



Reporte Técnico de Actividades Práctico-Experimentales Nro. 001

1. Datos de Identificación del Estudiante y la Práctica

Nombre del estudiante(s)	Javier Guarnizo y José Maldonado
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 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).

4. Procedimiento / Metodología Ejecutada

Metodología de aprendizaje: aprendizaje basado en problemas

Inicio

- Presentación del objetivo de la práctica.
 - Explicación de la relevancia de aplicar estructuras condicionales en la codificación de programas.
 - 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:
- $\text{Nota} \geq 9 \rightarrow$ “Excelente”
 - $\text{Nota} \geq 7 \text{ y } < 9 \rightarrow$ “Bueno”
 - $\text{Nota} \geq 5 \text{ y } < 7 \rightarrow$ “Regular”
 - $\text{Nota} < 5 \rightarrow$ “Deficiente”

Desarrollo

La solución del problema se desarrolla mediante el diseño de un algoritmo y su posterior implementación en un lenguaje de programación, empleando estructuras condicionales. Para ello, seguir los siguientes pasos:

1. **Análisis del problema:** identificar entradas, proceso y salidas.
2. **Diseño del algoritmo:** elaborar el algoritmo en diagrama de flujo con sus pruebas de escritorio para verificar su funcionamiento.
3. **Codificación:** trasladar la solución a un lenguaje de programación C.
4. **Pruebas:** compilar y ejecutar el programa en el IDE; verificar que los resultados sean correctos con el caso de prueba correspondiente a las notas reales del estudiante.
5. **Documentación:** elaborar informe PDF de acuerdo a la estructura del Reporte técnico.

Cierre

- Socialización de los resultados obtenidos.
- Retroalimentación docente sobre los aciertos y aspectos a mejorar en el diseño y codificación del algoritmo.

5. Resultados esperados

- Contextualización del problema.

El programa presentado tiene como propósito **calcular la nota final de la Unidad 1** de la materia de ciencias de la ingeniería de la computación. En este contexto académico, la calificación del estudiante no depende únicamente de un examen final, sino que se compone de **varios elementos de evaluación**, cada uno con distinto peso según el tipo de aprendizaje que representan.



El sistema de evaluación está estructurado en cuatro componentes fundamentales:

1. **Aprendizaje en Contacto con el Docente (ACD)**
Se refiere a actividades desarrolladas directamente con el profesor, como diseño de algoritmos y diseño de programas. Representa el **20%** de la calificación total.
2. **Aprendizaje Autónomo (AA)**
Agrupa actividades que el estudiante realiza por su cuenta, como el uso de herramientas digitales o la instalación de lenguajes de programación. También aporta el **20%** del total.
3. **Aprendizaje Práctico Experimental (APE)**
Incluye tareas de aplicación práctica, como construir algoritmos en pseudocódigo o elaborar programas en C. Este componente tiene un peso del **25%**.
4. **Evaluación Sumativa (ES)**
Consiste en productos evaluativos de mayor impacto, como un portafolio digital o una actividad de Aprendizaje Basado en Problemas (ABP). Se pondera internamente y aporta el **35%** de la nota final.

El programa solicita al estudiante ingresar sus calificaciones en cada una de estas actividades, calcula los promedios correspondientes y aplica los pesos establecidos por la malla curricular. Luego, suma los resultados para obtener la **nota final de la Unidad 1**, la cual es expresada sobre 10 puntos.

