



Reporte Técnico de Actividades Práctico-Experimentales Nro. 002

1. Datos de Identificación del Estudiante y la Práctica

Nombre del estudiante(s)	Valeria Idaly Agila Gomez
Asignatura	Teoría de la programación
Ciclo	1 A
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

- Herramientas de modelado de diagrama de flujo (Psient, Draw.io, Lucidchart, otros)
- IDE de programación: Visual Studio Code u otro entorno compatible.
- Lenguaje de programación: C (según los contenidos de la unidad).

4. Equipos y Herramientas

- Computador personal con sistema operativo Windows, Linux o macOS.
- Material de apoyo en el Entorno Virtual de Aprendizaje (EVA).



- Editores de texto (Word, Google Docs u otros) para la elaboración del informe técnico en formato PDF.
- Conexión a internet estable para acceder a recursos digitales y software en línea.
- Aula física asignada al paralelo.

5. Procedimiento / Metodología Ejecutada

Como primer paso, realicé un diagrama de flujo, en el cual representé de forma gráfica la lógica del programa, el orden en que se ejecutan las instrucciones y el proceso repetitivo necesario para calcular la nota final de cada estudiante. Este diagrama permitió organizar claramente las entradas, validaciones y cálculos del sistema.

Luego, como segundo paso, implementé esa lógica en el lenguaje de programación C, desarrollando el código que incluye un ciclo for para procesar a varios estudiantes y estructuras do...while para validar que cada nota ingresada se encuentre dentro del rango permitido. También programé los cálculos de los ponderados ACD, AA, APE y ES, así como la nota final y su clasificación.

Finalmente, como tercer paso, compilé y ejecuté el programa para comprobar su funcionamiento, detectar posibles errores y asegurar que los resultados obtenidos fueran correctos. Realicé varias pruebas ingresando diferentes notas para verificar la validez de los datos, el buen desempeño del ciclo repetitivo y el cálculo del promedio general.

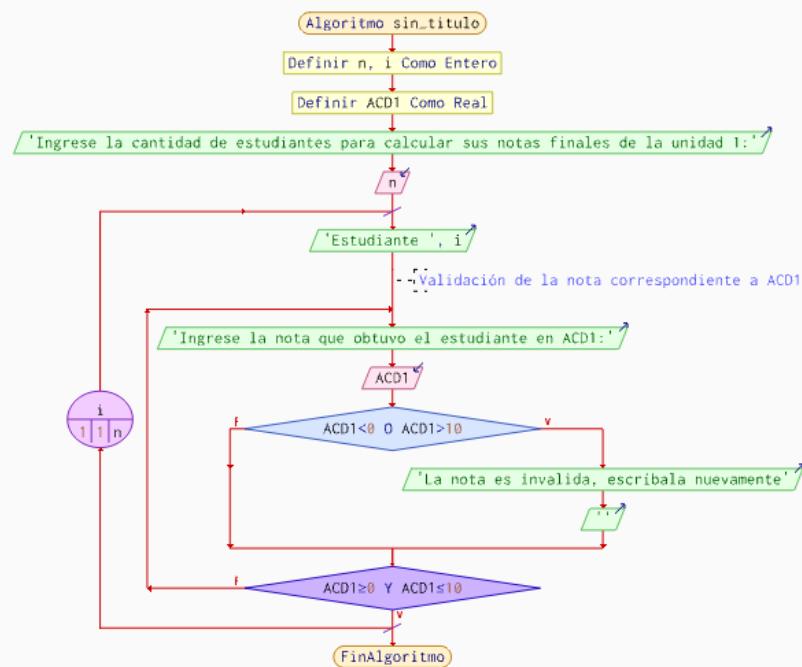
6. Resultados

• 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.

