11 }
12 void jugar() {
13     char tablero[3][3] = {{'1', '2', '3'}, {'4', '5', '6'}, {'7', '8', '9'}};
14     int turno = 0, fila, col, ganador = 0;
15     char jugador = 'X';
16     while (1) {
17         mostrar_tablero(tablero);
18         // Turno de jugador
19         cout << "Jugador " << jugador << ", ingresa fila (1-3) y columna (1-3): ";
20         cin >> fila >> col;
21         fila--; col--; // Ajuste de índice para 0 basado
22         if (movimiento_valido(tablero, fila, col)) {
23             tablero[fila][col] = jugador;
24             ganador = verificar_ganador(tablero);
25             if (ganador != 0) {
26                 mostrar_tablero(tablero);
27                 cout << ";El Jugador " << jugador << " ha ganado!" << endl;
28                 break;
29             }
30             // Cambiar de jugador
31             jugador = (jugador == 'X') ? 'O' : 'X';
32             turno++;
33             if (turno == 9) {
34                 mostrar_tablero(tablero);
35                 cout << "Es un empate" << endl;
36                 break;
37             }
38         }
39     }
40 }
```

Ejemplo 10: tres_en_raya.cc

```
tres_en_raya.cc
```

```
38         } else {
39             cout << "Movimiento inválido, intenta de nuevo." << endl;
40         }
41     }
42 }
43 void mostrar_tablero(char tablero[3][3]) {
44     cout << "\nTablero:" << endl;
45     for (int i = 0; i < 3; i++) {
46         for (int j = 0; j < 3; j++) {
47             cout << tablero[i][j] << " ";
48         }
49         cout << endl;
50     }
51 }
52 int verificar_ganador(char tablero[3][3]) {
53     // Verificar filas y columnas
54     for (int i = 0; i < 3; i++) {
55         if (tablero[i][0] == tablero[i][1] && tablero[i][1] == tablero[i][2])
56             return 1;
57         if (tablero[0][i] == tablero[1][i] && tablero[1][i] == tablero[2][i])
58             return 1;
59     }
60     // Verificar diagonales
61     if (tablero[0][0] == tablero[1][1] && tablero[1][1] == tablero[2][2])
62         return 1;
63     if (tablero[0][2] == tablero[1][1] && tablero[1][1] == tablero[2][0])
64         return 1;
65     return 0; // No hay ganador aún
66 }
67 int movimiento_valido(char tablero[3][3], int fila, int col) {
68     if (fila < 0 || fila >= 3 || col < 0 || col >= 3)
69         return 0; // Fuera de rango
70     if (tablero[fila][col] == 'X' || tablero[fila][col] == 'O')
71         return 0; // Casilla ocupada
72     return 1;
73 }
```

```

74 /*
75
76 EXPLICACION
77 Tablero: El tablero se representa como una matriz de 3x3
78 que contiene los números del 1 al 9 inicialmente.
79
80 Verificación de Ganador: Después de cada movimiento, se revisa
81 si hay un ganador, verificando las filas, columnas y las dos diagonales.
82
83 Entrada del Usuario: Los jugadores ingresan las filas y columnas
84 para colocar sus marcas (X u O).
85
86 Cambio de Turno: Después de cada movimiento válido, el programa
87 cambia el turno entre los jugadores.
88 */
89 // Jugador X, ingresa fila (1-3) y columna (1-3): 1 1
90 // Jugador O, ingresa fila (1-3) y columna (1-3): 2 2
91 // Jugador X, ingresa fila (1-3) y columna (1-3): 1 2
92 // Jugador O, ingresa fila (1-3) y columna (1-3): 3 3
93
94 // REFERENCIA
95 // https://chatgpt.com/share/67f6b4cb-0b00-800a-a1e4-4c80ddc8453d
96
97 // g++ tres_en_raya.cc -o tres_en_raya.exe
98

```

Ejemplo 10: tres_en_raya.cc

```

C:\code\cc\tres_en_raya>tres_en_raya.exe

Tablero:
1 2 3
4 5 6
7 8 9
Jugador X, ingresa fila (1-3) y columna (1-3): 1 1

Tablero:
X 2 3
4 5 6
7 8 9
Jugador O, ingresa fila (1-3) y columna (1-3): 1 3

Tablero:
X 2 0
4 5 6
7 8 9
Jugador X, ingresa fila (1-3) y columna (1-3): 1 2

Tablero:
X X 0
4 5 6
7 8 9
Jugador O, ingresa fila (1-3) y columna (1-3): 2 3

Tablero:
X X 0
4 5 0
7 8 9
Jugador X, ingresa fila (1-3) y columna (1-3): 2 1

Tablero:
X X 0
X 5 0
7 8 9
Jugador O, ingresa fila (1-3) y columna (1-3): 3 3

Tablero:
X X 0
X 5 0
7 8 0
¡El Jugador O ha ganado!

C:\code\cc\tres_en_raya>tres_en_raya.exe

```



C++



wxWidgets
Cross-Platform GUI Library

La forma más facil de trabajar con C++ y WxWidgets que es una libreria en C++ para el desarrollo de Interfaces gráficas, a criterio de este autor, es usar MINGW64 específicamente con el **Terminal UCRT64**, que es un entorno que simula Linux y que genera archivos ejecutables .exe nativos 100% compatibles con Windows.

Para que se ejecute correctamente cuando se hace un proyecto con interfaz gráfico, **el ejecutable principal en (.exe) debe ir acompañado de archivos (.dll) propios de wxWidgets**, así se puede distribuir el software creado sin la necesidad de instalar wxWidgets y MSYS2 en la máquina del cliente.

Con la ayuda de OpenAI (2025) se ha desarrollado un par de programas en el lenguaje de comandos para linux **Bash (.sh)** que facilita enormemente esta tarea, que a continuación se pasa a detallar. ***Están diseñados para trabajar solamente con la extensión .cpp***

empaqueta_dependencias.sh

```
1 #!/usr/bin/env bash
2
3 # Verificar argumento
4 if [ -z "$1" ]; then
5     echo "❌ Uso: $0 nombre_programa.exe"
6     exit 1
7 fi
8
9 EXE="$1"
10 BASENAME=$(basename "$EXE" .exe)
11 OUT_DIR="$BASENAME"
12
13 # Crear carpeta primero, como tú indicas
14 echo "📦 Preparando carpeta '$OUT_DIR/....'"
15 mkdir -p "$OUT_DIR"
16
17 # Verificar si el ejecutable existe
18 if [ ! -f "$EXE" ]; then
19     echo "❌ Error: no se encontró el archivo '$EXE'"
20     exit 1
21 fi
22
23 # Copiar el ejecutable
24 cp "$EXE" "$OUT_DIR/"
25
26 # Usar ldd para encontrar las DLL requeridas
27 echo "🔍 Buscando dependencias con ldd..."
28 ldd "$EXE" | grep '=> /ucrt64/bin/' | awk '{print $3}' > dlls.txt
29
30 echo "📄 Copiando DLLs necesarias a '$OUT_DIR/...."
31 while read dll; do
32     if [ -f "$dll" ]; then
33         echo "► $(basename "$dll")"
34         cp "$dll" "$OUT_DIR/"
35     fi
36 done < dlls.txt
```

enpaqueta_dependencias.sh

```
37 rm dlls.txt
38
39 echo "✅ Listo: carpeta '$OUT_DIR/' con ejecutable y DLLs empaquetadas."
40
41 # REFERENCIAS
42
43 # OpenAI. (2025). Script Bash para compilar y empaquetar aplicaciones
44 # wxWidgets con MSYS2-UCRT64 en Windows
45 # [Asistente conversacional, ChatGPT]. https://chat.openai.com/
46
47 # chmod +x empaqueta_dependencias.sh
48 #
49 #
50 # ./empaqueta_dependencias.sh mi_programa.exe
```



