



Reporte Técnico de Actividades Práctico-Experimentales Nro. 002

1. Datos de Identificación del Estudiante y la Práctica

Nombre del estudiante(s)	Alison Tapia
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 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 (PSeInt, 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.



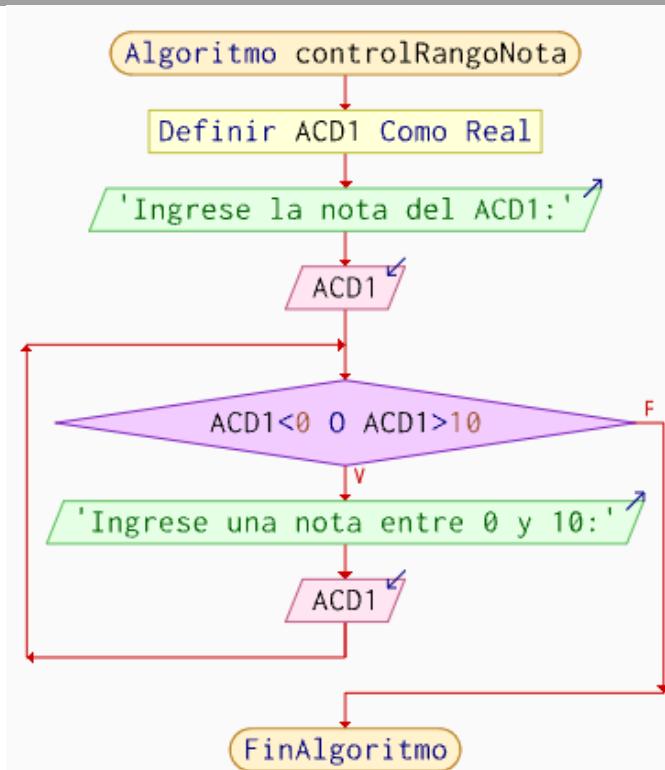
- Conexión a internet estable para acceder a recursos digitales y software en línea.
- Aula física asignada al paralelo.

4. Procedimiento / Metodología Ejecutada

1. Se analizó el problema y se definió los bucles a utilizar según las instrucciones solicitadas.
2. Se añadió el bucle while para validar la entrada de notas.
3. Se utilizó un bucle for para calcular el promedio de n estudiantes.
4. Se realizaron pruebas de escritorio.

5. 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:

```
C evaluacionCondicional.c > main()
1  #include <stdio.h>
2  int main(){
3      //Definición de variables
4      int nEstudiantes, i;
5      float promedioEstudiantes = 0;
6      float ACD1, ACD2, APE1, APE2, AA1, AA2, ES1, ES2;
7      float ponderadoACD, ponderadoAPE, ponderadoAA, ponderadoES, promedioFinal;
8      char * escala = "Excelente";
9
10     //Datos de entrada
11     printf("Ingrese el numero de estudiantes a calcular promedio: ");
12     scanf("%i", &nEstudiantes);
13     getchar();
14
15     for (i = 1; i <= nEstudiantes; i++){
16
17         //Ingreso notas de cada estudiante
18
19         printf("Ingrese las notas del estudiante %i\n", i);
20         printf("Ingrese la nota del ACD1: ");
21         scanf("%f", &ACD1);
22         while (ACD1 < 0 || ACD1 > 10){
23             printf("Ingrese una nota entre 0 y 10:\n");
24             scanf("%f", &ACD1);
25         }
26     }
27 }
```



UNL

Universidad
Nacional
de Loja
1859

FEIRNNR - Carrera de Computación

```
26         printf("Ingrese la nota del ACD2: ");
27         scanf("%f", &ACD2);
28         getchar();
29         while (ACD2 < 0 || ACD2 > 10){
30             printf("Ingrese una nota entre 0 y 10:\n");
31             scanf("%f", &ACD2);
32         }
33
34
35         printf("Ingrese la nota del APE1: ");
36         scanf("%f", &APE1);
37         getchar();
38         while (APE1 < 0 || APE1 > 10){
39             printf("Ingrese una nota entre 0 y 10:\n");
40             scanf("%f", &APE1);
41         }
42
43         printf("Ingrese la nota del APE2: ");
44         scanf("%f", &APE2);
45         getchar();
46         while (APE2 < 0 || APE2 > 10){
47             printf("Ingrese una nota entre 0 y 10:\n");
48             scanf("%f", &APE2);
49         }
50
51         printf("Ingrese la nota del AA1: ");
52         scanf("%f", &AA1);
53         getchar();
54         while (AA1 < 0 || AA1 > 10){
55             printf("Ingrese una nota entre 0 y 10:\n");
56             scanf("%f", &AA1);
57         }
58
59         printf("Ingrese la nota del AA2: ");
60         scanf("%f", &AA2);
61         getchar();
62         while (AA2 < 0 || AA2 > 10){
63             printf("Ingrese una nota entre 0 y 10:\n");
64             scanf("%f", &AA2);
65         }
66
67         printf("Ingrese la nota del Portafolio digital: ");
68         scanf("%f", &ES1);
69         getchar();
70         while (ES1 < 0 || ES1 > 10){
71             printf("Ingrese una nota entre 0 y 10:\n");
72             scanf("%f", &ES1);
73         }
```



```
74
75     printf("Ingrese la nota del Aprendizaje Basado en Problemas: ");
76     scanf("%f", &ES2);
77     getchar();
78     while (ES2 < 0 || ES2 > 10){
79         printf("Ingrese una nota entre 0 y 10:\n");
80         scanf("%f", &ES2);
81     }
82
83
84     //Cálculo ponderados promedio de cada estudiante
85
86     ponderadoACD = ((ACD1 + ACD2)/2) * 0.2;
87     ponderadoAPE = ((APE1 + APE2)/2) * 0.25;
88     ponderadoAA = ((AA1 + AA2)/2) * 0.2;
89     ponderadoES = ((ES1 * 0.4) + (ES2 * 0.6)) * 0.35;
90     promedioFinal = ponderadoACD + ponderadoAPE + ponderadoAA + ponderadoES;
91
92     //Muestra escala cualitativa en base al promedio
93
94     if(promedioFinal >= 9){
95         escala = "Excelente";
96     }else if(promedioFinal < 9 && promedioFinal >= 7){
97         escala = "Bueno";
98     }else if(promedioFinal < 7 && promedioFinal >= 5){
99         escala = "Regular";
100    }else if(promedioFinal < 5){
101        escala = "Deficiente";
102    }
103
104    //Suma promedios de los estudiantes
105
106    promedioEstudiantes = (promedioEstudiantes + promedioFinal);
107
108
109    //Muestra promedio individual y ponderados de cada estudiante
110
111    printf("La nota ponderada del ACD es: %.2f\n", ponderadoACD);
112    printf("La nota ponderada del APE es: %.2f\n", ponderadoAPE);
113    printf("La nota ponderada del AA es: %.2f\n", ponderadoAA);
114    printf("La nota ponderada del ES es: %.2f\n", ponderadoES);
115    printf("");
116    printf("La nota final de la Unidad 1 del estudiante %i es: %.2f\n", i, promedioFinal);
117    printf("La escala cualitativa es: %s\n", escala);
118    printf("\n");
119
120
121    //Cálculo promedio todos los estudiantes en general
122
123    promedioEstudiantes = promedioEstudiantes / nEstudiantes;
124    printf("El promedio de los estudiantes es: %.2f", promedioEstudiantes);
125 }
```

