



Reporte Técnico de Actividades Práctico-Experimentales Nro. 002

1. Datos de Identificación del Estudiante y la Práctica

Nombre del estudiante	José Daniel Maldonado Rodriguez
Asignatura	Teoría de la programación
Ciclo	1A
Unidad	2
Resultado de aprendizaje de la unidad	Aplica las estructuras de programación en la resolución de problemas básicos, bajo los principios de solidaridad, transparencia, responsabilidad y honestidad
Práctica Nro.	002
Tipo	Individual
Título de la Práctica	Aplicación de estructuras repetitivas en la resolución de problemas.
Nombre del Docente	Lissette Geoconda López Faicán
Fecha	Jueves 27 de noviembre del 2025 Jueves 04 de diciembre del 2025
Horario	10h30 – 13h30
Lugar	Aula física asignada al paralelo.
Tiempo planificado en el Sílabo	6 horas

2. Objetivo(s) de la Práctica

- Comprender y aplicar las estructuras repetitivas en la resolución de problemas.
- Diseñar y codificar un algoritmo que utilice bucles para resolver un problema de tipo iterativo.
- Validar el funcionamiento del programa mediante la ejecución práctica.

3. Materiales, Reactivos, Equipos y Herramientas

- Herramientas de modelado de diagrama de flujo (Psient).
- IDE de programación: Visual Studio Code.
- Lenguaje de programación: C (según los contenidos de la unidad).



4. Procedimiento / Metodología Ejecutada

Metodología de aprendizaje: Aprendizaje basado en problemas.

- **Objetivo:** Aplicación de estructuras repetitivas para la resolución de problemas.
- Las estructuras repetitivas, son fundamentales porque permiten ejecutar un conjunto de instrucciones varias veces sin tener que escribirlas repetidamente.
- **Contextualización del problema:** Basado en el ejercicio del “Cálculo de la nota final de la Unidad 1 mediante estructuras secuenciales en C”, se desea automatizar el proceso de cálculo para varios estudiantes utilizando estructuras repetitivas.
- El programa debe permitir ingresar la cantidad total de estudiantes, y mediante un bucle, repetir el proceso de lectura de calificaciones y cálculo de la nota final.
- En cada repetición, el programa solicitará los valores de los componentes (ACD, APE, AA y ES), calculará la nota final y mostrará el resultado antes de pasar al siguiente estudiante.
- Además, el programa debe validar que las notas ingresadas estén dentro del rango permitido (0 a 10). Si el usuario ingresa una nota fuera de este rango, el programa mostrará un mensaje de error y volverá a solicitar el dato hasta que sea correcto.
- No se requiere guardar las notas; el programa únicamente procesará y mostrará el resultado individual en cada iteración.

- **Desarrollo:**

Antes de empezar el código del programa se ha realizado un diagrama de flujo para entender el proceso y el funcionamiento de nuestro programa.

En el programa se solicita al usuario ingresar el número de estudiantes para calcular el promedio de sus notas. Mediante un bucle for, se repite el proceso de lectura de calificaciones y el cálculo de la nota final para cada estudiante, siguiendo la misma lógica previamente implementada en el programa “Cálculo de la nota final de la Unidad 1 mediante estructuras secuenciales en C”.

En cada iteración, el programa pide los valores de los componentes ACD, APE, AA y ES, calcula la nota final y muestra el resultado antes de continuar con el siguiente estudiante. Para cada componente se emplea un bucle do/while con el fin de validar que las notas ingresadas estén dentro del rango permitido (0 a 10). En caso de que el usuario introduzca un valor fuera de ese rango, el programa muestra el mensaje “La nota es inválida, escriba nuevamente” y repite la solicitud hasta obtener un dato correcto.

Una vez calculadas las notas finales de todos los estudiantes, el programa determina el promedio general del curso y lo imprime en pantalla con el mensaje: “EL PROMEDIO DEL AULA ES DE: (promedio)”.

