



Reporte Técnico de Actividades Práctico-Experimentales Nro. 00X

1. Datos de Identificación del Estudiante y la Práctica

Nombre del estudiante(s)	Christopher Alexander Pineda Rodas
Asignatura	Teoría de la programación
Ciclo	1A
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	28/10/2025
Horario	10:30 a 13:30
Lugar	Loja, Aula 324
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).



- 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. Inicio

- 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.

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):

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

Donde:

- C1 y C2 son las notas de los dos primeros certámenes.
- C3 es la nota del tercer certamen (la que se debe calcular).
- NL es la nota de laboratorio.
- NF es la nota final de la asignatura.

El programa debe permitir ingresar las notas de C1, C2 y NL; calcular automáticamente la nota mínima necesaria en C3 para que el estudiante apruebe la asignatura.

Nota: Si el resultado es negativo, significa que ya aprueba con las notas actuales.

Desarrollo:

La solución del problema se desarrolla mediante el diseño de un algoritmo y su posterior implementación en un lenguaje de programación, empleando exclusivamente estructuras secuenciales. Para ello, seguir los siguientes pasos:

1. Análisis del problema: identificar entradas, proceso y salidas.
2. Diseño del algoritmo:
 - Elaborar pseudocódigo en PSeInt con comentarios explicativos.
 - Diseñar el diagrama de flujo en una herramienta digital.
 - Definir pruebas de escritorio con al menos 3 casos y su validación con la ejecución del algoritmo.
3. Codificación: trasladar la solución a un lenguaje de programación C.
4. Pruebas: compilar y ejecutar el programa en el IDE; verificar que los resultados sean correctos con los mismos casos definidos en las pruebas de escritorio.



5. Documentación: elaborar informe PDF de acuerdo con la estructura del Reporte técnico. Cierre

- Socialización de los resultados obtenidos.
- Retroalimentación docente sobre los aciertos y aspectos a mejorar en el diseño y codificación del algoritmo

5. Resultados

Prueba de escritorio

Entrada	Necesitamos encontrar la nota c3 para que la nota final sea 60/100. Ingresar las variables que son calificaciones 1, 2 y la nota de laboratorio, la constante es NF=60.																										
Proceso	Usamos la formula: $NF = (NC \cdot 0.7) + (NL \cdot 0.3)$ Despejamos para que nos quede: $c3 = 3 * (NF - 0.3 * NL) / 0.7 - (c1 + c2)$																										
Salida	Nos tiene que dar la respuesta de c3																										
Ejemplos	<table border="1"><thead><tr><th colspan="4">Entrada</th><th>Proceso</th><th>Salida</th></tr><tr><th></th><th>c1</th><th>c2</th><th>nl</th><th>NF</th><th>c3</th></tr></thead><tbody><tr><td>1</td><td>90</td><td>75</td><td>50</td><td>60</td><td>$c3 = 3 * (60 - 0.3 * 50) / 0.7 - (90 + 75)$</td><td>27.857.143</td></tr><tr><td>2</td><td>45</td><td>30</td><td>10</td><td>60</td><td>$c3 = 3 * (60 - 0.3 * 10) / 0.7 - (45 + 30)$</td><td>La nota mínima necesaria es mayor a 100, lo cual es inválido</td></tr></tbody></table>	Entrada				Proceso	Salida		c1	c2	nl	NF	c3	1	90	75	50	60	$c3 = 3 * (60 - 0.3 * 50) / 0.7 - (90 + 75)$	27.857.143	2	45	30	10	60	$c3 = 3 * (60 - 0.3 * 10) / 0.7 - (45 + 30)$	La nota mínima necesaria es mayor a 100, lo cual es inválido
Entrada				Proceso	Salida																						
	c1	c2	nl	NF	c3																						
1	90	75	50	60	$c3 = 3 * (60 - 0.3 * 50) / 0.7 - (90 + 75)$	27.857.143																					
2	45	30	10	60	$c3 = 3 * (60 - 0.3 * 10) / 0.7 - (45 + 30)$	La nota mínima necesaria es mayor a 100, lo cual es inválido																					

Código de PSeInt:

```
EjercicioAPC2.psc X
1 Algoritmo EjercicioAPC2
2
3 // Declaración de variables
4 Definir c1, c2, c3, nl, NF Como Real
5
6 // Inicialización
7 NF ← 60 // Nota final deseada
8
9 // Entrada de datos
10 Escribir "Ingrese la calificación 1:"
11 Leer c1
12
13 Escribir "Ingrese la calificación 2:"
14 Leer c2
15
16 Escribir "Ingrese la nota de laboratorio:"
17 Leer nl
18
19 // Proceso: calcular la nota mínima necesaria en C3
20 c3 ← (3 * (NF - 0.3 * nl)) / 0.7 - (c1 + c2)
21
22 // Resultados con condiciones
23 Si c3 > 100 Entonces
24 | Escribir "La nota mínima necesaria es mayor a 100, lo cual es inválido."
25 Sino
26 | Escribir "La nota mínima necesaria en C3 es: ", c3
27 | Escribir "En caso de salir negativa, usted ya posee las notas requeridas."
28 FinSi
29
30 FinAlgoritmo
31
```

