



**ESPE**  
UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN  
EXAMEN DE: FUNDAMENTOS DE PROGRAMACIÓN  
FORMA A TEORIA Y PRACTICA (SOLUCIONARIO)

PERÍODO: octubre 2025-febrero 2026  
NOMBRE: Ederson Gualoto  
CARRERA: electrónica y Automatización

PARCIAL: 3  
CURSO (NRC): 29583  
FECHA: 28-01-2026

#### 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 1h.

#### Parte Práctica (20 Puntos)

##### Instrucciones.

- A continuación, se presentan los requisitos funcionales (RF) para el desarrollo de su programa
- Elabore la Tabla de Objetos
- Elabore el Seudocódigo
- Utilice arreglos unidimensionales (vectores), solamente,

#### 1. REQUISITOS FUNCIONALES

RF01 – Generación del número secreto: El programa debe generar un número entero aleatorio entre 1 y 100 al iniciar el juego.

RF02 – Ingreso de intentos: El programa debe permitir que el usuario ingrese hasta 5 intentos para adivinar el número.

RF03 – Registro de intentos: El programa debe almacenar cada número ingresado por el usuario en un vector.

RF04 – Comparación del intento: El programa debe comparar el número ingresado con el número secreto e indicar si es: Correcto, Demasiado bajo, Demasiado alto.

RF05 – Visualización de resultados: Al finalizar el juego, el sistema debe mostrar los números ingresados por el usuario y el número secreto.

#### 2. Tabla de Objetos

Nombre del Objeto	Tipo de Dato	Descripción
numeros	Numero	Números que utilizaremos ✓
numeroSecreto	Aleatorio	El número aleatorio secreto
intento	caracter ✓ 10	Numero de intentos ✓
i	numero ✓	Fila ✗
maxIntentos	caracter	intentos max. juego ✓

3. Seudocódigo  
Inicio

Fin

4. <u>Rùbrica</u>					
Criterio	4 puntos – Excelente	3 puntos – Bueno	2 puntos – Aceptable	1 punto – Deficiente	CALIFICACION/20
Inicialización y generación del número	El número aleatorio se genera 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.	1
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 del vector para almacenar datos	Vector implementado correctamente y se imprime al final.	Vector implementado, pero con errores menores.	Uso parcial del vector (no almacena todos).	No se usa el vector adecuadamente.	1
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 fallos en el mensaje.	Muestra el mensaje incluso cuando no acierta.	No se muestran resultados correctamente.	1

Elaborado por: Ing Jenny A Ruiz R  
Docente TC DCCO  
Fecha:28/01/2025



Inicio  
// Datos de

Definir  $i, j$  como 1; // 1 a 100

Alcatario ("numero de 1 a 100");

Leer numsecreto;

// Intentos  $\rightarrow 5$

escribir ("numero " %d, 100);

Leer  $x$ ; secreto;

Si numsecreto =  $x$  entonces;

escribir ("encontraste el numero");

Sino

escribir ("no es el numero");

Fin si

Intentos 5.

// Registro de intentos

max intentos  $\rightarrow$  intentos 5

vector ( $i \leq$  intentos; intentos =  $x$ ;  $x =$  numsecreto;  $i++$ );

max intentos  $\rightarrow$  intentos; numsecreto =  $x$ ;  $i++$ );

// Visualización

Escribir ("---- Registro ----");

Escribir ("Intentos | numsecreto |  $x$ ");

Tablar ("Intentos | numsecreto |  $x$ ");

Imprimir ("Tabla");

Fin

Fin



## Corrección

### Inicio

// Definir valores

Definir num-secreto, max-Intentos, i, num-Usuario, loAdivino  
como entero;

Definir vecnumUsuario [max-Intentos];

max-Intentos  $\rightarrow$  5

loAdivino = 0

num-secreto  $\Rightarrow$  numero Aleatorio Entre (1, 100);

// Mensaje

Escribir "Juego de adivinanza";

Escribir "Numeros enteros de 1 a 100 | max Intentos:"  
max-Intentos

Escribir "Se guardaran tus numeros"

### // Bucle

Para i desde 0 hasta max-Intentos - 1 Hacer:

Escribir "Intento", (i+1), " de ", max-Intentos, ": Ingresa  
numero entero: ";

Leer num-Usuario

// Valido de entero y rango

Mientras num-Usuario  $<$  1 o num-Usuario  $>$  100 Hacer

Escribir "Solo numeros enteros 1-100:";

Fin mientras

vecnumUsuario [i] = num-Usuario

Si num-Usuario == num-secreto

Entonces



Escribir "Adivinaste en el intento",  $(i+1)$ , "!"

loAdivino = 1

Salir del para

SINO SI num\_Usuario < num\_secreto Entonces

Escribir "El numero es grande"

SINO

Escribir "El numero es menor"

Fin si

Fin para

// Mensaje final

SI loAdivino == 0 Entonces

Escribir "Se acabaron los ", max-Intentos, "  
Intentos"

Escribir "El numero secreto era: " num\_secreto

Fin SI

Escribir "los numeros que tus Intentos Fueron  
(del vector):"

Para i desde 0 hasta max-Intentos - 1 Hacer

SI  $i \leq (\text{posicionUltimoIntento})$  Entonces

Escribir "Posición", i, "del vector:", vecnum\_Usuario  
[i]

SINO

Escribir "Posición", i, "del vector: Vacía"

Fin SI

Fin Para

Escribir "Maximo de intentos: ", max-Intentos

Fin

<https://onlinegdb.com/5dUCjKYdLQ>