

Ruleta De Pares

Nombre Completo de los Estudiantes: Kehit fernandez y tiziana sandoval

Nombre de la Institución: Escuela tecnica N°2

Curso: Laboratorio de Programación 5° 2° B

Profesor: Mansilla Muñoz York E.

Fecha: 27/10/2025

Objetivos: Algunos de los objetivos de nuestro programa son cosas como el desempate entre amigos (por alguna apuesta), el que gana decide que se hace,etc .

Básicamente nuestro objetivo es una ruleta divertida y que funcione con otras reglas.

Descripcion: El programa simula una ruleta virtual en la que hay cinco números pares y un número impar. El objetivo del jugador es evitar que salgan cinco números pares seguidos, ya que en ese caso pierde la partida. En cambio, si durante el juego sale el número impar, gana esa ronda y el contador de pares se reinicia. Cada vez que el usuario presiona ENTER, la ruleta "gira" y el programa genera un número al azar de entre los seis posibles para Luego mostrar el número obtenido y actualizar el conteo de los pares seguidos.

El programa continúa girando hasta que Aparece un número impar salta un mensaje de “¡GANASTE ESTA RONDA!”, O si se obtienen los cinco números pares consecutivos salta un mensaje de “¡PERDISTE!”

Herramientas: Word, Visual studio code, plantilla apa v7, github, g++, drive.

Resumen general de lo que hace el programa:

El juego de la ruleta par/impar en C++ simula una ruleta con más números pares que impares. El jugador presiona Enter para "girar" la ruleta, y si sale un número impar, gana la ronda. Sin embargo, si salen cinco números pares seguidos, pierde el juego. Despues de cada partida, se le pregunta al jugador si desea jugar de nuevo, permitiendo múltiples partidas hasta que decida salir. El objetivo es evitar que salgan cinco números pares consecutivos y tratar de obtener un número impar para ganar cada ronda.

Estructura:

```
1 #include <iostream>
2 #include <cstdlib>
3 #include <ctime>
4
5 using namespace std;
6
```

Esta imagen representa la estructura, esta se compone de 4 lineas las cuales son

- #include <iostream>
- #include <cstdlib>

- #include <ctime>
- using namespace std;

#include <iostream> : Esta línea es la librería de entrada y salida estándar. Gracias a ella podés usar comandos como “cout” que sirve para mostrar texto por pantalla, y “cin” que sirve para leer lo que el usuario escriba.

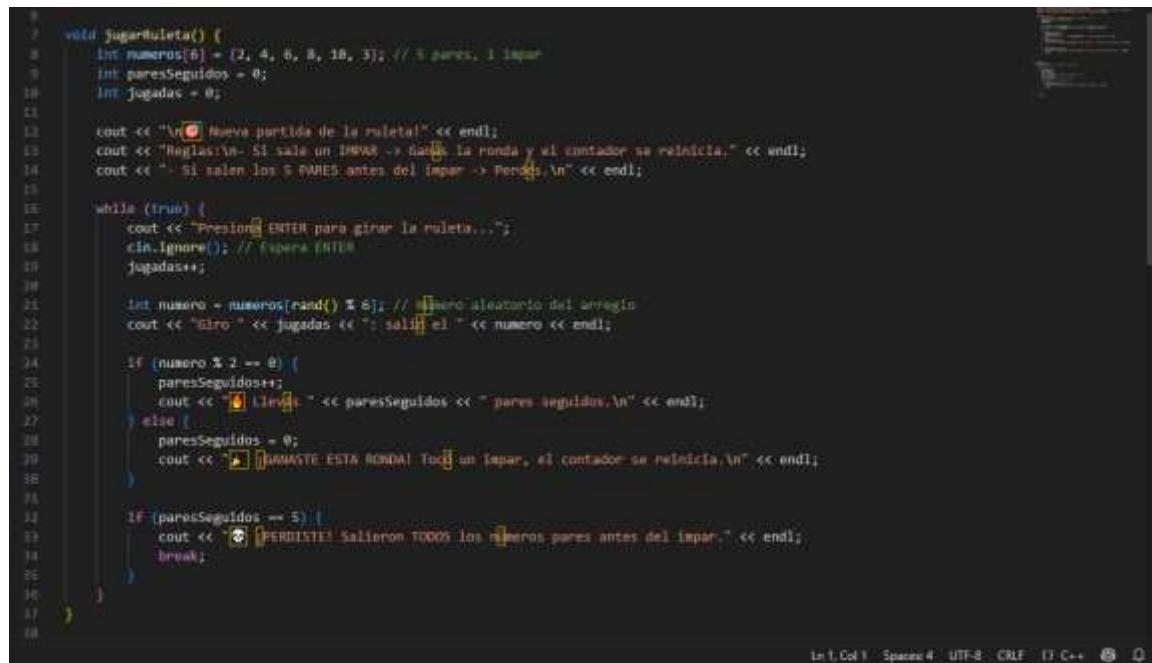
#include <cstdlib> : Incluye funciones útiles del lenguaje, entre ellas: rand() que genera un número aleatorio (por ejemplo 0, 1, 2...) y srand() que inicializa el generador de números aleatorios para que no repita siempre los mismos.

#include <ctime> : Se usa junto con srand(time(0)); time(0) devuelve el tiempo actual del sistema (en segundos). Eso se usa para que los números aleatorios cambien cada vez que ejecutás el programa.

using namespace std; : esta parte de la estructura sirve para no tener que escribir std:: antes de cosas como cout, cin, string, etc.

Sin esto, tendrías que escribir std::cout todo el tiempo.