5. Resultados



UNL

Universidad
Nacional
de Loja

1859

FEIRNNR - Carrera de Computación

• Código en C:

```
C BuclePromedio.c X
C BuclePromedio.c > main()
1  #include <stdio.h>
2
3  int main(){
4
5      //Definir las variables para cada actividad dentro de los ponderados ACD,AA, APE y ES
6      float ACD1, ACD2, AA1, AA2, APE1, APE2, ES1, ES2, ACD, AA, APE ,ES, NF, promedio, suma;
7      int estudiantes, c;
8      suma = 0;
9      //Los datos de entrada ingresados serán las calificaciones que tuvo cada actividad en su respectivo ponderado
10
11     printf("de cuantos estudiantes deseas calcular el promedio\n");
12     scanf("%i", &estudiantes);
13
14     for(c = estudiantes; c > 0; c--){
15         //Ponderado de los ACD(Aproximación con el docente)
16         printf("NOTA DEL ESTUDIANTE %i\n", c);
17         do{
18             printf("Ingrese la nota que obtuvo en su ACD1\n");
19             scanf("%f", &ACD1);
20             if(ACD1 > 10 || ACD1 < 0){
21                 printf("la nota es invalida, escriba nuevamente\n");
22                 printf("\n");
23             }
24         }while(ACD1 > 10 || ACD1 < 0);
25
26         do{
27             printf("Ingrese la nota que obtuvo en su ACD2\n");
28             scanf("%f", &ACD2);
29             if(ACD2 > 10 || ACD2 < 0){
30                 printf("la nota es invalida, escriba nuevamente\n");
31                 printf("\n");
32             }
33         }while(ACD2 > 10 || ACD2 < 0);
34
35         //Ponderado de los AA (Actividades autónomas)
36         do{
37             printf("Escriba la nota que tiene en su AA1\n");
38             scanf("%f", &AA1);
39             if(AA1 > 10 || AA1 < 0){
40                 printf("la nota es invalida, escriba nuevamente\n");
41                 printf("\n");
42             }
43         }while(AA1 > 10 || AA1 < 0);
```



UNL

Universidad
Nacional
de Loja

1859

```
44      do{
45          printf("Escriba la nota que tiene en su AA2\n");
46          scanf("%f", &AA2);
47          if(AA2 > 10 || AA2 < 0){
48              printf("la nota es invalida, escriba nuevamente\n");
49              printf("\n");
50          }
51      }while(AA2 > 10 || AA2 < 0);
52
53
54 //Ponderado de los APE (Actividades práctico experimentales)
55     do{
56         printf("Ingrese la nota que tiene en su APE1\n");
57         scanf("%f", &APE1);
58         if(APE1 > 10 || APE1 < 0){
59             printf("la nota es invalida, escriba nuevamente\n");
60             printf("\n");
61         }
62     }while(APE1 > 10 || APE1 < 0);
63
64     do{
65         printf("ingrese la nota que tiene en su APE2\n");
66         scanf("%f", &APE2);
67         if(APE2 > 10 || APE2 < 0){
68             printf("la nota es invalida, escriba nuevamente\n");
69             printf("\n");
70         }
71     }while(APE2 > 10 || APE2 < 0);
72
73 //Ponderado de ES (Evaluación sumativa)
74     do{
75         printf("ingrese la nota que tuvo en su ES1\n");
76         scanf("%f", &ES1);
77         if(ES1 > 10 || ES1 < 0){
78             printf("la nota es invalida, escriba nuevamente\n");
79             printf("\n");
80         }
81     }while(ES1 > 10 || ES1 < 0);
82
83     do{
84         printf("ingrese la nota que tuvo en su ES2\n");
85         scanf("%f", &ES2);
86         if(ES2 > 10 || ES2 < 0){
87             printf("la nota es invalida, escriba nuevamente\n");
88             printf("\n");
89         }
90     }while(ES2 > 10 || ES2 < 0);
```



