



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	Lissette 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



UNL

Universidad
Nacional
de Loja
1859

FEIRNNR - Carrera de Computación

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

Describa breve de los pasos que se siguió durante la ejecución de la práctica.

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

Para esta esté 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 quedó 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

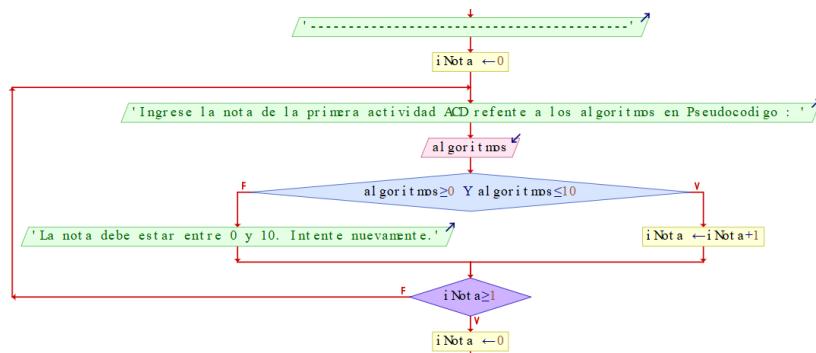


Contextualización del problema: Basado en el ejercicio del “Cálculo de la nota final de la Unidad 1 mediante estructuras secuenciales en C”, se desea automatizar el proceso de cálculo para varios estudiantes utilizando estructuras repetitivas:

- 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     float ACD, algoritmos, lenguajeC, porcentajeACD, porcentajeTotalACD ;
6     float AA, herramientas, instalacionLenguajes, porcentajeAA, porcentajeTotalAA;
7     float APE, construccionAlgoritmos, disenioAlgoritmos, porcentajeAPE, porcentajeTotalAPE;
8     float ES, portafolioDigital, problema, porcentajeES, porcentajeTotalES, portafolioDigitalTotal, problemaTotal ;
9
10    float notaFinal;
11
12    char escala[30];
13
14    int i;
15    int iNota;
16    int estudiantes;
17    float sumaNotasFinales = 0;
18    float promedioGeneral = 0;
19
20    printf("-----\n");
21    printf("¿cuantos Estudiantes Registraran su nota en este Programa? : \n");
22    scanf("%i", &estudiantes);
23    printf("-----\n");
```



UNL

Universidad
Nacional
de Loja

1859

FEIRNNR - Carrera de Computación

```
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
for (i = 1; i <= estudiantes;i++)
{
    printf("Estudiante %i Proceda a Ingresar sus respectivas notas: \n", i);
    printf("-----\n");

    iNota = 0;
    while(iNota < 1) {
        printf("Ingrese la nota de la primera actividad ACD refente a los algoritmos en Pseudocodigo : \n");
        scanf("%f", &algoritmos);

        if (algoritmos >= 0 && algoritmos <= 10){
            iNota++;
        }
        else
            printf("La nota debe estar entre 0 y 10. Intente nuevamente.\n");
    }

    iNota = 0;
    while(iNota < 1) {
        printf("Ingrese la nota de la segunda actividad ACD refente a la programacion en C : \n");
        scanf("%f", &lenguajeC);

        if (lenguajeC >= 0 && lenguajeC <= 10){
            iNota++;
        }
        else
            printf("La nota debe estar entre 0 y 10. Intente nuevamente.\n");
    }

    iNota = 0;
    while(iNota < 1) {
        printf("ingrese la nota del primer deber AA refente a las herramientas digitales para Pseudocodigo : \n");
        scanf("%f", &herramientas);

        if (herramientas >= 0 && herramientas <= 10){
            iNota++;
        }
        else
            printf("La nota debe estar entre 0 y 10. Intente nuevamente.\n");
    }

    iNota = 0;
    while(iNota < 1) {
        printf("Ingrse la nota del segundo deber AA refente a la Instalacion de Lenguajes de Programacion : \n");
        scanf("%f", &instacionLenguajes);

        if (instacionLenguajes >= 0 && instacionLenguajes <= 10){
            iNota++;
        }
        else
            printf("La nota debe estar entre 0 y 10. Intente nuevamente.\n");
    }

    iNota = 0;
    while(iNota < 1) {
        printf("ingrese la nota de la primera practica APE refente a la construccion de algoritmos con estructura secuencial : \n");
        scanf("%f", &construccionAlgoritmos);

        if (construccionAlgoritmos >= 0 && construccionAlgoritmos <= 10){
            iNota++;
        }
        else
            printf("La nota debe estar entre 0 y 10. Intente nuevamente.\n");
    }

    iNota = 0;
    while(iNota < 1) {
        printf(" Ingresa la nota de la segunda practica APE refente al diseo de algoritmos con estructura secuencial a la construccion del programa: \n");
        scanf("%f", &diseñoAlgoritmos);

        if (diseñoAlgoritmos >= 0 && diseñoAlgoritmos <= 10){
            iNota++;
        }
        else
            printf("La nota debe estar entre 0 y 10. Intente nuevamente.\n");
    }

    iNota = 0;
    while(iNota < 1) {
        printf("Ingrse la nota de su portafolio digital como ES: \n");
        scanf("%f", &portafolioDigital);

        if (portafolioDigital >= 0 && portafolioDigital <= 10){
            iNota++;
        }
        else
            printf("La nota debe estar entre 0 y 10. Intente nuevamente.\n");
    }

    iNota = 0;
    while(iNota < 1) {
```



UNL

Universidad
Nacional
de Loja

1859

FEIRNNR - Carrera de Computación

