



1859

unl
computaciónUniversidad
Nacional
de Loja

FEIRNNR

-

Carrera

de

Reporte Técnico de Actividades Práctico-Experimentales Nro. 002

1. Datos de Identificación del Estudiante y la Práctica

Nombre del estudiante(s)	Selena Castillo
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 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, 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). FEIRNNR - Carrera de Computación.



- 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

Primeramente, revisé el código de la evaluación 1 y con las indicaciones de la docente, analicé las posibles soluciones y una vez con la idea clara, comencé a redactar el código en C en Visual Studio.

Luego, realicé las pruebas necesarias para comprobar el correcto funcionamiento del programa.

Finalmente, utilicé la herramienta Pseint para graficar el diagrama de flujo simplificado y procedí a redactar el presente informe.

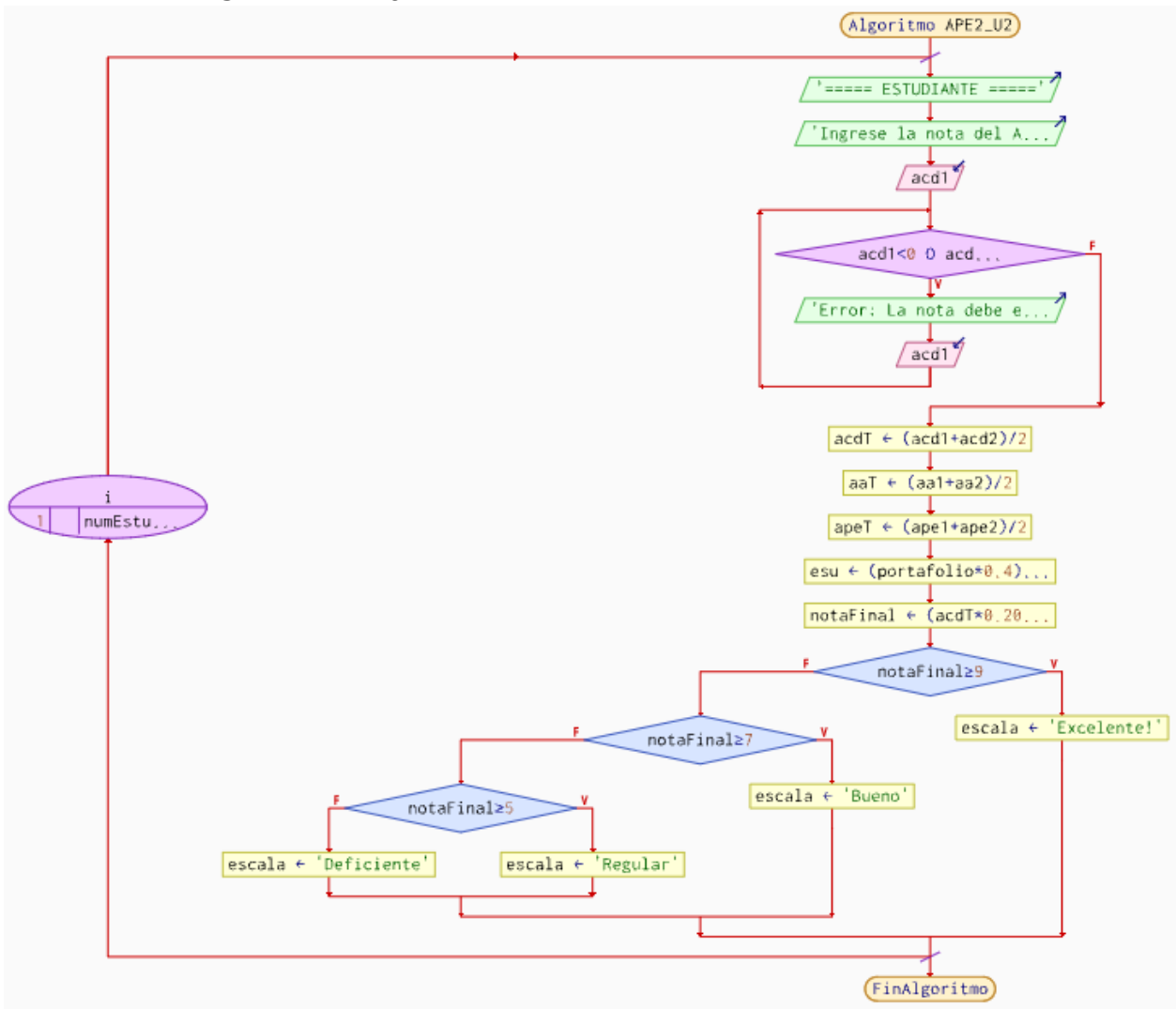
5. Resultados

- **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.

• Diagrama de flujo:



• Código en C

```
//insertamos las librerias necesarias
```

```
#include <stdio.h>
```

```
#include <float.h>
```

```
int main() {
```

```
    //Definimos las variables a utilizar
```

```
    float acd1, acd2, acdT, aa1, aa2, aaT, ape1, ape2, apeT, portafolio, abp,
```

```
    es, notaFinal, promSimple=0;
```

```
    int numEstudiantes;
```

```
    char * escala = "Excelente";
```

```
    //Solicitamos el numero de estudiantes al usuario y leemos
```

```
    printf("Ingrese el numero de estudiantes para calcular promedio:\n");
```

```
    scanf("%d", &numEstudiantes);
```

```
    for ( int i = 1; i <= numEstudiantes; i++) {
```



```
//Encabezado para mostrar el numero de estudiante
printf("\n===== ESTUDIANTE %d =====\n", i);

//Proceso de lectura de notas usando el ciclo while para validacion de
datos
//Solicitamos las notas de ACD al usuario y leemos
printf("Ingrese la nota del ACD de algoritmos:\n");
scanf("%f", &acd1);
while(acd1 < 0 || acd1 > 10){
printf("Error: La nota debe estar entre 0 y 10. Ingrese nuevamente la
nota del ACD de algoritmos:\n");
scanf("%f", &acd1);
}

printf("Ingrese la nota del ACD de lenguaje C:\n");
scanf("%f", &acd2);
while(acd2 < 0 || acd2 > 10){
printf("Error: La nota debe estar entre 0 y 10. Ingrese nuevamente la
nota del ACD de lenguaje C:\n");
scanf("%f", &acd2);
}

//Solicitamos las notas de AA al usuario y leemos
printf("Ingrese la nota del AA de herramientas:\n");
scanf("%f", &aa1);
while(aa1 < 0 || aa1 > 10){
printf("Error: La nota debe estar entre 0 y 10. Ingrese nuevamente la
nota del AA de herramientas:\n");
scanf("%f", &aa1);
