



Reporte Técnico de Actividades Práctico-Experimentales Nro. 00X

1. Datos de Identificación del Estudiante y la Práctica

Nombre del estudiante(s)	Joaquin Emilio Moscol Castillo
Asignatura	Teoría de la programación
Ciclo	1 A
Unidad	1
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 o Grupal
Título de la Práctica	Aplicación de estructuras condicionales en la resolución de problemas.
Nombre del Docente	Lisette Geoconda López Faicán
Fecha	Jueves 13 de noviembre del 2025 Jueves 20 de noviembre 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 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).
- 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

Realizamos el diagrama de flujo de este código correctamente.

Hacemos este diagrama de flujo en el código de C

Una vez lo acabamos, realizamos las pruebas de escritorio y corroborar esto.

Una vez nos de los resultados esperados completamos el informe.

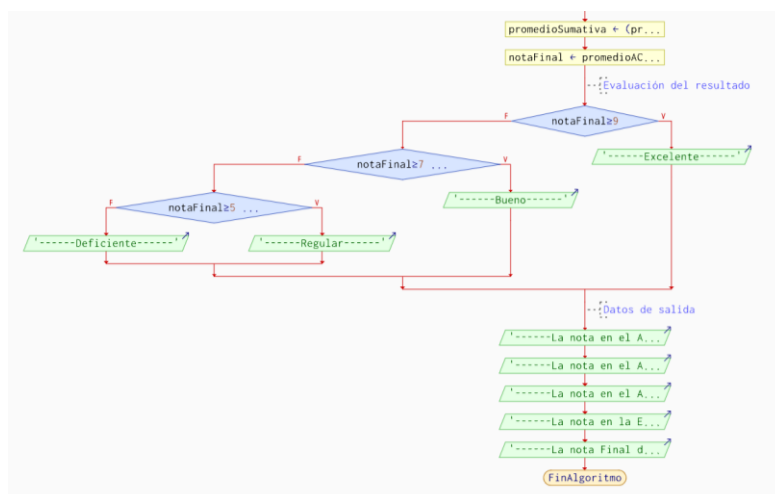
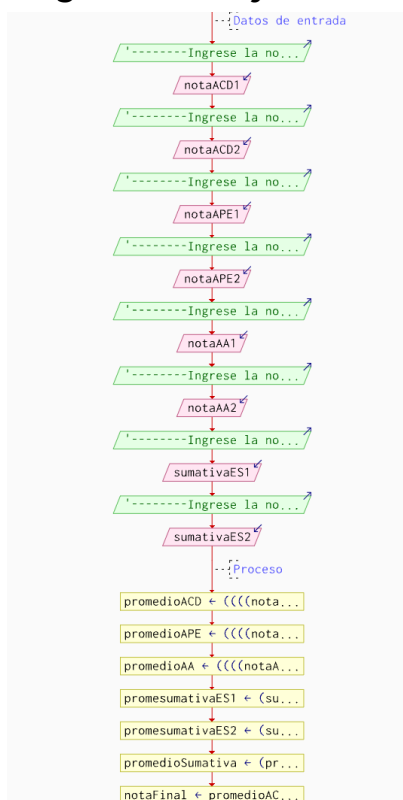
5. Resultados

Tenemos que crear un código el cual nos permita sacar nuestro promedio final de las Unidades, Ingresando las dos notas de ACD, APE, AA, y ES que ingresemos cada nota de las evaluaciones y multiplicar cada uno por lo que vale que es 0.4 y 0.6.

Luego que estas dos se promedien y se multipliquen por lo que vale cada una como $ACD * 0.2$, $APE * 0.25$, $AA * 0.2$ y $ES * 0.35$, y una vez obtenido estos resultados se sumen en una nota ya que ese numero es nuestra nota final.

También que en base a esta suma final se muestre un texto dependiendo de la nota obtenida, como si se obtiene 9 imprima "Excelente" y así sucesivamente, ≥ 7 y $< 9 \rightarrow$ "Bueno", ≥ 5 y $< 7 \rightarrow$ "Regular" y por ultimo $< 5 \rightarrow$ "Deficiente"

Diagrama de Flujo:





unl

Universidad
Nacional
de Loja

Código en C:

```
#include <stdio.h>

int main() {

    // Definición de variables
    double notaACD1, notaACD2, notaAPE1, notaAPE2, notaAA1, notaAA2;
    double sumativaES1, sumativaES2, notaFinal;
    double promedioACD, promedioAPE, promedioAA, promedioSumativa, promesumativaES1,
    promesumativaES2;
    double pondeACD = 0.02, pondeAPE = 0.025, pondeAA = 0.02, pondeES = 0.35;

    // Datos de entrada
    printf("-----Ingrese la nota del ACD Nro1:-----\n");
    scanf("%lf", &notaACD1);
    getchar();

    printf("-----Ingrese la nota del ACD Nro2:-----\n");
    scanf("%lf", &notaACD2);
    getchar();

    printf("-----Ingrese la nota del APE Nro1:-----\n");
    scanf("%lf", &notaAPE1);
    getchar();

    printf("-----Ingrese la nota del APE Nro2:-----\n");
    scanf("%lf", &notaAPE2);
    getchar();

    printf("-----Ingrese la nota del AA Nro1:-----\n");
    scanf("%lf", &notaAA1);
    getchar();

    printf("-----Ingrese la nota del AA Nro2:-----\n");
    scanf("%lf", &notaAA2);
    getchar();

    printf("-----Ingrese la nota de la Evaluacion Sumativa Nro1:-----\n");
    scanf("%lf", &sumativaES1);
    getchar();

    printf("-----Ingrese la nota de la Evaluacion Sumativa Nro2:-----\n");
    scanf("%lf", &sumativaES2);
    getchar();

    // Proceso
    promedioACD = (((notaACD1 + notaACD2) / 2) * 100) / 10 * pondeACD;
    promedioAPE = (((notaAPE1 + notaAPE2) / 2) * 100) / 10 * pondeAPE;
```



```
promedioAA = (((notaAA1 + notaAA2) / 2) * 100) / 10) * pondeAA;
promesumativaES1 = (sumativaES1 * 0.4);
promesumativaES2 = (sumativaES2 * 0.6);
promedioSumativa = (promesumativaES1 + promesumativaES2) * pondeES;
notaFinal = promedioACD + promedioAPE + promedioAA + promedioSumativa;

// Datos de salida
printf("-----La nota en el Aprendizaje en Contacto con el Docente (ACD) es:----- %f\n",
promedioACD);
printf("-----La nota en el Aprendizaje Práctico Experimental (APE) es:----- %f\n",
promedioAPE);
printf("-----La nota en el Autoaprendizaje (AA) es:----- %f\n", promedioAA);
printf("-----La nota en la Evaluacion Sumativa (ES) es:----- %f\n", promedioSumativa);
printf("-----La nota Final de la UNIDAD 1 es:----- %f\n", notaFinal);

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

getchar();
getchar();

return 0;
}
```

Prueba de Escritorio con Nuestras Propias Notas:

```
-----Ingrese la nota del ACD Nro1:-----
9
-----Ingrese la nota del ACD Nro2:-----
0
-----Ingrese la nota del APE Nro1:-----
9.75
-----Ingrese la nota del APE Nro2:-----
7
-----Ingrese la nota del AA Nro1:-----
0
-----Ingrese la nota del AA Nro2:-----
9
-----Ingrese la nota de la Evaluacion Sumativa Nro1:-----
7
-----Ingrese la nota de la Evaluacion Sumativa Nro2:-----
9
-----La nota en el Aprendizaje en Contacto con el Docente (ACD) es:----- 0.900000
-----La nota en el Aprendizaje Práctico Experimental (APE) es:----- 2.093750
-----La nota en el Autoaprendizaje (AA) es:----- 0.900000
-----La nota en la Evaluacion Sumativa (ES) es:----- 2.870000
-----La nota Final de la UNIDAD 1 es:----- 6.763750
-----Regular-----
```

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 o relacional que, al ser evaluada por el computador, da como resultado un valor de verdad (Verdadero o Falso). Su función principal es determinar el flujo de ejecución del programa; es decir, decide si un bloque específico de instrucciones se ejecuta o se omite, permitiendo que el software tome decisiones automáticas basándose en los datos de entrada.

- ¿Qué diferencia existe entre una estructura condicional simple, doble y múltiple?

Simple (if): Evalúa una condición y ejecuta un bloque de instrucciones solo si la condición es verdadera. Si es falsa, el programa continúa sin hacer nada específico respecto a esa estructura.

Doble (if-else): Evalúa una condición y ofrece dos caminos excluyentes: ejecuta un bloque de acciones si es verdadera y otro bloque distinto si es falsa.

Múltiple (if-else if, switch): Permite elegir entre tres o más caminos posibles. Evalúa varias condiciones sucesivamente o el valor de una variable para determinar cuál de las múltiples opciones ejecutar.

- ¿Qué es una estructura condicional anidada?

Una estructura condicional anidada consiste en colocar una sentencia condicional (if o else) dentro de otra estructura condicional principal. Resulta útil cuando se requiere evaluar condiciones secundarias que dependen de un requisito previo.

Ejemplo en la práctica: Fue necesaria para determinar el rango de la nota (Excelente, Bueno, Regular, Deficiente). No bastaba con saber si aprobó o no; dentro de las notas aprobatorias, tuvimos que "anidar" nuevas preguntas para saber si era mayor a 9 o menor a 7.

7. Conclusiones

Se logró implementar correctamente estructuras condicionales anidadas (if-else if-else) para clasificar la nota final del estudiante en categorías cualitativas, cumpliendo con el objetivo de la práctica. Se verificó la importancia de la jerarquía en el código; ubicar las condicionales después de los cálculos matemáticos es esencial para evaluar el dato final correcto.

8. Recomendaciones

Es recomendable validar que las notas ingresadas por el usuario estén dentro del rango de 0 a 10 antes de realizar los cálculos para evitar errores lógicos. Utilizar nombres de variables descriptivos y comentarios en el código (como se hizo en esta práctica) facilita la detección de errores en las fórmulas de ponderación.