• Esquema lógico simplificado



• Código fuente en lenguaje C

```
#include <stdio.h>

int main(){

    //Definir las variables para cada actividad dentro de los ponderados ACD,AA, APE y ES
    float ACD1, ACD2, AA1, AA2, APE1, APE2, ES1, ES2, ACD, AA, APE ,ES, NF;
    float sumapromedios=0;
    int n;
    // Solicitud la cantidad de estudiantes a evaluar
    printf("Ingresar la cantidad de estudiantes para calcular sus notas finales de la unidad 1:\n");
    scanf("%d", &n);

    for(int i=1; i<=n; i++){
        printf("Estudiante %d:\n", i);
        // Validación de la nota correspondiente a ACD (Aproximación con el Docente)
        do{
            printf("Ingresar la nota que obtuvo el estudiante en ACD1:\n");
            scanf("%f", &ACD1);
            if(ACD1 > 10 || ACD1 < 0){
                printf("La nota es invalida, escribala nuevamente\n");
                printf("\n");
            }
        }while(ACD1 > 10 || ACD1 < 0);
        do{
            printf("Ingresar la nota que obtuvo el estudiante en ACD2:\n");
            scanf("%f", &ACD2);
            if(ACD2 > 10 || ACD2 < 0){
                printf("La nota es invalida, escribala nuevamente\n");
                printf("\n");
            }
        }while(ACD2 > 10 || ACD2 < 0);
    }
}
```



UNL

Universidad
Nacional
de Loja
1859

FEIRNNR - Carrera de Computación

```
 }while(ACD2 > 10 || ACD2 < 0);
 // Validación de la nota correspondiente a AA (Actividades autónomas)
 do{
    printf("Ingrese la nota que obtuvo el estudiante en AA1:\n");
    scanf("%f", &AA1);
    if(AA1 > 10 || AA1 < 0){
        printf("La nota es invalida, escribala nuevamente\n");
        printf("\n");
    }
}while(AA1 > 10 || AA1 < 0);
do{
    printf("Ingrese la nota que obtuvo el estudiante en AA2:\n");
    scanf("%f", &AA2);
    if(AA2 > 10 || AA2 < 0){
        printf("La nota es invalida, escribala nuevamente\n");
        printf("\n");
    }
}while(AA2 > 10 || AA2 < 0);
// Validación de la nota correspondiente a APE (Actividades práctico experimentales)
do{
    printf("Ingrese la nota que obtuvo el estudiante en APE1:\n");
    scanf("%f", &APE1);
    if(APE1 > 10 || APE1 < 0){
        printf("La nota es invalida, escribala nuevamente\n");
        printf("\n");
    }
}while(APE1 > 10 || APE1 < 0);
do{
    printf("Ingrese la nota que obtuvo el estudiante en APE2:\n");
    scanf("%f", &APE2);
    if(APE2 > 10 || APE2 < 0){
        printf("La nota es invalida, escribala nuevamente\n");
        printf("\n");
    }
}while(APE2 > 10 || APE2 < 0);
```



UNL

Universidad
Nacional
de Loja

1859

FEIRNNR - Carrera de Computación

```
 }while(APE2 > 10 || APE2 < 0);
 //Validación de la nota correspondiente de ES (Evaluación sumativa)
 do{
     printf("Ingrese la nota que obtuvo el estudiante en ES1(Portafolio digital):\n");
     scanf("%f", &ES1);
     if(ES1 > 10 || ES1 < 0){
         printf("La nota es invalida, escribala nuevamente\n");
         printf("\n");
     }
 }while(ES1 > 10 || ES1 < 0);
 do{
     printf("Ingrese la nota que obtuvo el estudiante en ES2(Evaluacion):\n");
     scanf("%f", &ES2);
     if(ES2 > 10 || ES2 < 0){
         printf("La nota es invalida, escribala nuevamente\n");
         printf("\n");
     }
 }while(ES2 > 10 || ES2 < 0);

 // Cálculo de ponderados
 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;
 //Calculo del promedio general de la clase
 sumapromedios = sumapromedios+ NF;
 // Clasificación del desempeño del estudiante según su nota final
 if(NF >= 9 ){
     printf("La nota del estudiante %.2f es excelente ", NF);
 }else if(NF >= 7 && NF < 9){
     printf("La nota del estudiante %.2f es buena ", NF);
 }else if(NF >= 5 && NF <7){
     printf("La nota del estudiante %.2f es regular ", NF);
 }else if(NF < 5){
     printf("La nota del estudiante %.2f es deficiente ", NF);
 }

 // Mostrar resultados
 printf("\nEl promedio del ponderado ACD es %.2f\n", ACD);
 printf("El promedio del ponderado AA es %.2f\n", AA);
 printf("El promedio del ponderado APE es %.2f\n", APE);
 printf("El promedio del ponderado ES es %.2f\n", ES);
 printf("La nota final del estudiante de la Unidad 1 es %.2f\n", NF);
```