```

compilar_y_empaquetar.sh ✘ x
1  #!/usr/bin/env bash
2
3  # 📁 Verificación de argumentos
4  if [ -z "$1" ]; then
5      echo "✖ Uso: $0 archivo_fuente.cpp"
6      exit 1
7  fi
8
9  CPP_FILE="$1"
10 BASENAME=$(basename "$CPP_FILE" .cpp)
11 EXE="$BASENAME.exe"
12 OUT_DIR="$BASENAME"
13
14 echo "🛠️ Compilando $CPP_FILE..."
15 g++ "$CPP_FILE" -o "$EXE" $(wx-config --cxxflags --libs)
16 if [ $? -ne 0 ]; then
17     echo "✖ Error en la compilación"
18     exit 1
19 fi
20
21 echo "📦 Preparando carpeta '$OUT_DIR/'..."
22 mkdir -p "$OUT_DIR"
23 cp "$EXE" "$OUT_DIR/"
24
25 echo "🔍 Buscando dependencias con ldd..."
26 ldd "$EXE" | grep '^=> /ucrt64/bin/' | awk '{print $3}' > dlls.txt
27
28 echo "📄 Copiando DLLs necesarias a '$OUT_DIR/'..."
29 while read dll; do
30     if [ -f "$dll" ]; then
31         echo "▶ $(basename "$dll")"
32         cp "$dll" "$OUT_DIR/"
33     fi
34 done < dlls.txt
35
36 rm dlls.txt

```

37
38 echo "📦 Empaquetando en '\$OUT_DIR.zip'..."
39 zip -r "\$OUT_DIR.zip" "\$OUT_DIR" > /dev/null
40
41 echo "✅ Listo: ejecutable y DLLs empaquetados en '\$OUT_DIR/' y '\$OUT_DIR.zip'"
42
43 # REFERENCIA
44 # OpenAI. (2025). Script Bash para compilar y empaquetar
45 # aplicaciones wxWidgets con MSYS2-UCRT64 en Windows
46 # [Asistente conversacional, ChatGPT].
47 # <https://chat.openai.com/>
48
49 # chmod +x compilar_y_empaquetar.sh
50 # ./compilar_y_empaquetar.sh mi_programa.cpp

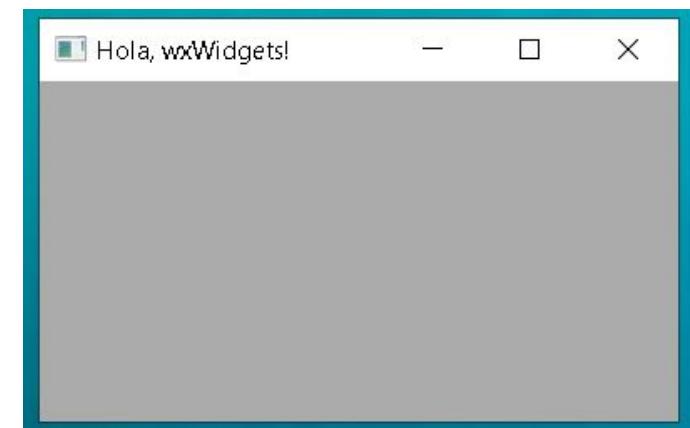
compilar_y_empaquetar.sh



minimal.cpp

```
1 // minimal.cpp
2 // LINUX
3 // g++ minimal.cpp -o minimal `wx-config --cxxflags --libs`
4 // ./minimal
5 // WINDOWS
6 // g++ minimal.cpp -o minimal.exe $(wx-config --cxxflags --libs)
7 // minimal.exe
8
9 #include <wx/wx.h>
10
11 class MyApp : public wxApp
12 {
13 public:
14     virtual bool OnInit();
15 };
16
17 class MyFrame : public wxFrame
18 {
19 public:
20     MyFrame() : wxFrame(NULL, wxID_ANY, "Hola, wxWidgets!") {}
21 };
22
23 wxIMPLEMENT_APP(MyApp);
24
25 bool MyApp::OnInit()
26 {
27     MyFrame *frame = new MyFrame();
28     frame->Show(true);
29     return true;
30 }
31
32 // REFERENCIA
33 // OpenAI. (2025, 2 de abril). ChatGPT (versión del 2 de abril de 2025)
34 // [Modelo de lenguaje de gran tamaño].
35 // https://chatgpt.com/share/67edeb7f-f6d8-800a-a08a-0c3255e6a0dd
```

Ejemplo 11: minimal.cpp

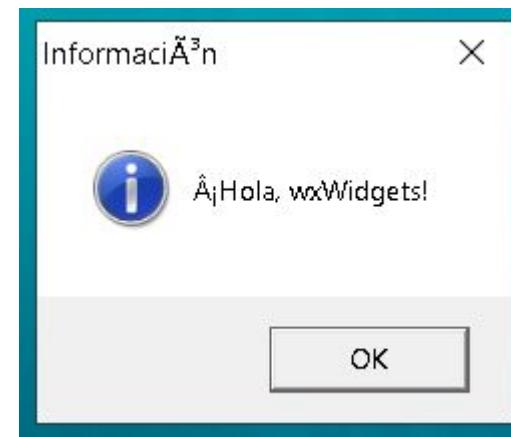
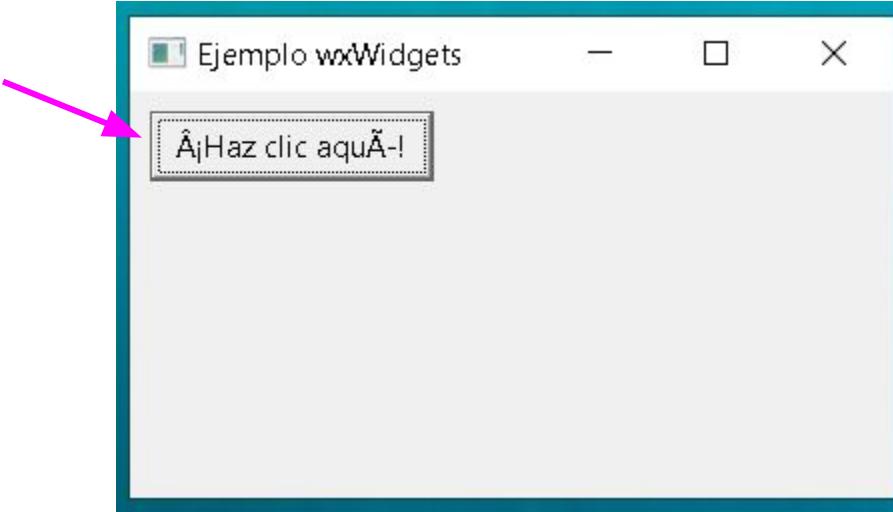


```
4     {
5         public:
6             virtual bool OnInit();
7     };
8
9     // Definir la clase de la ventana
10    class MyFrame : public wxFrame
11    {
12        public:
13            // Constructor de la ventana
14            MyFrame(const wxString& title);
15
16        private:
17            // Manejo del clic en el botón
18            void OnButtonClick(wxCommandEvent& event);
19    };
20
21    // Implementación de la clase MyApp
22    wxIMPLEMENT_APP(MyApp);
23
24    bool MyApp::OnInit()
25    {
26        MyFrame* frame = new MyFrame("Ejemplo wxWidgets");
27        frame->Show(true); // Mostrar la ventana
28        return true;
29    }
30
```

Ejemplo 12: boton.cpp

```
31 // Implementación de la clase MyFrame
32 MyFrame::MyFrame(const wxString& title)
33 : wxFrame(nullptr, wxID_ANY, title)
34 {
35     // Crear un panel dentro de la ventana
36     wxPanel* panel = new wxPanel(this, wxID_ANY);
37
38     // Crear un botón
39     wxButton* button = new wxButton(panel, wxID_ANY, ";Haz clic aquí!", wxDefaultPosition);
40
41     // Asociar el evento de clic con el método OnButtonClick
42     button->Bind(wxEVT_BUTTON, &MyFrame::OnButtonClick, this);
43 }
44
45 // Manejo del evento de clic en el botón
46 void MyFrame::OnButtonClick(wxCommandEvent& event)
47 {
48     wxMessageBox(";Hola, wxWidgets!", "Información", wxOK | wxICON_INFORMATION);
49 }
50
51 // REFERENCIA
52 // https://chatgpt.com/share/67f1cf3-b4bc-800a-8e01-e26731232b06
53
54 // LINUX
55 // g++ `wx-config --cxxflags` -o ejemplo02 ejemplo02.cpp `wx-config --libs`
56 // ./ejemplo2
57 //
58 // WINDOWS
59 // g++ boton.cpp -o boton.exe $(wx-config --cxxflags --libs)
60 // boton.exe
```

Ejemplo 12: botón.cpp



mi_programa.cpp

