



Reporte Técnico de Actividades Práctico-Experimentales Nro. 00X

1. Datos de Identificación del Estudiante y la Práctica

Nombre del estudiante(s)	Teoría de la programación
Asignatura	Teoría de la programación
Ciclo	1 A
Unidad	2
Resultado de aprendizaje de la unidad	Aplica las estructuras de programación en la resolución de problemas básicos, bajo los principios de solidaridad, transparencia, responsabilidad y honestidad
Práctica Nro.	002
Tipo	Individual
Título de la Práctica	Aplicación de estructuras repetitivas en la resolución de problemas.
Nombre del Docente	Lisette Geoconda López Faicán
Fecha	Jueves 27 de noviembre del 2025 Jueves 04 de diciembre del 2025
Horario	10h30 – 13h30
Lugar	Aula física asignada al paralelo.
Tiempo planificado en el Sílabo	6 horas

2. Objetivo(s) de la Práctica

- Comprender y aplicar las estructuras repetitivas en la resolución de problemas.
- Diseñar y codificar un algoritmo que utilice bucles para resolver un problema de tipo iterativo.
- Validar el funcionamiento del programa mediante la ejecución práctica.

3. Materiales, Reactivos, Equipos y Herramientas

- Herramientas de modelado de diagrama de flujo (Psient, Draw.io, Lucidchart, otros)
- IDE de programación: Visual Studio Code u otro entorno compatible.
- Lenguaje de programación: C (según los contenidos de la unidad).

4. Procedimiento / Metodología Ejecutada

Describe brevemente los pasos que se siguió durante la ejecución de la práctica.

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

Para esta APE debemos implementar las mejoras del algoritmo ya realizado en la unidad 1 debemos primero agregar las nuevas variables de entrada, salida y proceso,

- **Datos de Entrada:** estudiantes
- **Datos de Proceso y Proceso o Estructuras:** i, iNota, sumaNotasFinales.

sumaNotasFinales = sumaNotasFinales + notaFinal;

promedioGeneral = sumaNotasFinales / estudiantes;

- **Datos de Salida:** promedioGeneral

2. Diseño del algoritmo: elaborar un boceto simplificado o esquema lógico de los procesos repetitivos utilizados (no es necesario el diagrama de flujo completo).

Una vez analizados los datos para el nuevo algoritmo podemos proceder a analizarlo de forma estructurada es decir a través de estructuras como lo son diagramas de flujo, en mi caso utilice PSeInt para graficar como se vería el nuevo algoritmo

Nota: Esta es la estructura de la verificación de notas, es decir, la estructura que verifica las notas una por una para verificar que sean mayores o iguales a cero o menores o iguales a 10:

El diagrama de Flujo completo se encontrará en los resultados esperados:

3. Codificación: trasladar la solución a un lenguaje de programación C.

Una vez realizada y analizado el diagrama de flujo podemos trasladarlo al lenguaje C, ya aplicando todos los cambios (El código se podrá observar en la sección de resultados esperados)

4. Pruebas: compilar y ejecutar el programa en el IDE; verificar que los resultados sean correctos con 3 casos de prueba correspondientes a las notas reales de los estudiantes.

Para la prueba de este algoritmo se les pidió a unos compañeros que ingresen sus notas dentro del algoritmo y la tabla a continuación para poder verificar que el algoritmo funciona correctamente y así fue como quedo la tabla.

Nro	ACD 1	ACD 2	AA 1	AA 2	APE 1	APE 2	ES 1 (Portafolio)	ES2 (Evaluación)	Total
1	9	8	10	10	10	9.5	9	5.5	8.55
2	10	8	10	10	10	9	9	9.5	9.43
3	8	10	9.5	10	8.5	9.5	6	8	8.52
4	10	8	9.5	8.5	7.5	9	9	9	8.81

De forma manual todas las notas totales se deben sumar y dividir para la cantidad de estudiantes que ingresaron su nota a lo cual, esta fórmula se representa así:

$$(8.55 + 9.43 + 8.52 + 8.81) / 4 = 8.83$$

Por lo que en cuestión este promedio final de “8.83” debería salir en la consola como nota final para verificar (El resultado en consola se mostrara en la sección de resultados esperados)

