



Reporte Técnico de Actividades Práctico-Experimentales Nro. 002

1. Datos de Identificación del Estudiante y la Práctica

Nombre del estudiante(s)	Domenica Narvaez
Asignatura	Teoría de la programación
Ciclo	1 A
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 28 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.



- 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.

Enunciado del problema:

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.

Pasos durante la práctica:

1) Análisis del problema:

Se identificaron las variables (C1, C2, C3 y NL), los datos de entrada (notas de certámenes C1, C2 y la nota de laboratorio NL), el proceso (cálculo de la nota necesaria en el tercer certamen C3) y la salida (resultado de la nota mínima para aprobar la asignatura con 60/100).

2) Diseño del algoritmo:

Se elaboró el pseudocódigo en Pseint, utilizando estructuras secuenciales y comentarios que indican las partes del algoritmo.

Posteriormente, se ejecutó el diagrama de flujo en Pseint, representando el pseudocódigo ya desarrollado anteriormente.

Finalmente, se realizaron las tres pruebas de escritorio solicitadas, con diferentes valores de entrada para verificar el funcionamiento correcto del algoritmo antes de su implementación.

3) Codificación:

La solución se trasladó al lenguaje de programación C, implementando la fórmula para el cálculo de C3 en el entorno de desarrollo Visual Studio Code. Se aplicaron buenas prácticas de programación y se incluyeron comentarios que indican las partes del código fuente desarrollado.

4) Pruebas de programa:

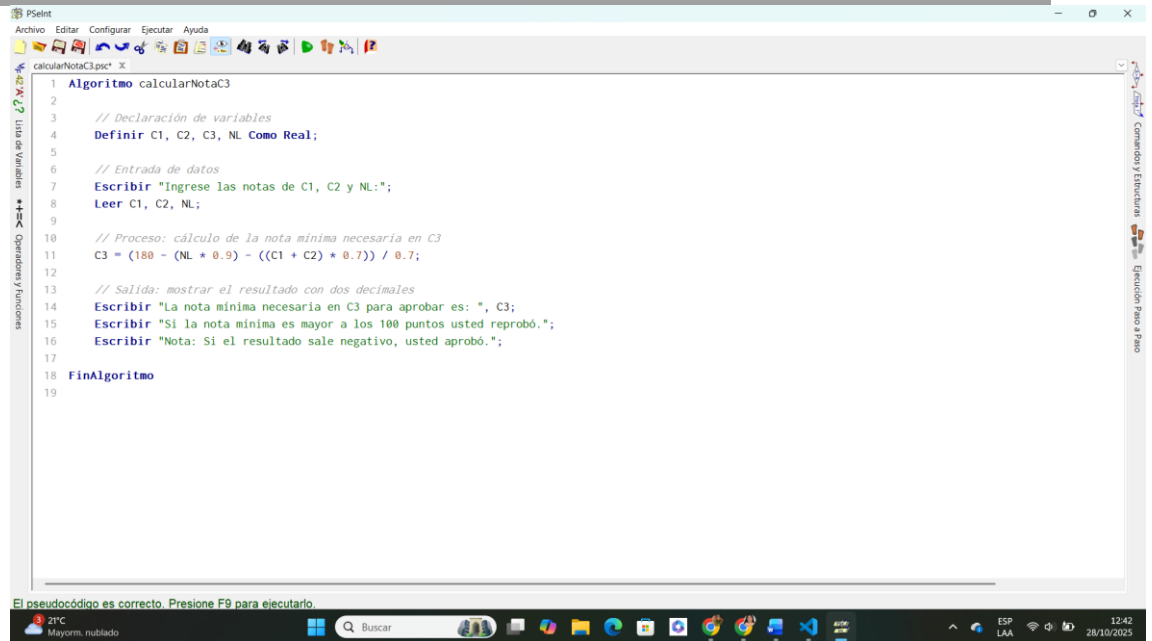
Se compiló y ejecutó el programa en el IDE, ingresando los mismos valores utilizados en las pruebas de escritorio para comprobar que los resultados fueran coherentes, precisos y certeros.

5) Documentación:

Se elaboró el informe técnico, que incluye análisis del problema, el diseño del algoritmo, la codificación, los resultados obtenidos de la práctica y las conclusiones sobre esta.

5. Resultados

- **Pseudocódigo en PSeInt**, correctamente estructurado y comentado, que refleje la solución planteada al problema.