```
1 // mi_programa.cpp
2
3 #include <wx/wx.h>
4
5 // Definición de la clase MyApp que hereda de wxApp
6 class MyApp : public wxApp
7 {
8     public:
9         virtual bool OnInit();
10    };
11
12 // Definición de la clase MyFrame que hereda de wxFrame
13 class MyFrame : public wxFrame
14 {
15     public:
16         MyFrame(const wxString& title);
17
18     private:
19         void OnButtonClicked(wxCommandEvent& event);
20
21         wxPanel* m_panel;
22         wxStaticText* m_label;
23         wxTextCtrl* m_textCtrl;
24         wxButton* m_button;
25    };
26
27 // Implementación de la macro para inicializar la aplicación
28 wxIMPLEMENT_APP(MyApp);
29
30 // Implementación del método OnInit de MyApp
31 bool MyApp::OnInit()
32 {
33     MyFrame* frame = new MyFrame("Aplicación wxWidgets con Controles");
34     frame->Show(true);
35     return true;
36 }
37
```

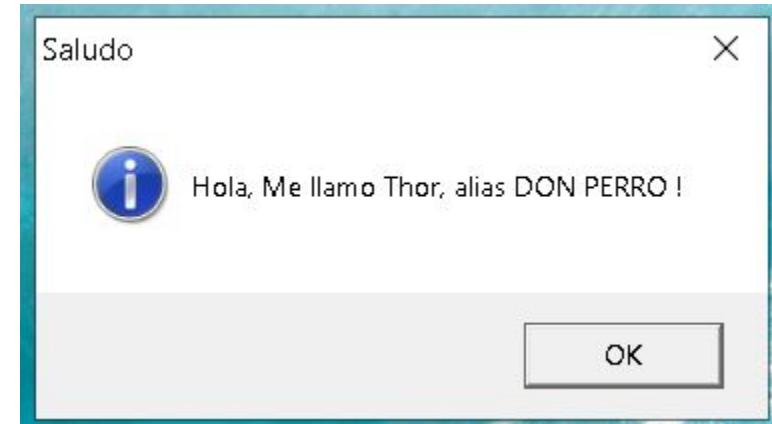
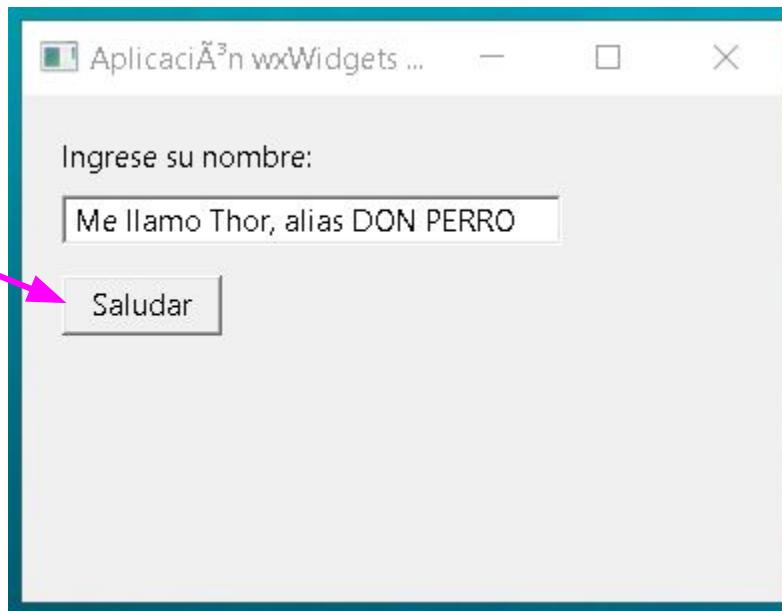
Ejemplo 13: mi_programa.cpp

Ejemplo 13: mi_programa.cpp

```
38 // Constructor de MyFrame
39 MyFrame::MyFrame(const wxString& title)
40   : wxFrame(NULL, wxID_ANY, title, wxDefaultPosition, wxSize(400, 300))
41 {
42     // Crear el panel principal
43     m_panel = new wxPanel(this, wxID_ANY);
44
45     // Crear una etiqueta
46     m_label = new wxStaticText(m_panel, wxID_ANY, "Ingrese su nombre:",
47                               wxDefaultPosition,
48                               wxDefaultSize);
49
50     // Crear un cuadro de texto
51     m_textCtrl = new wxTextCtrl(m_panel, wxID_ANY, "",
52                               wxDefaultPosition,
53                               wxDefaultSize);
54
55     // Crear un botón
56     m_button = new wxButton(m_panel, wxID_ANY, "Saludar",
57                             wxDefaultPosition,
58                             wxDefaultSize);
59
60     // Conectar el evento del botón con el manejador de eventos
61     m_button->Bind(wxEVT_BUTTON, &MyFrame::OnButtonClicked, this);
62 }
63
64 // Manejador de eventos para el botón
65 void MyFrame::OnButtonClicked(wxCommandEvent& event)
66 {
67     wxString nombre = m_textCtrl->GetValue();
68     wxString mensaje = "Hola, " + nombre + "!";
69     wxMessageBox(mensaje, "Saludo", wxOK | wxICON_INFORMATION);
70 }
```

```
69 // REFERENCIA
70 // OpenAI. (2025, 2 de abril). ChatGPT (versión del 2 de abril de 2025)
71 // [Modelo de lenguaje de gran tamaño].
72 // https://chat.openai.com/chat/67edeb7f-f6d8-800a-a08a-0c3255e6a0dd
73
74 // LINUX
75 // g++ mi_programa.cpp -o mi_programa `wx-config --cxxflags --libs`
76 // ./mi_programa
77 // WINDOWS
78 // g++ mi_programa.cpp -o mi_programa.exe $(wx-config --cxxflags --libs)
79 // ./mi_programa.exe
```

Ejemplo 13: mi_programa.cpp



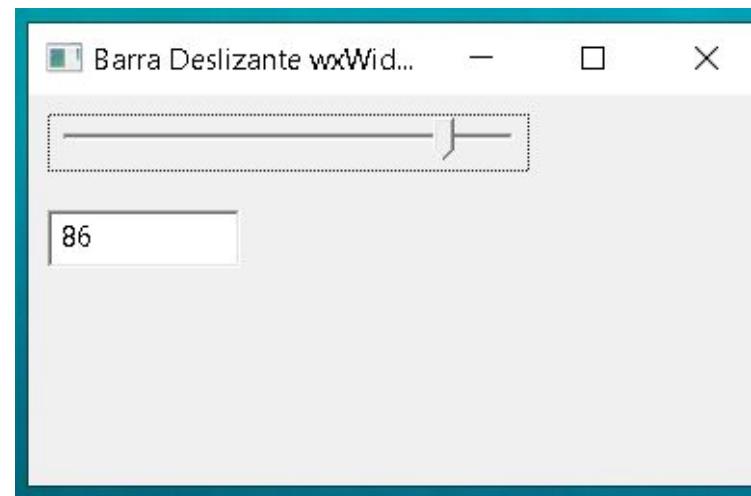
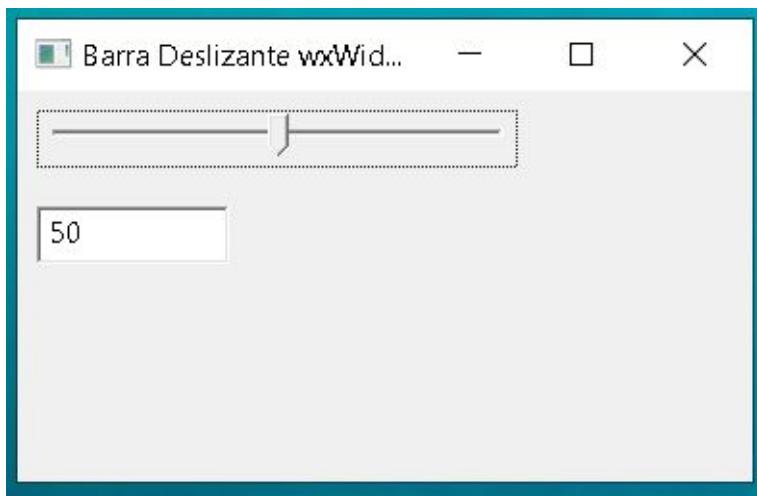
Ejemplo 14: app_slider.cpp

```
1 //include <wx/wx.h>
2
3 class MyApp : public wxApp
4 {
5     public:
6         virtual bool OnInit();
7     };
8
9
10 class MyFrame : public wxFrame
11 {
12     public:
13         MyFrame(const wxString& title);
14
15     private:
16         void OnSliderScroll(wxCommandEvent& event); // Evento para manejar el deslizamiento
17         wxSlider* m_slider; // Barra deslizante
18         wxTextCtrl* m_textCtrl; // Cuadro de texto para mostrar el valor
19         wxBoxSizer* m_sizer;
20
21         WXDECLARE_EVENT_TABLE();
22     };
23
24 // Mapa de eventos
25 wxBEGIN_EVENT_TABLE(MyFrame, wxFrame)
26     EVT_SLIDER(wxID_ANY, MyFrame::OnSliderScroll)
27 wxEND_EVENT_TABLE()
28
29 wxIMPLEMENT_APP(MyApp);
30
31 bool MyApp::OnInit()
32 {
33     MyFrame* frame = new MyFrame("Barra Deslizante wxWidgets");
34     frame->Show(true);
35     return true;
36 }
37
```

Ejemplo 14: app_slider.cpp

```
38 MyFrame::MyFrame(const wxString& title)
39     : wxFrame(nullptr, wxID_ANY, title)
40 {
41     // Crear un panel para la ventana
42     wxPanel* panel = new wxPanel(this);
43
44     // Crear la barra deslizante (slider)
45     m_slider = new wxSlider(panel, wxID_ANY, 50, 0, 100, wxPoint(20, 20), wxSize(250, 30));
46
47     // Crear un cuadro de texto para mostrar el valor de la barra deslizante
48     m_textCtrl = new wxTextCtrl(panel, wxID_ANY, "50", wxPoint(20, 70), wxSize(100, 30));
49
50     // Crear un sizer para organizar los controles
51     m_sizer = new wxBoxSizer(wxVERTICAL);
52     m_sizer->Add(m_slider, 0, wxALL, 10);
53     m_sizer->Add(m_textCtrl, 0, wxALL, 10);
54
55     panel->SetSizerAndFit(m_sizer);
56 }
57
58 // Evento que maneja el deslizamiento
59 void MyFrame::OnSliderScroll(wxCommandEvent& event)
60 {
61     // Obtener el valor de la barra deslizante
62     int value = m_slider->GetValue();
63
64     // Actualizar el cuadro de texto con el valor actual del slider
65     m_textCtrl->SetValue(wxString::Format("%d", value));
66 }
67
```