5. Resultados

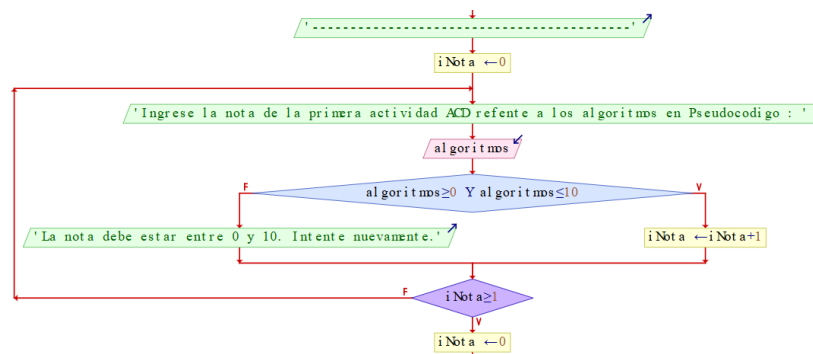
- Contextualización del problema

Contextualización del problema: Basado en el ejercicio del “Cálculo de la nota final de la Unidad 1 mediante estructuras secuenciales en C”, se desea automatizar el proceso de cálculo para varios estudiantes utilizando estructuras repetitivas:

- El programa debe permitir ingresar la cantidad total de estudiantes, y mediante un bucle, repetir el proceso de lectura de calificaciones y cálculo de la nota final.
- En cada repetición, el programa solicitará los valores de los componentes (ACD, APE, AA y ES), calculará la nota final y mostrará el resultado antes de pasar al siguiente estudiante.
- Además, el programa debe validar que las notas ingresadas estén dentro del rango permitido (0 a 10). Si el usuario ingresa una nota fuera de este rango, el programa mostrará un mensaje de error y volverá a solicitar el dato hasta que sea correcto.
- No se requiere guardar las notas; el programa únicamente procesará y mostrará el resultado individual en cada iteración.

Ahora con todo este contexto del problema podemos proceder a Analizar e Identificar las soluciones o métodos para poder darle una solución a los problemas o mejoras planteadas de esta práctica.

- Esquema lógico.



