

(8)
Parte Práctica

PERÍODO: 2025 - 2026
NOMBRE: Francisco Vargas
CARRERA: Electrónica y Automatización

PARCIAL: 3
CURSO (NRC): 29583
FECHA: 21/01/2026

INDICACIONES GENERALES:

- La evaluación es personal y no se permite intercambiar información con sus compañeros, si es llamado la atención por una vez, se procederá a ponerle CERO y no recibirá su evaluación.
- La parte práctica debe subirlo en la carpeta de GitHub y formato indicado por el Sr. Docente (Apellidos Nombres_U2_Examen),
- Dispone de 2h.

Parte Práctica (20 Puntos)

Instrucciones.

1. A continuación, se presentan los requisitos funcionales (RF) para el desarrollo de su programa
2. Recuerde que en el código elaborado por Ud. Debe comentar donde desarrolla el RF solicitado, en este documento coloque el código y al final las capturas de pantalla que evidencien los RF.
3. Desde ya éxitos en su evaluación.
4. Este documento debe ser subido en formato PDF con Apellidos Nombres_U2_Examen, a la tarea al AULA VIRTUAL

1. REQUISITOS FUNCIONALES

Talla Datos

Eskumar en Pseudocódigo

RF01 – Configuración dinámica del rango: El programa debe permitir que el usuario defina el rango-mínimo y máximo del número secreto.

RF02 – Configuración dinámica de intentos: El programa debe permitir que el usuario seleccione el número máximo de intentos (hasta 10).

RF03 – Validación de rango y duplicados: El programa debe validar que cada intento esté dentro del rango y no se repita.

RF04 – Registro de intentos en matriz: El programa debe almacenar en una matriz el número de intento, el valor ingresado y el resultado (0 bajo, 1 alto, 2 correcto).

RF05 – Visualización de resumen detallado: Al terminar, el programa debe mostrar una tabla con cada intento y el resultado textual, y revelar el número secreto si no se adivinó.

2. RUBRICA DE CALIFICACIÓN

Criterio	4 puntos – Excelente	3 puntos – Bueno	2 puntos – Aceptable	1 punto – Deficiente	EVALUACION
Inicialización y generación del número aleatorio	Número aleatorio generado correctamente y una sola vez.	Se genera bien, pero fuera de lugar lógico.	Se generan varios números o el rango es incorrecto.	No se genera el número correctamente.	3
Ingreso y validación de datos	Captura todos los intentos y valida correctamente.	Captura intentos, pero tiene validaciones limitadas.	Captura intentos, pero con errores menores.	No captura correctamente los intentos.	3

Uso de la matriz para almacenar datos	Matriz implementada correctamente y se imprime al final.	Matriz implementada, pero con errores menores.	Uso parcial de la matriz.	No se usa la matriz adecuadamente.	3
Condiciones y control de flujo	Comparación exacta, con mensajes adecuados y flujo lógico.	Mensajes adecuados con mínimo error de flujo.	Comparación básica con mensajes genéricos.	Lógica confusa o incorrecta.	2
Mensajes finales y condición secreta	Mensaje secreto visible solo si acierta, resumen completo.	Muestra el resumen, pero con <u>fallos en el mensaje</u> .	Muestra el mensaje incluso cuando no acierta.	No se muestran resultados correctamente.	3
TOTAL, SOBRE 20 PUNTOS					15

Elaborado por: Ing Jenny A Ruiz R
 Docente TC DCCO
 Fecha: 21/01/2026

Universidad De Las Fuerzas Armadas

Nombre: Francisco Vargas

Fecha: 21/01/2026

NRC: 29583

Tabla De Datos

Variáble	Tipo De Dato	Descripción
Rng - Min	Entero	Límite mínimo definido por el usuario
Rng - Máx	Entero	Límite máximo definido por el usuario
Número Secreto	Entero	Número aleatorio generado dentro del rango
Intento	Entero	Número ingresado por el usuario
Contador de Intentos	Entero	Cuenta la cantidad de intentos realizados
Máx - Intentos	Entero	Límite de intentos permitidos (10)
Historial de Intentos	Matriz / Arreglo	Guarda cada intento realizado
Resultado Intento	Cadena	Indica si el intento fue mayor, menor o correcto
Adivina	Booleano	Indica si el usuario adivinó el número

Ejercicio en Pseudocódigo

1. Inicio
2. Define Variables: Rng - Min, Rng - Máx, Número Secreto, Intento, Contador de Intentos, Historial de Intentos, Resultado Intentos
3. Mostrar "Ingrese el Rng - Min"
4. Leer Rng - Min
5. Mostrar "Ingrese el Rng - Máx"

6. Leer Rng - Mín /
7. Generar un número aleatorio secreto entre el rango mínimo y máximo.
8. Mostrar "Ingrese el número Max - Intentos"
9. Leer Max - Intentos
10. Contador Intentos $\leftarrow 0$
- 11.
- 12.
- 13.
- 14.
- 15.
- 16.
- 17.
- 18.



UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE



Nombre:Francisco Vargas

NRC: 29583

Fecha:23/01/2026

Tema: Corrección de la prueba parcial

LINK DE GDB ONLINE: <https://onlinegdb.com/JGoAW-pV8>

CODIGO

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

int main() {
    int min, max;
    int maxIntentos;
    int numeroSecreto;
    int intento;

    // Matriz RF04: registro de intentos
    int intentos[10][1];

    // Nueva matriz para RF05: resultados textuales
    char resultados[10][30];

    int contadorIntentos = 0;

    // -----
    // RF01: Configuración dinámica del rango
    // -----
    printf("==== CONFIGURACION DEL RANGO ====\n");
    printf("Ingrese el minimo del rango: ");
    scanf("%d", &min);
    printf("Ingrese el maximo del rango: ");
    scanf("%d", &max);

    if (min >= max) {
        printf("ERROR: El minimo debe ser menor que el maximo.\n");
        return 1;
    }

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

    // -----
    // RF02: Configuración dinámica de intentos
    // -----
    printf("\n==== CONFIGURACION DE INTENTOS ====\n");
    printf("Seleccione el maximo de intentos (1 a 10): ");
    scanf("%d", &maxIntentos);

    if (maxIntentos < 1 || maxIntentos > 10) {
        printf("ERROR: Los intentos deben estar entre 1 y 10.\n");
        return 1;
    }
```

```

}

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

// -----
// Ciclo principal del juego
// -----
while (contadorIntentos < maxIntentos) {
    printf("\nIntento %d de %d. Ingrese un numero: ",
           contadorIntentos + 1, maxIntentos);
    scanf("%d", &intento);

    // RF03: Validar rango
    if (intento < min || intento > max) {
        printf("ERROR: Numero fuera del rango [%d - %d]\n", min, max);
        continue;
    }

    // RF03: Validar duplicados
    int duplicado = 0;
    for (int i = 0; i < contadorIntentos; i++) {
        if (intentos[i][0] == intento) {
            duplicado = 1;
            break;
        }
    }
    if (duplicado) {
        printf("ERROR: Ya ingresaste este numero antes.\n");
        continue;
    }

    // RF04: Registrar intento en matriz
    intentos[contadorIntentos][0] = intento;

    // Determinar resultado textual (para RF05)
    if (intento == numeroSecreto) {
        sprintf(resultados[contadorIntentos], "Correcto");
    } else if (intento < numeroSecreto) {
        sprintf(resultados[contadorIntentos], "Menor que el secreto");
    } else {
        sprintf(resultados[contadorIntentos], "Mayor que el secreto");
    }

    contadorIntentos++;

    // Mensajes del juego
    if (intento == numeroSecreto) {
        printf(";FELICIDADES! Adivinaste el numero secreto.\n");
        break;
    } else if (intento < numeroSecreto) {
        printf("El numero secreto es MAYOR.\n");
    } else {
        printf("El numero secreto es MENOR.\n");
    }
}

```

```

// -----
// RF05: TABLA DETALLADA DEL RESUMEN
// -----
printf("\n=====\\n");
printf("      RESUMEN DETALLADO DEL JUEGO\\n");
printf("=====\\n");

printf("Rango elegido: [%d - %d]\\n", min, max);
printf("Numero secreto: %d\\n", numeroSecreto);
printf("Intentos permitidos: %d\\n", maxIntentos);
printf("Intentos realizados: %d\\n", contadorIntentos);
printf("Intentos restantes: %d\\n\\n", maxIntentos - contadorIntentos);

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

for (int i = 0; i < contadorIntentos; i++) {
    printf("| %-10d | %-15d | %-20s |\\n",
          i + 1,
          intentos[i][0],
          resultados[i]);
}

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

if (contadorIntentos == maxIntentos && intentos[contadorIntentos - 1][0] != numeroSecreto)
{
    printf("\\nNo lograste adivinar el numero. ¡Sigue intentando!\\n");
}

return 0;
}

```