UNL

Universidad
Nacional
de Loja

1859

FEIRNNR - Carrera de Computación

```
ACD = (((ACD1 + ACD2) / 2) * 0.2);
AA = (((AA1 + AA2) / 2) * 0.2);
APE = (((APE1 + APE2) / 2) * 0.25);
ES1 = ES1 * 0.4;
ES2 = ES2 * 0.6;
ES = (((ES1 + ES2 )) * 0.35);
NF = ACD + AA + APE + ES;

//Condicionales para que salga la reseña acerca de la nota final de su promedio

if(NF >= 9 ){
    printf("Su nota es %f excelente :D\n", NF);
}else if(NF >= 7 && NF < 9){
    printf("Su nota es %f buena c:\n", NF);
}else if(NF >= 5 && NF <7){
    printf(" Su nota es %f regular :)\n", NF);
}else if(NF < 5){
    printf(" Su nota es %f deficiente :c\n", NF);
}
suma = suma + NF;

//Datos de salida
//Aquí mostramos los resultados de los promedios de cada ponderado y la nota final de Unidad sumando

printf("El promedio del ponderado ACD es %f\n", ACD);
printf("El promedio del ponderado AA es %f\n", AA);
printf("El promedio del ponderado APE es %f\n", APE);
printf("El promedio del ponderado ES es %f\n", ES);
printf("Su nota final de Unidad es %f\n", NF);
printf("\n");

promedio = suma / estudiantes;
printf("EL PROMEDIO DEL AULA ES DE: %.2f", promedio);
return 0;
}
```

- **Pruebas en la terminal:**



UNL

Universidad
Nacional
de Loja

1859

FEIRNNR - Carrera de Computación

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
● PS C:\Users\josed\OneDrive\Documentos\UNL\TEORIA DE PROGRAMACIÓN\EJERCICIOS CON C> ./BuclePromedio.exe
de cuantos estudiantes deseas calcular el promedio
3
NOTA DEL ESTUDIANTE 3
Ingrese la nota que obtuvo en su ACD1
10
Ingrese la nota que obtuvo en su ACD2
10
Escriba la nota que tiene en su AA1
10
Escriba la nota que tiene en su AA2
10
Ingrese la nota que tiene en su APE1
9.5
ingrese la nota que tiene en su APE2
8
ingrese la nota que tuvo en su ES1
10
ingrese la nota que tuvo en su ES2
9.75
Su nota es 9.635000 excelente :D
El promedio del ponderado ACD es 2.000000
El promedio del ponderado AA es 2.000000
El promedio del ponderado APE es 2.187500
El promedio del ponderado ES es 3.447500
Su nota final de Unidad es 9.635000

