



Reporte Técnico de Actividades Práctico-Experimentales Nro. 001

1. Datos de Identificación del Estudiante y la Práctica

Nombre del estudiante	Kiara Salomé Condoy Morocho
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.	001
Tipo	Individual
Título de la Práctica	Aplicación de estructuras condicionales en la resolución de problemas.
Nombre del Docente	Lissette Geoconda López Faicán
Fecha	Jueves 20 de noviembre del 2025
Horario	10h30 – 13h30
Lugar	Aula física asignada al paralelo.
Tiempo planificado en el Sílabo	6 horas

2. Objetivos de la Práctica

- Comprender y aplicar las estructuras condicionales simples, dobles y múltiples en la resolución de problemas.
- Diseñar y codificar un algoritmo que utilice sentencias de decisión para analizar y clasificar información.
- Validar el funcionamiento del programa mediante la ejecución práctica.

3. 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). 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

La práctica se desarrolló siguiendo una metodología basada en la resolución de problemas, aplicando estructuras condicionales en lenguaje C. A continuación, se detallan los pasos realizados:

1. **Apertura del entorno de desarrollo:** Se inició abriendo Visual Studio Code. Dentro del entorno, se localizó el archivo correspondiente a la evaluación de la Unidad 1, el cual contenía el código base. Este código fue copiado y pegado en un nuevo archivo para trabajar específicamente en la Unidad 2.
2. **Análisis del problema:** Se revisaron las instrucciones de la práctica para identificar las condiciones necesarias para clasificar el promedio final del estudiante según la escala cualitativa establecida.
3. **Diseño del algoritmo:** Se elaboró un diagrama de flujo que representara la lógica del programa, permitiendo visualizar de forma clara la estructura condicional múltiple que se debía implementar.
4. **Codificación:** Con base en el diagrama de flujo, se procedió a codificar el programa en lenguaje C, utilizando estructuras condicionales if y else if para determinar la calificación cualitativa según el promedio calculado.
5. **Pruebas de funcionamiento:** Se ejecutó el programa con un caso de prueba personal, utilizando las siguientes notas:
 - **AA 1:** 10 **AA 2:** 10
 - **APE 1:** 10 **APE 2:** 10
 - **ACD 1:** 10 **ACD 2:** 10
 - **Evaluación Sumativa:** 9
 - **Portafolio:** 10

El resultado obtenido coincidió correctamente con el promedio final esperado, validando así el funcionamiento del programa.

6. **Documentación:** Finalmente, se procedió a documentar todo el proceso en el presente informe, incluyendo el análisis, el diseño del algoritmo, el código fuente, las pruebas realizadas y la reflexión correspondiente.