```
68 // REFERENCIA
69 // https://chatgpt.com/share/67f1cf3-b4bc-800a-8e01-e26731232b06
70
71 // LINUX
72 // g++ `wx-config --cxxflags` -o app_slider app_slider.cpp `wx-config --libs`
73 // ./app_slider
74 //
75 // WINDOWS
76 // g++ app_slider.cpp -o app_slider.exe $(wx-config --cxxflags --libs)
77 // app_slider.exe
```



Ejemplo 14: app_slider.cpp

```
app_checklist.cpp ✘ x
1
2     #include <wx/wx.h>
3     #include <wx/checklst.h>
4
5     class MyApp : public wxApp
6     {
7         public:
8             virtual bool OnInit();
9     };
10
11    class MyFrame : public wxFrame
12    {
13        public:
14            MyFrame(const wxString& title);
15
16        private:
17            void OnButtonClick(wxCommandEvent& event); // Maneja el evento de clic en el botón
18
19            wxCheckListBox* m_checkListBox; // Lista de selección múltiple
20            wxBoxSizer* m_sizer; // Contenedor (Sizer)
21            wxDECLARE_EVENT_TABLE();
22    };
23
24    // Mapa de eventos
25    wxBEGIN_EVENT_TABLE(MyFrame, wxFrame)
26        EVT_BUTTON(wxID_ANY, MyFrame::OnButtonClick)
27    wxEND_EVENT_TABLE()
28
29    wxIMPLEMENT_APP(MyApp);
30
31    bool MyApp::OnInit()
32    {
33        MyFrame* frame = new MyFrame("Menú de Selección Múltiple wxWidgets");
34        frame->Show(true);
35        return true;
36    }
37
```

Ejemplo 15: app_checklist.cpp

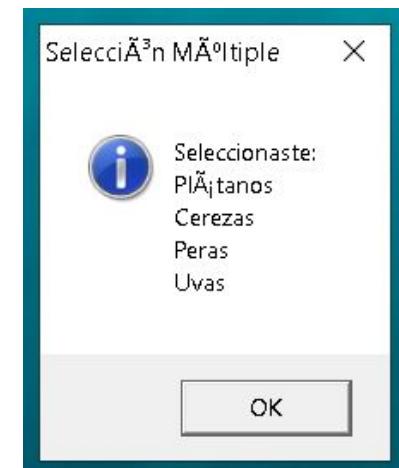
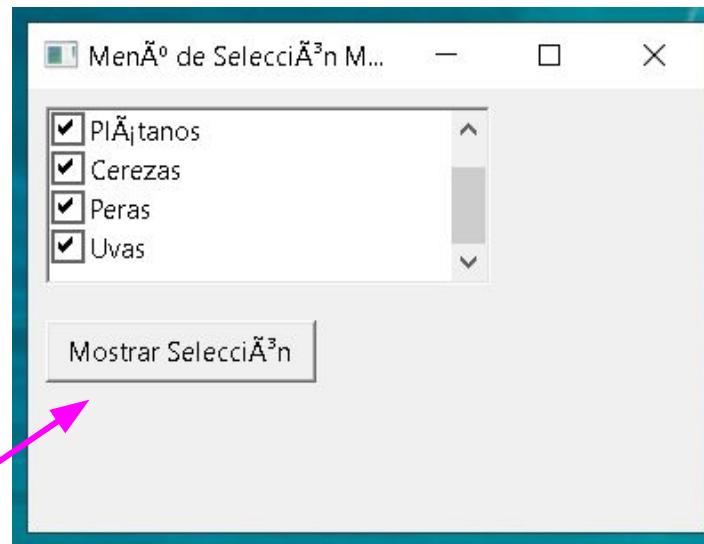
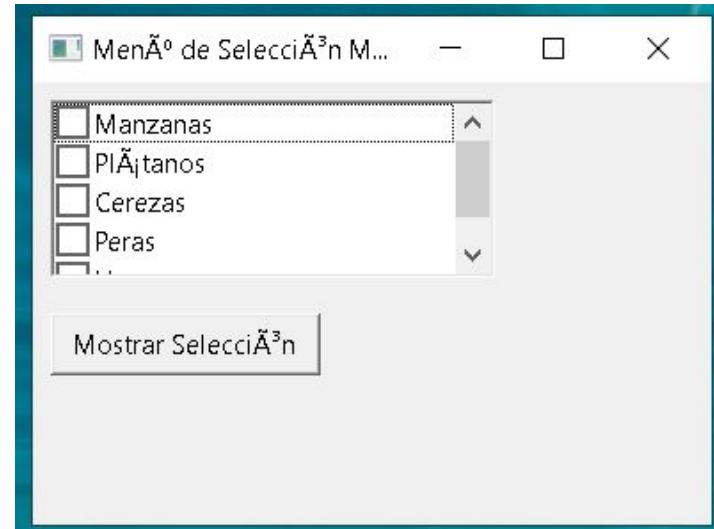
Ejemplo 15: app_checklist.cpp

```
38     MyFrame::MyFrame(const wxString& title)
39         : wxFrame(nullptr, wxID_ANY, title)
40     {
41         // Crear un panel para la ventana
42         wxPanel* panel = new wxPanel(this);
43
44         // Crear la lista de selección múltiple (wxCheckListBox)
45         wxArrayString choices;
46         choices.Add("Manzanas");
47         choices.Add("Plátanos");
48         choices.Add("Cerezas");
49         choices.Add("Peras");
50         choices.Add("Uvas");
51
52         m_checkListBox = new wxCheckListBox(panel, wxID_ANY, wxPoint(20, 20), wxSize(250, 100), choices);
53
54         // Crear un botón
55         wxButton* button = new wxButton(panel, wxID_ANY, "Mostrar Selección", wxPoint(20, 130));
56
57         // Crear un sizer para organizar los controles
58         m_sizer = new wxBoxSizer(wxVERTICAL);
59         m_sizer->Add(m_checkListBox, 0, wxALL, 10);
60         m_sizer->Add(button, 0, wxALL, 10);
61
62         panel->SetSizerAndFit(m_sizer);
63     }
64 }
```

Ejemplo 15: app_checklist.cpp

```
65 // Maneja el clic en el botón
66 void MyFrame::OnButtonClick(wxCommandEvent& event)
67 {
68     // Crear un array para almacenar las selecciones
69     wxArrayInt selections;
70
71     // Obtener las selecciones de la lista de selección múltiple
72     m_checkListBox->GetCheckedItems(selections);
73
74     wxString message = "Seleccionaste:\n";
75
76     // Construir un mensaje con las selecciones
77     for (size_t i = 0; i < selections.GetCount
78         () ; ++i)
79     {
80         message += m_checkListBox->GetString(selections[i]) + "\n";
81     }
82
83     // Mostrar las selecciones en un cuadro de mensaje
84     wxMessageBox(message, "Selección Múltiple", wxOK | wxICON_INFORMATION);
85 }
86
87 // REFERENCIA
88 // https://chatgpt.com/share/67f1cf3-b4bc-800a-8e01-e26731232b06
89 //
90 // LINUX
91 // g++ `wx-config --cxxflags` -o app_checklist app_checklist.cpp `wx-config --libs`
92 // ./app_checklist
93 //
94 // WINDOWS
95 // g++ app_checklist.cpp -o app_checklist.exe $(wx-config --cxxflags --libs)
96 // app_checklist.exe
```

Ejemplo 15: app_checklist.cpp



```
tictactoe.cpp ✘ x
1 #include <wx/wx.h>
2 #include <array>
3
4 class MyApp : public wxApp
5 {
6     public:
7         virtual bool OnInit();
8 };
9
10 class MyFrame : public wxFrame
11 {
12     public:
13         MyFrame(const wxString& title);
14
15     private:
16         void OnButtonClick(wxCommandEvent& event);
17         void ResetGame(wxCommandEvent& event); // Reiniciar juego
18         void OnExit(wxCommandEvent& event); // Salir del juego
19         bool CheckWinner();
20         void UpdateButtonLabel(int row, int col);
21
22         wxButton* m_buttons[3][3]; // Matriz de botones (3x3)
23         wxBoxSizer* m_sizer; // Sizer para organizar los controles
24         wxString m_turn; // Turno actual ("X" o "O")
25         std::array<std::array<wxString, 3>, 3> m_board; // Tablero 3x3
26         bool m_gameOver; // Si el juego terminó
27
28     wxDECLARE_EVENT_TABLE();
29 };
30
31 // Mapa de eventos
32 wxBEGIN_EVENT_TABLE(MyFrame, wxFrame)
33     EVT_BUTTON(wxEVT_BUTTON, MyFrame::OnButtonClick)
34     EVT_BUTTON(wxEVT_BUTTON, MyFrame::ResetGame) // Manejamos el evento del botón de reinicio
35     EVT_BUTTON(wxEVT_BUTTON, MyFrame::OnExit) // Manejamos el evento del botón de salida
36 wxEND_EVENT_TABLE()
37
```

Ejemplo 16: tictactoe.cpp

Ejemplo 16: tictactoe.cpp