```
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         printf("La nota debe estar entre 0 y 10. Intente nuevamente.\n");
138     }
139
140
141     ACD = (algoritmos + lenguajeC) / 2 ;
142
143     AA = (herramientas + instalacionLenguajes) / 2 ;
144
145     APE = (construccionAlgoritmos + disenoAlgoritmos) / 2 ;
146
147     ES = (portafolioDigital * 0.4) + (problema * 0.6);
148
149     porcentajeACD = ACD/10 ;
150     porcentajeTotalACD = porcentajeACD * 2;
151
152     porcentajeAA = AA/10;
153     porcentajeTotalAA = porcentajeAA * 2;
154
155     porcentajeAPE = APE/10;
156     porcentajeTotalAPE = porcentajeAPE * 2.5;
157
158     porcentajeES = ES / 10;
159     porcentajeTotalES = porcentajeES * 3.5;
160
161     notaFinal = porcentajeTotalACD + porcentajeTotalAA + porcentajeTotalAPE + porcentajeTotalES;
162
163     if (notaFinal >= 9)
164     {
165         strcpy(escala, "Excelente!");
166         getchar();
167     }
168     else if (notaFinal >= 7 && notaFinal < 9)
169     {
170         strcpy(escala, "Buena");
171         getchar();
172     }
173     else if (notaFinal >= 5 && notaFinal < 7)
174     {
175         strcpy(escala, "Regular");
176         getchar();
177     }
178     else
179     {
180         strcpy(escala, "Deficiente");
181         getchar();
182     }
183
184     printf("-----\n");
185     printf("\n-- RESULTADOS --\n");
186     printf("ACD (2 pts): %.2f\n", porcentajeTotalACD);
187     printf("AA (2 pts): %.2f\n", porcentajeTotalAA);
188     printf("APE (2.5 pts): %.2f\n", porcentajeTotalAPE);
189     printf("ES (3.5 pts): %.2f\n", porcentajeTotalES);
190     printf("NOTA FINAL (sobre 10) del estudiante %i es: %.2f la cual se considera %s \n", i, notaFinal, escala);
191     printf("-----\n");
192
193     sumaNotasFinales = sumaNotasFinales + notaFinal;
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
204 }
```

- Pruebas.



UNL

Universidad
Nacional
de Loja

1859

FEIRNNR - Carrera de Computación

```
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 refente a los algoritmos en Pseudocodigo :
9
Ingrese la nota de la segunda actividad ACD refente a la programacion en C :
8
Ingrese la nota del primer deber AA refente a las herramientas digitales para Pseudocodigo :
10
Ingrese la nota del segundo deber AA refente a la Instalacion de Lenguajes de Programacion :
10
Ingrese la nota de la primera practica APE refente a la construccion de algoritmos con estructura secuencial :
10
Ingrese la nota de la segunda practica APE refente 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 refente a los algoritmos en Pseudocodigo :
10
Ingrese la nota de la segunda actividad ACD refente a la programacion en C :
8
Ingrese la nota del primer deber AA refente a las herramientas digitales para Pseudocodigo :
10
Ingrese la nota del segundo deber AA refente a la Instalacion de Lenguajes de Programacion :
10
Ingrese la nota de la primera practica APE refente a la construccion de algoritmos con estructura secuencial :
10
Ingrese la nota de la segunda practica APE refente 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 ---
AA (2 pts): 1.80
AA (2 pts): 2.00
APE (2.5 pts): 2.38
ES (3.5 pts): 2.25
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 refente a los algoritmos en Pseudocodigo :
8
Ingrese la nota de la segunda actividad ACD refente a la programacion en C :
10
Ingrese la nota del primer deber AA refente a las herramientas digitales para Pseudocodigo :
9.5
Ingrese la nota del segundo deber AA refente a la Instalacion de Lenguajes de Programacion :
10
Ingrese la nota de la primera practica APE refente a la construccion de algoritmos con estructura secuencial :
8.5
Ingrese la nota de la segunda practica APE refente 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 refente a los algoritmos en Pseudocodigo :
10
Ingrese la nota de la segunda actividad ACD refente a la programacion en C :
8
Ingrese la nota del primer deber AA refente a las herramientas digitales para Pseudocodigo :
8.5
Ingrese la nota del segundo deber AA refente a la Instalacion de Lenguajes de Programacion :
8.5
Ingrese la nota de la primera practica APE refente a la construccion de algoritmos con estructura secuencial :
7.5
Ingrese la nota de la segunda practica APE refente 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): 2.00
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



UNL

Universidad
Nacional
de Loja
1859

FEIRNNR - Carrera de Computación

- [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>