

REPETITIVAS

PERÍODO: Abril 2025 – Agosto 2025
NOMBRE: Mateo Velecela
CARRERA: Electronica y automatizacion

PARCIAL: 2do 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>
```

```
int main(){
    int numsec;
    int intento;
    int gano=0;
    int intento1=0, intento2=0, intento3=0, intento4=0, intento5=0;
    srand(time(NULL)); //R.F.1
    numsec=(rand()%100)+1; //99+1
    for(int i=1; i<=5; i++){
        printf("intento %d:ingresa un numero entre el 1 y el 100: ",i); //R.F.2
        scanf("%d",&intento);
        if(i==1)intento1=intento;
        else if(i==2)intento2=intento;
        else if(i==3)intento3=intento;
        else if(i==4)intento4=intento;
        else if(i==5)intento5=intento;
        if(intento==numsec){
            printf("Adivinaste el numero\n");
            printf("mensaje secreto: 123456789\n");
            gano=1;
        }
    }
}
```

```

        break;
    } else if(intento < numsec){
        printf("muy bajo \n");
    } else {
        printf("muy alto\n "); //R.F.3
    }
}
printf("intentos realizados: \n");
if (intento1) printf("1:%d\n", intento1);
if (intento2) printf("2:%d\n", intento2);
if (intento3) printf("3:%d\n", intento3);
if (intento4) printf("4:%d\n", intento4);
if (intento5) printf("5:%d\n", intento5); //R.F.5

if(!gano){
    printf("No lograste adivinar el numero, el numero era: %d\n", numsec); //R.F.6
}
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: Ingrese 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					

Carpeta Unidad 2

Apellidos_Nombres_EvalN1.c