}

printf("Ingrese la nota del AA de instalacion de lenguajes:\n");
scanf("%f", &aa2);
while(aa2 < 0 || aa2 > 10){
printf("Error: La nota debe estar entre 0 y 10. Ingrese nuevamente la
nota del AA de instalacion de lenguajes:\n");
scanf("%f", &aa2);
}

//Solicitamos las notas de APE al usuario y leemos
printf("Ingrese la nota del APE de algoritmos:\n");
scanf("%f", &ape1);
while(ape1 < 0 || ape1 > 10){
printf("Error: La nota debe estar entre 0 y 10. Ingrese nuevamente la
nota del APE de algoritmos:\n");
scanf("%f", &ape1);
}

printf("Ingrese la nota del APE de lenguaje C:\n");
```

```
scanf("%f", &ape2);
while(ape2 < 0 || ape2 > 10){
printf("Error: La nota debe estar entre 0 y 10. Ingrese nuevamente la
nota del APE de lenguaje C:\n");
scanf("%f", &ape2);
}

//Solicitamos las notas de Evaluación Sumativa al usuario y leemos
printf("Ingrese la nota del portafolio:\n");
scanf("%f", &portafolio);
while(portafolio < 0 || portafolio > 10){
printf("Error: La nota debe estar entre 0 y 10. Ingrese nuevamente la
nota del portafolio:\n");
scanf("%f", &portafolio);
}

printf("Ingrese la nota del ABP:\n");
scanf("%f", &abp);
while(abp < 0 || abp > 10){
printf("Error: La nota debe estar entre 0 y 10. Ingrese nuevamente la
nota del ABP:\n");
scanf("%f", &abp);
}

//Proceso
//Calculamos la nota final de ACD
acdT= (acd1 + acd2)/2;
//Calculamos la nota final de AA
aaT=(aa1+aa2)/2;
//Calculamos la nota final de APE
apeT=(ape1+ape2)/2;
//Calculamos la nota final de Evaluación Sumativa
es= (portafolio * 0.4) + (abp * 0.6);
//Con todos los datos listos, calculamos y mostramos la nota final de
la Unidad 1
notaFinal= (acdT * 0.20) + (aaT * 0.20) + (apeT * 0.25) + (es * 0.35);

//Condiciona para calificación cualitativa
if (notaFinal >= 9){
    escala="Excelente!";
} else if (notaFinal >= 7 && notaFinal < 9){
    escala="Bueno";
} else if (notaFinal >= 5 && notaFinal < 7){
    escala="Regular";
} else {
    escala="Deficiente";
}

//Mostrar resultados
```

```
printf("Su nota final de ACD en la Unidad 1 es: %.2f\n", acdT);
printf("Su nota final de AA en la Unidad 1 es: %.2f\n", aaT);
printf("Su nota final de APE en la Unidad 1 es: %.2f\n", apeT);
printf("Su nota final de Evaluacion Sumativa en la Unidad 1 es: %.2f\n",
es);
printf("Su nota final de la Unidad 1 es: %.2f\n", notaFinal);
printf("La escala cualitativa es: %s\n", escala);

promSimple = promSimple + notaFinal;
}

//Calculamos y mostramos el promedio simple de notas finales de la Unidad 1
de todos los estudiantes
promSimple = promSimple / numEstudiantes;
printf("El promedio de notas finales de la Unidad 1 de los %d estudiantes
es: %.2f\n", numEstudiantes, promSimple);

return 0;
}
```

