Reporte Técnico de Actividades Práctico-Experimentales Nro. 001

1. Datos de Identificación del Estudiante y la Práctica

Nombre del estudiante(s)	José Valencia
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	20/11/2025
Horario	10h30 – 13h30
Lugar	Aula Asignada al paralelo
Tiempo planificado en el Sílabo	6 horas

2. Objetivo(s) 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

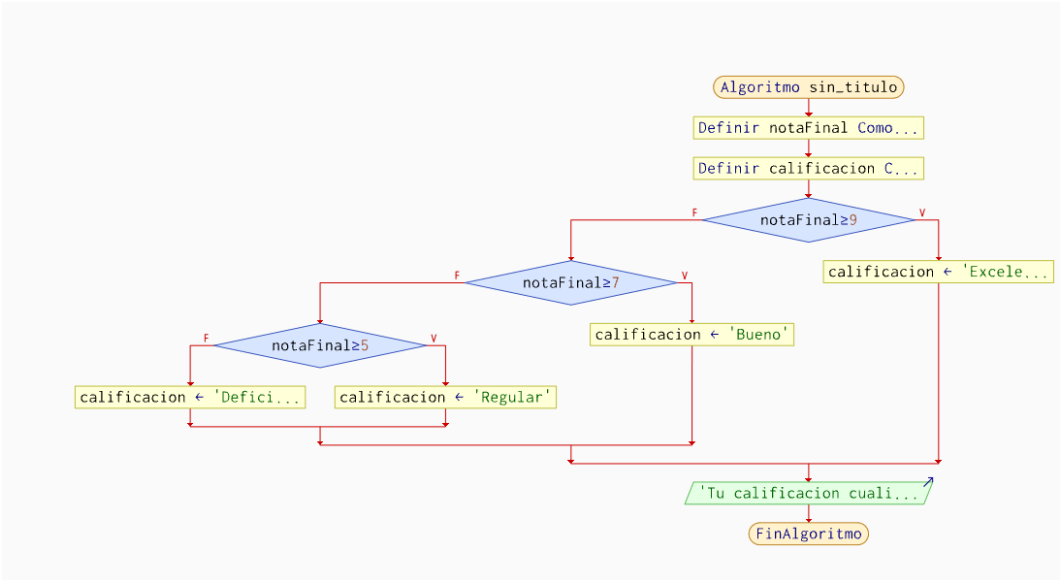
1) Análisis del Problema

Los pasos que se siguió son:

Estructurar la condición en un diagrama de flujo, completamos el programa en C en base al algoritmo ya realizado, y finalmente realizamos una prueba de escritorio con nuestras notas

5. Resultados

Diagrama de Flujo:



Codificación:

```c
#include <stdio.h>

int main(){

    //Definicion de Variables

    printf("Nota final de Teoria de la Programacion - Unidad 1\n\n");

    float acd1, acd2, aa1, aa2, ape1, ape2, es1, es2, notaAPE, notaACD, notaAA, notaES, notaFinal;
    char *calificacion;

    //Datos de Entrada

    printf("Aprendizaje en Contacto con el Docente (ACD)\n");

    printf("Introduce la calificacion de tu primer ACD (0-10): ");
    scanf("%f", &acd1);
    getchar();

    printf("Introduce la calificacion de tu segundo ACD (0-10): ");
    scanf("%f", &acd2);
    getchar();

    printf("Aprendizaje Autonomo (AA)\n");

    printf("Introduce la calificacion de tu primer AA (0-10): ");
    scanf("%f", &aa1);
    getchar();

    printf("Introduce la calificacion de tu segundo AA (0-10): ");
    scanf("%f", &aa2);
    getchar();

    printf("Aprendizaje Practico Experimental (APE)\n");

    printf("Introduce la calificacion de tu primer APE (0-10): ");
    scanf("%f", &ape1);
    getchar();

    printf("Introduce la calificacion de tu segundo APE (0-10): ");
    scanf("%f", &ape2);
    getchar();
```

```
    printf("Introduce la calificacion de tu segundo APE (0-10): ");
    scanf("%f", &ape2);
    getchar();

    printf("Evaluacion Sumativa (ES)\n");

    printf("Introduce la calificacion de tu primera ES (0-10): ");
    scanf("%f", &es1);
    getchar();

    printf("Introduce la calificacion de tu segunda ES (0-10): ");
    scanf("%f", &es2);
    getchar();

    //Proceso

    notaAPE = ((ape1 + ape2) / 2) * 0.25;
    notaACD = ((acd1 + acd2) / 2) * 0.2;
    notaAA = ((aa1 + aa2) / 2) * 0.2;
    notaES = ((es1 * 0.4) + (es2 * 0.6)) * 0.35;

    notaFinal = notaAPE + notaACD + notaAA + notaES;

    // Estructura del Condicional para calificacion cualitativa

    if (notaFinal >= 9) {
        calificacion = "Excelente";
    } else if (notaFinal < 9 && notaFinal >= 7) {
        calificacion = "Bueno";
    } else if (notaFinal < 7 && notaFinal >= 5) {
        calificacion = "Regular";
    } else {
        calificacion = "Deficiente";
    }

   //Salida

    printf("Su nota en el APE es de %.2f", notaAPE);
    printf("\nSu nota en el ACD es de %.2f", notaAPE);
    printf("\nSu nota en el AA es de %.2f", notaAA);
    printf("\nSu nota en el ES es de %.2f\n", notaES);
    printf("Tu calificacion final en la asignatura de Teoria de la Programacion es de: %.2f/10", notaFinal);

    printf("\nTu calificacion cualitativa es: %s\n", calificacion);


    return 0;
}
```

