

Reporte Técnico de Actividades Práctico-Experimentales Nro. 002

1. Datos de Identificación del Estudiante y la Práctica

Nombre del estudiante(s)	Pilar Valentina Naranjo Quizhpe
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	Lisette 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.

2. Materiales, Reactivos, Equipos y Herramientas

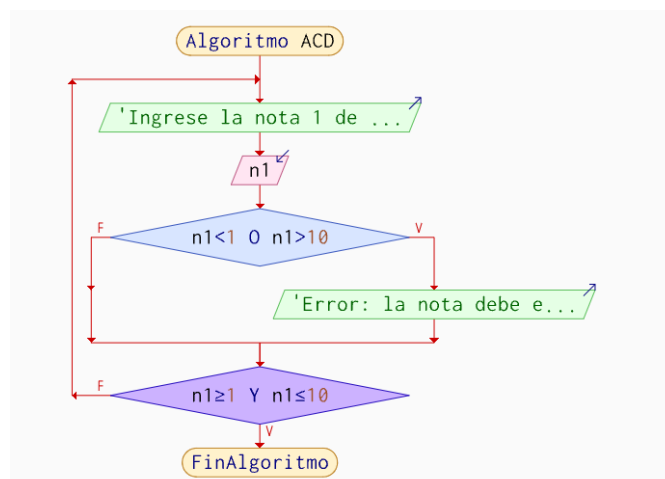
- Herramientas de modelado de diagram de flujo (Psient, Draw.io, Lucidchart, otros)
- IDE de programación: Visual Studio Code u otro entorno compatible.
- Lenguaje de programación: C (según los contenidos de la unidad).
- Computador personal con sistema operativo Windows, Linux o macOS.
- Material de 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.

3. Procedimiento / Metodología Ejecutada

- Se analizó el problema definiendo los datos de entrada (notas de los estudiantes), el proceso mediante ponderaciones y la salida correspondiente a la nota final.
- Se elaboró un diagrama de flujo para comprender mejor los bucles utilizados y cómo permiten repetir el cálculo de manera ordenada para varios estudiantes.
- Finalmente, se realizaron pruebas con diferentes conjuntos de notas reales para verificar que el programa funcione correctamente en distintos escenarios.

3. 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

```
1  #include <stdio.h>
2  int main() {
3      // Declaración de variables
4      float acd, n1, n2, aa, n3, n4, ape, n5, n6, es, portafolio, evaluacion, promedio;
5      int estudiantes, i;
6      int acumulador = 0;
7      i=estudiantes;
8      // Entrada de datos
9      printf("Ingrese el numero de estudiantes:\n");
10     scanf("%d", &estudiantes);
11
12     for(i = 1; i <= estudiantes; i++){
13         printf(" Notas del estudiante %d\n", i);
14
15
16         //ACD
17         do {
18             printf("Ingrese la nota 1 de ACD: ");
19             scanf("%f", &n1);
20             if (n1 < 1 || n1 > 10) printf("Error: la nota debe estar entre 1 y 10.\n");
21         } while (n1 < 1 || n1 > 10);
22
23         do {
24             printf("Ingrese la nota 2 de ACD:");
25             scanf("%f", &n2);
26             if (n2 < 1 || n2 > 10) printf("Error: la nota debe estar entre 1 y 10.\n");
27         } while (n2 < 1 || n2 > 10);
28
29         //AA
30         do {
31             printf("Ingrese la nota 1 de AA: ");
32             scanf("%f", &n3);
33             if (n3 < 1 || n3 > 10) printf("Error: la nota debe estar entre 1 y 10.\n");
34         } while (n3 < 1 || n3 > 10);
35
36         do {
37             printf("Ingrese la nota 2 de AA:");
38             scanf("%f", &n4);
39             if (n4 < 1 || n4 > 10) printf("Error: la nota debe estar entre 1 y 10.\n");
40         } while (n4 < 1 || n4 > 10);
41
42         //APE
43         do {
44             printf("Ingrese la nota 1 de APE: ");
45             scanf("%f", &n5);
46             if (n5 < 1 || n5 > 10) printf("Error: la nota debe estar entre 1 y 10.\n");
47         } while (n5 < 1 || n5 > 10);
48
49         do {
50             printf("Ingrese la nota 2 de APE:");
51             scanf("%f", &n6);
52             if (n6 < 1 || n6 > 10) printf("Error: la nota debe estar entre 1 y 10.\n");
53         } while (n6 < 1 || n6 > 10);
54
55         //Portafolio
56         do {
57             printf("Ingrese su nota del portafolio:");
58             scanf("%f", &portafolio);
```

```

59         if (portafolio < 1 || portafolio > 10) printf("Error: la nota debe estar entre 1 y 10.\n");
60     } while (portafolio < 1 || portafolio > 10);
61
62     //Evaluacion
63     do {
64         printf("Ingrese su nota de la evaluacion:");
65         scanf("%f", &evaluacion);
66         if (evaluacion < 1 || evaluacion > 10) printf("Error: la nota debe estar entre 1 y 10.\n");
67     } while (evaluacion < 1 || evaluacion > 10);
68
69     // Proceso
70     acd = ((n1 + n2) / 2) * 0.2;
71     aa = ((n3 + n4) / 2) * 0.2;
72     ape = ((n5 + n6) / 2) * 0.25;
73     es = ((portafolio * 0.4) + (evaluacion * 0.6)) * 0.35;
74     promediof = acd + aa + ape + es;
75
76     acumulador += promediof;
77
78     if (promediof >= 9) {
79         printf("Desempeño: Excelente\n");
80     } else if (promediof >= 7) {
81         printf("Desempeño: Bueno\n");
82     } else if (promediof >= 5) {
83         printf("Desempeño: Regular\n");
84     } else {
85     }
86
87     // Datos de salida
88
89     printf("Nota ponderada ACD: %.2f\n", acd);
90     printf("Nota ponderada AA : %.2f\n", aa);
91     printf("Nota ponderada APE: %.2f\n", ape);
92     printf("Nota ponderada ES : %.2f\n", es);
93     printf("Su nota final de la Unidad 1 es: %.2f\n", promediof );
94
95 }
96
97 float total = acumulador / estudiantes;
98 printf("El promedio general de estos estudinates es: %.2f\n", total);
99
100 return 0;
101 }