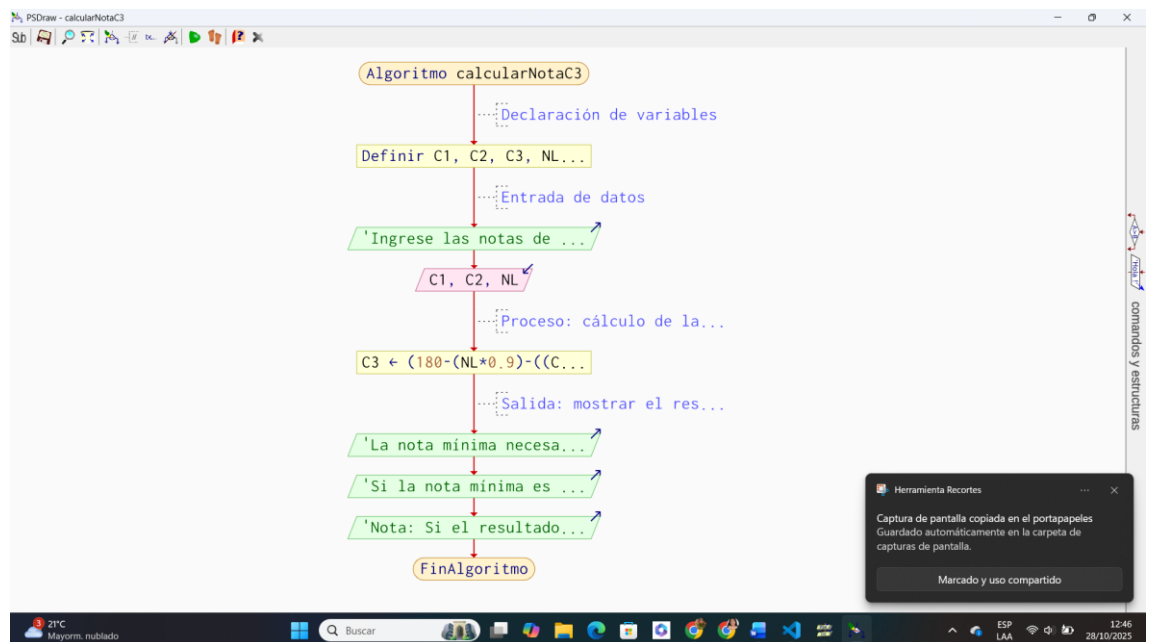


```

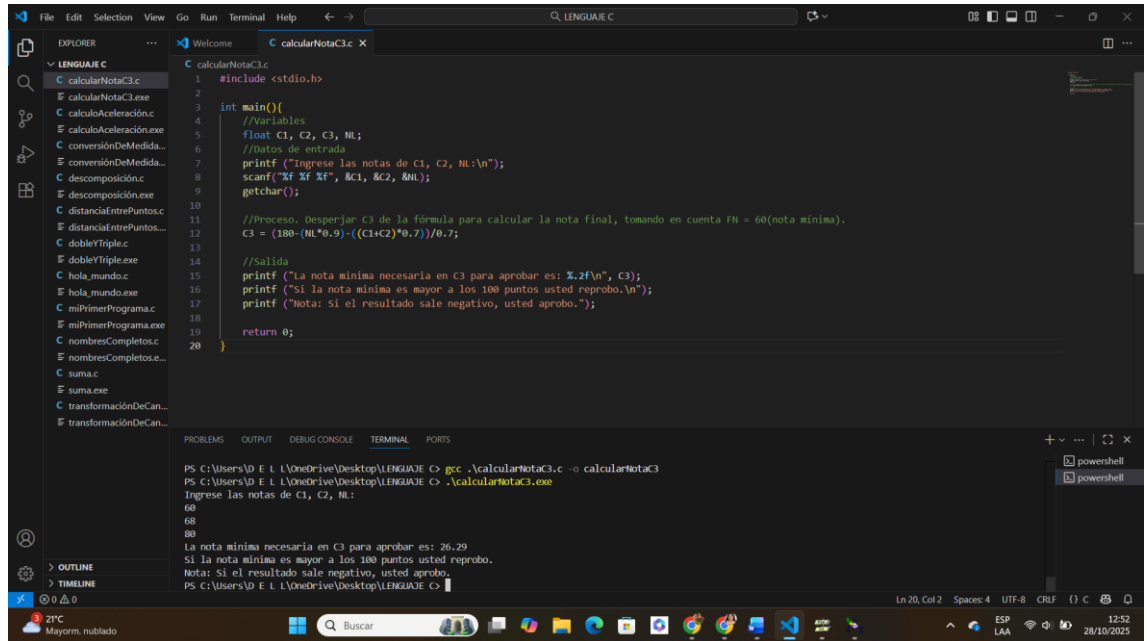
1 Algoritmo calcularNotaC3
2
3 // Declaración de variables
4 Definir C1, C2, C3, NL Como Real;
5
6 // Entrada de datos
7 Escribir "Ingrese las notas de C1, C2 y NL:";
8 Leer C1, C2, NL;
9
10 // Proceso: cálculo de la nota mínima necesaria en C3
11 C3 ← (180 - (NL * 0.9) - ((C1 + C2) * 0.7)) / 0.7;
12
13 // Salida: mostrar el resultado con dos decimales
14 Escribir "La nota mínima necesaria en C3 para aprobar es: ", C3;
15 Escribir "Si la nota mínima es mayor a los 100 puntos usted reprobó.";
16 Escribir "Nota: Si el resultado sale negativo, usted aprobó.";
17
18 FinAlgoritmo
19