- **Pruebas de escritorio:**

Compañeros de clase:

```
Ingrese el numero de estudiantes para calcular promedio:
3

===== ESTUDIANTE 1 =====
Ingrese la nota del ACD de algoritmos:
10
Ingrese la nota del ACD de lenguaje C:
10
Ingrese la nota del AA de herramientas:
9.5
Ingrese la nota del AA de instalacion de lenguajes:
9.5
Ingrese la nota del APE de algoritmos:
10
Ingrese la nota del APE de lenguaje C:
7.5
Ingrese la nota del portafolio:
9.5
Ingrese la nota del ABP:
10
Su nota final de ACD en la Unidad 1 es: 10.00
Su nota final de AA en la Unidad 1 es: 9.50
Su nota final de APE en la Unidad 1 es: 8.75
Su nota final de Evaluacion Sumativa en la Unidad 1 es: 9.80
Su nota final de la Unidad 1 es: 9.52
La escala cualitativa es: Excelente!

===== ESTUDIANTE 2 =====
Ingrese la nota del ACD de algoritmos:
8
Ingrese la nota del ACD de lenguaje C:
7
```



1859

unl
computación

Universidad
Nacional
de Loja

FEIRNNR

-

Carrera

de

```
Ingrese la nota del AA de herramientas:
10
Ingrese la nota del AA de instalacion de lenguajes:
9
Ingrese la nota del APE de algoritmos:
9.5
Ingrese la nota del APE de lenguaje C:
7.5
Ingrese la nota del portafolio:
10
Ingrese la nota del ABP:
7
Su nota final de ACD en la Unidad 1 es: 7.50
Su nota final de AA en la Unidad 1 es: 9.50
Su nota final de APE en la Unidad 1 es: 8.50
Su nota final de Evaluacion Sumativa en la Unidad 1 es: 8.20
Su nota final de la Unidad 1 es: 8.39
La escala cualitativa es: Bueno
```