```
38 wxIMPLEMENT_APP(MyApp);
39
40 bool MyApp::OnInit()
41 {
42     MyFrame* frame = new MyFrame("3 en Raya - Tic-Tac-Toe");
43     frame->Show(true);
44     return true;
45 }
46
47 MyFrame::MyFrame(const wxString& title)
48 : wxFrame(nullptr, wxID_ANY, title), m_turn("X"), m_gameOver(false)
49 {
50     // Crear un panel para la ventana
51     wxPanel* panel = new wxPanel(this);
52
53     // Crear un sizer para la disposición de los botones
54     m_sizer = new wxBoxSizer(wxVERTICAL);
55
56     // Crear la cuadrícula de botones 3x3
57     for (int i = 0; i < 3; ++i)
58     {
59         wxBoxSizer* rowSizer = new wxBoxSizer(wxHORIZONTAL);
60         for (int j = 0; j < 3; ++j)
61         {
62             m_buttons[i][j] = new wxButton(panel, wxID_ANY, "", wxDefaultPosition, wxSize(100, 100));
63             rowSizer->Add(m_buttons[i][j], 0, wxALL, 5);
64             m_board[i][j] = ""; // Inicializar el tablero vacío
65         }
66         m_sizer->Add(rowSizer, 0, wxALIGN_CENTER);
67     }
68
69     // Añadir un botón para reiniciar el juego
70     wxButton* resetButton = new wxButton(panel, wxID_ANY, "Reiniciar", wxDefaultPosition, wxSize(100, 50));
71     m_sizer->Add(resetButton, 0, wxALL | wxALIGN_CENTER, 10);
72 }
```

Ejemplo 16: tictactoe.cpp

```
73 // Añadir un botón para salir
74 wxButton* exitButton = new wxButton(panel, wxID_EXIT, "Salir", wxDefaultPosition, wxSize(100, 50));
75 m_sizer->Add(exitButton, 0, wxALL | wxALIGN_CENTER, 10);
76
77 panel->SetSizerAndFit(m_sizer);
78
79 // Conectar el evento de clic del botón de reinicio
80 resetButton->Bind(wxEVT_BUTTON, &MyFrame::ResetGame, this);
81
82 // Conectar el evento de clic del botón de salida
83 exitButton->Bind(wxEVT_BUTTON, &MyFrame::OnExit, this);
84 }
85
86 // Maneja el clic en los botones del tablero
87 void MyFrame::OnButtonClick(wxCommandEvent& event)
88 {
89     if (m_gameOver) return; // Si el juego terminó, no hacer nada
90
91     // Buscar el botón que fue presionado
92     for (int i = 0; i < 3; ++i)
93     {
94         for (int j = 0; j < 3; ++j)
95         {
96             if (event.GetEventObject() == m_buttons[i][j] && m_board[i][j] == "")
97             {
98                 // Marcar el tablero con el símbolo del jugador actual
99                 m_board[i][j] = m_turn;
100                UpdateButtonLabel(i, j);
101
102                // Comprobar si hay un ganador
103                if (CheckWinner())
104                {
105                    wxMessageBox(m_turn + " ha ganado!", "¡Victoria!", wxOK | wxICON_INFORMATION);
106                    m_gameOver = true;
107                }
108            else
```

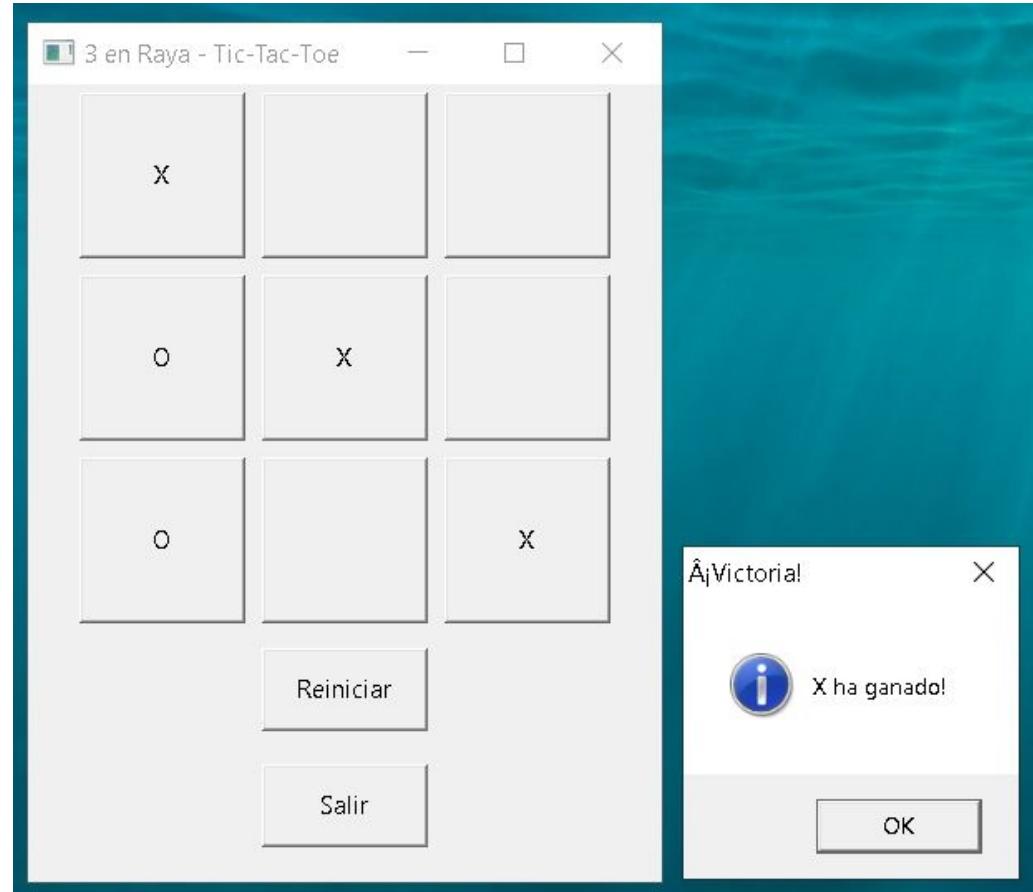
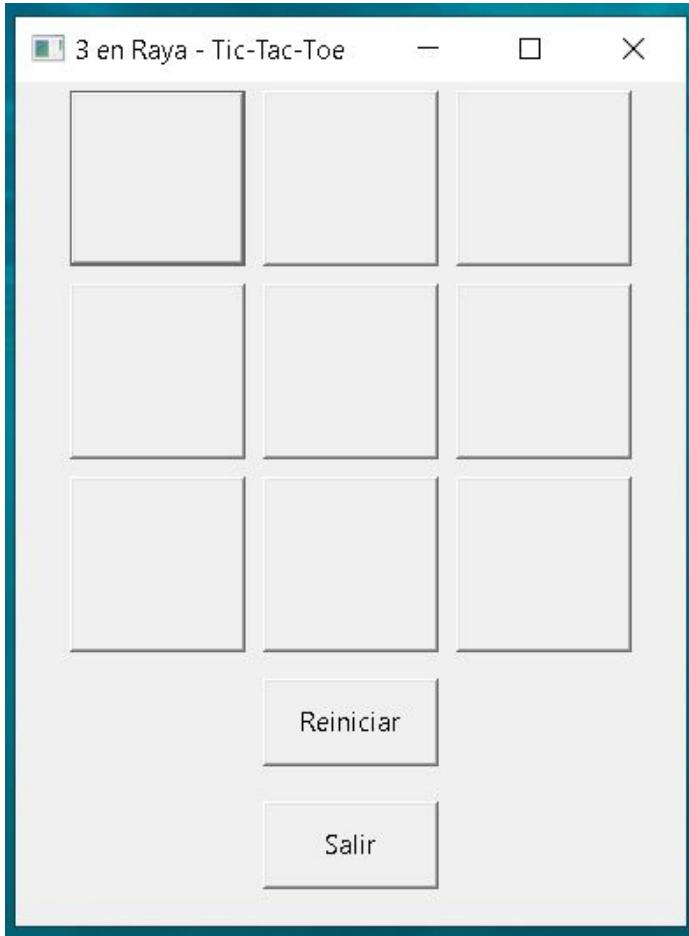
Ejemplo 16: tictactoe.cpp

```
109
110
111     {
112         // Cambiar turno
113         m_turn = (m_turn == "X") ? "O" : "X";
114     }
115     return;
116 }
117 }
118
119 // Actualizar la etiqueta de un botón con el valor del turno
120 void MyFrame::UpdateButtonLabel(int row, int col)
121 {
122     m_buttons[row][col]->SetLabel(m_board[row][col]);
123 }
124
125 // Comprobar si hay un ganador
126 bool MyFrame::CheckWinner()
127 {
128     // Comprobar filas y columnas
129     for (int i = 0; i < 3; ++i)
130     {
131         if (m_board[i][0] == m_board[i][1] && m_board[i][1] == m_board[i][2] && m_board[i][0] != "")
132             return true;
133         if (m_board[0][i] == m_board[1][i] && m_board[1][i] == m_board[2][i] && m_board[0][i] != "")
134             return true;
135     }
136
137     // Comprobar diagonales
138     if (m_board[0][0] == m_board[1][1] && m_board[1][1] == m_board[2][2] && m_board[0][0] != "")
139         return true;
140     if (m_board[0][2] == m_board[1][1] && m_board[1][1] == m_board[2][0] && m_board[0][2] != "")
141         return true;
142
143     return false;
144 }
```

Ejemplo 16: tictactoe.cpp