UNL

Universidad
Nacional
de Loja
1859

FEIRNNR - Carrera de Computación

```
}

// Cálculo del promedio general de la clase
float promediogeneral = sumapromedios / n;

printf("\n");
printf("El promedio general de la clase es %.2f\n", promediogeneral);

return 0;
}
```

●Pruebas

Compilación

```
PS D:\Desktop\lenguaje C> gcc unidad1.c -o unidad1
PS D:\Desktop\lenguaje C> ./unidad1.exe
```

Estudiante 1: José Valencia

```
Ingrese la cantidad de estudiantes para calcular sus notas finales de la unidad 1:
3
Estudiante 1:
Ingrese la nota que obtuvo el estudiante en ACD1:
10
Ingrese la nota que obtuvo el estudiante en ACD2:
10
Ingrese la nota que obtuvo el estudiante en AA1:
10
Ingrese la nota que obtuvo el estudiante en AA2:
10
Ingrese la nota que obtuvo el estudiante en APE1:
8.5
Ingrese la nota que obtuvo el estudiante en APE2:
8.5
Ingrese la nota que obtuvo el estudiante en ES1(Portafolio digital):
9.5
Ingrese la nota que obtuvo el estudiante en ES2(Evaluacion):
9.5
La nota del estudiante 9.45 es excelente
El promedio del ponderado ACD es 2.00
El promedio del ponderado AA es 2.00
El promedio del ponderado APE es 2.13
El promedio del ponderado ES es 3.32
La nota final del estudiante de la Unidad 1 es 9.45
```



UNL

Universidad
Nacional
de Loja

1859

FEIRNNR - Carrera de Computación

Estudiante 2: Selena Castillo

```
Estudiante 2:  
Ingrese la nota que obtuvo el estudiante en ACD1:  
10  
Ingrese la nota que obtuvo el estudiante en ACD2:  
10  
Ingrese la nota que obtuvo el estudiante en AA1:  
10  
Ingrese la nota que obtuvo el estudiante en AA2:  
9  
Ingrese la nota que obtuvo el estudiante en APE1:  
11  
La nota es invalida, escribala nuevamente  
  
Ingrese la nota que obtuvo el estudiante en APE1:  
10  
Ingrese la nota que obtuvo el estudiante en APE2:  
9  
Ingrese la nota que obtuvo el estudiante en ES1(Portafolio digital):  
9.5  
Ingrese la nota que obtuvo el estudiante en ES2(Evaluacion):  
9.5  
La nota del estudiante 9.60 es excelente  
El promedio del ponderado ACD es 2.00  
El promedio del ponderado AA es 1.90  
El promedio del ponderado APE es 2.38  
El promedio del ponderado ES es 3.32  
La nota final del estudiante de la Unidad 1 es 9.60
```