5. Resultados

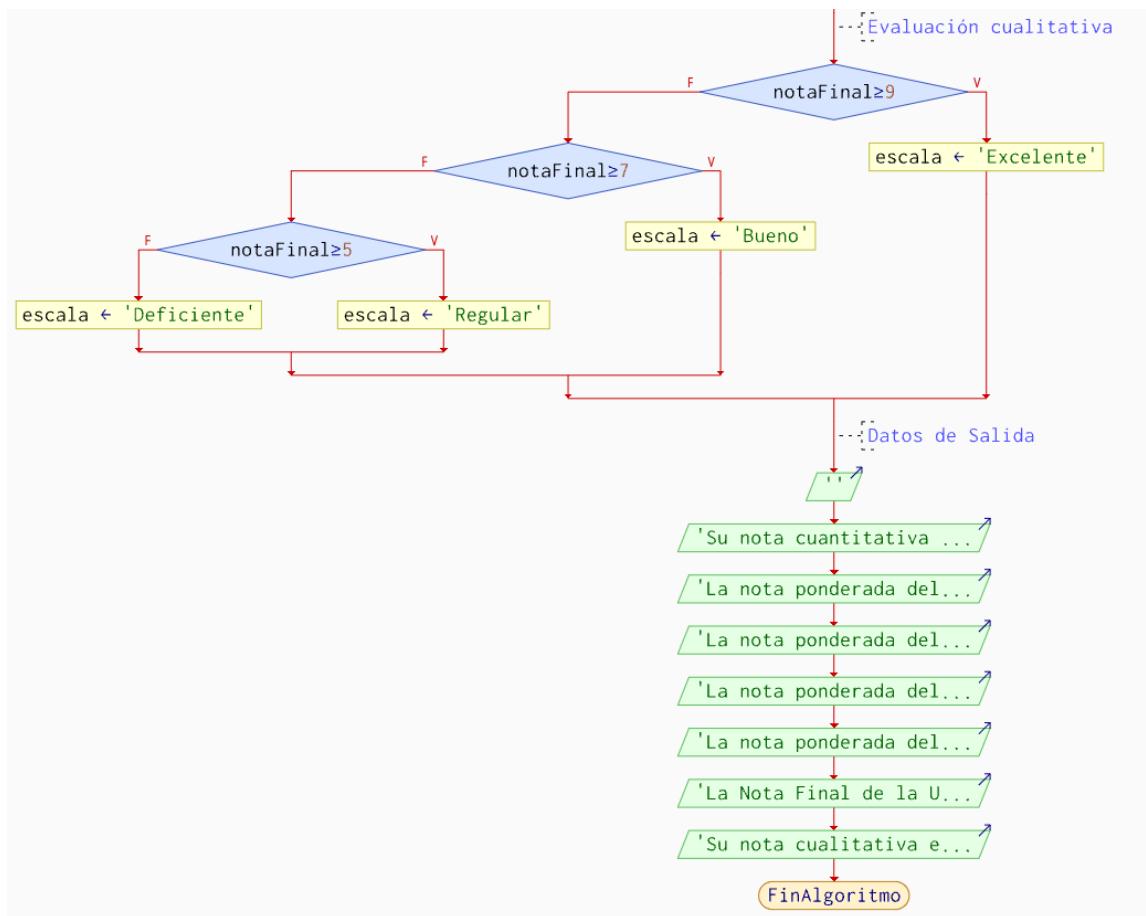
5.1. Contextualización del Problema

Basado en el ejercicio del “Cálculo de la nota final de la Unidad 1 mediante estructuras secuenciales en C”, mostrar el nivel de desempeño del promedio de la unidad 1, aplicando las siguientes condiciones:

- Nota $\geq 9 \rightarrow$ “Excelente”
- Nota ≥ 7 y $< 9 \rightarrow$ “Bueno”
- Nota ≥ 5 y $< 7 \rightarrow$ “Regular”
- Nota $< 5 \rightarrow$ “Deficiente”



5.2. Diagrama de Flujo



5.3. Código fuente en lenguaje C.

```
C U2APE1.c > main()
1  #include <stdio.h>
2
3  int main (){
4
5      //Definición de variables
6      float acd1, acd2, ape1, ape2, aa1, aa2, evaluacion, portafolio;
7      float acdPonderado, apePonderado, aaPonderado, esPonderado, notaFinal;
8      char *escala;
9
10
11     //Datos de Entrada
12     printf("Ingrese las notas del Aprendizaje en Contacto con el Docente (ACD) 1 y 2\n");
13     scanf("%f, %f", &acd1, &acd2);
14
15     printf("Ingrese las notas del Aprendizaje Practico Experimental (APE) 1 y 2\n");
16     scanf(" %f, %f", &ape1 ,&ape2);
17
18     printf("Ingrese las notas del Aprendizaje Autonomo (AA) 1 y 2\n");
19     scanf(" %f, %f", &aa1, &aa2);
20
21     printf("Ingrese la nota del Portafolio (Evaluacion Sumativa 1)\n");
22     scanf(" %f", &portafolio);
23
24     printf("Ingrese la nota de la Evaluacion Final (Evaluacion Sumativa 2)\n");
25     scanf(" %f", &evaluacion);
26 }
```



```
27
28 //Proceso
29 acdPonderado = ((acd1+acd2)/2)*0.2;
30 apePonderado = ((ape1+ape2)/2)*0.25;
31 aaPonderado = ((aa1+aa2)/2)*0.2;
32 esPonderado = ((evaluacion*0.6) + (portafolio*0.4))*0.35;
33
34 notaFinal = acdPonderado + apePonderado + aaPonderado + esPonderado;
35
36 //Evaluación Cualitativa
37 if (notaFinal>=9){
38     escala = "Excelente";
39 } else if (notaFinal>=7 && notaFinal<9){
40     escala = "Bueno";
41 } else if (notaFinal>= 5 && notaFinal<7){
42     escala = "Regular";
43 } else if (notaFinal<5){
44     escala = "Deficiente";
45 }
46
47 //Datos de Salida
48 printf("\nSu nota cuantitativa es:\n");
49 printf("La nota ponderada del APE es de %.2f\n", apePonderado);
50 printf("La nota ponderada del ACD es de %.2f\n", acdPonderado);
51 printf("La nota ponderada del AA es de %.2f\n", aaPonderado);
52 printf("La nota ponderada del ES es de %.2f\n", esPonderado);
53 printf("La Nota Final de la Unidad 1 del estudiante es de %.2f/10\n\n", notaFinal);
54 printf("Su nota cualitativa es: %s", escala);
55
56 return 0;
57 }
```