Prueba de Escritorio:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\Usuario iTC\Documents\io> gcc calculoUnidad1.c -o calculoUnidad1
PS C:\Users\Usuario iTC\Documents\io> .\calculoUnidad1.exe
Nota final de Teoria de la Programacion - Unidad 1

Aprendizaje en Contacto con el Docente (ACD)
Introduce la calificacion de tu primer ACD (0-10): 10
Introduce la calificacion de tu segundo ACD (0-10): 10
Aprendizaje Autonomo (AA)
Introduce la calificacion de tu primer AA (0-10): 10
Introduce la calificacion de tu segundo AA (0-10): 10
Aprendizaje Practico Experimental (APE)
Introduce la calificacion de tu primer APE (0-10): 8.5
Introduce la calificacion de tu segundo APE (0-10): 8.5
Introduce la calificacion de tu segundo APE (0-10): 8.5
Introduce la calificacion de tu segundo APE (0-10): 8.5
Evaluacion Sumativa (ES)
Introduce la calificacion de tu primera ES (0-10): 9.5
Introduce la calificacion de tu segunda ES (0-10): 9.5
Su nota en el APE es de 2.13
Su nota en el ACD es de 2.13
Su nota en el AA es de 2.00
Su nota en el ES es de 3.33
Tu calificacion final en la asignatura de Teoria de la Programacion es de: 9.45/10
Tu calificacion cualitativa es: Excelente
PS C:\Users\Usuario iTC\Documents\io>
```

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 ser verdadera o falsa, y se utiliza en programación para decidir qué camino debe seguir un programa. Su función es evaluar una situación específica para determinar si se cumple o no. Por ejemplo, si una persona tiene 18 años y desea saber si es mayor de edad, se formula una condición como: “si la edad es menor que 18, es menor de edad; de lo contrario, es mayor de edad”. De esta manera, la condición permite controlar el flujo del programa según el valor evaluado.

● ¿Qué diferencia existe entre una estructura condicional simple, doble y multiple?

La estructura condicional simple ejecuta un bloque de instrucciones únicamente cuando la condición es verdadera. En cambio, la estructura condicional doble (if–else) evalúa una condición y permite que el programa tome uno de dos caminos posibles: uno si es verdadera y otro si es falsa. Por último, la estructura condicional múltiple permite evaluar varias condiciones diferentes y seleccionar entre múltiples

alternativas, ya sea mediante varios else-if encadenados o usando estructuras como switch-case.

- **¿Qué es una estructura condicional anidada y en qué casos resulta útil emplearla dentro de un programa?**

Una estructura condicional anidada es aquella donde un condicional se encuentra dentro de otro condicional. Se utiliza cuando es necesario evaluar decisiones que dependen de resultados previos, es decir, situaciones jerárquicas en las que una condición conduce a otra más específica. Este tipo de estructura es útil cuando se requiere un mayor nivel de detalle en la toma de decisiones, como clasificar datos en categorías más precisas o validar varias reglas dentro de un mismo proceso.

7. Conclusiones

En resumen, comprender y aplicar las estructuras condicionales en programación es fundamental para desarrollar algoritmos capaces de tomar decisiones de manera lógica y ordenada. A través del uso de condiciones simples, dobles, múltiples y anidadas, es posible controlar el flujo de un programa según diferentes escenarios y valores ingresados por el usuario. Estas herramientas permiten que un software responda adecuadamente a cada situación, fortaleciendo la precisión, eficiencia y claridad del código.

8. Recomendaciones

Se recomienda profundizar en la práctica de estructuras condicionales mediante la resolución de problemas progresivamente más complejos. Esto permitirá fortalecer el pensamiento lógico, mejorar la capacidad de análisis y favorecer una programación más organizada y eficiente.

9. Declaracion de uso de IA

Para la elaboración de este trabajo utilicé herramientas de inteligencia artificial únicamente como apoyo para aclarar dudas, mejorar la redacción y complementar mi comprensión del tema. El contenido final ha sido revisado, adaptado y aprobado por mí.