- Código fuente en lenguaje C.

```
1 #include <stdio.h>
2 #include <string.h>
3
4 int main() {
5
6     float ACD, algoritmos, lenguajeC, porcentajeACD, porcentajeTotalACD ;
7
8     float AA, herramientas, instalacionLenguajes, porcentajeAA, porcentajeTotalAA;
9
10    float APE, construccionAlgoritmos, disenoAlgoritmos, porcentajeAPE, porcentajeTotalAPE;
11
12    float ES, portafolioDigital, problema, porcentajeES, porcentajeTotalES, portafolioDigitalTotal, problemaTotal ;
13
14    float notaFinal;
15
16    char escala[30];
17
18    int i;
19    int iNota;
20    int estudiantes;
21    float sumaNotasFinales = 0;
22    float promedioGeneral = 0;
23
24    printf("-----\n");
25    printf("¿Cuántos Estudiantes Registraran su nota en este Programa? : \n");
26    scanf("%i", &estudiantes);
27    printf("-----\n");
```

```
29 for (i = 1; i <= estudiantes;i++)
30 {
31
32     printf("Estudiante %i Proceda a Ingresar sus respectivas notas: \n", i);
33     printf("-----\n");
34
35
36     iNota = 0;
37     while(iNota < 1) {
38         printf("Ingrese la nota de la primera actividad ACD referente a los algoritmos en Pseudocodigo : \n");
39         scanf("%f", &algoritmos);
40
41         if (algoritmos >= 0 && algoritmos <= 10){
42             iNota++;
43         }
44         else
45             printf("La nota debe estar entre 0 y 10. Intente nuevamente.\n");{
46         }
47     }
48
49     iNota = 0;
50     while(iNota < 1) {
51         printf("Ingrese la nota de la segunda actividad ACD referente a la programacion en C : \n");
52         scanf("%f", &lenguajeC);
53
54         if (lenguajeC >= 0 && lenguajeC <= 10){
55             iNota++;
56         }
57         else
58             printf("La nota debe estar entre 0 y 10. Intente nuevamente.\n");{
59         }
60     }
61
62     iNota = 0;
63     while(iNota < 1) {
64
65         printf("Ingrese la nota del primer deber AA referente a las herramientas digitales para Pseudocodigo : \n");
66         scanf("%f", &herramientas);
67
68         if (herramientas >= 0 && herramientas <= 10){
69             iNota++;
70         }
71         else
72             printf("La nota debe estar entre 0 y 10. Intente nuevamente.\n");{
73         }
74
75         iNota = 0;
76         while(iNota < 1) {
77             printf("Ingrese la nota del segundo deber AA referente a la Instalacion de Lenguajes de Programacion : \n");
78             scanf("%f", &instalacionLenguajes);
79
80             if (instalacionLenguajes >= 0 && instalacionLenguajes <= 10){
81                 iNota++;
82             }
83             else
84                 printf("La nota debe estar entre 0 y 10. Intente nuevamente.\n");{
85             }
86         }
87
88         iNota = 0;
89         while(iNota < 1) {
90             printf("Ingrese la nota de la primera practica APE referente a la construccion de algoritmos con estructura secuencial : \n");
91             scanf("%f", &construccionAlgoritmos);
92
93
94             if (construccionAlgoritmos >= 0 && construccionAlgoritmos <= 10){
95                 iNota++;
96             }
97             else
98                 printf("La nota debe estar entre 0 y 10. Intente nuevamente.\n");{
99             }
100         }
101
102         iNota = 0;
103         while(iNota < 1) {
104             printf("Ingrese la nota de la segunda practica APE referente al diseño de algoritmos con estructura secuencial a la construccion del programa: \n");
105             scanf("%f", &diseñoAlgoritmos);
106
107             if (diseñoAlgoritmos >= 0 && diseñoAlgoritmos <= 10){
108                 iNota++;
109             }
110             else
111                 printf("La nota debe estar entre 0 y 10. Intente nuevamente.\n");{
112             }
113         }
114
115         iNota = 0;
116         while(iNota < 1) {
117             printf("Ingrese la nota de su portafolio digital como ES: \n");
118             scanf("%f", &portafolioDigital);
119
120             if (portafolioDigital >= 0 && portafolioDigital <= 10){
121                 iNota++;
122             }
123             else
124                 printf("La nota debe estar entre 0 y 10. Intente nuevamente.\n");{
125             }
126         }
127
128         iNota = 0;
129         while(iNota < 1) {
```

```

130     printf("Ingrese la nota de su evaluacion sumativa como ES: \n");
131     scanf("%f", &problema);
132
133     if (portafolioDigital >= 0 && portafolioDigital <= 10){
134         iNota++;
135     }
136     else
137     {
138         printf("La nota debe estar entre 0 y 10. Intente nuevamente.\n");
139     }
140 }
141
142 ACD = (algoritmos + lenguajeC) / 2 ;
143
144 AA = (herramientas + instalacionLenguajes) / 2 ;
145
146 APE = (construccionAlgoritmos + disenoAlgoritmos) / 2 ;
147
148 ES = (portafolioDigital * 0.4) + (problema * 0.6);
149
150 porcentajeACD = ACD/10 ;
151 porcentajeTotalACD = porcentajeACD * 2;
152
153 porcentajeAA = AA/10;
154 porcentajeTotalAA = porcentajeAA * 2;
155
156 porcentajeAPE = APE/10;
157 porcentajeTotalAPE = porcentajeAPE * 2.5;
158
159 porcentajeES = ES / 10;
160 porcentajeTotalES = porcentajeES * 3.5;
161
162 notaFinal = porcentajeTotalACD + porcentajeTotalAA + porcentajeTotalAPE + porcentajeTotalES;
163
164 if (notaFinal >= 9)
165 {
166     strcpy(escala, "Excelente!");
167     getchar();
168 }else if (notaFinal >= 7 && notaFinal < 9)
169 {
170     strcpy(escala, "Buena");
171     getchar();
172 }else if (notaFinal >= 5 && notaFinal < 7)
173 {
174     strcpy(escala, "Regular");
175     getchar();
176 }else
177 {
178     strcpy(escala, "Deficiente");
179     getchar();
180 }
181
182 printf("-----\n");
183 printf("\n--- RESULTADOS ---\n");
184 printf("ACD (2 pts): %.2f\n", porcentajeTotalACD);
185 printf("AA (2 pts): %.2f\n", porcentajeTotalAA);
186 printf("APE (2.5 pts): %.2f\n", porcentajeTotalAPE);
187 printf("ES (3.5 pts): %.2f\n", porcentajeTotalES);
188 printf("NOTA FINAL (sobre 10) del estudiante %i es: %.2f la cual se considera %s \n", i, notaFinal, escala);
189 printf("-----\n");
190
191 sumaNotasFinales = sumaNotasFinales + notaFinal;
192
193
194
195
196
197 promedioGeneral = sumaNotasFinales / estudiantes;
198
199 printf("\n===== \n");
200 printf("El promedio total entre los estudiantes que ingresaron sus notas fue de: %.2f / 10\n", promedioGeneral);
201 printf("===== \n");
202
203 return 0;
204 }
205

