



Reporte Técnico de Actividades Práctico-Experimentales Nro. 002

1. Datos de Identificación del Estudiante y la Práctica

Nombre del estudiante(s)	Jeancarlos Fernando Aguirre Romero
Asignatura	Teoría de la programación
Ciclo	1 A
Unidad	Unidad: 1
Resultado de aprendizaje de la unidad	Identifica los conceptos fundamentales de la teoría de la programación, bajo los principios de solidaridad, transparencia, responsabilidad y honestidad.
Práctica Nro.	002
Tipo	Individual
Título de la Práctica	Del diseño del algoritmo con estructuras secuenciales a la construcción del programa.
Nombre del Docente	Lisette Geoconda López Faicán
Fecha	Martes 23 de octubre del 2025
Horario	10h30 – 13h30
Lugar	Aula física asignada al paralelo.
Tiempo planificado en el Sílabo	6 horas

2. Objetivo(s) de la Práctica

- Desarrollar la capacidad de transformar un problema en una solución computacional.
- Aplicar estructuras secuenciales en el diseño del algoritmo.
- Validar la lógica del algoritmo mediante pruebas de escritorio.
- Implementar y ejecutar la solución en un lenguaje de programación.

3. Materiales, Reactivos, Equipos y Herramientas

- Herramienta de pseudocódigo y diagramación de algoritmos: PSeInt.
- 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.

4. Procedimiento / Metodología Ejecutada

Metodología de aprendizaje: Aprendizaje basado en problemas.

- Presentación del objetivo de la práctica: explicación de los propósitos formativos de la actividad y la relevancia de aplicar estructuras secuenciales en el diseño de algoritmos.
- Contextualización del problema a resolver: explicación del enunciado que plantea la situación práctica.

El Problema Planteado para esta Actividad es el siguiente:

Un estudiante necesita saber qué calificación debe obtener en el tercer certamen (C3) para aprobar la asignatura con una nota final de 60/100 puntos.

Para calcular la nota final se utilizan las siguientes fórmulas:

1. Promedio de certámenes (NC):

$$NC = \frac{C1 + C2 + C3}{3}$$

2. Nota final del ciclo (NF):

$$NF = (NC \cdot 0.7) + (NL \cdot 0.3)$$

Para Realizar ese algoritmo lo primero que debemos hacer es reconocer los datos Involucrados dentro del algoritmo

- **Datos de Entrada:** certamen1, certamen2, notaLaboratorio,

- **Datos de Proceso:** certamen3, notaFinal
- **Datos de Salida:** certamen3

Una vez reconocidos los datos Utilizados dentro del código debemos Interpretar las formulas dadas para la resolución de este ejercicio uy a la conclusión que se llevo es que se debe despejar el certamen3 e incluir la formula del promedio de certámenes dentro de la formula del primer ciclo, la cual ya despejado y organizada se presenta de la siguiente forma:

$$\text{certamen3} = (((\text{notaFinal} - (\text{notaLaboratorio} * 0.3)) / 0.7) * 3) - (\text{certamen1} + \text{certamen2})$$

Ya con la formula Exacta determina lo que procederemos a realizar es realizar el código en PSeInt y Realizar las Pruebas de escritorio.

5. Resultados

Desarrollo del Algoritmo:

Algoritmo Calculo_de_notas

```
//Primero debemos declarar las Variables que ocuparemos
Definir certamen1, certamen2, notaLaboratorio, certamen3 Como Real
notaFinal = 60

//Luego le pediremos al usuario que ingrese los datos que necesitamos
Escribir 'Ingrese la nota del certamen 1:'
Leer certamen1
Escribir 'Ingrese la nota del certamen 2:'
Leer certamen2
Escribir 'Ingrese la nota de laboratorio:'
Leer notaLaboratorio

//Procedemos a realizar los respectivos cálculos con la formula
certamen3 = (((notaFinal - (notaLaboratorio * 0.3)) / 0.7 ) * 3) - (certamen1 +
certamen2)

//Presentamos los resultados en Pantalla
Escribir 'La nota mínima necesaria en el tercer certamen para aprobar es: ',
certamen3
```

