

Reporte Técnico de Actividades Prácticas-Experimentales Nro. 001

1. Datos de Identificación del Estudiante y la Práctica

Nombre del estudiante(s)	Christopher Alexander Pineda Rodas
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 I
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	10:30 a 13:30
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 (Pseint, 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

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:
- $\text{Nota} \geq 9 \rightarrow \text{"Excelente"}$
- $\text{Nota} \geq 7 \text{ y } < 9 \rightarrow \text{"Bueno"}$
- $\text{Nota} \geq 5 \text{ y } < 7 \rightarrow \text{"Regular"}$
- $\text{Nota} < 5 \rightarrow \text{"Deficiente"}$

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, procesos 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.

Pasos Realizados:

1. Identifique las variables que debería de utilizar y qué estructura tendrá el algoritmo.
2. Realizar las fórmulas correspondientes para sacar las notas con sus resultados correctos.
3. Ejecutarlo y revisar que no haya fallas en su programación.
4. Realizar las correcciones correspondientes.

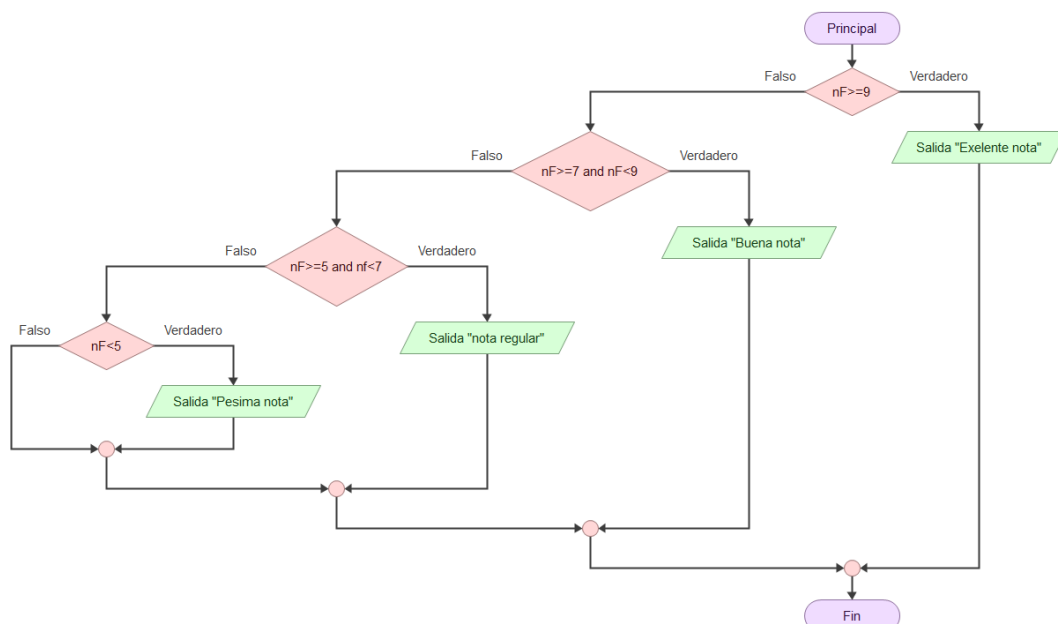
5. Resultados

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"

Diagrama de flujo:



Código:

```
C evaluacionunidad1.c > main()
1  #include <stdio.h>
2  #include <string.h>
3
4  int main (){
5
6      //Declaramos las variables
7      double aCD, acd1, acd2;
8      double aPE, ape1, ape2;
9      double aA, aa1, aa2;
10     double es, es1PortafolioVirtual, es2ActividadDeAprendizajeBasadoEnProblemas;
11     double nF;
12     char * escala;
13
14     //Ingreso de datos
15     printf("Ingrese la nota de la primera actividad de ACD: \n");
16     scanf("%lf", &acd1);
17
18     printf("Ingrese la nota de la segunda actividad de ACD: \n");
19     scanf("%lf", &acd2);
20
21     printf("Ingrese la nota de el primer deber de AA: \n");
22     scanf("%lf", &aa1);
23
24     printf("Ingrese la nota de el segundo deber de AA: \n");
25     scanf("%lf", &aa2);
26
27     printf("Ingrese la nota de la primera pactica de APE: \n");
28     scanf("%lf", &ape1);
29
30     printf("Ingrese la nota de la segunda pactica de APE: \n");
31     scanf("%lf", &ape2);
32
33     printf("Ingrese la nota de la la primera evaluacion de ES: \n");
34     scanf("%lf", &es1PortafolioVirtual);
35
36     printf("Ingrese la nota de la segunda evaluacion de ES: \n");
37     scanf("%lf", &es2ActividadDeAprendizajeBasadoEnProblemas);
38
```

```
39
40 //Proceso
41 aCD = ((acd1 + acd2)/2)*20/100;
42
43 aA = ((aa1 + aa2)/2)*20/100;
44
45 aPE = ((ape1 + ape2)/2)*25/100;
46
47 es1PortafolioVirtual = es1PortafolioVirtual*0.40;
48
49 es2ActividadDeAprendizajeBasadoEnProblemas = es2ActividadDeAprendizajeBasadoEnProblemas*0.60;
50
51 eS = ((es1PortafolioVirtual + es2ActividadDeAprendizajeBasadoEnProblemas)*35)/100;
52
53 nF = aCD + aA + aPE + eS;
54
55 //Datos de salida
56 printf("La nota de las actividades ACD es: \n%f\n", aCD);
57 printf("La nota de las tareas AA es: \n%f\n", aA);
58 printf("La nota de las practicas APE es: \n%f\n", aPE);
59 printf("La nota de las Evaluaciones ES es: \n%f\n", eS);
60 printf("La nota final de la Unidad 1 es: \n%f\n", nF);
61
62 if (nF >= 9)
63 {
64     escala="Excelente";
65 } else if (nF >= 7 && nF < 9)
66 {
67     escala="Buena";
68 } else if (nF >= 5 && nF < 7)
69 {
70     escala="Regular";
71 }else if (nF < 5)
72 {
73     escala="Deficiente";
74 }
75 printf("Su nota es: \n%s\n", escala);
76 }
```

Terminar:

```
PS C:\Users\Usuario ITC\Documents\Teoria de la programacion\ActividadesEnC> gcc evaluacionunidad1.c -o evaluacionunidad1
PS C:\Users\Usuario ITC\Documents\Teoria de la programacion\ActividadesEnC> ./evaluacionunidad1
Ingrese la nota de la primera actividad de ACD:
10
Ingrese la nota de la segunda actividad de ACD:
8
Ingrese la nota de el primer deber de AA:
9.5
Ingrese la nota de el segundo deber de AA:
8.5
Ingrese la nota de la primera pactica de APE:
7.5
Ingrese la nota de la segunda pactica de APE:
9
Ingrese la nota de la la primera evaluacion de ES:
9
Ingrese la nota de la segunda evaluacion de ES:
9
La nota de las actividades ACD es:
1.800000
La nota de las tareas AA es:
1.800000
La nota de las practicas APE es:
2.062500
La nota de las Evaluaciones ES es:
3.150000
La nota final de la Unidad 1 es:
8.812500
Su nota es:
Buena
```

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 verdadero o falso, trata de realizar la comparación de dos valores o variables, usando operadores como: <, >, ==, etc. Ayuda a determinar qué bloque de código toca seguir.

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

En la simple solo muestra su bloque de código si se cumple su condición.

En la doble Hay un bloque de código por si se cumple o no se cumpla la condición.

La múltiple permite realizar varias condiciones, donde se ejecuta la primera que se cumpla.

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

Una estructura condicional anidada pasa cuando una estructura condicional (simple, doble o múltiple) se coloca dentro del bloque de código que se ejecuta por otra estructura condicional.

Resulta útil cuando se deben examinar varias variables en modo de una jerarquía.

7. Conclusiones

- El aprendizaje sobre las estructuras condicionales, es una gran función que ayuda a resolver problemas complejos de programación.
- El ejercicio realizado ayudó a reafirmar mis conocimientos sobre el uso de las estructuras condicionales.

8. Recomendaciones

- Realizar más prácticas para mejorar la habilidad de los estudiantes en la resolución de problemas.