Reporte Técnico de Actividades Práctico-Experimentales Nro. 00X

1. Datos de Identificación del Estudiante y la Práctica

Nombre del estudiante(s)	José Daniel Maldonado Rodriguez
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	Lissette Geoconda López Faicán
Fecha	Jueves 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).

4. Procedimiento / Metodología Ejecutada

Metodología de aprendizaje: aprendizaje basado en problemas.

Procedimiento:

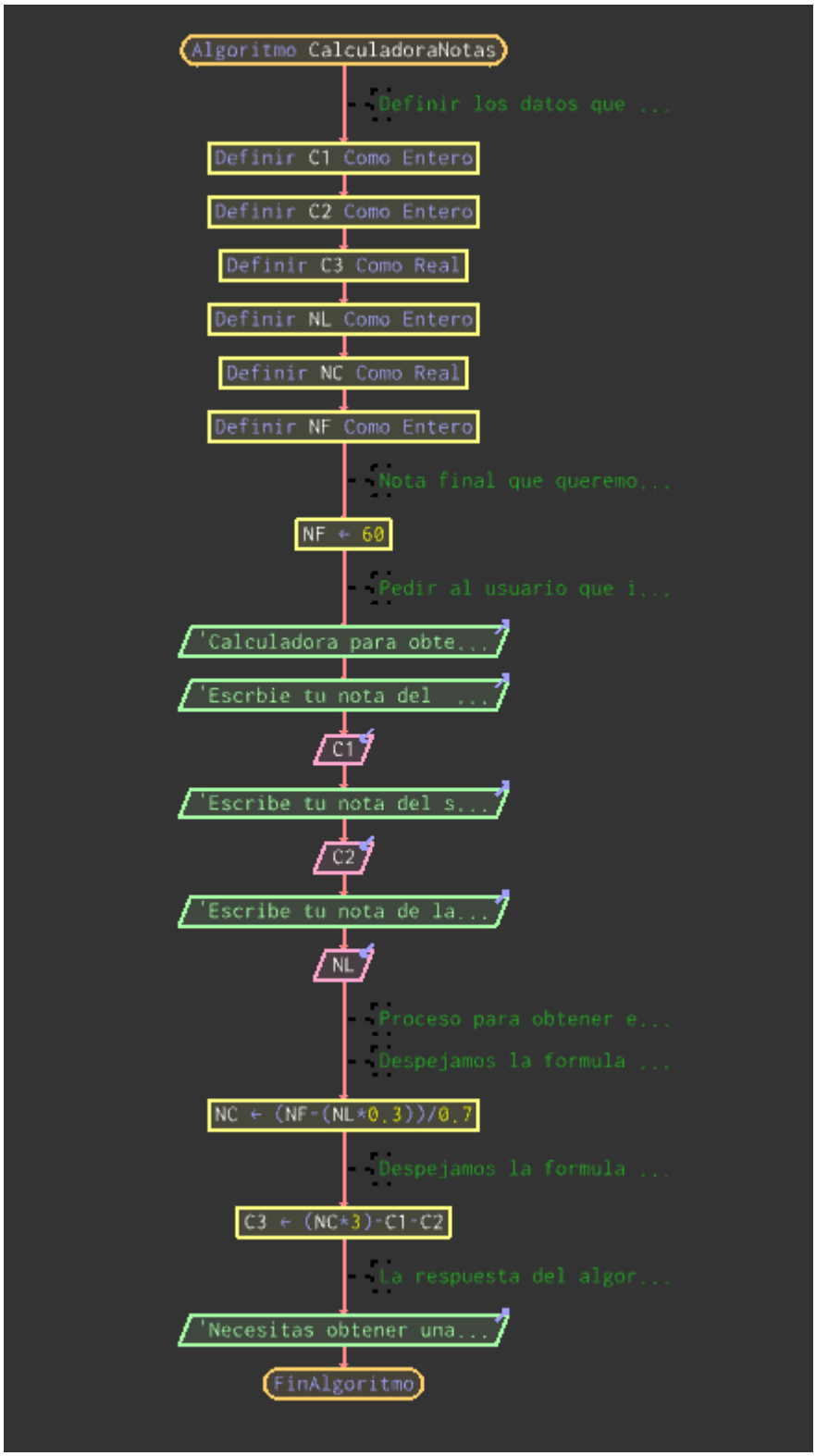
- Analizar el problema planteado en la actividad.
- Plantear un algoritmo con pseudocódigo en PSeInt para resolver el problema
- Diseñar el diagrama de flujo para explicar el algoritmo.
- Realizar pruebas de escritorio para comprobar la funcionalidad del algoritmo.
- Trasladar el algoritmo en pseudocódigo al lenguaje de programación C.