FinAlgoritmo

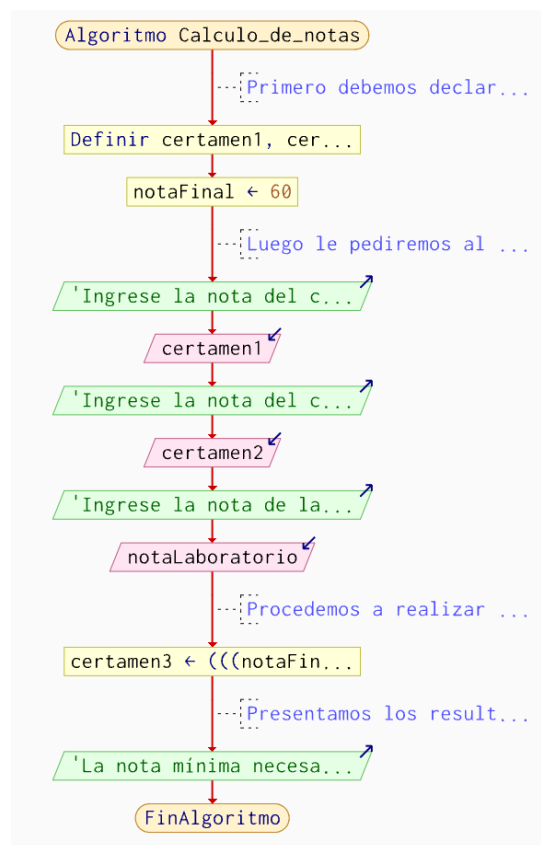
Desarrollo del Algoritmo la interfaz de PSeInt:

```

1  Algoritmo Calculo_de_notas
2
3      //Primero debemos declarar las Variables que ocuparemos
4      Definir certamen1, certamen2, notaLaboratorio, certamen3 Como Real
5      notaFinal = 60
6      //Luego le pediremos al usuario que ingrese los datos que necesitamos
7      Escribir 'Ingrese la nota del certamen 1:'
8      Leer certamen1
9      Escribir 'Ingrese la nota del certamen 2:'
10     Leer certamen2
11     Escribir 'Ingrese la nota de laboratorio:'
12     Leer notaLaboratorio
13     //Procedemos a realizar los respectivos calculos con la formula
14     certamen3 = (((notaFinal - (notaLaboratorio * 0.3)) / 0.7 ) * 3) - (certamen1 + certamen2)
15     //Presentamos los resultados en Pantalla
16     Escribir 'La nota minima necesaria en el tercer certamen para aprobar es: ', certamen3
17
18  FinAlgoritmo

```

Desarrollo del Algoritmo como diagrama de Flujo dentro de PSeInt:



Pruebas de escritorio Validando el Proceso y resultados:



certamen1	certamen2	notaLaboratorio	Formula	Certamen3
70	60	80	$\frac{((\text{notaFinal} - (\text{notaLaboratorio} * 0.3)) / 0.7) * 3}{(\text{certamen1} + \text{certamen2})}$	24.28
90	85	95	$\frac{((\text{notaFinal} - (\text{notaLaboratorio} * 0.3)) / 0.7) * 3}{(\text{certamen1} + \text{certamen2})}$	-40 (Ya puede aprobar)
40	55	60	$\frac{((\text{notaFinal} - (\text{notaLaboratorio} * 0.3)) / 0.7) * 3}{(\text{certamen1} + \text{certamen2})}$	85

Desarrollo del Algoritmo en el Lenguaje de programación C.