Diagrama de Flujo:

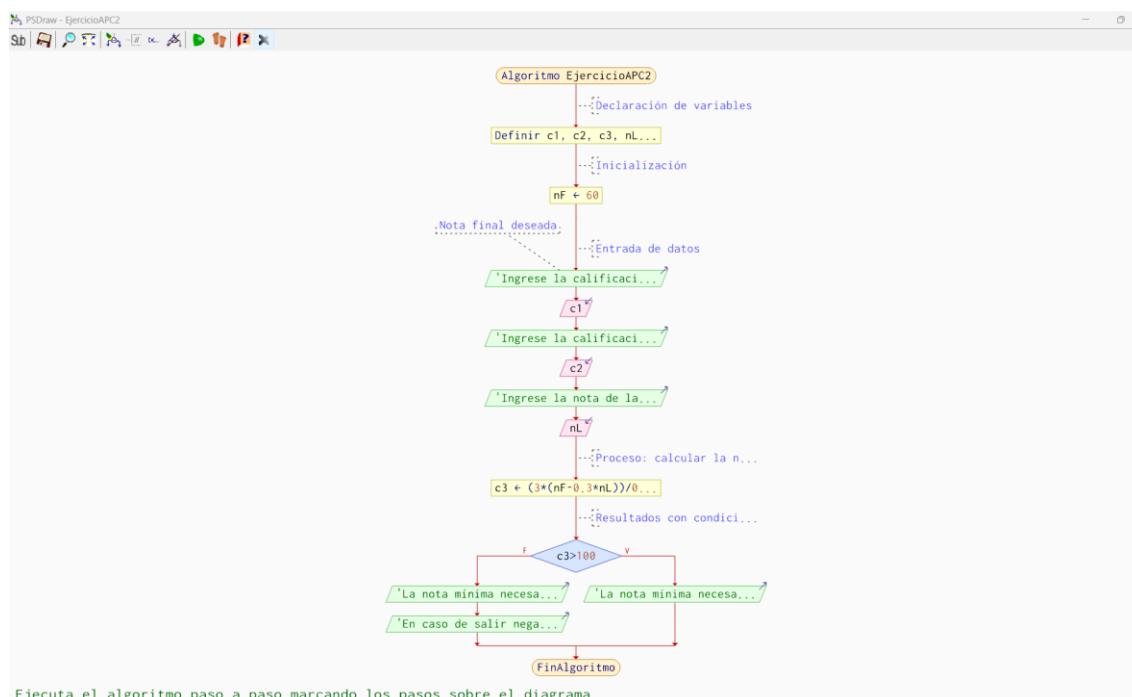


UNL

Universidad
Nacional
de Loja

1859

FEIRNNR - Carrera de Computación



Ejecuta el algoritmo paso a paso marcando los pasos sobre el diagrama.

Código de C

```
< ACTIVIDADES EN C
  C conversorDeMedidas...
  E conversorDeMedidas...
  C dobleTripeDeNumer...
  E dobleTripeDeNumer...
  C ejercicio1.c
  E ejercicio1.exe
  C ejercicio2.c
  E ejercicio2.exe
  C ejercicio3.c
  E ejercicio3.exe
  C ejercicio4.c
  E ejercicio4.exe
  C EjercicioAPC2.c
  E EjercicioAPC2.exe
  C holaMundo.c
  E holaMundo.exe
  C miPrimerPrograma.c
  E miPrimerPrograma.exe
  C segundoPrograma.c
  E segundoPrograma.exe
  C sumaDosNumeros.c
  E sumaDosNumeros.exe

C EjercicioAPC2.c > main()
1 #include <stdio.h>
2 #include <string.h>
3
4 int main (){
5
6     //Declaramos las variables
7     double c1, c2, c3, nL, nF=60;
8
9     //Ingreso de datos
10    printf("Ingrese la calificacion 1 : \n");
11    scanf("%lf", &c1);
12
13    printf("Ingrese la calificacion 2 : \n");
14    scanf("%lf", &c2);
15
16    printf("Ingrese la nota de laboratorio: \n");
17    scanf("%lf", &nL);
18
19    //Procesos
20    c3 = 3*(nF-0.3*nL)/0.7-(c1+c2);
21
22
23    //Resultado
24    if (c3>100)
25    {
26        printf("La nota mínima necesaria es mayor a 100, lo cual es invalido.\n");
27    } else if (c3<=100) {
28        printf("La nota mínima necesaria es: %lf\n", c3);
29        printf("En caso de salir negativo usted ya posee las notas requeridas\n");
30    }
31
32
33    return 0;
```

