DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
EXAMEN DE: FUNDAMENTOS DE PROGRAMACIÒN
UNIDAD 2 TEMA VECTORES CON Y SIN ESTRUCTURAS REPETITIVAS

PERÍODO: Abril 2025 – Agosto 2025
NOMBRE: Alvear Alexander
CARRERA: Electronica y automatización
PARCIAL: 2° Parcial
CURSO (NRC): 20823
FECHA: 27/06/2025

Nivel 1: Adivina el número con ciclo for (sin vector)

Objetivo: Aplicar un ciclo 'for' para realizar varios intentos con lógica condicional.

Requisitos funcionales – Nivel 1

/*

Requisitos funcionales – Nivel 1 SIN VECTORES

- RF2.1 – debe generar un número aleatorio entre 1 y 100.
- RF2.2 – debe permitir hasta 5 intentos mediante un ciclo 'for'.
- RF2.3 – debe indicar si el intento es correcto, bajo o alto.
- RF2.4 – Al final del juego, debe mostrar todos los intentos realizados.
- RF2.5 – Mostrar un mensaje secreto si el jugador adivina el número correctamente

Código Nivel 1

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

#define MAX_INTENTOS 5

int main() {
   int numeroSecreto, intento;
   int intentos[MAX_INTENTOS];
   srand(time(NULL));

   numeroSecreto = rand() % 100 + 1; //Debe generar un número aleatorio entre 1 y 100

   printf("Adivina el numero secreto entre 1 y 100.\n");

   for (int i = 0; i < MAX_INTENTOS; i++) { //Debe permitir hasta 5 intentos mediante un ciclo 'for'
      printf("Intento %d: ", i + 1);
      scanf("%d", &intento);
      intentos[i] = intento;

      //Debe indicar si el intento es correcto, bajo o alto
```

```
        if (intento == numeroSecreto) {
            printf("¡Felicidades! Adivinaste el numero secreto :D\n");
            printf("Mensaje secreto: Eres un genio :v\n"); //Mostrar un mensaje secreto si el jugador
adivina el número correctamente
            break;
        } else if (intento < numeroSecreto) {
            printf("El numero es mas alto.\n");
        } else {
            printf("El numero es mas bajo.\n");
        }
    }

    //Al final del juego, debe mostrar todos los intentos realizados
    printf("\nIntentos realizados:\n");
    for (int i = 0; i < MAX_INTENTOS; i++) {
        printf("%d: %d\n", i + 1, intentos[i]);
    }

    if (intento != numeroSecreto) {
        printf("\nLo siento, no adivinaste el numero :C Era: %d\n", numeroSecreto);
    }

    return 0;
}
```

Nivel 2: Adivina el número con ciclo for y vector

Objetivo: Usar un vector para almacenar los intentos realizados en el ciclo.

Requisitos funcionales – Nivel 2

- RF2.1 – El sistema debe generar un número aleatorio entre 1 y 100.
- RF2.2 – El sistema debe permitir hasta 5 intentos mediante un ciclo 'for'.
- RF2.3 – El sistema debe almacenar cada intento en un vector.
- RF2.4 – El sistema debe indicar si el intento es correcto, bajo o alto.
- RF2.5 – Al final del juego, debe mostrar todos los intentos realizados.
- RF2.6 – Mostrar un mensaje secreto si el jugador adivina el número correctamente.

Código Nivel 2

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

int main() {
    int numsec; // RF2.1: Número secreto
    int intento;
    int gano = 0;
    int intentos[5]; // RF2.3: Vector para almacenar los intentos
```

```
    srand(time(NULL));
    numsec = (rand() % 100) + 1; // RF2.1: Número aleatorio entre 1 y 100

    // RF2.2: Permitir hasta 5 intentos usando un ciclo for
    for (int i = 0; i < 5; i++) {
        printf("Intento %d: Ingresa un número entre 1 y 100: ", i + 1);
        scanf("%d", &intento);
        intentos[i] = intento; // RF2.3: Guardar intento en vector

        // RF2.4: Evaluar si es correcto, bajo o alto
        if (intento == numsec) {
            printf("¡Adivinaste el número!\n");
            printf("Mensaje secreto: 123456789\n"); // RF2.6
            gano = 1;
            break;
        } else if (intento < numsec) {
            printf("Muy bajo.\n");
        } else {
            printf("Muy alto.\n");
        }
    }

    // RF2.5: Mostrar todos los intentos realizados
    printf("\nIntentos realizados:\n");
    for (int i = 0; i < 5; i++) {
        if (intentos[i] != 0) {
            printf("Intento %d: %d\n", i + 1, intentos[i]);
        }
    }

    if (!gano) {
        printf("No lograste adivinar el número. El número secreto era: %d\n", numsec);
    }

    return 0;
}
```

Rúbrica de Evaluación

Criterio	4 pts – Excelente	3 pts – Bueno	2 pts – Aceptable	1 pt – Deficiente	EVAL
Captura de datos	Lee correctamente los valores ingresados.	Lee datos pero con errores menores.	Errores en la captura de datos.	No se realiza lectura o es incorrecta.	

Uso de condicionales	Condicionales anidados bien estructurados y funcionales.	Uso correcto con ligeros errores.	Uso parcial de condicionales.	No se aplican correctamente.	
Mensajes adecuados	Mensajes claros para cada caso (alto, bajo, correcto).	Mensajes claros con mínimos errores.	Mensajes confusos o repetitivos.	No se muestran o son incorrectos.	
Lógica de los intentos	Evalúa correctamente hasta cinco intentos.	Evalúa dos intentos correctamente.	Evalúa uno solo correctamente.	Lógica incompleta o confusa.	
Identificación de acierto	Reconoce el número correcto en cualquier intento.	Reconoce el acierto parcialmente.	Reconocimiento limitado del acierto.	No reconoce cuando se acierta.	
CALIFICACION /20 PTOS					