```

- **Diagrama de flujo digital**, elaborado en una herramienta adecuada (PSeInt, Lucidchart, Draw.io u otra), con símbolos correctos y una lógica secuencial clara.



- **Código fuente en lenguaje C**, funcional y documentado con comentarios explicativos.



```

1 #include <stdio.h>
2
3 int main(){
4     //Variables
5     float C1, C2, C3, NL;
6     //Datos de entrada
7     printf("Ingrese las notas de C1, C2, NL:\n");
8     scanf("%f %f %f", &C1, &C2, &NL);
9     getchar();
10
11     //Proceso. Despejar C3 de la fórmula para calcular la nota final, tomando en cuenta FN = 60(nota mínima).
12     C3 = (180-(NL*0.9)-((C1+C2)*0.7))/0.7;
13
14     //Salida
15     printf("La nota mínima necesaria en C3 para aprobar es: %.2f\n", C3);
16     printf("Si la nota mínima es mayor a los 100 puntos usted reprobó.\n");
17     printf("Nota: Si el resultado sale negativo, usted aprobó.");
18
19     return 0;
20 }
    
```

```

PS C:\Users\D E L L\OneDrive\Desktop\LENGUAJE > gcc .\calcularNotaC3.c -o calcularNotaC3
PS C:\Users\D E L L\OneDrive\Desktop\LENGUAJE > .\calcularNotaC3.exe
Ingrese las notas de C1, C2, NL:
60
68
80
La nota mínima necesaria en C3 para aprobar es: 26.29
Si la nota mínima es mayor a los 100 puntos usted reprobó.
Nota: Si el resultado sale negativo, usted aprobó.
PS C:\Users\D E L L\OneDrive\Desktop\LENGUAJE >
    
```

• Pruebas de escritorio.

C1	C2	NL	Constante NC	Cálculo C3	Salida
60	68	80	60	$C3 = (180 - (80 \cdot 0.9) - ((60 + 68) \cdot 0.7)) / 0.7$	26.28
50	65	70	60	$C3 = (180 - (70 \cdot 0.9) - ((50 + 65) \cdot 0.7)) / 0.7$	52.14
40	50	80	60	$C3 = (180 - (80 \cdot 0.9) - ((40 + 50) \cdot 0.7)) / 0.7$	64.28

• Caso 1:

PSeInt:

PSeInt - Ejecutando proceso CALCULARNOTAC3

*** Ejecución Iniciada. ***

Ingrese las notas de C1, C2 y NL:

> 60

> 68

> 80

La nota mínima necesaria en C3 para aprobar es: 26.2857142857

Si la nota mínima es mayor a los 100 puntos usted reprobó.

Nota: Si el resultado sale negativo, usted aprobó.

*** Ejecución Finalizada. ***

C:

```
PS C:\Users\D E L L\OneDrive\Desktop\LENGUAJE > gcc .\calcularNotaC3.c -o calcularNotaC3
```

```
PS C:\Users\D E L L\OneDrive\Desktop\LENGUAJE > .\calcularNotaC3.exe
```

Ingrese las notas de C1, C2, NL:

60

68

80

La nota mínima necesaria en C3 para aprobar es: 26.29

Si la nota mínima es mayor a los 100 puntos usted reprobó.

