



# Reporte Técnico de Actividades Práctico-Experimentales Nro. 00X

## 1. Datos de Identificación del Estudiante y la Práctica

<b>Nombre del estudiante(s)</b>	Alison Micaela Tapia Morocho
<b>Asignatura</b>	Teoría de la programación
<b>Ciclo</b>	1 A
<b>Unidad</b>	1
<b>Resultado de aprendizaje de la unidad</b>	Identifica los conceptos fundamentales de la teoría de la programación, bajo los principios de solidaridad, transparencia, responsabilidad y honestidad.
<b>Práctica Nro.</b>	002
<b>Tipo</b>	Individual
<b>Título de la Práctica</b>	Del diseño del algoritmo con estructuras secuenciales a la construcción del programa.
<b>Nombre del Docente</b>	Lisette Geoconda López Faicán
<b>Fecha</b>	Martes 28 de octubre del 2025
<b>Horario</b>	10h30 – 13h30
<b>Lugar</b>	Aula física asignada al paralelo.
<b>Tiempo planificado en el Sílabo</b>	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

1. Se leyó la guía proporcionada por la docente.

2. Se realizó el análisis del problema, identificando los datos de entrada, las constantes y el valor a calcular.
3. Se determinó el proceso y las fórmulas a utilizar, realizando los respectivos despejes.
4. Se diseñó el algoritmo en PSeInt.
5. Se realizaron las pruebas de escritorio.
6. Se implementó el código en lenguaje C utilizando Visual Studio Code.
7. Se comprobó el correcto funcionamiento del código, verificando que los resultados coincidieran con los obtenidos en las pruebas de escritorio.

## 5. Resultados

### • Código PSeInt

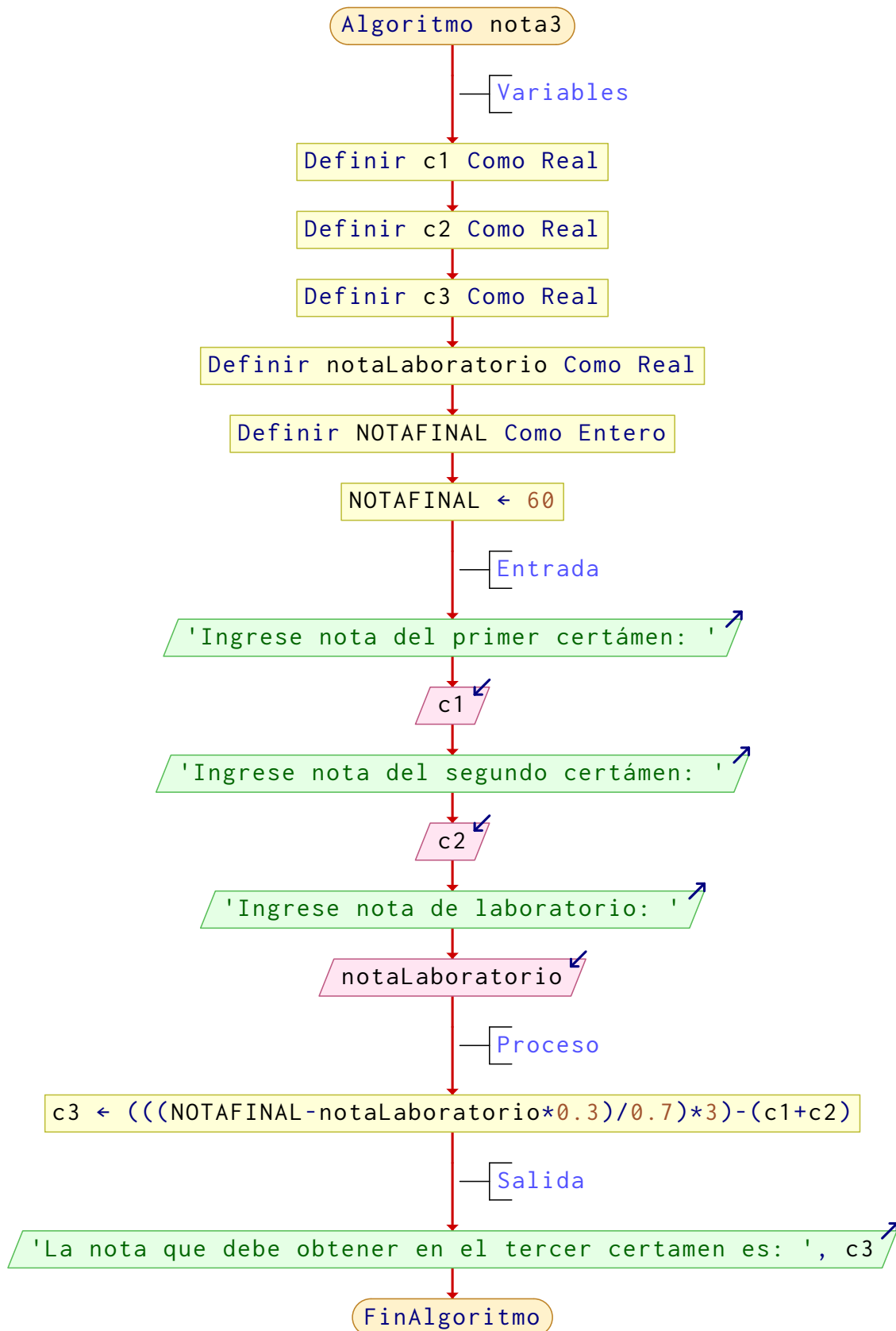
```

1  Algoritmo nota3
2      // Variables
3      Definir c1 Como Real;
4      Definir c2 Como Real;
5      Definir c3 Como Real;
6      Definir notaLaboratorio Como Real;
7      Definir NOTAFINAL como Entero;
8      NOTAFINAL = 60;
9
10     // Entrada
11     Escribir "Ingrese nota del primer certámen: ";
12     Leer c1;
13     Escribir "Ingrese nota del segundo certámen: ";
14     Leer c2;
15     Escribir "Ingrese nota de laboratorio: ";
16     Leer notaLaboratorio;
17
18     // Proceso
19     c3 = (((NOTAFINAL-notaLaboratorio*0.3)/0.7)*3) - (c1+c2);
20
21     // Salida
22     Escribir "La nota que debe obtener en el tercer certamen es: ", c3;
23 FinAlgoritmo
  
