



Guía de Actividades Práctico-Experimentales Nro. 002

Asignatura: Teoría de la programación
Ciclo: 1 A
Unidad: 1
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
Horario: 10h30 – 13h30

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.

1. Análisis del problema: identificar entradas, proceso y salidas.

Entrada: C1, C2, NL

Proceso: $C3 = (((60 - (NL * 0.3)) / 0.7) * 3) - (C1 + C2)$

Salida: C3

2. Diseño del algoritmo:

```
<sin_titulo>* X
1  Algoritmo clase
2      Definir C1, C2, NL, C3 como real
3
4      Escribir "Ingrese la nota del primer certamen (C1):"
5      Leer C1
6
7      Escribir "Ingrese la nota del segundo certamen (C2):"
8      Leer C2
9
10     Escribir "Ingrese la nota del laboratorio (NL):"
11     Leer NL
12
13     // Cálculo de la nota mínima en C3
14     // Fórmula:  $C3 = (((60 - (NL * 0.3)) / 0.7) * 3) - (C1 + C2)$ 
15      $C3 = (((60 - (NL * 0.3)) / 0.7) * 3) - (C1 + C2)$ 
16      $NF = Redon(NF * 10) / 10$ 
17
18
19     Escribir "La nota mínima necesaria en el tercer certamen (C3) es: ", C3
20     Escribir "Si la nota es negativa, aprueba con las notas actuales "
21     Escribir "Si no lo es, No aprueba la asignatura"
22
23
24
25  Fin Algoritmo
```



Pruebas de Escritorio (Pseint)

Caso	C1	C2	NL	Resultado Esperado	Resultado Obtenido
1	50	60	55	76.42857143	76.4285714286
2	70	80	75	10.71428571	10.7142857143
3	100	100	80	-45.7142857	-45.7142857143

```
PSeInt - Ejecutando proceso CLASE
Ingrese la nota del laboratorio (NL):
> 55
La nota mínima necesaria en el tercer certamen (C3) es: 76.4285714286
Si la nota es negativa, aprueba con las notas actuales
Si no lo es, No aprueba la asignatura
*** Ejecución Finalizada. ***
☐ No cerrar esta ventana ☐ Siempre visible 
```

Caso 1

```
PSeInt - Ejecutando proceso CLASE
Ingrese la nota del laboratorio (NL):
> 75
La nota mínima necesaria en el tercer certamen (C3) es: 10.7142857143
Si la nota es negativa, aprueba con las notas actuales
Si no lo es, No aprueba la asignatura
*** Ejecución Finalizada. ***
☐ No cerrar esta ventana ☐ Siempre visible 
```

Caso 2

```
PSeInt - Ejecutando proceso CLASE
Ingrese la nota del laboratorio (NL):
> 80
La nota mínima necesaria en el tercer certamen (C3) es: -45.7142857143
Si la nota es negativa, aprueba con las notas actuales
Si no lo es, No aprueba la asignatura
*** Ejecución Finalizada. ***
☐ No cerrar esta ventana ☐ Siempre visible 
```

Caso 3

3. Codificación

```
C clase1.c U X  clase1.exe U
src > C clase1.c > ...
1  #include <stdio.h>
2
3  int main(){
4      float C1, C2, NL, C3;
5
6      printf("Ingrese la nota del primer certamen (C1): ");
7      scanf("%f", &C1);
8
9      printf("Ingrese la nota del segundo certamen (C2): ");
10     scanf("%f", &C2);
11
12     printf("Ingrese la nota del laboratorio (NL): ");
13     scanf("%f", &NL);
14
15     C3 = ((60 - (NL * 0.3)) / 0.7 * 3) - (C1 + C2);
16
17     printf("\nLa nota minima necesaria en el tercer certamen (C3) es: %.1f\n", C3);
18     printf("Si el valor es negativo, ya aprueba con las notas actuales.\n");
19
20
21     return 0;
22 }
```

4. Pruebas (En C)

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS C:\Users\Usuario\Documents\Emerson_VSC\src> .\clase1.exe
Ingrese la nota del primer certamen (C1): 50
Ingrese la nota del segundo certamen (C2): 60
Ingrese la nota del laboratorio (NL): 55

La nota minima necesaria en el tercer certamen (C3) es: 76.4
Si el valor es negativo, ya aprueba con las notas actuales.
PS C:\Users\Usuario\Documents\Emerson_VSC\src> █
```

Caso 1 hecho en lenguaje C

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS C:\Users\Usuario\Documents\Emerson_VSC\src> .\clase1.exe
Ingrese la nota del primer certamen (C1): 70
Ingrese la nota del segundo certamen (C2): 80
Ingrese la nota del laboratorio (NL): 75

La nota minima necesaria en el tercer certamen (C3) es: 10.7
Si el valor es negativo, ya aprueba con las notas actuales.
PS C:\Users\Usuario\Documents\Emerson_VSC\src> █
```

Caso 2 hecho en lenguaje C



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\Usuario\Documents\Emerson_VSC\src> .\clase1.exe
Ingrese la nota del primer certamen (C1): 100
Ingrese la nota del segundo certamen (C2): 100
Ingrese la nota del laboratorio (NL): 80

La nota minima necesaria en el tercer certamen (C3) es: -45.7
Si el valor es negativo, ya aprueba con las notas actuales.
PS C:\Users\Usuario\Documents\Emerson_VSC\src> |
```

Caso 3 hecho en lenguaje C

Cierre

Socialización de Resultados

La práctica fue exitosa al transformar el problema en una solución computacional y aplicar estructuras secuenciales.

- **Análisis Correcto:** Se identificaron C1, C2, NL como entradas y C3 como salida.
- **Proceso Clave:** Se implementó la fórmula despejada para C3 en una sola línea.
- **Validación:** Las Pruebas de Escritorio (PSeInt) y las Pruebas en C confirmaron la **precisión** de la fórmula, demostrando que el resultado obtenido coincide con el esperado en todos los casos (ej., Caso 1: 76.42857143).

Preguntas de Control:

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

En el análisis de un problema computacional, se deben identificar fundamentalmente tres elementos: las Entradas (los datos necesarios para iniciar el proceso, como C1, C2, C3, el Proceso (la lógica o las operaciones que transforman las entradas, como la fórmula de despeje para C3, y las Salidas (los resultados que el programa debe generar, como la nota C3)

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

Validar un algoritmo mediante pruebas de escritorio es crucial para asegurar la lógica de este. Este proceso permite verificar de forma manual que la secuencia de pasos y las fórmulas funcionen correctamente para diferentes casos, garantizando que el Resultado Obtenido coincida con el Resultado Esperado antes de la implementación final en código



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

El traslado de un algoritmo en pseudocódigo a un lenguaje de programación (como C) se realiza mediante la codificación. Esto implica reemplazar las palabras y estructuras genéricas del pseudocódigo (Definir, Leer, Escribir) con la sintaxis y funciones específicas del lenguaje de programación (ej., usando float para real, y scanf/printf para entrada/salida), manteniendo la lógica de las estructuras secuenciales.

Bibliografía

- [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-bf84bddd293f>
- [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=93328>