NOTA DEL ESTUDIANTE 2
Ingrese la nota que obtuvo en su ACD1
8
Ingrese la nota que obtuvo en su ACD2
8
Escriba la nota que tiene en su AA1
9.5
Escriba la nota que tiene en su AA2
8.5
Ingrese la nota que tiene en su APE1
9
ingrese la nota que tiene en su APE2
9
ingrese la nota que tuvo en su ES1
9
ingrese la nota que tuvo en su ES2
8.5
```



UNL

Universidad
Nacional
de Loja

1859

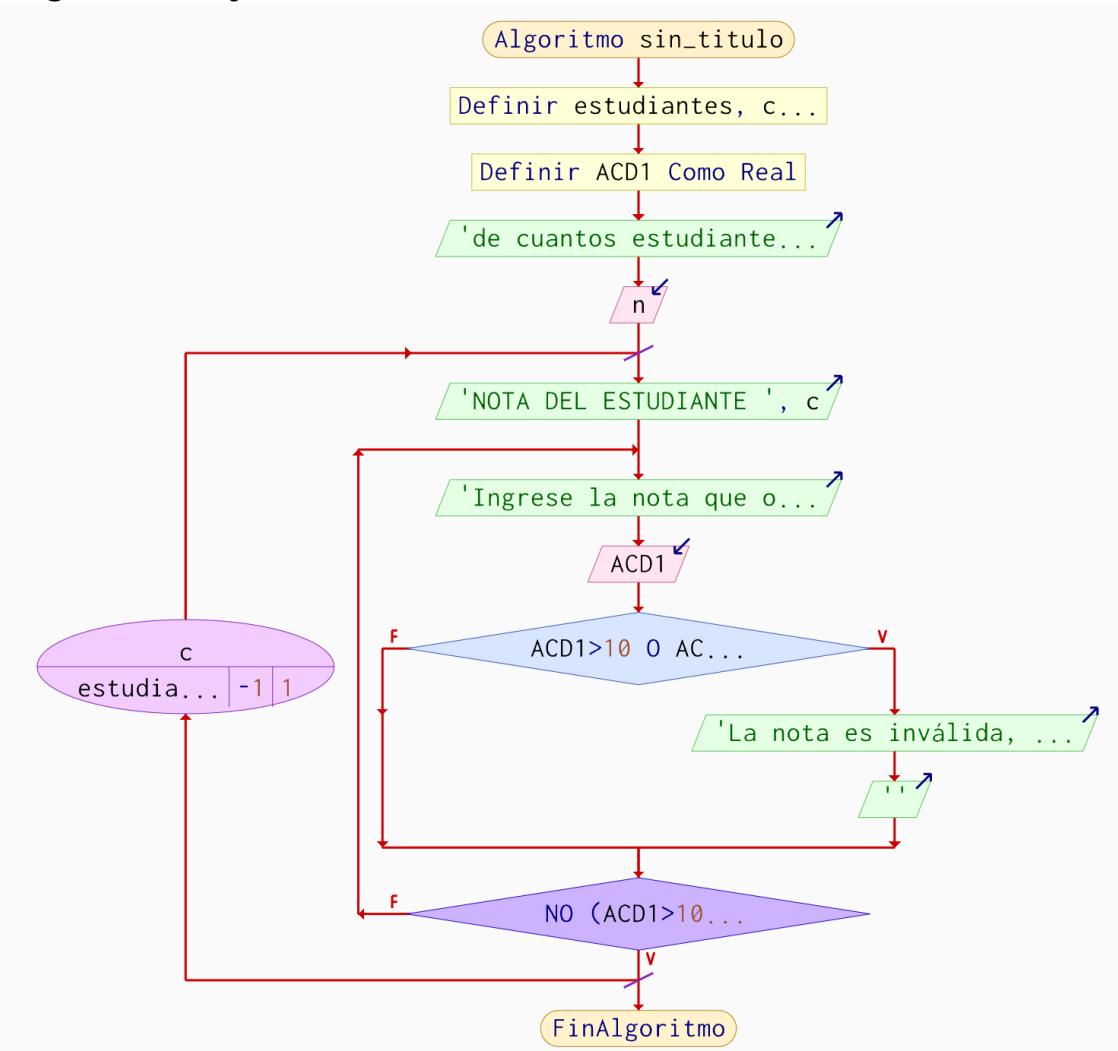
FEIRNNR - Carrera de Computación

```
PS C:\Users\josed\OneDrive\Documentos\UNL\TEORIA DE PROGRAMACIÓN\EJERCICIOS CON C> ./BuclePromedio.exe
Su nota es 8.695000 buena c:
El promedio del ponderado ACD es 1.600000
El promedio del ponderado AA es 1.800000
El promedio del ponderado APE es 2.250000
El promedio del ponderado ES es 3.045000
Su nota final de Unidad es 8.695000

NOTA DEL ESTUDIANTE 1
Ingrese la nota que obtuvo en su ACD1
10
Ingrese la nota que obtuvo en su ACD2
10
Escriba la nota que tiene en su AA1
9.5
Escriba la nota que tiene en su AA2
9.25
Ingrese la nota que tiene en su APE1
9.5
ingrese la nota que tiene en su APE2
9
ingrese la nota que tuvo en su ES1
9
ingrese la nota que tuvo en su ES2
8
Su nota es 9.127501 excelente :D
El promedio del ponderado ACD es 2.000000
El promedio del ponderado AA es 1.875000
El promedio del ponderado APE es 2.312500
El promedio del ponderado ES es 2.940000
Su nota final de Unidad es 9.127501