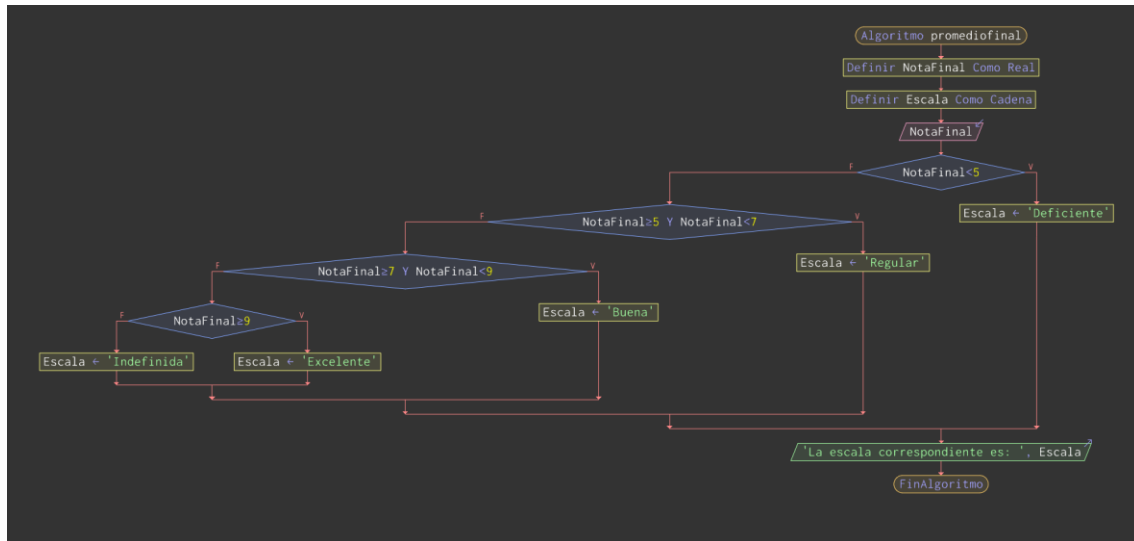
Finalmente, el sistema clasifica la nota obtenida en una categoría cualitativa:

- **Deficiente:** menos de 5
- **Regular:** entre 5 y menos de 7
- **Buena:** entre 7 y menos de 9
- **Excelente:** 9 o más

Esta clasificación permite al estudiante interpretar mejor su rendimiento académico.

En resumen, este programa automatiza el proceso de cálculo de calificaciones siguiendo el modelo pedagógico institucional, asegurando precisión y facilitando que el estudiante conozca de forma clara y rápida su desempeño global en la unidad de estudio.

o Diagrama de flujo.



o Código fuente en lenguaje C.

```

1  #include <stdio.h>
2  #include <math.h>
3
4  int main(){
5      //Definicion de variables
6      float AprendizajeContactoDocente, AprendizajeAutonomo, AprendizajePracticoExperimental, EvaluacionSumativa;
7      float NotaFinal;
8      float ACD1, ACD2, AA1, AA2, APE1, APE2, ES1, ES2;
9      char *Escala;
10
11     //Datos de entrada
12     printf("Notas de Aprendizaje en Contacto con el Docente\n");
13     printf("Ingrese su nota del ACD 1: Diseno de algoritmos\n");
14     scanf("%f", &ACD1);
15     printf("Ingrese su nota del ACD 2: Diseno de programas\n");
16     scanf("%f", &ACD2);
17
18     printf("Notas de Aprendizaje Autonomo\n");
19     printf("Ingrese su nota del AA 1: Herramientas digitales para algoritmos\n");
20     scanf("%f", &AA1);
21     printf("Ingrese su nota del AA 2: Instalacion de lenguajes de programacion\n");
22     scanf("%f", &AA2);
23
24     printf("Notas de Aprendizaje Practico Experimental\n");
25     printf("Ingrese su nota del APE 1: Algoritmos en pseudocodigo\n");
26     scanf("%f", &APE1);
27     printf("Ingrese su nota del APE 2: Construccion del programa en C\n");
28     scanf("%f", &APE2);
29
30     printf("Notas de la Evaluacion Sumativa\n");
31     printf("Ingrese su nota del ES 1: Portafolio digital\n");
32     scanf("%f", &ES1);
33     printf("Ingrese su nota del ES 2: Actividad de Aprendizaje Basado en Problemas\n");
34     scanf("%f", &ES2);
35
  
```

```
36 //Proceso
37
38 AprendizajeContactoDocente = ((ACD1 + ACD2) / 2) * 0.2;
39
40 AprendizajeAutonomo = ((AA1 + AA2) / 2) * 0.2;
41
42 AprendizajePracticoExperimental = ((APE1 + APE2) / 2) * 0.25;
43
44 EvaluacionSumativa = ((ES1 * 0.4) + (ES2 * 0.6)) * 0.35;
45
46 NotaFinal = AprendizajeContactoDocente + AprendizajeAutonomo + AprendizajePracticoExperimental + EvaluacionSumativa;
47
48 if (NotaFinal < 5) {
49     Escala = "Deficiente";
50 } else if (NotaFinal >= 5 && NotaFinal < 7) {
51     Escala = "Regular";
52 } else if (NotaFinal >= 7 && NotaFinal < 9) {
53     Escala = "Buena";
54 } else if (NotaFinal >= 9) {
55     Escala = "Excelente";
56 } else {
57     Escala = "Indefinida";
58 }
59
60 //Datos de salida
61
62 printf("Su nota de ACD es de %.2f, su nota de AA es de %.2f, su nota de APE es de %.2f, su nota de ES es de %.2f \n",
63        printf("Su nota final de la unidad 1 es de %.2f sobre 10 puntos\n", NotaFinal);
64        printf("La escala cualitativa es %s", Escala);
65
66
67 return 0;
68
69 }
```