- **Pruebas:**



```
Ingrese el numero de estudiantes a calcular promedio: 3
Ingrese las notas del estudiante 1
Ingrese la nota del ACD1: 8
Ingrese la nota del ACD2: 7
Ingrese la nota del APE1: 10
Ingrese la nota del APE2: 9
Ingrese la nota del AA1: 9.5
Ingrese la nota del AA2: 9.5
Ingrese la nota del Portafolio digital: 10
Ingrese la nota del Aprendizaje Basado en Problemas: 7
La nota ponderada del ACD es: 1.50
La nota ponderada del APE es: 2.38
La nota ponderada del AA es: 1.90
La nota ponderada del ES es: 2.87
La nota final de la Unidad 1 del estudiante 1 es: 8.64
La escala cualitativa es: Bueno

Ingrese las notas del estudiante 2
Ingrese la nota del ACD1: 10
Ingrese la nota del ACD2: 10
Ingrese la nota del APE1: 10
Ingrese la nota del APE2: 9
Ingrese la nota del AA1: 10
Ingrese la nota del AA2: 9
Ingrese la nota del Portafolio digital: 9.5
Ingrese la nota del Aprendizaje Basado en Problemas: 9.5
La nota ponderada del ACD es: 2.00
La nota ponderada del APE es: 2.38
La nota ponderada del AA es: 1.90
La nota ponderada del ES es: 3.33
La nota final de la Unidad 1 del estudiante 2 es: 9.60
La escala cualitativa es: Excelente

Ingrese las notas del estudiante 3
Ingrese la nota del ACD1: 10
Ingrese la nota del ACD2: 10
Ingrese la nota del APE1: 10
Ingrese la nota del APE2: 9
Ingrese la nota del AA1: 10
Ingrese la nota del AA2: 10
Ingrese la nota del Portafolio digital: 9
Ingrese la nota del Aprendizaje Basado en Problemas: 9.5
La nota ponderada del ACD es: 2.00
La nota ponderada del APE es: 2.38
La nota ponderada del AA es: 2.00
La nota ponderada del ES es: 3.26
La nota final de la Unidad 1 del estudiante 3 es: 9.63
La escala cualitativa es: Excelente
```

6. Preguntas de Control

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



Se diferencian principalmente en que la estructura repetitiva evalúa una condición múltiples veces y ejecuta un bloque de instrucciones mientras dicha condición se cumpla; en cambio, la estructura condicional evalúa una condición solo una vez para decidir qué bloque de código se ejecutará.

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

La diferencia es:

- **for** se utiliza para repetir código basado en una condición cuando se conoce el número exacto de iteraciones o se tiene un contador definido.
- **while** repite código mientras una condición sea verdadera, y se usa cuando no se sabe cuántas iteraciones habrá.
- **do...while** también repite código en función de una condición, pero ejecuta el bloque al menos una vez antes de evaluar dicha condición, siendo útil cuando se necesita que el ciclo corra mínimo una vez.

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

Es importante incluir validaciones porque permiten controlar la calidad y corrección de los datos ingresados, evitando errores y asegurando que el programa funcione de manera adecuada y segura.

7. Conclusiones

- Se comprendió y aplicó las estructuras repetitivas en la resolución de problemas.
- Se diseñó y codificó un algoritmo utilizando bucles resolviendo un problema de tipo iterativo.
- Se validó el funcionamiento del programa mediante la ejecución práctica.

8. Recomendaciones

- Se recomienda que al utilizar bucles se redacte correctamente la condición y se incluya el incremento o actualización correspondiente. Además, es importante asegurarse de que la condición llegue a ser falsa en algún momento para evitar la creación de un bucle infinito.
- Se recomienda analizar correctamente el problema y establecer de forma adecuada el contador, asegurando que se inicialice correctamente y se actualice de manera coherente para evitar errores en la lógica del ciclo.