Desarrollo del código:



```

1 void jugarRuleta() {
2     int numeros[6] = {2, 4, 6, 8, 10, 3}; // 3 pares, 1 impar
3     int paresSeguidos = 0;
4     int jugadas = 0;
5
6     cout << "\n[NUEVA PARTIDA DE LA RULETA!]\n" << endl;
7     cout << "Reglas:\n- Si sale un IMPAR -> GANAS la ronda y el contador se reinicia.\n- Si salen los 5 PARES antes del IMPAR -> Perdes.\n" << endl;
8
9     while (true) {
10         cout << "Presiona ENTER para girar la ruleta..."; 
11         cin.ignore(); // Espera ENTER
12         jugadas++;
13
14         int numero = numeros[rand() % 6]; // [numero aleatorio del arreglo]
15         cout << "[numero]" << jugadas << ": salió el " << numero << endl;
16
17         if (numero % 2 == 0) {
18             paresSeguidos++;
19             cout << "[LLEVAS]" << paresSeguidos << "[PARES SEGUIDOS]\n" << endl;
20         } else {
21             paresSeguidos = 0;
22             cout << "[PERDISTE ESTA RONDA!] Todo un IMPAR, el contador se reinicia.\n" << endl;
23         }
24
25         if (paresSeguidos == 5) {
26             cout << "[PERDISTE!] Salieron TODOS los numeros pares antes del IMPAR.\n" << endl;
27             break;
28         }
29     }
30 }

```

void jugarRuleta() {} : Esto declara una función llamada jugarRuleta, que no devuelve nada (void). Dentro de esta función está el código de una partida del juego.

-Arreglo de números y variables:

- int numeros[6] = {2, 4, 6, 8, 10, 3};
- int paresSeguidos = 0;
- int jugadas = 0;

int numeros[6] : es un arreglo (o lista) de 6 valores que pueden salir en la ruleta, Hay 5 pares (2, 4, 6, 8, 10) y 1 impar (3).

Esto significa que basicamente hay más probabilidades de perder que de ganar.

paresSeguidos : cuenta cuántos números pares seguidos salieron en tu partida.

jugadas : cuenta cuántas veces giraste la ruleta.

Introducción del juego:

- cout << "\n"  Nueva partida de la ruleta!" << endl;
- cout << "Reglas:\n- Si sale un IMPAR -> Ganás la ronda y el contador se reinicia." << endl;
- cout << "- Si salen los 5 PARES antes del impar -> Perdés.\n" << endl;

Esto lo que hace es mostrar las reglas que uno como desarrollador elige para el juego por pantalla.

\n y << endl significan un salto de línea.

Cabe aclarar que el programa usa emojis para hacerlo más divertido. :D

Primer Bucle de la ruleta:

```
while (true) {
    cout << "Presioná ENTER para girar la ruleta...";
    cin.ignore(); // Espera que el jugador presione ENTER
    jugadas++;
    int numero = numeros[rand() % 6]; // Número aleatorio del arreglo
    cout << "Giro " << jugadas << ": salió el " << numero << endl;
}
```

while (true): es un bucle infinito que se repetirá hasta que algo lo interrumpa con break.

cin.ignore(): este comando hace una pausa esperando que presiones ENTER.

rand() % 6 : genera un número aleatorio entre 0 y 5 (porque % es el operador “resto”).

-Si sale 0 → elige numeros[0] (que es 2)

-Si sale 5 → elige numeros[5] (que es 3)

Así se simula que la ruleta “gira” y elige un número para Luego mostrarlo en pantalla.

Funcionamiento de la logica de ganar o perder:

numero % 2 == 0 significa que si “¿el número es divisible por 2?” (o sea, ¿es par?), Si sí, suma 1 al contador de pares (paresSeguidos++), Si no (es impar), reinicia el contador a 0, y muestra que ganaste esa ronda.

Luego el segundo if pregunta, Si llegaste a tener 5 pares seguidos, entonces perdiste y el break rompe el bucle osea que termina la partida.

Final del codigo:

```
39 int main() {
40     srand(time(0)); // Inicializa números aleatorios
41
42     while (true) {
43         jugarRuleta();
44         char opcion;
45         cout << "\nQuieres jugar otra vez? (s/n): ";
46         cin >> opcion;
47         cin.ignore(); // limpia el buffer
48
49         if (tolower(opcion) != 's') {
50             cout << "\nGracias por jugar, hasta la próxima!" << endl;
51             break;
52         }
53     }
54
55     return 0;
56 }
```

int main() {

srand(time(0)); // Inicializa los números aleatorios

Esto se ejecuta una sola vez al comienzo.

time(0) devuelve un número distinto cada segundo, y srand lo usa para que rand() no repita los mismos resultados.

Funcionamiento del Repetir el juego:

```

while (true) {

    jugarRuleta(); // llama a la función anterior

    char opcion;

    cout << "\n¿Querés jugar otra vez? (s/n): ";

    cin >> opcion;

    cin.ignore(); // limpia el buffer del teclado


    if (tolower(opcion) != 's') {

        cout << "\n👋 Gracias por jugar, ¡hasta la próxima!" << endl;

        break;

    }

}

```

Después de cada partida, el programa pregunta si querés volver a jugar. Si escribís s (de "sí"), vuelve a empezar. Si escribís cualquier otra cosa, termina con un mensaje de despedida.

`tolower(opcion)` convierte la letra a minúscula, para que funcione igual si ponés S o s.

Fin del programa:

```
return 0;
```

Indica que el programa finalizó correctamente.

Github: el siguiente enlace te va a llevar al repositorio de github de nuestro proyecto.

https://github.com/kehit19/Ruleta_De_Pares.git