Estudiante 3: Darío Chillogallo

```
Estudiante 3:  
Ingrese la nota que obtuvo el estudiante en ACD1:  
10  
Ingrese la nota que obtuvo el estudiante en ACD2:  
10  
Ingrese la nota que obtuvo el estudiante en AA1:  
9.5  
Ingresar la nota que obtuvo el estudiante en AA2:  
9.25  
Ingresar la nota que obtuvo el estudiante en APE1:  
9.5  
Ingresar la nota que obtuvo el estudiante en APE2:  
9  
Ingresar la nota que obtuvo el estudiante en ES1(Portafolio digital):  
8  
Ingresar la nota que obtuvo el estudiante en ES2(Evaluacion):  
9  
La nota del estudiante 9.20 es excelente  
El promedio del ponderado ACD es 2.00  
El promedio del ponderado AA es 1.88  
El promedio del ponderado APE es 2.31  
El promedio del ponderado ES es 3.01  
La nota final del estudiante de la Unidad 1 es 9.20  
  
El promedio general de la clase es 9.42
```



7. Preguntas de Control

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

Una estructura condicional permite tomar decisiones dentro del programa según se cumpla o no una condición; solo ejecuta un bloque específico de instrucciones una vez.

En cambio, una estructura repetitiva ejecuta un conjunto de instrucciones varias veces mientras una condición es verdadera. Es decir, la condicional decide qué hacer, mientras que la repetitiva decide cuántas veces hacerlo.

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

for: Se usa cuando se conoce de antemano la cantidad de repeticiones. Incluye inicialización, condición e incremento en una misma línea, lo que lo hace ideal para ciclos contados.

while: Se emplea cuando no se sabe cuántas veces se repetirá el ciclo. Evalúa la condición antes de ejecutar el bloque, por lo que podría no ejecutarse nunca.

do...while: Ejecuta el bloque al menos una vez porque la condición se evalúa al final del ciclo. Es útil cuando se necesita que el usuario ingrese un valor o acción antes de validar.

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

Las validaciones son importantes porque aseguran que los datos ingresados sean correctos, estén dentro del rango permitido y no produzcan errores en el programa. Además, evitan resultados incorrectos, mantienen la integridad del proceso y mejoran la experiencia del usuario al guiarlo cuando comete un error.

8. Conclusiones

Se concluye lo siguiente:

- Se logró comprender y aplicar adecuadamente las estructuras repetitivas, ya que el programa utilizó ciclos para procesar múltiples estudiantes, cumpliendo el objetivo de resolver un problema de tipo iterativo de manera eficiente.
- El algoritmo fue diseñado y codificado de forma correcta, integrando bucles y validaciones que garantizan la correcta captura de datos y un flujo lógico organizado para el cálculo de los diferentes componentes de la nota final.
- La ejecución práctica permitió validar el funcionamiento del programa, demostrando que las operaciones de cálculo, las validaciones y el uso de estructuras repetitivas producen resultados precisos y coherentes para cada estudiante.

9. Recomendaciones

Se recomienda lo siguiente:

- Practicar con una mayor variedad de ejercicios que utilicen diferentes tipos de bucles (while, do-while, for) para reforzar el dominio de cada estructura. Esto permitirá identificar con mayor facilidad cuál es la más adecuada según el tipo de repetición que requiera el problema, optimizando así la lógica y el rendimiento del algoritmo.



-
- Elaborar primero el algoritmo en pseudocódigo o mediante un diagrama de flujo antes de escribir el código final, ya que estas herramientas facilitan visualizar el recorrido del programa, prever posibles errores y asegurar una secuencia lógica coherente. Este paso es esencial para evitar fallas durante la codificación y mejorar la claridad del proceso de diseño.
 - Realizar pruebas incrementales y continuas del programa, ejecutándolo por partes para verificar que cada sección del ciclo funcione correctamente. Este método ayuda a detectar a tiempo errores relacionados con las condiciones de entrada, salida o control de iteraciones, garantizando que el programa final cumpla con los objetivos planteados de manera confiable.

Uso de la Inteligencia Artificial

En la realización de este informe se emplearon herramientas de inteligencia artificial que ayudaron a redactar y estructurar adecuadamente el contenido.