```
145
146 // Reiniciar el juego
147 void MyFrame::ResetGame(wxCommandEvent& event)
148 {
149     m_gameOver = false;
150     m_turn = "X";
151
152     // Limpiar el tablero
153     for (int i = 0; i < 3; ++i)
154     {
155         for (int j = 0; j < 3; ++j)
156         {
157             m_board[i][j] = "";
158             m_buttons[i][j]->SetLabel("");
159         }
160     }
161 }
162
163 // Salir del juego
164 void MyFrame::OnExit(wxCommandEvent& event)
165 {
166     Close(true); // Cierra la ventana y termina la aplicación
167 }
168
169 // REFERENCIA
170 // OPENAI. chatGPT
171 // https://chatgpt.com/share/67f1fcf3-b4bc-800a-8e01-e26731232b06
172
173 // LINUX
174 // g++ `wx-config --cxxflags` -o tictactoe tictactoe.cpp `wx-config --libs`
175 // ./tictactoe
176 //
177 // WINDOWS
178 // g++ tictactoe.cpp -o tictactoe.exe $(wx-config --cxxflags --libs)
179
```

Ejemplo 16: tictactoe.cpp



Lenguaje C. Videos Complementarios



LENGUAJES DE PROGRAMACIÓN
Explicación

- ◆ Codificación
- ◆ Código fuente
- ◆ Lenguaje máquina

➡ 58 видео

Estudia con Marisol (2018)



Programación ATS (2016)



Curso
Básico
de C

Código Facilito

1

➡ 24 видео

Código Facilito (2017)



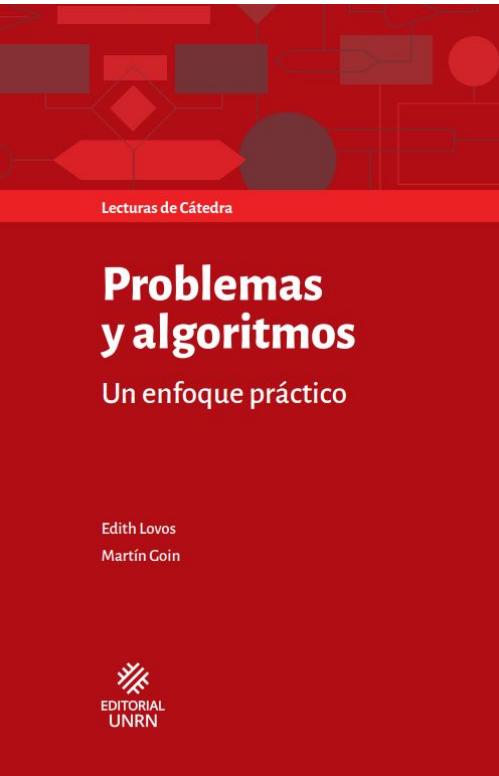
APRENDE
A PROGRAMAR
EN “C”

CURSO COMPLETO

4:12:21

Lucas Moy (2022)

Lenguaje C. Libros



Goin y Lovos (2020)



Goin (2022)



Ruiz-Linares (2023)

Lenguaje C. Libros



Universidad Nacional de Catamarca
Facultad de Tecnología y Ciencias Aplicadas
Ingeniería Electrónica

Lenguaje de Programación en C

Apuntes de cátedra: Informática

Docente: Lic. Ana María del Prado

Del Prado (s.f.)

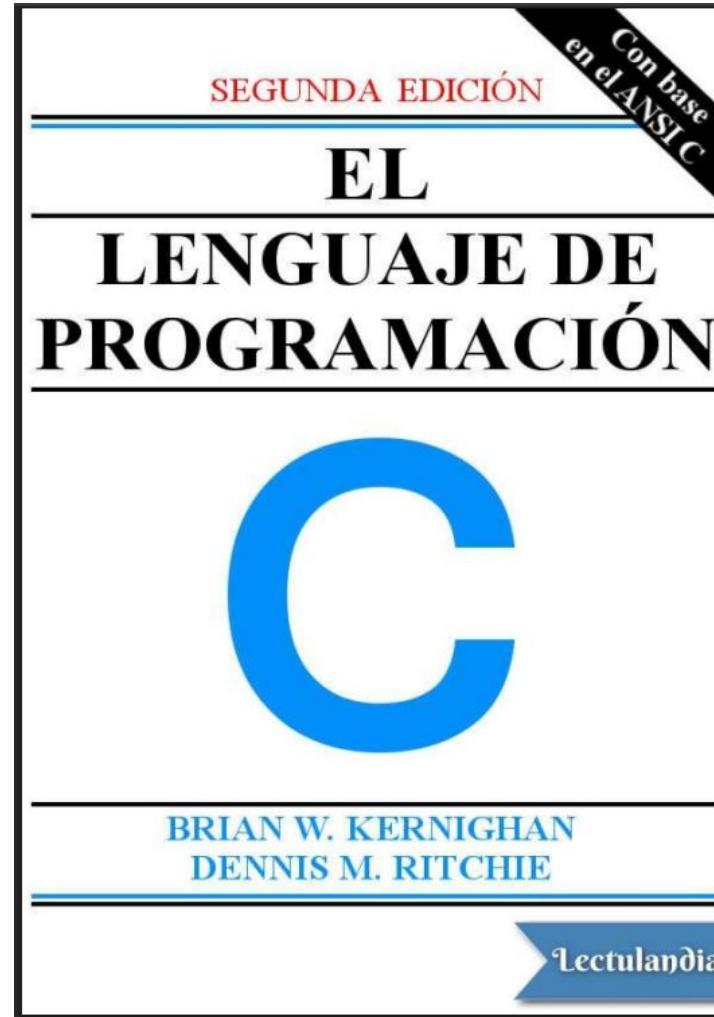


Guerra, Ramos y Vallejo (2023)



Muñoz-Guerrero (2021)

Lenguaje C. Libro



Kernighan y Ritchie (1988)

Lenguaje C++. Videos Complementarios



Programación ATS (2018)



Pildoras Informáticas (2022)

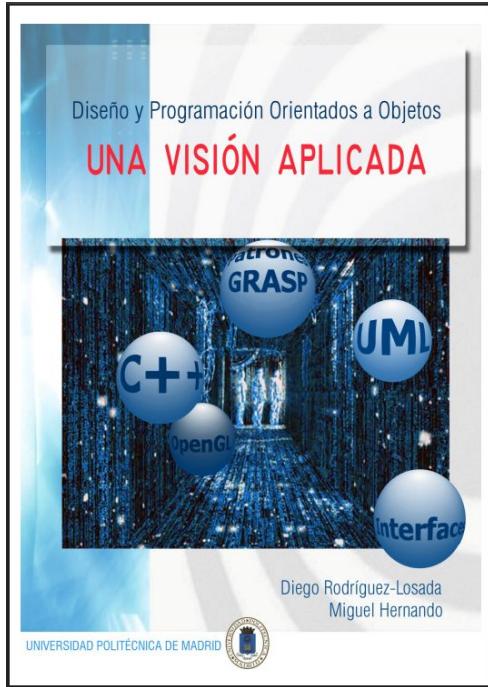


Aprende a Programar (2024)



Lucas Moy (2023)

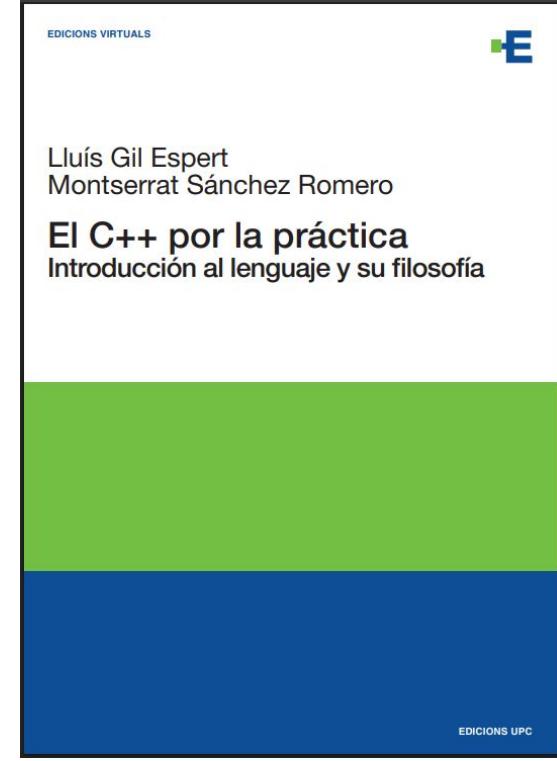
Lenguaje C++. Libros



Rodríguez-Losada y Hernando-Gutiérrez (2021)



Miguel Hernando Gutiérrez (2020)



Gil-Espert y Sánchez-Romero (1999)

Tu trabajo ocupará gran parte de tu vida y la única manera de estar verdaderamente satisfecho es hacer lo que crees que es un gran trabajo. Y la única manera de hacer un gran trabajo es amar lo que haces. Si aún no lo has encontrado, sigue buscando. No te conformes. Como con todo lo que atañe al corazón, lo sabrás cuando lo encuentres.

Ceremonia de graduación de Stanford University el 12 de junio de 2005.

– Steve Jobs

Lenguaje C++. Libros

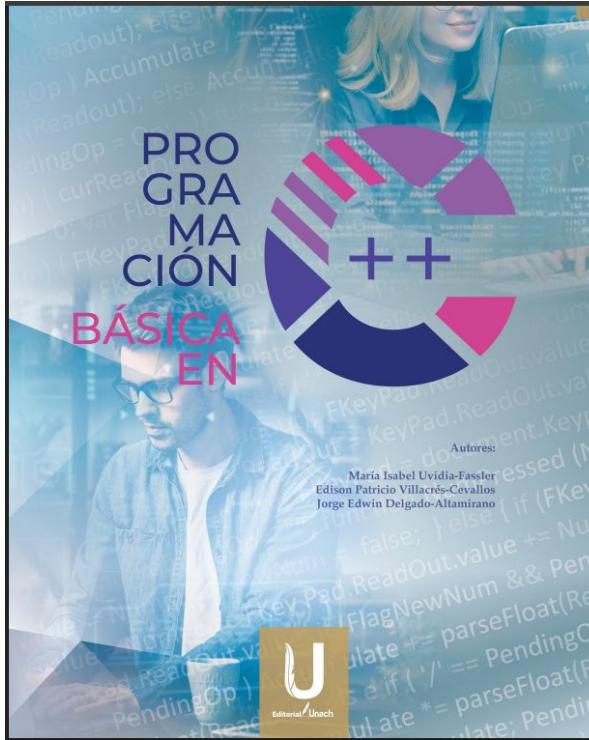


UNIVERSIDAD DE JAÉN
Departamento de Informática

FUNDAMENTOS BÁSICOS DE PROGRAMACIÓN EN
C++

Francisco Martínez del Río

Martínez del Río (2019)



Uvidia, Villacrez y Delgado (2024)



UNIVERSIDAD DE MÁLAGA
Dpto. Lenguajes y C.C. Computación
E.I.T. de Información

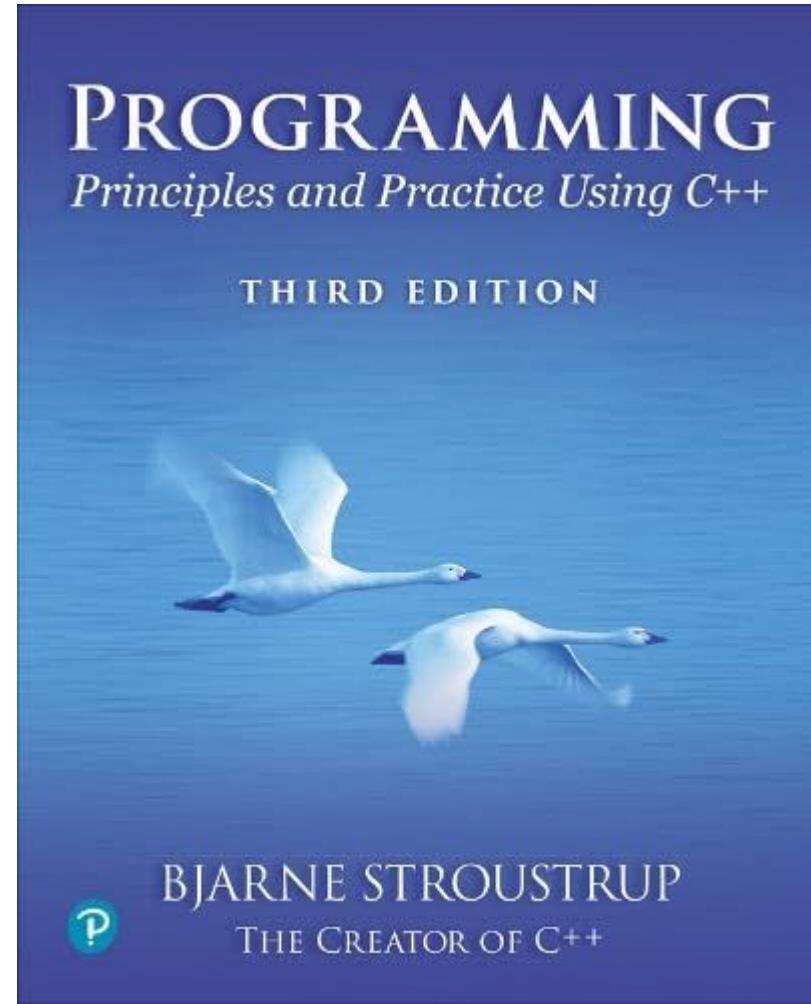
Fundamentos de Programación
con
el Lenguaje de Programación
C++

Vicente Benjumea y Manuel Roldán

13 de mayo de 2024

Benjumea y Roldán (2024)

Lenguaje C++. Libro





СПАСИБО
БОЛЬШОЕ

до свидания

REFERENCIAS

Perplexity AI. (2025, abril 4). *Importancia del lenguaje C y C++ en el mantenimiento industrial y sus aplicaciones estadísticas*. Respuesta generada por un asistente virtual.

<https://www.perplexity.ai/search/cual-es-la-importancia-del-len-KDFdfp.uTXWcFVp2F2moVw>

Raylib (2024, 26 Abr.). *Raylib projects showcase 2024* [Video]. YouTube.

<https://www.youtube.com/watch?v=GdIGE-eDlzg>

Recluit (s.f.). *¿Cuál es la historia de C++?* <https://recluit.com/cual-es-la-historia-de-c/>

Digital News Asia (2012, 6 Sep.). *Honoring the Life of Dennis Richie. UNIX co-creator and father of C.* [Foto]

<https://www.digitalnewsasia.com/digital-economy/honoring-the-life-of-dennis-ritchie-unix-co-creator-and-father-of-c>

Stroustrup (s.f.), *Welcome to Bjarne Stroustrup's homepage!* <https://www.stroustrup.com/>

Programador X (2024, 16 Abr.). *Aprende C en 45 minutos.* [Video]. YouTube

<https://www.youtube.com/watch?v=1-qjMS6C960>

Programación ATS (2016. 21 Feb.). *Programación en C - Presentación del curso.* [Video]. YouTube.

https://www.youtube.com/watch?v=5O1srQNyJXo&list=PLWtYZ2ejMVJmUTNE2QVaCd1y_6GslOeZ6

Alejandro Taboada Sanchez (2025). *Programación.* <https://programacionats.newzenler.com/>

Picando Código (2023, 10 Oct.). *¿Qué es el lenguaje de programación C?* [Video]. YouTube.