o Pruebas.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SERIAL MONITOR

10
Ingrese su nota del APE 2: Construccion del programa en C
10
Notas de la Evaluacion Sumativa
Ingrese su nota del ES 1: Portafolio digital
10
Ingrese su nota del ES 2: Actividad de Aprendizaje Basado en Problemas
10
Su nota de ACD es de 2.00, su nota de AA es de 2.00, su nota de APE es de 2.50, su nota de ES es de 3.50
Su nota final de la unidad 1 es de 10.00 sobre 10 puntos
La escala cualitativa es Excelente

10
Ingrese su nota del APE 2: Construccion del programa en C
8
Notas de la Evaluacion Sumativa
Ingrese su nota del ES 1: Portafolio digital
7
Ingrese su nota del ES 2: Actividad de Aprendizaje Basado en Problemas
8.75
Su nota de ACD es de 2.00, su nota de AA es de 2.00, su nota de APE es de 2.25, su nota de ES es de 2.82
Su nota final de la unidad 1 es de 9.07 sobre 10 puntos
La escala 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 en programación es una expresión lógica que puede ser verdadera (true) o falsa (false).

Sirve para que el programa pueda tomar decisiones según el valor o situación que se esté evaluando.

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

Una estructura condicional simple evalúa una única condición y ejecuta una acción si se cumple; la doble ofrece dos caminos, uno si la condición es verdadera y otro si es falsa; mientras que la múltiple permite evaluar varias condiciones secuenciales, ofreciendo múltiples rutas de ejecución según el valor analizado. Estas estructuras permiten controlar el flujo del programa y tomar decisiones lógicas según diferentes escenarios

- **¿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 coloca un if o else dentro de otro if o else, permitiendo evaluar decisiones que dependen de una condición previa y creando niveles jerárquicos de lógica; es útil cuando un resultado solo tiene sentido si antes se cumple otra condición.

7. Conclusiones

- El programa demuestra un uso adecuado de condicionales múltiples (if – else - else) para determinar la escala cualitativa de la nota final. Gracias a estas estructuras, el sistema puede clasificar de manera precisa el rendimiento del estudiante según rangos establecidos, mostrando cómo los condicionales permiten tomar decisiones lógicas basadas en valores numéricos.
- La estructura condicional principal permite que, dependiendo del rango en el que se encuentre la nota final, el programa seleccione automáticamente una categoría como “Deficiente”, “Regular”, “Buena” o “Excelente”. Esto evidencia cómo los condicionales influyen directamente en el flujo final del programa, adaptando la salida según las condiciones evaluadas.
- La estructura condicional múltiple aplicada evita redundancias y permite una lectura clara de cómo se toma la decisión final. Esta organización demuestra la importancia de los condicionales para mantener un código legible, modular y fácil de mantener.

8. Recomendaciones

- Es importante diseñar primero el diagrama de flujo para visualizar claramente el proceso completo y asegurar que los rangos y cálculos estén correctamente estructurados antes de programarlos.



-
- También es fundamental validar las entradas del usuario para evitar notas fuera de rango que afecten los cálculos y la clasificación final.
 - Finalmente, mantener una estructura condicional organizada y coherente garantiza que el programa clasifique correctamente la nota final y facilita la lectura y el mantenimiento del código.

9. Declaración de IA

“Para la construcción de este informe contamos con el acompañamiento de ChatGPT, una herramienta que me sirvió como guía en la redacción, revisión y pulido de ideas. Su apoyo facilitó expresar con claridad los razonamientos técnicos y dar una presentación más ordenada y coherente al contenido desarrollado.”