



Reporte Técnico de Actividades Práctico-Experimentales Nro. 002

1. Datos de Identificación del Estudiante y la Práctica

Nombre del estudiante(s)	Isaac Alejandro Vire Poma
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 324
Tiempo planificado en el Sílabo	6 horas



2. Objetivos 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. Procedimiento / Metodología Ejecutada

Comprensión del problema planteado para identificar variables, constantes, el proceso a llevar, etc. y proceder a realizar la solución en pseudocódigo, comprobándolo por prueba y error y de esa manera por pasarlo a un lenguaje de programación.

4. Resultados

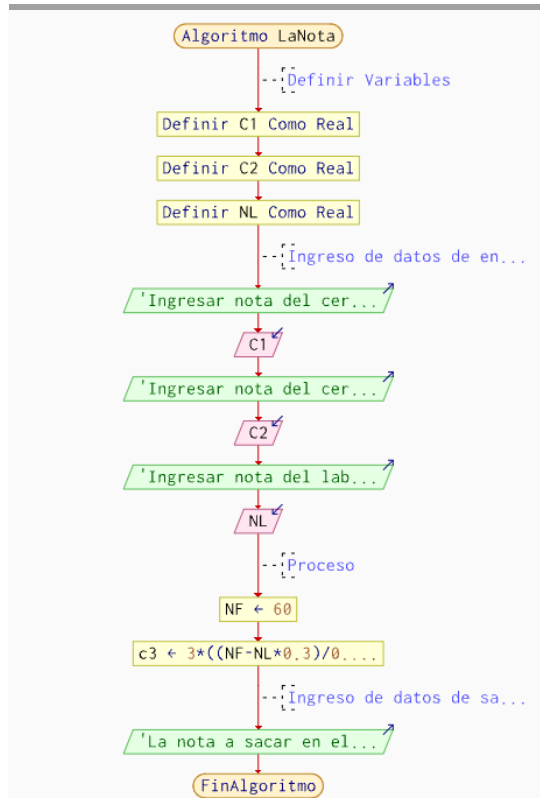
Planteado el problema se procedió a hacer el pseudocódigo en Pseint, dando el siguiente resultado:

Algoritmo LaNota

```
// Definir Variables
Definir C1 Como Real;
Definir C2 Como Real;
Definir NL Como Real;
// Ingreso de datos de entrada
Escribir 'Ingresar nota del certamen 1';
Leer C1;
Escribir 'Ingresar nota del certamen 2';
Leer C2;
Escribir 'Ingresar nota del laboratorio';
Leer NL;
// Proceso
NF = 60;
 $c3 = 3 * ((NF - NL * 0.3) / 0.7) - C1 - C2$ ;
// Ingreso de datos de salida
Escribir 'La nota a sacar en el certamen 3 es:', c3;
```

FinAlgoritmo

Y su correspondiente diagrama de flujo es:



Pruebas de escritorio

Son útiles para comprobar si nuestro código cumple con lo requerido. Se utilizarán tres casos para las pruebas.

Casos	Datos de entrada	Proceso	Datos de salida
1	c1= 80 c2= 90 NL= 70	$c3=3*((NF-NL*0.3)/0.7)-C1-C2$	-2.85
2	c1= 70 c2= 70 NL= 80	$c3=3*((NF-NL*0.3)/0.7)-C1-C2$	14.28
3	c1= 75 c2= 60 NL= 90	$c3=3*((NF-NL*0.3)/0.7)-C1-C2$	6.42

1.

```
PSelnt - Ejecutando proceso LANOTA
> 80
Ingresar nota del certamen 2
> 90
Ingresar nota del laboratorio
> 70
La nota a sacar en el certamente 3 es:-2.85714
-----
2.
```

```
PSelnt - Ejecutando proceso LANOTA
> 70
Ingresar nota del certamen 2
> 70
Ingresar nota del laboratorio
> 80
La nota a sacar en el certamente 3 es:14.28571
3.
```

```
PSelnt - Ejecutando proceso LANOTA
> 75
Ingresar nota del certamen 2
> 60
Ingresar nota del laboratorio
> 90
La nota a sacar en el certamente 3 es:6.428571
```

Algoritmo en C

Ahora se procede a traspasar el pseudocódigo a un lenguaje de programación.

```
#include <stdio.h>
int main(){
//Declaracion de variables
float c2, c1, NL, c3;
int NF;
// Ingreso de datos de entrada
printf("Ingresar nota del certamen 1 \n");
scanf("%f", &c1);
printf("Ingresar nota del certamen 2 \n");
scanf("%f", &c2);
printf("Ingresar nota del laboratorio \n");
scanf("%f", &NL);
```

```
//Proceso
NF= 60;
c3 = 3 * ((NF - NL * 0.3) / 0.7)-c1-c2;
// Ingreso de datos de salida
printf("La nota a sacar en el certamen 3 es: %.2f", c3);

return 0;
}
```

5. Preguntas de Control

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

Aquellos con los cuales se podrá dar paso a la creación de la solución, variables, constantes, formulas, etc. Esto ayudara a saber que es lo que se debe realizar y de manera.

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

Porque de esa manera se puede comprobar por mano propia si el algoritmo está cumpliendo su función correctamente.

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

Se transcriben los datos al lenguaje que el programa entienda, adaptándose a la manera que este tiene de entender las cosas, variables, constantes, signos, etc.

6. Conclusiones

El proceso de escribir estructuras secuenciales en programas básicos a escribirlas en lenguajes de programación es tardado y frustrante, pues hay muchas cosas que se tienen que considerar, palabras, bibliotecas, signos, etc. Y aun así, al final, como parte del proceso, ayuda a entrenar la mente y que pueda adaptarse más fácilmente a cambios en la programación, algo muy común dentro del rubro y a lo que hay que acostumbrarse.

7. Recomendaciones

Considerar leer con más atención los datos que se ofrecen en el problema a resolver, si no se conoce una fórmula, entenderla primero.