```

- Pruebas.



```
PS C:\Users\USUARIO\Documents\Programacion\Lenguaje C> ./MEJORADO_evaluacionunidad1.exe
-----
¿Cuantos Estudiantes Registraran su nota en este Programa? :
4
-----
Estudiante 1 Proceda a Ingresar sus respectivas notas:
-----
Ingrese la nota de la primera actividad ACD referente a los algoritmos en Pseudocodigo :
9
Ingrese la nota de la segunda actividad ACD referente a la programacion en C :
8
Ingrese la nota del primer deber AA referente a las herramientas digitales para Pseudocodigo :
10
Ingrese la nota del segundo deber AA referente a la Instalacion de Lenguajes de Programacion :
10
Ingrese la nota de la primera practica APE referente a la construccion de algoritmos con estructura secuencial :
10
Ingrese la nota de la segunda practica APE referente al diseño de algoritmos con estructura secuencial a la construccion del programa:
9.5
Ingrese la nota de su portafolio digital como ES:
9
Ingrese la nota de su evaluacion sumativa como ES:
5.5

--- RESULTADOS ---
ACD (2 pts): 1.70
AA (2 pts): 2.00
APE (2.5 pts): 2.44
ES (3.5 pts): 2.41
NOTA FINAL (sobre 10) del estudiante 1 es: 8.55 la cual se considera Buena
-----

Estudiante 2 Proceda a Ingresar sus respectivas notas:
-----
Ingrese la nota de la primera actividad ACD referente a los algoritmos en Pseudocodigo :
10
Ingrese la nota de la segunda actividad ACD referente a la programacion en C :
8
Ingrese la nota del primer deber AA referente a las herramientas digitales para Pseudocodigo :
10
Ingrese la nota del segundo deber AA referente a la Instalacion de Lenguajes de Programacion :
10
Ingrese la nota de la primera practica APE referente a la construccion de algoritmos con estructura secuencial :
10
Ingrese la nota de la segunda practica APE referente al diseño de algoritmos con estructura secuencial a la construccion del programa:
9
Ingrese la nota de su portafolio digital como ES:
9
Ingrese la nota de su evaluacion sumativa como ES:
9.5

--- RESULTADOS ---
ACD (2 pts): 1.80
AA (2 pts): 2.00
APE (2.5 pts): 2.38
ES (3.5 pts): 2.26
NOTA FINAL (sobre 10) del estudiante 2 es: 9.43 la cual se considera Excelente!
-----

Estudiante 3 Proceda a Ingresar sus respectivas notas:
-----
Ingrese la nota de la primera actividad ACD referente a los algoritmos en Pseudocodigo :
9
Ingrese la nota de la segunda actividad ACD referente a la programacion en C :
10
Ingrese la nota del primer deber AA referente a las herramientas digitales para Pseudocodigo :
9.5
Ingrese la nota del segundo deber AA referente a la Instalacion de Lenguajes de Programacion :
10
Ingrese la nota de la primera practica APE referente a la construccion de algoritmos con estructura secuencial :
8.5
Ingrese la nota de la segunda practica APE referente al diseño de algoritmos con estructura secuencial a la construccion del programa:
9.5
Ingrese la nota de su portafolio digital como ES:
6
Ingrese la nota de su evaluacion sumativa como ES:
8

--- RESULTADOS ---
ACD (2 pts): 1.80
AA (2 pts): 1.95
APE (2.5 pts): 2.25
ES (3.5 pts): 2.52
NOTA FINAL (sobre 10) del estudiante 3 es: 8.52 la cual se considera Buena
-----

Estudiante 4 Proceda a Ingresar sus respectivas notas:
-----
Ingrese la nota de la primera actividad ACD referente a los algoritmos en Pseudocodigo :
10
Ingrese la nota de la segunda actividad ACD referente a la programacion en C :
8
Ingrese la nota del primer deber AA referente a las herramientas digitales para Pseudocodigo :
9.5
Ingrese la nota del segundo deber AA referente a la Instalacion de Lenguajes de Programacion :
8.5
Ingrese la nota de la primera practica APE referente a la construccion de algoritmos con estructura secuencial :
7.5
Ingrese la nota de la segunda practica APE referente al diseño de algoritmos con estructura secuencial a la construccion del programa:
9
Ingrese la nota de su portafolio digital como ES:
9
Ingrese la nota de su evaluacion sumativa como ES:
9

--- RESULTADOS ---
ACD (2 pts): 1.80
AA (2 pts): 1.80
APE (2.5 pts): 2.06
ES (3.5 pts): 3.15
NOTA FINAL (sobre 10) del estudiante 4 es: 8.81 la cual se considera Buena
-----

=====
El promedio total entre los estudiantes que ingresaron sus notas fue de: 8.83 / 10
=====
PS C:\Users\USUARIO\Documents\Programacion\Lenguaje C> |
```