Nota: Si el resultado sale negativo, usted aprobó.

```
PS C:\Users\D E L L\OneDrive\Desktop\LENGUAJE > |
```

• Caso 2:

PSeInt:

PSInt - Ejecutando proceso CALCULARNOTAC3

*** Ejecución Iniciada. ***

Ingrese las notas de C1, C2 y NL:

> 50

> 65

> 70

La nota mínima necesaria en C3 para aprobar es: 52.1428571429

Si la nota mínima es mayor a los 100 puntos usted reprobó.

Nota: Si el resultado sale negativo, usted aprobó.

*** Ejecución Finalizada. ***

C:

```
PS C:\Users\D E L L\OneDrive\Desktop\LENGUAJE C> gcc .\calcularNotaC3.c -o calcularNotaC3
```

```
PS C:\Users\D E L L\OneDrive\Desktop\LENGUAJE C> .\calcularNotaC3.exe
```

Ingrese las notas de C1, C2, NL:

50

65

70

La nota mínima necesaria en C3 para aprobar es: 52.14

Si la nota mínima es mayor a los 100 puntos usted reprobó.

Nota: Si el resultado sale negativo, usted aprobó.

```
PS C:\Users\D E L L\OneDrive\Desktop\LENGUAJE C> |
```

- **Caso 3:**

PSInt:

PSInt - Ejecutando proceso CALCULARNOTAC3

*** Ejecución Iniciada. ***

Ingrese las notas de C1, C2 y NL:

> 40

> 50

> 80

La nota mínima necesaria en C3 para aprobar es: 64.2857142857

Si la nota mínima es mayor a los 100 puntos usted reprobó.

Nota: Si el resultado sale negativo, usted aprobó.

*** Ejecución Finalizada. ***

C:

```
PS C:\Users\D E L L\OneDrive\Desktop\LENGUAJE C> gcc .\calcularNotaC3.c -o calcularNotaC3
```

```
PS C:\Users\D E L L\OneDrive\Desktop\LENGUAJE C> .\calcularNotaC3.exe
```

Ingrese las notas de C1, C2, NL:

40

50

80

La nota mínima necesaria en C3 para aprobar es: 64.29

Si la nota mínima es mayor a los 100 puntos usted reprobó.

Nota: Si el resultado sale negativo, usted aprobó.

```
PS C:\Users\D E L L\OneDrive\Desktop\LENGUAJE C> |
```

Microsoft Edge

6. Preguntas de Control

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

Se deben identificar 3 procesos:

Entradas: los datos que necesita el programa para funcionar.

Procesos: las operaciones o cálculos que se realizan con estos datos.

Salidas: los resultados que el programa debe mostrar

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

Validar un algoritmo con pruebas de escritorio es importante porque permite detectar errores lógicos antes de programar.

Al simular manualmente la ejecución con diferentes datos, se puede comprobar que el algoritmo sigue los pasos correctos y produce los resultados esperados, evitando fallos en la etapa de codificación.

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

El algoritmo en pseudocódigo se traslada a un lenguaje de programación siguiendo la misma lógica y estructura, pero usando la sintaxis específica del lenguaje (por ejemplo, C, Python o Java). Cada instrucción del pseudocódigo (leer, escribir, calcular, condicionales, etc.) se convierte en una instrucción real del lenguaje, respetando sus reglas de escritura y formato.

7. Conclusiones

- La práctica me permitió comprender el proceso de transformación de un problema real en una solución computacional, aplicando correctamente las estructuras secuenciales en el diseño del algoritmo.
- El uso de PSeInt facilitó la representación del algoritmo en pseudocódigo y diagrama de flujo, lo que ayudó a verificar la lógica del programa antes de codificarlo en lenguaje C.
- Al implementar el algoritmo en C y realizar pruebas de escritorio y de ejecución, se comprobó que el programa calcula correctamente la nota mínima necesaria en C3 y demuestra la importancia de validar cada etapa del desarrollo.

8. Recomendaciones

- Es importante realizar pruebas de escritorio para verificar que la función del algoritmo sea correcto.
- Realizar comentarios permite tener claro el proceso que se ejecuto a lo largo del algoritmo, ya que podemos olvidar y confundir lo que hicimos.

9. Anexos

No hay anexos para este informe.

10. Declaración de IA.

Declaro que en la elaboración de este informe se utilizó la herramienta ChatGPT (modelo GPT-5 de OpenAI) únicamente como apoyo educativo para la redacción, revisión gramatical y explicación de conceptos teóricos relacionados con la práctica.

El desarrollo del algoritmo, la codificación en lenguaje C, las pruebas y la interpretación de resultados fueron realizados de manera personal y responsable, garantizando la autenticidad del trabajo presentado.