===== ESTUDIANTE 3 =====

```
Ingrese la nota del ACD de algoritmos:
10
Ingrese la nota del ACD de lenguaje C:
10
Ingrese la nota del AA de herramientas:
10
Ingrese la nota del AA de instalacion de lenguajes:
9
Ingrese la nota del APE de algoritmos:
10
Ingrese la nota del APE de lenguaje C:
9
```

```
Ingrese la nota del portafolio:
9.5
Ingrese la nota del ABP:
9.5
Su nota final de ACD en la Unidad 1 es: 10.00
Su nota final de AA en la Unidad 1 es: 9.50
Su nota final de APE en la Unidad 1 es: 9.50
Su nota final de Evaluacion Sumativa en la Unidad 1 es: 9.50
Su nota final de la Unidad 1 es: 9.60
La escala cualitativa es: Excelente!
El promedio de notas finales de la Unidad 1 de los 3 estudiantes es: 9.17
```

Tres estudiantes: (Prueba del funcionamiento del while)

```
Su nota final de AA en la Unidad 1 es: 6.00
Su nota final de APE en la Unidad 1 es: 8.50
Su nota final de Evaluacion Sumativa en la Unidad 1 es: 8.60
Su nota final de la Unidad 1 es: 7.84
La escala cualitativa es: Bueno

===== ESTUDIANTE 3 =====
Ingrese la nota del ACD de algoritmos:
6
Ingrese la nota del ACD de lenguaje C:
5
Ingrese la nota del AA de herramientas:
8
Ingrese la nota del AA de instalacion de lenguajes:
9
Ingrese la nota del APE de algoritmos:
6
Ingrese la nota del APE de lenguaje C:
7
Ingrese la nota del portafolio:
90
Error: La nota debe estar entre 0 y 10. Ingrese nuevamente la nota del portafolio:
9
Ingrese la nota del ABP:
10
Su nota final de ACD en la Unidad 1 es: 5.50
Su nota final de AA en la Unidad 1 es: 8.50
Su nota final de APE en la Unidad 1 es: 6.50
Su nota final de Evaluacion Sumativa en la Unidad 1 es: 9.60
Su nota final de la Unidad 1 es: 7.79
La escala cualitativa es: Bueno
El promedio de notas finales de la Unidad 1 de los 3 estudiantes es: 7.88
```

6. Preguntas de Control

- **¿En qué se diferencia una estructura repetitiva de una condicional?**
Se diferencian principalmente porque una condicional decide si un bloque se ejecuta o no según una condición y una estructura repetitiva ejecuta un bloque varias veces mientras se cumple una condición.
- **¿Qué diferencia existe entre las estructuras for, while y do...while en cuanto a su funcionamiento y uso?**
 - For: Se usa cuando se conoce cuántas veces se repetirá el proceso.
 - While: Se usa cuando no se sabe cuántas repeticiones habrá y se evalúa antes de ejecutar.
 - Do...while: Ejecuta el bloque al menos una vez, porque la condición se evalúa al final.
- **¿Por qué es importante incluir validaciones dentro de un programa cuando se solicitan datos al usuario?**
Es importante porque permiten evitar errores, asegurar que los datos ingresados sean correctos y garantizar un funcionamiento seguro y confiable del programa.

7. Conclusiones

- En esta práctica se logró entender mejor cómo funcionan las estructuras repetitivas, aplicándolas para repetir procesos como el ingreso y validación de datos.
- Se pudo diseñar y programar un algoritmo sencillo y correcto, usando bucles para automatizar tareas repetitivas y evitar errores.
- Al ejecutar el programa, se comprobó que funciona correctamente, ya que valida los datos ingresados y realiza los cálculos esperados.



8. Recomendaciones

- Seguir practicando el uso de bucles y validaciones, ya que ayudan a escribir programas más ordenados y seguros.
- Probar el programa con diferentes datos de entrada, para detectar posibles errores y mejorar su funcionamiento.