6. Preguntas de Control

- **¿En qué se diferencia una estructura repetitiva de una condicional?**

Una estructura condicional permite decidir entre dos o más caminos según se cumpla una condición, mientras que una estructura repetitiva ejecuta un bloque de instrucciones varias veces hasta que se cumpla una condición de salida. La condicional selecciona qué hacer, mientras que la repetitiva define cuántas veces hacerlo.

- **¿Qué diferencia existe entre las estructuras for, while y do...while en cuanto a su funcionamiento y uso?**

El for se utiliza cuando se conoce el número exacto de iteraciones, ya que la variable de control, la condición y el incremento están integrados en una sola línea. El while se emplea cuando no se sabe cuántas veces se repetirá el ciclo; evalúa la condición antes de entrar al bloque. El do...while garantiza que el bloque se ejecute al menos una vez, pues evalúa la condición después de ejecutar el ciclo.

Cada una se usa según la necesidad de control y la lógica del problema.

- **¿Por qué es importante incluir validaciones dentro de un programa cuando se solicitan datos al usuario?**

Las validaciones aseguran que los datos ingresados sean correctos, evitando errores lógicos, cálculos incorrectos o comportamientos inesperados en el programa. Además, garantizan que el sistema sea confiable, robusto y que la información con la que trabaja el programa sea coherente para producir resultados válidos.

7. Conclusiones

La práctica permitió comprender y aplicar estructuras condicionales y repetitivas para resolver un problema real relacionado con el cálculo de notas, asegurando además la correcta validación de datos ingresados por el usuario. Se logró consolidar el uso de ciclos anidados, acumuladores, promedios y estructuras de control, cumpliendo con el objetivo de desarrollar un programa funcional y confiable capaz de calcular la nota final de varios estudiantes y obtener un promedio general.

8. Recomendaciones

Se recomienda seguir practicando la implementación de validaciones y el uso de estructuras repetitivas y condicionales para fortalecer la lógica de programación, así como explorar mejoras en la organización del código y manejo de errores para aplicarlo de manera efectiva en proyectos reales y de mayor complejidad.

9. Bibliografía / Referencias



-
- [1] 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
- [2] 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>