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 **PARCIAL: 2 do**
NOMBRES: Brayan Miguitama, Jordy Cevallos, Ana Ulloa **CURSO (NRC): 20823**
CARRERA: Electronica y automatizacion **FECHA:** 27/6/2025

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>



#define MAX_INTENTOS 5



int main() {

  // RF2.1 - Generar un numero aleatorio entre 1 y 100

  int numeroSecreto = rand() % 100 + 1; // numero aleatorio entre 1 y 100



  int intentoUsuario;

  int intentosRealizados = 0;
```

```
int adivinado = 0;

printf("Adivina el numero entre 1 y 100. Tienes %d intentos.\n", MAX_INTENTOS);

// RF2.2 - Permitir hasta 5 intentos usando un ciclo for
for (int i = 1; i <= MAX_INTENTOS; ) {
   printf("Intento %d: ", i);
   scanf("%d", &intentoUsuario);

   // Validar que el numero este en el rango permitido (1-100)
   if (intentoUsuario < 1 || intentoUsuario > 100) {
      printf("Por favor ingresa un numero entre 1 y 100.\n");
      continue; // no cuenta este intento
   }

   // RF2.4 - Mostrar cada intento del usuario durante el juego
   printf("Ingresaste: %d\n", intentoUsuario);

   intentosRealizados = i;

   // RF2.3 - Indicar si el intento es correcto, bajo o alto
   if (intentoUsuario == numeroSecreto) {
      // RF2.5 - Mostrar mensaje secreto si acierta
      printf("Correcto! Has adivinado el numero.\n");
      printf("Mensaje secreto: Eres un genio!\n");
```

```
            printf("Felicidades, eres un ganador! Sigue asi!\n");
            adivinado = 1;
            break;
        } else if (intentoUsuario < numeroSecreto) {
            printf("Muy bajo.\n");
        } else {
            printf("Muy alto.\n");
        }

        i++; // solo se incrementa si el intento fue valido
    }

    // RF2.4 - Mostrar cantidad de intentos usados al final
    printf("Intentos realizados: %d\n", intentosRealizados);

    // RF2.5 - Si no adivina, mostrar mensaje motivador
    if (!adivinado) {
        printf("No adivinaste el numero, era %d.\n", numeroSecreto);
        printf("SIGUE PARTICIPANDO LA PROXIMA TE IRA MEJOR ;).\n");
    }

    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

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	Reconoce el acierto parcialmente.	Reconocimiento limitado del acierto.	No reconoce cuando se acierta.	

	intento.				
CALIFICACION /20 PTOS					

Carpeta Unidad 2

Apellidos_Nombres_EvalN1.c