https://www.youtube.com/watch?v=sy1AuLyHc5Q&list=PLO8IO9oepSLvAvnxCn_uCsFVCAYCDsGTK

REFERENCIAS

Estudia con Marisol (2018, 21 May.). *¿Qué es CODIFICAR, CÓDIGO FUENTE y LENGUAJE MÁQUINA? - Lenguajes de Programación #1* [Video]. YouTube.

https://www.youtube.com/watch?v=fkiOskQbewM&list=PL46-B5QR6sHmlim_g4RzKqnH_5AIG7wP1

Lucas Moy (2022, 30 Nov.). *Curso de Programación en C (Desde 0 actualizado a 2023)* [Video]. ATL Academy. YouTube.

https://www.youtube.com/watch?v=v=0OTpkPo_TY

Código Facilito (2017, 9 Ene.). *Curso Básico de C - Introducción* [Video]. YouTube.

https://www.youtube.com/watch?v=rEsSxd0L4GI&list=PLpOqH6AE0tNggnxjMAJ8bX_L1a7lnBaH

Programación ATS (2018, 2 Abr.). *Aprende Programación en C++* [Video]. YouTube.

https://www.youtube.com/watch?v=dJzLmjSJc2c&list=PLWtYZ2ejMVJIUu1rEHLC0i_oibctkl0Vh

Aprende a Programar (2024, 2 Feb.). *Curso Maestro de C++ De Cero a Programador Master #1* [Video]. YouTube.

<https://www.youtube.com/watch?v=RGSEnAzS4XY&list=PL5YuOOJ-FGGjjK8j50TkEcjCcZ8dRAiPy>

Pildoras Informáticas (2022, 10 Sep.). *Curso C++. ¿Por qué aprender C++? Vídeo 1* [Video]. YouTube.

<https://www.youtube.com/watch?v=EaTeni7YCyM&list=PLU8oAIHdN5BmoxxnllWdZfRcffMPCNck5>

Lucas Moy (2023, 30 May.). *Curso Completo de C++ para Principiantes (2023)* [Video]. ATL Academy. YouTube.

<https://www.youtube.com/watch?v=VQo6gj7-hw8>

REFERENCIAS

- Goin, M. y Lovos, E. (2020, 3 Set.). *Problemas y Algoritmos: Un enfoque práctico*. Colección: Lecturas de Cátedra. pp. 120. ISBN: 978-987-4960-31-3
https://editorial.unrn.edu.ar/index.php/catalogo/346/view_bl/62/lecturas-de-catedra/94/problemas-y-algoritmos-un-enfoque-practico?tab=getmybooksTab&is_show_data=1
- Goin, M. (2022, 11 May.). *Caminando junto al Lenguaje C*. Colección: Lecturas de Cátedra. pp. 188. ISBN: 978-987-4960-76-4
https://editorial.unrn.edu.ar/index.php/catalogo/346/view_bl/62/lecturas-de-catedra/26/caminando-junto-al-lenguaje-c?tab=getmybooksTab&is_show_data=1
- Ruiz-Linares, A. E. (2023, 28 Nov.). *Programación en C para Ingenieros electrónicos*. Tecnológico Nacional de México. Instituto Tecnológico de Minatitlán. <https://rinacional.tecnm.mx/handle/TecNM/6243>
- Del Prado, A. M. (s.f.). *Lenguaje de Programación en C. Apuntes de Catedra. Informática*. Universidad Nacional de Catamarca – Secretaría de Ciencia y Tecnología Editorial Científica Universitari. a ISBN: 978-987-661-127-5
<https://editorial.unca.edu.ar/Publicacione%20on%20line/ CUADERNOS%20DE%20CATEDRA/Ana%20Maria%20del%20Prado/ApunteDeCatedralinformatica2.pdf>
- Guerra, J. E., Ramos, M. V. y Vallejo, G. E. (2023, 22). *Programando en C desde la práctica: Problemas Resueltos*. Puerto Madero Editorial. <https://puertomaderoeeditorial.com.ar/index.php/pmea/catalog/book/36>
- Muñoz-Guerrero, L. E. (2021). *Introducción a la programación con C*. Editorial CIMTED. [Descargar PDF].
<https://memoriascimted.com/wp-content/uploads/2022/04/Libro-Introduccion-a-la-programacion-con-C.pdf>

REFERENCIAS

ASNetworking (2025, 12 May.). COPIA archivos y carpetas con MÁXIMO rendimiento con este comando.

https://editorial.unrn.edu.ar/index.php/catalogo/346/view_bl/62/lecturas-de-catedra/94/problemas-y-algoritmos-un-enfoque-practico?tab=getmybooksTab&is_show_data=1

Miguel Hernando Gutiérrez (2020, Feb.). *C++ estándar. Apuntes de Informática Industrial y Comunicaciones*. Escuela Técnica Superior de Ingeniería y Diseño Industrial.

<https://blogs.upm.es/miguelhernando/wp-content/uploads/sites/734/2022/11/libroC2020conPortada.pdf>

Rodríguez-Losada, D y Hernando-Gutierrez, M. (2021). Diseño y programación Orientados a Objetos. Una Visión Aplicada. 2da. Revisión.

https://blogs.upm.es/miguelhernando/wp-content/uploads/sites/734/2022/11/DisenoyProgramacionOrientadaObjetos_UnaVisionAplicada_2021.pdf

Gil-Espert. L. y Sánchez-Romero, M. (1999). *El C++ por la práctica: introducción al lenguaje y su filosofía*. Edicions UPC.
<https://upcommons.upc.edu/handle/2099.3/36408>

Martínez del Río, F. (2019). *Fundamentos Básicos de Programación en C++*. Universidad de Jaén.
<https://www4.ujaen.es/~fmartin/apuntesC++.pdf>

Uvidia, M. I., Villacrez, E. P. y Delgado J. E. (2024). *Programación Básica en C++*. Editorial UNACH. Universidad Nacional de Chimborazo. <https://editorial.unach.edu.ec/index.php/Editorial/catalog/book/253>

Benjumea, V., y Roldán, M. (2024). *Fundamentos de Programación con el Lenguaje de Programación C++*. Universidad de Málaga. Recuperado de http://www.lcc.uma.es/~vicente/docencia/cppdoc/programacion_cxx.pdf

