

## 1. DATOS INFORMATIVOS

Carrera: Electronica y automatizacion

Asignatura: Fundamentos de programacion

Tema del taller: Correccion de la conjunta

Integrantes: Richard Casa

Fecha: 23/01/2026 Paralelo: 29583

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <time.h>
```

```
#include <string.h>
```

```
#define MAX_INTENTOS 10
```

```
#define TAM_TEXTO 30
```

```
typedef struct {
```

```
    int valor;
```

```
    char resultado[TAM_TEXTO]; // "Correcto", "Menor que el secreto", "Mayor que el secreto"
```

```
} Intento;
```

```
// --- Prototipos ---
```

```
int leerEntero(const char *mensaje);
```

```
int leerEnteroEnRango(const char *mensaje, int min, int max);
```

```
int esDuplicado(const Intento intentos[], int usados, int valor);

void imprimirResumen(int min, int max, int secreto, const Intento intentos[], int usados, int
maxIntentos);

int main(void) {

    int min, max;

    int maxIntentos;

    int numeroSecreto;

    Intento intentos[MAX_INTENTOS];

    int usados = 0;

    // -----

    // RF01: Configuración del rango

    // -----

    printf("=== CONFIGURACION DEL RANGO ===\n");

    min = leerEntero("Ingrese el minimo del rango: ");

    max = leerEntero("Ingrese el maximo del rango: ");

    if (min >= max) {

        printf("ERROR: El minimo debe ser menor que el maximo.\n");

        return 1;

    }

    // Inicializar RNG y secreto

    srand((unsigned)time(NULL));
```

```
numeroSecreto = (rand() % (max - min + 1)) + min;

// -----

// RF02: Configuración de intentos

// -----

printf("\n=== CONFIGURACION DE INTENTOS ===\n");

maxIntentos = leerEnteroEnRango("Seleccione el maximo de intentos (1 a 10): ", 1,
MAX_INTENTOS);

printf("\n=== COMIENZA EL JUEGO ===\n");

// -----

// Ciclo principal (RF03, RF04)

// -----

while (usados < maxIntentos) {

    int valor;

    printf("\nIntento %d de %d. Ingrese un numero: ", usados + 1, maxIntentos);

    if (scanf("%d", &valor) != 1) {

        // Limpiar buffer en caso de entrada no numérica

        int c; while ((c = getchar()) != '\n' && c != EOF) {}

        printf("ERROR: Entrada no valida. Intente nuevamente.\n");

        continue;

    }

    // Validar rango (RF03)
```

```
if (valor < min || valor > max) {  
  
    printf("ERROR: Numero fuera del rango [%d - %d]\n", min, max);  
  
    continue;  
  
}  
  
// Validar duplicado (RF03)  
  
if (esDuplicado(intentos, usados, valor)) {  
  
    printf("ERROR: Ya ingresaste este numero antes.\n");  
  
    continue;  
  
}  
  
// Registrar intento (RF04)  
  
intentos[usados].valor = valor;  
  
// Calcular resultado textual para RF05  
  
if (valor == numeroSecreto) {  
  
    strncpy(intentos[usados].resultado, "Correcto", TAM_TEXTO - 1);  
  
    intentos[usados].resultado[TAM_TEXTO - 1] = '\0';  
  
} else if (valor < numeroSecreto) {  
  
    strncpy(intentos[usados].resultado, "Menor que el secreto", TAM_TEXTO - 1);  
  
    intentos[usados].resultado[TAM_TEXTO - 1] = '\0';  
  
} else {  
  
    strncpy(intentos[usados].resultado, "Mayor que el secreto", TAM_TEXTO - 1);  
  
    intentos[usados].resultado[TAM_TEXTO - 1] = '\0';  
  
}
```

```
// Mensaje interactivo

if (valor == numeroSecreto) {

    printf("¡FELICIDADES! Adivinaste el numero secreto.\n");

    usados++; // se cuenta el intento ganador

    break;

} else if (valor < numeroSecreto) {

    printf("El numero secreto es MAYOR.\n");

} else {

    printf("El numero secreto es MENOR.\n");

}

usados++;

}

// -----

// RF05: Resumen detallado en tabla

// -----

imprimirResumen(min, max, numeroSecreto, intentos, usados, maxIntentos);

if (usados == maxIntentos && (usados == 0 || intentos[usados - 1].valor != numeroSecreto)) {

    printf("\nNo lograste adivinar el numero. ¡Sigue intentando!\n");

}

return 0;
```

```
}
```

```
// -----
```

```
// Implementaciones auxiliares
```

```
// -----
```

```
int leerEntero(const char *mensaje) {
```

```
    int v;
```

```
    for (;;) {
```

```
        printf("%s", mensaje);
```

```
        if (scanf("%d", &v) == 1) {
```

```
            return v;
```

```
        }
```

```
        // limpiar buffer
```

```
        int c; while ((c = getchar()) != '\n' && c != EOF) {}
```

```
        printf("Entrada no valida. Intente nuevamente.\n");
```

```
    }
```

```
}
```

```
int leerEnteroEnRango(const char *mensaje, int min, int max) {
```

```
    int v;
```

```
    for (;;) {
```

```
        printf("%s", mensaje);
```

```
        if (scanf("%d", &v) == 1 && v >= min && v <= max) {
```

```
            return v;
```

```
}

int c; while ((c = getchar()) != '\n' && c != EOF) {}

printf("ERROR: Debe estar entre %d y %d.\n", min, max);

}

}

int esDuplicado(const Intento intentos[], int usados, int valor) {

    for (int i = 0; i < usados; ++i) {

        if (intentos[i].valor == valor) {

            return 1;

        }

    }

    return 0;

}

void imprimirResumen(int min, int max, int secreto, const Intento intentos[], int usados, int
maxIntentos) {

    printf("\n===== \n");

    printf(" RESUMEN DETALLADO DEL JUEGO \n");

    printf("===== \n");

    printf("Rango elegido: [%d - %d] \n", min, max);

    printf("Numero secreto: %d \n", secreto);

    printf("Intentos permitidos: %d \n", maxIntentos);

    printf("Intentos realizados: %d \n", usados);

    printf("Intentos restantes: %d \n \n", maxIntentos - usados);
```

```
// Tabla

printf("-----\n");

printf("%-10s %-15s %-20s\n", "Intento", "Valor", "Resultado");

printf("-----\n");

for (int i = 0; i < usados; ++i) {

    printf("%-10d %-15d %-20s\n", i + 1, intentos[i].valor, intentos[i].resultado);

}

printf("-----\n");

}
```

<https://onlinegdb.com/MoHc24qDm>