EL PROMEDIO DEL AULA ES DE: 9.15
```

Diagrama de flujo:



6. Preguntas de Control

- ¿En qué se diferencia una estructura repetitiva de una condicional?

Una estructura condicional permite tomar decisiones dentro de un programa, ejecutando un bloque de instrucciones únicamente si se cumple una condición específica. En cambio, una estructura repetitiva está diseñada para repetir un conjunto de instrucciones varias veces, ya sea mientras se cumpla una condición o durante un número determinado de iteraciones.

- ¿Qué diferencia existe entre las estructuras for, while y do...while en cuanto a su funcionamiento y uso?

for:

Se utiliza cuando se conoce de antemano cuántas veces debe repetirse un proceso. Incluye inicialización, condición e incremento en una sola línea, lo que facilita el control de ciclos contados o de iteraciones definidas.



while:

Ejecuta un bloque de instrucciones mientras una condición sea verdadera. Es ideal cuando no se conoce cuántas repeticiones serán necesarias y el número de iteraciones depende directamente de una condición que puede cambiar en tiempo de ejecución.

Do/while:

A diferencia de un bucle while, el bloque se ejecuta al menos una vez, y la condición se evalúa al final. Esto lo hace útil cuando necesitamos que el usuario ingrese un dato o que se ejecute una acción antes de validar la condición.

- **¿Por qué es importante incluir validaciones dentro de un programa cuando se solicitan datos al usuario?**

Porque así se asegura que la información ingresada por el usuario sea correcta y coherente con lo que el programa necesita. Las validaciones permiten evitar errores, prevenir resultados incorrectos, proteger la integridad del sistema y garantizar que el programa funcione de manera estable. Además, guían al usuario cuando ingresa valores fuera del rango permitido, asegurando que los datos finales sean confiables y adecuados para los cálculos o procesos que realizará el programa.

7. Conclusiones

La práctica permitió comprender y aplicar las estructuras repetitivas en la resolución de problemas de programación. A través del desarrollo del algoritmo, se evidenció cómo los bucles facilitan la automatización de procesos, especialmente cuando se requiere ejecutar múltiples veces una misma operación, como el cálculo de notas para varios estudiantes. Asimismo, la implementación de validaciones mediante estructuras do/while reforzó la importancia de controlar los datos ingresados por el usuario, garantizando resultados correctos y evitando errores durante la ejecución. Finalmente, la práctica cumplió el objetivo planteado al demostrar que el uso adecuado de estructuras repetitivas permite construir programas más eficientes y adaptados a situaciones reales.

8. Recomendaciones

Se recomienda seguir implementando validaciones en futuros programas para asegurar que los datos ingresados por el usuario sean precisos y evitar fallos en la ejecución. También es aconsejable realizar más ejercicios que combinen bucles, condicionales y cálculos, con el fin de fortalecer la lógica de programación y mejorar la capacidad de resolver problemas de mayor complejidad. Para optimizar la práctica en contextos reales. Finalmente, se sugiere probar el programa con diferentes tipos de entradas para verificar su estabilidad y asegurar que responda correctamente ante errores o valores inesperados.

Declaración del Uso de IA:

Para la elaboración de este informe se utilizó apoyo de herramientas de inteligencia artificial generativa (ChatGPT) únicamente con fines de redacción, mejora de la claridad textual y organización de ideas. La herramienta no



UNL

Universidad
Nacional
de Loja

1859

FEIRNNR - Carrera de Computación

reemplazó el razonamiento ni la ejecución de la actividad, sino que se empleó como un recurso complementario para optimizar la presentación del informe.