REFERENCIAS

Kernighan, B. W. y Ritchie, D. M. (1988). *El Lenguaje de Programación C* (2.^a ed., Versión en español). Trad. disponible en InfoBiblioteca Digital. Recuperado de <https://info-biblioteca.mincyt.gob.ve/books/el-lenguaje-de-programacion-c-espanol/>

Bjarne Stroustrup (2024). *Programming -- Principles and Practice Using C++* (3rd Edition).
<https://www.stroustrup.com/programming.html>

Canva (s.f.). *Diseño de Carátula*. <https://www.canva.com>

Howard, C (2013, Oct. 5). *F-35 Joint Strike Fighter benefits from modern software testing, quality assurance*.
Military-Aerospace Electronics.
<https://www.militaryaerospace.com/home/article/16715746/f-35-joint-strike-fighter-benefits-from-modern-software-testing-quality-assurance>

open62541 Project. (s.f.). *open62541 – Open Source OPC UA (IEC62541) implementation*. <https://www.open62541.org/>

libmodbus Project. (s.f.). *libmodbus – A Modbus library for Linux, Mac OS X, FreeBSD, QNX and Windows*. <https://libmodbus.org/>

FreeCAD Project. (s.f.). *FreeCAD: Your own 3D parametric modeler*. <https://www.freecad.org/>

FreeCAD Project. (s.f.a). *FreeCAD (Version en desarrollo) [Código fuente]*. GitHub. <https://github.com/FreeCAD/FreeCAD>

REFERENCIAS

7-Zip. (s.f.). *7-Zip*. <https://www.7-zip.org/>

WinMD5. (s.f.). *WinMD5 Free*. <https://www.winmd5.com/>

Notepad++ Team. (s.f.). *Notepad++*. <https://notepad-plus-plus.org/>

Gnuplot Project. (s.f.). *Gnuplot*. <http://www.gnuplot.info/>

Nomacs Project. (s.f.). *Nomacs – Image Lounge*. <https://nomacs.org/>

XnView Software. (s.f.). *XnView Software*. <https://www.xnview.com/en/>

Krzysztof Kowalczyk. (s.f.). *SumatraPDF*. <https://www.sumatrapdfreader.org/free-pdf-reader>

Git Project. (s.f.). *Git for Windows*. <https://git-scm.com/downloads/win>

GNU Project. (s.f.). *GCC, the GNU Compiler Collection*. <https://gcc.gnu.org/>

Clang Team. (s.f.). *Clang: a C language family frontend for LLVM*. <https://clang.org/>

Embarcadero Technologies. (s.f.). *Embarcadero*. <https://www.embarcadero.com/>

Microsoft. (s.f.). *Visual Studio: precios y suscripciones*. <https://visualstudio.microsoft.com/vs/pricing/?tab=paid-subscriptions>

REFERENCIAS

IBM. (s.f.). *IBM Open XL C/C++ for AIX*. <https://www.ibm.com/products/open-xl-cpp-aix-compiler-power>

MinGW-w64 Project. (s.f.). *MinGW-w64*. <https://www.mingw-w64.org/>

MSYS2 Team. (s.f.). *MSYS2*. <https://www.msys2.org/>

wxWidgets Project. (s.f.). *wxWidgets: Cross-Platform GUI Library*. <https://www.wxwidgets.org/>

CMake Project. (s.f.). *CMake*. <https://cmake.org/>

Neovim Project. (s.f.). *Neovim*. <https://neovim.io/>

Davis E. King. (s.f.). *dlib C++ Library*. <https://dlib.net/>

GNU Project. (s.f.). *GNU Scientific Library (GSL)*. <https://www.gnu.org/software/gsl/>

Guennebaud, G., Jacob, B., y colaboradores. (s.f.). *Eigen*. <https://eigen.tuxfamily.org/>

mlpack Developers. (s.f.). *mlpack: fast, flexible machine learning library*. <https://www.mlpack.org/>

ImageMagick Studio LLC. (s.f.). *ImageMagick*. <https://imagemagick.org/index.php>

SDL Project. (s.f.). *Simple DirectMedia Layer*. <https://www.libsdl.org/>

REFERENCIAS

raylib Technologies. (s.f.). *raylib: a simple and easy-to-use library to enjoy videogames programming.* <https://www.raylib.com/>

Wireshark Foundation. (s.f.). *Wireshark.* <https://www.wireshark.org/>

Perl Foundation. (s.f.). *The Perl Programming Language.* <https://www.perl.org/>

Curl Project. (s.f.). *cURL.* <https://curl.se/>

Python Software Foundation. (s.f.). *Welcome to Python.org.* <https://www.python.org/>

pygame Community. (s.f.). *pygame.* <https://www.pygame.org/>

Arma Developer Team. (s.f.). *ARMA.* <https://arma.sourceforge.net/>

OpenCV.org. (s.f.). *OpenCV.* <https://opencv.org/>

Leenissen, R. (s.f.). *Code::Blocks.* <https://leenissen.dk/>

Boost Community. (s.f.). *Boost C++ Libraries.* <https://www.boost.org/>

Microsoft. (s.f.). *Microsoft.* <https://www.microsoft.com/>

OpenAI (2025, 29 Abr.). *Script Bash para compilar y empaquetar aplicaciones wxWidgets con MSYS2-UCRT64 en Windows [Asistente conversacional, ChatGPT].* <https://chat.openai.com/>

REFERENCIAS

Briana. (s.f.). *Lenguaje C* [PDF]. Cheatography. <https://cheatography.com/briana/cheat-sheets/lenguaje-c/pdf/>

Tche Fonseca Alfaro. (s.f.). *GUÍA DE SINTAXIS DEL LENGUAJE C++* [PDF]. WordPress.
<https://tchefonsecalfaro.wordpress.com/wp-content/uploads/2009/03/manualc.pdf>