```

• Pruebas

1. Roy Gordillo

```

Ingrese el numero de estudiantes:
3
Notas del estudiante 1
Ingrese la nota 1 de ACD: 7
Ingrese la nota 2 de ACD:10
Ingrese la nota 1 de AA: 9.5
Ingrese la nota 2 de AA:9
Ingrese la nota 1 de APE: 9.25
Ingrese la nota 2 de APE:6
Ingrese su nota del portafolio:7
Ingrese su nota de la evaluacion:9
Desempeño: Bueno
Nota ponderada ACD: 1.70
Nota ponderada AA : 1.85
Nota ponderada APE: 1.91
Nota ponderada ES : 2.87
Su nota final de la Unidad 1 es: 8.33

```

2. Arlette Quezada

```

Notas del estudiante 2
Ingrese la nota 1 de ACD: 10
Ingrese la nota 2 de ACD:10
Ingrese la nota 1 de AA: 10
Ingrese la nota 2 de AA:9
Ingrese la nota 1 de APE: 9.75
Ingrese la nota 2 de APE:6.5
Ingrese su nota del portafolio:9.5
Ingrese su nota de la evaluacion:9
Desempeño: Excelente
Nota ponderada ACD: 2.00
Nota ponderada AA : 1.90
Nota ponderada APE: 2.03
Nota ponderada ES : 3.22
Su nota final de la Unidad 1 es: 9.15

```



José Soto

```
Notas del estudiante 3
Ingrese la nota 1 de ACD: 10
Ingrese la nota 2 de ACD:10
Ingrese la nota 1 de AA: 10
Ingrese la nota 2 de AA:10
Ingrese la nota 1 de APE: 9.25
Ingrese la nota 2 de APE:6.5
Ingrese su nota del portafolio:8.5
Ingrese su nota de la evaluacion:9
Desempeño: Excelente
Nota ponderada ACD: 2.00
Nota ponderada AA : 2.00
Nota ponderada APE: 1.97
Nota ponderada ES : 3.08
Su nota final de la Unidad 1 es: 9.05
El promedio general de estos estudinates es:8.84
```

4. Preguntas de Control

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

Las estructuras condicionales permiten decidir qué instrucciones ejecutar dependiendo de si una condición es verdadera o falsa. Es decir, solo se ejecutan una vez, y únicamente si la condición se cumple.

En cambio, las estructuras repetitivas permiten ejecutar un conjunto de instrucciones varias veces, se repite continuamente hasta que la condición deje de cumplirse.

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

El ciclo **for** se usa cuando el número de repeticiones es conocido, ya que permite definir en una sola línea la inicialización, la condición y el incremento. El **while** es útil cuando no sabemos cuántas veces se repetirá el proceso, pero sí conocemos la condición que debe mantenerse verdadera. En cambio, el **do...while** asegura que el bloque se ejecute al menos una vez antes de validar la condición, ideal cuando primero se necesita ingresar o procesar un dato.

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

Para prevenir fallos en el manejo, confirmar que el usuario proporcione datos precisos y garantizar que los resultados sean confiables.

5. Conclusiones

En esta práctica se utilizaron correctamente estructuras repetitivas para procesar varios estudiantes y se aplicaron validaciones que aseguraron que las notas ingresadas fueran válidas. El programa demuestra cómo los bucles ayudan a automatizar tareas repetitivas y a realizar cálculos de manera ordenada. Gracias a esto, se cumplió el objetivo principal: automatizar el cálculo de notas de forma clara y eficiente.