Terminar



```
PS C:\Users\Usuario iTC\Documents\Teoria de la programacion\Actividades en C> .\EjercicioAPC2
Ingrese la calificacion 1 :
90
Ingrese la calificacion 2 :
75
Ingrese la nota de laboratorio:
50
La nota minima necesaria es:
27.857143
En caso de ser Negativo usted ya posee las notas requeridas
PS C:\Users\Usuario iTC\Documents\Teoria de la programacion\Actividades en C> .\EjercicioAPC2
Ingrese la calificacion 1 :
45
Ingrese la calificacion 2 :
30
Ingrese la nota de laboratorio:
10
La nota minima necesaria es mayor a 100, lo cual es invalido.
PS C:\Users\Usuario iTC\Documents\Teoria de la programacion\Actividades en C> []
```

6. Preguntas de Control

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

Se debe identificar el problema a solucionar, los datos de entrada, que soluciones se pueden implementar.

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

Facilita encontrar los posibles problemas antes de ejecutar el algoritmo, comprender de mejor manera el algoritmo, además de verificar que si resuelva el problema.

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

Se debe entender el nuevo lenguaje para poder cambiar correctamente las funciones y que no haya errores, para mantener la misma metodología de solución.

7. Conclusiones

- Realizar prácticas con problemas complejos ayuda a desarrollar el pensamiento analítico de los estudiantes, y mejorando su criterio para futuros problemas.
- El uso de las pruebas de escritorio ayuda a una mejor resolución al momento de hacer algoritmo.

8. Recomendaciones

- Realizar siempre las pruebas de escritorio para que no haya complicaciones al realizar el algoritmo
- Si no se puede avanzar en el proceso de algoritmo, descansar ayudara a que lo veas desde otro ángulo para poder resolverlo.

