



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
EXAMEN DE: FUNDAMENTOS DE PROGRAMACIÓN
FORMA A PRACTICA

PERÍODO: 2025-2026

NOMBRE: JHOSUE ANGAMARCA

CARRERA: ELECTRONICA Y AUTOMATIZACION

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 recibirle su evaluación.
- La parte práctica debe subirle En la carpeta de GitHub y formato indicado por el Sr. Docente (Apellidos Nombres_U2_Examen).
- Dispone de 2h.

Parte Práctica (20 Puntos)

Instrucciones.

- A continuación, se presentan los requisitos funcionales (RF) para el desarrollo de su programa
- 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.
- Desde ya éxitos en su evaluación.
- Este documento debe ser subido en formato PDF con Apellidos Nombres_U2_Examen, a la tarea al AULA VIRTUAL

1. REQUISITOS FUNCIONALES

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.	3
Mensajes finales y condición secreta	Mensaje secreto visible solo si acierta, resumen completo.	Muestra el resumen, pero con fallos en el mensaje.	Muestra el mensaje incluso cuando no acierta.	No se muestran resultados correctamente.	4
TOTAL, SOBRE 20 PUNTOS					16

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

UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE

JOHOSUE ANGAMARCA

FUNDAMENTOS. PROGRA.

ING. JENNY RUIZ.

NRC: 29583

21/01/2026

TABLA DE DATOS:

REQUISITOS	Descripción	Parámetros
Configuración dinámica del rango	Permitir al usuario definir el rango de números	Maximo 10 elementos
Configuración de intentos y tiempo	configurar el tiempo de ejecución y el número máximo de intentos	Maximo 10 intentos
Validación de entrada	Validar que cada intento este dentro del rango y no sea duplicado	en el rango y no sea duplicado
Registro de intentos en la matriz	Almacenar cada intento en una matriz.	Al finalizar mostrar tabla de intentos y resultados. Si no se acierta mostrar el número secreto.

SEUDOCODIGO:

Inicio

Escribir "Minimo del rango"

Leer minimo

Escribir "Maximo del rango"

Leer maximo.

Si $\text{maximo} - \text{minimo} + 1 > 10$ entonces

escribir "El rango no puede tener mas de 10 números"

Fin Si

Escribir "Maximo de intentos hasta 10 números"

Leer max intentos

Si max intentos > 10 entonces

max intentos = 10

secreto = Número Aleatorio mínimo, máximo

tabla = lista vacia

intento = 1

acertado = Falso

Mientras intento \leq max - intentos y acertado = Falso

Escribir "Intento", intento, " Adivina"

Leer num

Si num $<$ minimo o num $>$ maximo entonces

Escribir "Fuera de rango"

continuar

Fin Si

duplicado Falso

Para cada fila en tabla hacer

si fila[i] = num Entonces

duplicado = verdadero

Fin Si

Si duplicado = Verdadero Entonces

Escribir "Número repetido"

continuar

Fin Si

Agregar intento, num, resultado, a tabla

intento = intento + 1

Fin Mientras

Escribir resultados intentos

Escribir "Intento (valor) Resultado

Para cada fila en tabla hacer

Escribir fila 0 fila 1 fila 2

Fin

si acerto = falso entonces

Escribir el número secreto era

Escribir "¡Felicidades!"

Fin si

Fin



UNIVERSIDAD DE LAS FUERZAS ARMADAS- ESPE



NOMBRE: JHOSUE ANGAMARCA

FECHA: 26/01/2026

NRC: 29583

TEMA: CORRECCIÓN

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

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

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

```
// RF01
```

```
typedef struct {
```

```
    int id;
```

```
    int valor;
```

```
    char resultado[15];
```

```
} Registro;
```

```
int main() {
```

```
    int min, max, secreto, num_usuario, max_intentos;
```

```
    int acertado = 0;
```

```
    int total_realizados = 0;
```

```
    Registro tabla[10];
```

```
    printf(" JUEGO ADIVINA EL NUMERO SECRETO \n");
```

```
// RF01
```

```
    printf("Introduce el rango ejemplo: 1 10: ");
```

```
    scanf("%d %d", &min, &max);
```

```
// RF02
```

```
    printf("¿Cuántos intentos quieres tener? (Maximo 10): ");
```

```
    scanf("%d", &max_intentos);
```

```

if (max_intentos > 10) {
    printf("Ajustando intentos a 10 (maximo permitido).\n");
    max_intentos = 10;
}

// RF03
secreto = (rand() % (max - min + 1)) + min;
printf("\nJuego listo. Tienes %d intentos para adivinar.\n", max_intentos);

for (int i = 0; i < max_intentos; i++) {
    printf("\nIntento %d/%d: ", i + 1, max_intentos);
    scanf("%d", &num_usuario);

    // RF04
    tabla[i].id = i + 1;
    tabla[i].valor = num_usuario;
    total_realizados++;

    if (num_usuario == secreto) {
        strcpy(tabla[i].resultado, "Correcto");
        printf("¡Felicidades! Ganaste.\n");
        acertado = 1;
        break;
    } else if (num_usuario < secreto) {
        strcpy(tabla[i].resultado, "Muy bajo");
        printf("Pista: El numero secreto es MAYOR.");
    } else {
        strcpy(tabla[i].resultado, "Muy alto");
        printf("Pista: El numero secreto es MENOR.");
    }
}
}

```

```

printf("\n\n=====");
printf("\n RESUMEN DE INTENTOS");
printf("\n\n=====");
printf("\nIntento | Valor | Resultado");
printf("\n-----");

for (int j = 0; j < total_realizados; j++) {
    printf("\n%-8d | %-7d | %s", tabla[j].id, tabla[j].valor, tabla[j].resultado);
}

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

if (!acertado) {
    printf("Se agotaron los intentos. El numero era: %d\\n", secreto);
}

return 0;
}

```