



Universidad Autónoma de Querétaro

Facultad de Informática

MINI PROYECTO

ESTRUCTURA DE DATOS

Alumno: Erick Jesús Cruz Martínez

Profesor: Roberto Becerril

6 de marzo de 2023



1. Descripción del problema

El programa comienza generando un número aleatorio entre 1 y 100, utilizando la función `rand()` de C++ y la función `time(NULL)` para inicializar el generador de números aleatorios con la hora actual. Luego, el programa solicita al usuario que ingrese un número y lo compara con el número secreto generado aleatoriamente.

Si el número ingresado por el usuario es igual al número secreto, el programa felicita al usuario y termina. Si el número ingresado es incorrecto, el programa informa si el número secreto es mayor o menor que el número ingresado y permite que el usuario haga más intentos hasta que se acabe el número máximo de intentos permitidos.

Si el usuario no adivina el número secreto dentro del número de intentos permitido, el programa revela el número secreto y termina.

El programa también proporciona mensajes de bienvenida y de despedida, así como mensajes de ayuda para guiar al usuario a través del proceso de adivinanza.

2. Paso a Paso

1. El programa comienza por incluir las librerías necesarias: `iostream`, `cstdlib` y `ctime`. La librería `iostream` se usa para entrada/salida de datos, la librería `cstdlib` se usa para generar números aleatorios y la librería `ctime` se usa para inicializar el generador de números aleatorios.

```
1  #include <iostream>
2  #include <cstdlib>
3  #include <ctime>
4
```

2. Se define la función `main()` que es la función principal del programa.
3. Dentro de la función `main()`, se utiliza la función `srand()` de la librería `ctime` para inicializar el generador de números aleatorios con la hora actual. Esto se hace para que se genere un número aleatorio diferente cada vez que se ejecute el programa.



```
int main() {  
    srand(time(NULL));
```

4. Se genera un número aleatorio entre 1 y 100 utilizando la función `rand()` de la librería `cstdlib`.

```
const int num_intentos = 5; // número máximo de intentos  
int numero_secreto = rand() % 100 + 1; // número secreto aleatorio entre 1 y 100  
int intentos_restantes = num_intentos; // número de intentos restantes
```

5. Se muestra un mensaje de bienvenida al usuario con una breve descripción del juego.

```
cout << "Bienvenido al juego de adivinar el número.\n";  
cout << "Tienes " << num_intentos << " intentos para adivinar el número secreto.\n";
```

6. Se inicializa una variable `intentos` con el valor de 5. Esto indica el número máximo de intentos permitidos para adivinar el número secreto.

```
const int num_intentos = 5; // número máximo de intentos
```

7. Se utiliza un bucle `do-while` para permitir al usuario hacer intentos de adivinanza hasta que adivine el número secreto o hasta que se acaben los intentos.
8. Dentro del bucle `do-while`, se muestra un mensaje solicitando al usuario que ingrese un número.

```
do {
    cout << "Introduce un numero entre 1 y 100: ";
    cin >> numero_usuario;

    if (numero_usuario == numero_secreto) {
        cout << "¡Felicidades, has adivinado el numero secreto!\n";
        return 0; // salir del programa
    }
    else if (numero_usuario < numero_secreto) {
        cout << "El numero ingresado es menor que el número secreto.\n";
    }
    else {
        cout << "El numero ingresado es mayor que el número secreto.\n";
    }

    intentos_restantes--;
    if (intentos_restantes == 0) {
        cout << "Lo siento, has agotado todos tus intentos.\n";
        cout << "El número secreto era " << numero_secreto << ".\n";
        return 0; // salir del programa
    }
    else {
        cout << "Te quedan " << intentos_restantes << " intentos restantes.\n";
    }
} while (intentos_restantes > 0);
```

9. Se utiliza la función cin de la librería iostream para leer el número ingresado por el usuario.

```
cout << "Introduce un numero entre 1 y 100: ";
cin >> numero_usuario;
```

10. Se compara el número ingresado por el usuario con el número secreto generado aleatoriamente. Si son iguales, se muestra un mensaje felicitando al usuario y se sale del bucle do-while.

```
if (numero_usuario == numero_secreto) {
    cout << "¡Felicidades, has adivinado el numero secreto!\n";
    return 0; // salir del programa
}
else if (numero_usuario < numero_secreto) {
    cout << "El numero ingresado es menor que el número secreto.\n";
}
else {
    cout << "El numero ingresado es mayor que el número secreto.\n";
}
```

11. Si el número ingresado por el usuario es incorrecto, se muestra un mensaje indicando si el número secreto es mayor o menor que el número ingresado por el usuario. Además, se resta 1 de la variable intentos.



-
12. **Si el usuario agota los intentos sin adivinar el número secreto, se muestra un mensaje revelando el número secreto y se sale del bucle do-while.**
 13. **Se muestra un mensaje de despedida al usuario y se cierra el programa.**