5. Resultados

Algoritmo en PseInt:

```
Algoritmo CalculadoraNotas
//Definir los datos que vamos a utilziar
Definir C1 Como Entero
Definir C2 Como Entero
Definir C3 Como Real
Definir NL Como Entero
Definir NC Como Real
Definir NF Como Entero

//Nota final que queremos conseguir (60/100)
NF = 60

//Pedir al usuario que ingrese los datos que necesita el algortimo para que se ejecute el codigo
Escribir "Calculadora para obtener el resultado de cuanto necesitas sacar en el tercer certamen para sacar una nota final de 60/100"

Escribir "Escrbie tu nota del  primer certamen"
Leer C1

Escribir "Escribe tu nota del segundo certamen"
Leer  C2

Escribir  "Escribe tu nota de laboratorio"
Leer NL

//Proceso para obtener el valor del certamen C3
//Despejamos la formula NF = (NC * 0.7) + (NL * 0.3)
NC = (NF - (NL * 0.3)) / 0.7

//Despejamos la formula NC = (C1 + C2 + C3) / 3
C3 = (NC*3) - C1 - C2

//La respuesta del algoritmo
Escribir "Necesitas obtener una puntuacion en el certamen 3 de: ", C3 " para pasar con una nota final de 60/100"

FinAlgoritmo
```

Diagrama de flujo:



Resultado:

```
PSeInt - Ejecutando proceso CALCULADORANOTAS
*** Ejecución Iniciada. ***
Calculadora para obtener el resultado de cuanto necesitas sacar en el tercer certamen para sacar una nota final de 60/100
Escrbie tu nota del  primer certamen
> 50
Escribe tu nota del segundo certamen
> 60
Escribe tu nota de laboratorio
> 100
Necesitas obtener una puntuacion en el certamen 3 de: 18.5714285714 para pasar con una nota final de 60/100
*** Ejecución Finalizada. ***
```

Programa en C:

CalculadoraNotas.c > main()

```c
#include <stdio.h>
//Calculadora para obtener el resultado de cuanto necesitas sacar en el tercer certamen para sacar una nota final de 60/100

//Definir los datos que vamos a utilziar
int main(){
int C1, C2, NL;
float C3, NC, NF;

//Nota final que queremos conseguir (60/100)
NF = 60;

//Pedir al usuario que ingrese los datos que necesita el algortimo para que se ejecute el codigo
printf("Escrbie tu nota del  primer certamen:\n");
scanf("%d", &C1);

printf("Escribe tu nota del segundo certamen:\n");
scanf("%d", &C2);

printf("Escribe tu nota de laboratorio:\n");
scanf("%d", &NL);

//Proceso para obtener el valor del certamen C3
//Despejamos la formula NF = (NC * 0.7) + (NL * 0.3)
NC = (NF - (NL * 0.3)) / 0.7;

//Despejamos la formula NC = (C1 + C2 + C3) / 3
C3 = (NC*3) - C1 - C2;

//La respuesta del algoritmo
printf("Necesitas obtener una puntuacion en el certamen 3 de: %f para pasar con una nota final de 60/100", C3);

return 0;
}
```

Resultado:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  PORTS  TERMINAL

PS C:\Users\josed\OneDrive\Documentos\UNL\TEORIA DE PROGRAMACIÓN\EJERCICIOS CON C> gcc CalculadoraNotas.c -o CalculadoraNotas
PS C:\Users\josed\OneDrive\Documentos\UNL\TEORIA DE PROGRAMACIÓN\EJERCICIOS CON C> ./CalculadoraNotas.exe
Escrbie tu nota del  primer certamen:
50
Escribe tu nota del segundo certamen:
60
Escribe tu nota de laboratorio:
100
Necesitas obtener una puntuacion en el certamen 3 de: 18.571430 para pasar con una nota final de 60/100
PS C:\Users\josed\OneDrive\Documentos\UNL\TEORIA DE PROGRAMACIÓN\EJERCICIOS CON C>
```

Pruebas de escritorio:

Nm de Prueba	Certamen 1	Certamen 2	Nota de laboratorio	Nota final	Promedio de los certámenes	Respuesta final (Certamen 3)
1	50	60	100	60	42.85	18.57
2	80	90	30	60	72.86	48.58
3	50	40	55	60	62.14	96.43

6. Preguntas de Control

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

Primero hay que identificar cuales son las variables del problema y que resultado queremos obtener. Posterior a eso hay que identificar que procesos debemos aplicar para resolver el problema y por ultimo los resultados o salida de datos.

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

Porque de esta forma se puede asegurar el correcto funcionamiento de nuestro algoritmo antes de programarlo en un lenguaje de programación.

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

Primero se empieza con la definición de las variables para posteriormente convertir las instrucciones del pseudocódigo en una sintaxis valida de un lenguaje de programación en este caso C. Por último, queda probar el programa y depurar errores.

7. Conclusiones

Mediante esta actividad se ha desarrollado la capacidad de poder transformar un problema en una solución computacional mediante la aplicación de estructuras secuenciales en el algoritmo atreves de la herramienta digital PSeInt.

También se aprendió a comprobar el funcionamiento del algoritmo mediante pruebas de escritorio y la implementación de este algoritmo al lenguaje de programación C.

Estas nuevas habilidades aprendidas permiten al estudiante comprender mejor problemas computaciones y como resolverlos atreves de la lógica.

8. Recomendaciones

- Analizar cuidadosamente el enunciado del problema antes de diseñar el algoritmo.

- Utilizar correctamente las estructuras secuenciales en el pseudocódigo, asegurando que los pasos sigan un orden lógico.
- Validar siempre el algoritmo mediante pruebas de escritorio antes de implementarlo en un lenguaje de programación, con el fin de detectar errores lógicos.
- Verificar las fórmulas y operaciones matemáticas con ejemplos reales para confirmar la exactitud de los resultados.
- Mantener una adecuada identificación y declaración de variables, empleando nombres significativos que faciliten la comprensión del código.
- Documentar y comentar el código durante su desarrollo, para facilitar futuras revisiones o mejoras.
- Probar el programa en diferentes casos de entrada, comprobando que funcione correctamente bajo distintas condiciones.