Ahora ya que hemos realizado el respectivo código en PSeInt procederemos a realizar el código en el lenguaje de Programación C.

```
#include <stdio.h>
int main() {
    float certamen1, certamen2, notaLaboratorio, certamen3;

    printf("Ingrese la nota del certamen 1: ");
    scanf("%f", &certamen1);
    printf("Ingrese la nota del certamen 2: ");
    scanf("%f", &certamen2);
    printf("Ingrese la nota de laboratorio: ");
    scanf("%f", &notaLaboratorio);

    certamen3 = ((60 - (notaLaboratorio * 0.3)) / 0.7 * 3) - (certamen1 + certamen2);
    printf("La nota mínima necesaria en el tercer certamen para aprobar es: %f", certamen3);
    return 0;
}
```

Desarrollo del Algoritmo en la interfaz de VS Code:

```
C: Calculo_de_Certamenes.c > ...
1  #include <stdio.h>
2
3  int main() {
4
5      float certamen1, certamen2, notaLaboratorio, certamen3;
6
7      printf("Ingrese la nota del certamen 1: ");
8      scanf("%f", &certamen1);
9      printf("Ingrese la nota del certamen 2: ");
10     scanf("%f", &certamen2);
11     printf("Ingrese la nota de laboratorio: ");
12     scanf("%f", &notaLaboratorio);
13
14     certamen3 = ((60 - (notaLaboratorio * 0.3)) / 0.7 * 3) - (certamen1 + certamen2);
15
16     printf("La nota mínima necesaria en el tercer certamen para aprobar es: %f", certamen3);
17     return 0;
18 }
19
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\USUARIO\Documents\Programacion\Lenguaje C> gcc Calculo_de_Certamenes.c -o Calculo_de_Certamenes
PS C:\Users\USUARIO\Documents\Programacion\Lenguaje C> gcc Calculo_de_Certamenes.c -o Calculo_de_Certamenes
PS C:\Users\USUARIO\Documents\Programacion\Lenguaje C> ./Calculo_de_Certamenes.exe
Ingrese la nota del certamen 1: 70
Ingrese la nota del certamen 2: 60
Ingrese la nota de laboratorio: 80
La nota mínima necesaria en el tercer certamen para aprobar es: 24.285715
PS C:\Users\USUARIO\Documents\Programacion\Lenguaje C> 
```

Como se puede ver en la Imagen el algoritmo funciona correctamente, además de ya estar compilado y guardado adecuadamente

6. Preguntas de Control

- **¿Qué elementos deben identificarse en el análisis de un problema computacional?**

En el análisis de un problema computacional se deben identificar claramente las entradas, los procesos y las salidas. De esta forma sabremos identificar todos los datos necesarios para la creación de los algoritmos de forma eficiente.

- **¿Por qué es importante validar un algoritmo mediante pruebas de escritorio?**

La validación mediante pruebas de escritorio permite detectar errores lógicos y verificar el funcionamiento del algoritmo antes de su implementación en un lenguaje de programación. Lo que es bastante útil para minimizar errores u otras cuestiones que presenten los algoritmos

- **¿Cómo se traslada un algoritmo en pseudocódigo a un lenguaje de programación?**

El traslado de un algoritmo en pseudocódigo a un lenguaje de programación se realiza manteniendo la estructura lógica original y adaptando las instrucciones a la sintaxis específica del lenguaje elegido. Por ejemplo, la sintaxis de PSeInt "LEER" en C se escribe `scanf` y no solo estas instrucciones sino otras normas o reglas que tienen los lenguajes de programación dentro de su estructura.

7. Conclusiones

La práctica permitió aplicar de forma efectiva los conceptos de estructuras secuenciales en la resolución de un problema real, reforzando la importancia del análisis previo, la validación mediante pruebas de escritorio y la correcta traducción del pseudocódigo al lenguaje C.

8. Recomendaciones

Como recomendación es necesario analizar cuidadosamente cada problema antes de programarlo, realizar pruebas de escritorio para verificar la lógica del algoritmo y documentar adecuadamente el código. Estas acciones aseguran una implementación más clara, eficiente y libre de errores.