```

### • Pruebas de escritorio

C1	C2	NL	$c3 = (((60 - (NL * 0.3)) / 0.7) * 3) - (c1 + c2)$	Resultado
60	37	69	$c3 = (((60 - (80 * 0.3)) / 0.7) * 3) - (50 + 80)$	71.43
23	76	82	$c3 = (((60 - (90 * 0.3)) / 0.7) * 3) - (80 + 85)$	52.72
47	53	38	$c3 = (((60 - (40 * 0.3)) / 0.7) * 3) - (70 + 60)$	108.28

- Diagrama de flujo



- **Código en C**

```
C nota3.c > ...
1  #include <stdio.h>
2  int main (){
3      //Variables
4      float c1, c2, c3, notaLaboratorio;
5      int const NOTAFINAL = 60;
6      //Entarda
7      printf("Ingrese nota del primer certamen: ");
8      scanf("%f", &c1);
9      printf("Ingrese nota del segundo certamen: ");
10     scanf("%f", &c2);
11     printf("Ingrese nota de laboratorio: ");
12     scanf("%f", &notaLaboratorio);
13
14     //Proceso
15     c3 = (((NOTAFINAL - notaLaboratorio * 0.3)/0.7)*3) - (c1 + c2);
16
17     //Salida
18     printf("La nota que debe obtener en el tercer certamen es: %.2f", c3);
19 }
```

## 6. Preguntas de Control

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

En el análisis de un problema computacional deben identificarse los datos de entrada, las constantes, los procesos o fórmulas necesarias y los resultados o datos de salida.

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

La validación de un algoritmo mediante pruebas de escritorio es importante porque permite verificar el funcionamiento del algoritmo y la pronta detección de errores antes de la aplicación en el lenguaje de programación.

- **¿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 traduciendo las instrucciones lógicas y estructuras del algoritmo a la sintaxis propia del lenguaje de programación.

## 7. Conclusiones

Se desarrolló la capacidad de transformar un problema en una solución computacional, mediante el diseño de un algoritmo que aplica estructuras secuenciales, validando su funcionamiento a través de pruebas de escritorio e implementando posteriormente la solución en el lenguaje C.



---

## 8. Recomendaciones

Se recomienda identificar correctamente las variables de entrada, realizar el despeje adecuado de las fórmulas y efectuar pruebas de escritorio para verificar la exactitud de los resultados, además de comprobar que todas las variables hayan sido declaradas correctamente.