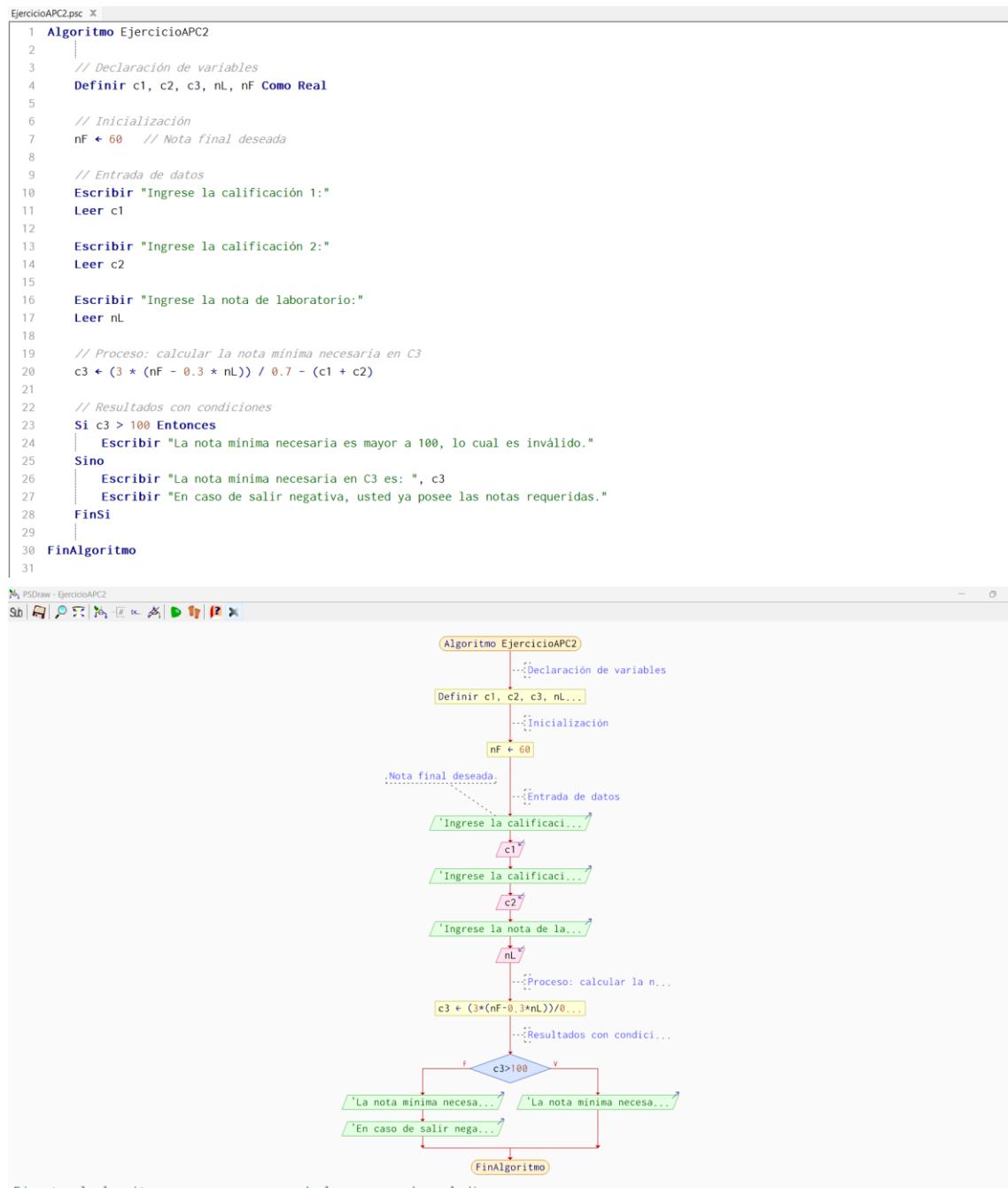


9. Bibliografía / Referencias

- [1] M. M. Arteaga Martínez, Lógica de programación con PSeInt. Enfoque práctico, 1st ed. Medellín, Colombia: Fondo Editorial Remington, 2023. [Online]. Available: https://research.ebsco.com/linkprocessor/plink?id=0c1115b8-e552-38e4-bc75-bf84_bbdd293f
- [2] M. Goin, Caminando junto al Lenguaje C. Río Negro, Argentina: Editorial UNRN, 2022. [Online]. Available: https://editorial.unrn.edu.ar/index.php/catalogo/346/view.bl/62/lecturas-de-catedra/26/caminando-junto-al-lenguaje-c?tab=getmybooksTab&is_show_data=1
- [3] J. E. Guerra Salazar, M. V. Ramos Valencia, and G. E. Vallejo Vallejo, Programando en C desde la práctica: problemas resueltos. Puerto Madero: Puerto Madero Editorial, 2023. [Online]. Available: <https://dialnet.unirioja.es/servlet/libro?codigo=933288>



10. Anexos





UNL

Universidad
Nacional
de Loja

1859

FEIRNNR - Carrera de Computación

```
ACTIVIDADES EN C
C conversorDeMedidas...
E conversorDeMedidas...
C dobleTripeDeNumer...
E dobleYTripeDeNumer...
C ejercicio1.c
E ejercicio1.exe
C ejercicio2.c
E ejercicio2.exe
C ejercicio3.c
E ejercicio3.exe
C ejercicio4.c
E ejercicio4.exe
C EjercicioAPC2.c
E EjercicioAPC2.exe
C holaMundo.c
E holaMundo.exe
C miPrimerPrograma.c
E miPrimerPrograma.exe
C segundoPrograma.c
E segundoPrograma.exe
C sumaDosNumeros.c
E sumaDosNumeros.exe

C EjercicioAPC2.c > ⚡ main()
1 #include <stdio.h>
2 #include <string.h>
3
4 int main (){
5
6     //Declaramos las variables
7     double c1, c2, c3, nL, nF=60;
8
9     //Ingreso de datos
10    printf("Ingrese la calificacion 1 : \n");
11    scanf("%lf", &c1);
12
13    printf("Ingrese la calificacion 2 : \n");
14    scanf("%lf", &c2);
15
16    printf("Ingrese la nota de laboratorio: \n");
17    scanf("%lf", &nL);
18
19    //Procesos
20    c3 = 3*(nF-0.3*nL)/0.7-(c1+c2);
21
22
23    //Resultado
24    if (c3>100)
25    {
26        printf("La nota minima necesaria es mayor a 100, lo cual es invalido.\n");
27    } else if (c3<=100) {
28        printf("La nota minima necesaria es: \n%lf\n", c3);
29        printf("En caso de salir negativo usted ya posee las notas requeridas\n");
30    }
31}
```

```
PS C:\Users\Usuario iTC\Documents\Teoria de la programacion\Actividades en C> .\EjercicioAPC2
Ingrese la calificacion 1 :
90
Ingrese la calificacion 2 :
75
Ingrese la nota de laboratorio:
50
La nota minima necesaria es:
27.857143
En caso de ser Negativo usted ya posee las notas requeridas
PS C:\Users\Usuario iTC\Documents\Teoria de la programacion\Actividades en C> .\EjercicioAPC2
Ingrese la calificacion 1 :
45
Ingrese la calificacion 2 :
30
Ingrese la nota de laboratorio:
10
La nota minima necesaria es mayor a 100, lo cual es invalido.
PS C:\Users\Usuario iTC\Documents\Teoria de la programacion\Actividades en C> █
```