

Corrección de la Prueba U3

1. DATOS INFORMATIVOS

Carrera: Ing. Electronica y Automatizacion

Asignatura: Fund. PRogramacion

Tema del taller: Corrección de Prueba

Docente: Jenny Ruiz

Nombre: Andrango Tobias,

Fecha: 19/11/2025 Paralelo: 29583

Codigo Corregido

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <stdbool.h>

int main() {
    srand(time(0));

    int min, max, maxIntentos, numeroSecreto;
    int matrizResultados[10][3];
    int intentosRealizados = 0;
    bool adivinado = false;

    printf("== Adivina el Numero Secreto ==\n\n");

    // --- Requisito 1: Definir rango (Validando > 0) ---
    do {
        printf("1. Ingresa el rango MINIMO : ");
        scanf("%d", &min);
        printf(" Ingresa el rango MAXIMO : ");
        scanf("%d", &max);

        if (min <= 0 || max <= 0) {
            printf("[Error] Solo se permiten numeros POSITIVOS y mayores que cero.\n");
        }

        else if (min >= max) {
            printf("[Error] El minimo debe ser menor que el maximo.\n");
        }
    } while (min <= 0 || max <= 0 || min >= max);

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

    // --- Requisito 2: Seleccionar intentos (Validando > 0 y <= 10) ---
    do {
        printf("2. Cuantos intentos deseas tener (1-10)? ");
        scanf("%d", &maxIntentos);

        if (maxIntentos <= 0) {
            printf("[Error] El numero de intentos debe ser mayor a 0.\n");
        }
    } while (maxIntentos <= 0);
```

```

} else if (maxIntentos > 10) {
    printf("[Error] El maximo de intentos permitidos es 10.\n");
}
} while (maxIntentos <= 0 || maxIntentos > 10);

printf("\n--- COMIENZA EL JUEGO ---\n");
printf("Adivina el numero entre %d y %d.\n", min, max);

for (int i = 0; i < maxIntentos; i++) {
    int intentoUsuario;
    bool entradaValida = false;

    // --- Requisito 3: Validación de rango, repetidos y ceros ---
    while (!entradaValida) {
        printf("\nIntento #%d: ", i + 1);
        scanf("%d", &intentoUsuario);

        if (intentoUsuario <= 0) {
            printf("[Error] No se permite el cero ni numeros negativos.\n");
            continue;
        }

        if (intentoUsuario < min || intentoUsuario > max) {
            printf("[Cuidado] El numero esta fuera del rango definido (%d-%d).\n", min, max);
            continue;
        }
    }

    bool repetido = false;
    for (int j = 0; j < i; j++) {
        if (matrizResultados[j][1] == intentoUsuario) {
            repetido = true;
            break;
        }
    }

    if (repetido) {
        printf("[Error] Ya intentaste ese numero antes. Prueba otro.\n");
    } else {
        entradaValida = true; // El número es válido
    }
}

// --- Requisito 4: Almacenar en Matriz ---
matrizResultados[i][0] = i + 1;
matrizResultados[i][1] = intentoUsuario;

int codigoResultado;

if (intentoUsuario < numeroSecreto) {
    codigoResultado = 0;
    printf("-> Demasiado BAJO.\n");
} else if (intentoUsuario > numeroSecreto) {
    codigoResultado = 1;
    printf("-> Demasiado ALTO.\n");
} else {
    codigoResultado = 2;
    printf("-> CORRECTO! Has ganado.\n");
    adivinado = true;
}

matrizResultados[i][2] = codigoResultado;
intentosRealizados++;

```

```

        if (adivinado) {
            break;
        }
    }

// --- Requisito 5: Mostrar tabla final ---
printf("\n\n=====\\n");
printf("      Tabla de Resultados \\n");
printf("=====\\n");
printf("%-10s %-10s %-15s\\n", "Intento", "Valor", "Resultado");
printf("-----\\n");

for (int i = 0; i < intentosRealizados; i++) {
    char descripcion[15];
    switch (matrizResultados[i][2]) {
        case 0: sprintf(descripcion, "Bajo"); break;
        case 1: sprintf(descripcion, "Alto"); break;
        case 2: sprintf(descripcion, "CORRECTO"); break;
    }
    printf("%-10d %-10d %-15s\\n", matrizResultados[i][0], matrizResultados[i][1], descripcion);
}
printf("=====\\n");

if (!adivinado) {
    printf("\\nLo siento, se acabaron los intentos.\\n");
    printf("El NUMERO SECRETO era: %d\\n", numeroSecreto);
} else {
    printf("\\nFelicidades por encontrar el numero!\\n");
}

return 0;
}

```