5.4. Pruebas

```
PS C:\Users\ASUS\Downloads\UNL\1erCiclo\Teoria de la Programación\Unidad 2\Ejercicios C> gcc U2APE1.c -o U2APE1
PS C:\Users\ASUS\Downloads\UNL\1erCiclo\Teoria de la Programación\Unidad 2\Ejercicios C> ./U2APE1.exe
Ingrese las notas del Aprendizaje en Contacto con el Docente (ACD) 1 y 2
10,10
Ingrese las notas del Aprendizaje Practico Experimental (APE) 1 y 2
10,10
Ingrese las notas del Aprendizaje Autonomo (AA) 1 y 2
10,10
Ingrese la nota del Portafolio (Evaluacion Sumativa 1)
9
Ingrese la nota de la Evaluacion Final (Evaluacion Sumativa 2)
10
Su nota cuantitativa es:
La nota ponderada del APE es de 2.50
La nota ponderada del ACD es de 2.00
La nota ponderada del AA es de 2.00
La nota ponderada del ES es de 3.36
La Nota Final de la Unidad 1 del estudiante es de 9.86/10

Su nota cualitativa es: Excelente
```

6. Preguntas de Control

- ¿Qué es una condición y cuál es su función dentro de una estructura condicional en programación?**

Una condición es una expresión lógica que puede evaluarse como verdadera o falsa. Dentro de una estructura condicional en programación, su función es determinar si se deben ejecutar ciertas instrucciones. Es decir, permite que el programa tome decisiones y siga diferentes caminos según el resultado de la evaluación.



- **¿Qué diferencia existe entre una estructura condicional simple, doble y múltiple?**
 - Condicional simple: Utiliza únicamente la instrucción if. Si la condición se cumple (es verdadera), se ejecutan las instrucciones asociadas; de lo contrario, el programa continúa sin realizar ninguna otra acción.
 - Condicional doble: Emplea if y else. Si la condición del if no se cumple, el bloque else se ejecuta como alternativa, permitiendo manejar dos posibles escenarios.
 - Condicional múltiple: Incluye if, else if y else. Esta estructura permite evaluar varias condiciones de forma secuencial. Si la primera no se cumple, se verifica la siguiente, y así sucesivamente, hasta encontrar una condición verdadera o ejecutar el bloque else si ninguna se cumple.
- **¿Qué es una estructura condicional anidada y en qué casos resulta útil emplearla dentro de un programa?**

Una estructura condicional anidada es aquella en la que se incluye una condición dentro de otra. Es decir, dentro de un bloque if, else if o else, se coloca otra estructura condicional. Este tipo de estructura resulta útil cuando se necesita tomar decisiones más complejas que dependen de múltiples niveles de evaluación.

7. Conclusiones

- Se evidenció el dominio en el uso de estructuras condicionales simples, dobles y múltiples, aplicadas correctamente en el desarrollo de la práctica y evidenciando su utilidad en escenarios de la vida real.
- Se logró diseñar y codificar de manera efectiva el algoritmo propuesto, cumpliendo con los requerimientos del problema planteado y utilizando buenas prácticas de programación como la correcta tabulación y estructuración del código.
- La ejecución del programa con un caso real, basado en las notas del estudiante, permitió validar la lógica implementada y comprobar que el resultado obtenido coincide con el promedio esperado.

8. Recomendaciones

- Se sugiere reforzar el uso de estructuras condicionales anidadas mediante ejercicios adicionales, que permitan resolver problemas con múltiples niveles de decisión.
- Es conveniente realizar pruebas con diferentes casos reales, no solo con promedios altos, para verificar el correcto funcionamiento del programa en todos los rangos de evaluación.