

Reporte Técnico de Actividades Práctico-Experimentales Nro. 00X

1. Datos de Identificación del Estudiante y la Práctica

Nombre del estudiante(s)	Pilar Valentina Naranjo Quizhpe.
Asignatura	Teoría de la programación
Ciclo	1A
Unidad	3
Resultado de aprendizaje de la unidad	Desarrolla aplicaciones utilizando el principio de la programación modular y estructuras de datos simples y/o estáticas compuestas, bajo los principios de solidaridad, transparencia, responsabilidad y honestidad.
Práctica Nro.	001
Tipo	Individual
Título de la Práctica	Construcción de funciones y procedimientos en un lenguaje de programación.
Nombre del Docente	Lissette Geoconda López Faicán
Fecha	Jueves 8 de enero del 2026 Jueves 15 de enero del 2026
Horario	10h30 – 13h30
Lugar	Aula física asignada al paralelo.
Tiempo planificado en el Sílabo	6 horas

2. Objetivo(s) de la Práctica

- Aplicar los fundamentos de la programación modular mediante la construcción y uso de funciones y procedimientos, para resolver un problema real, garantizando un código estructurado, reutilizable y correctamente documentado.

3. Materiales, Reactivos, Equipos y Herramientas

- Computador personal con sistema operativo Windows, Linux o macOS.
- Material de apoyo en el Entorno Virtual de Aprendizaje (EVA).
- Editores de texto (Word, Google Docs u otros) para la elaboración del informe técnico en formato PDF.



- Conexión a internet estable para acceder a recursos digitales y software en línea.
- Aula física asignada al paralelo.

4. Procedimiento / Metodología Ejecutada

Los pasos para realizar esta práctica fueron los siguientes:

- a. Análisis del problema y determinación de los requerimientos necesarios.
- b. Diseño de la solución aplicando el enfoque de programación modular.
- c. Desarrollo del programa utilizando el lenguaje de programación C.
- d. Compilación y ejecución del código para comprobar su funcionamiento.
- e. Verificación de los resultados obtenidos mediante un caso de prueba real.

5. Resultados

- **Contextualización del problema:**

Se requiere desarrollar un programa que calcule la nota final de un estudiante, aplicando funciones y procedimientos para cada componente evaluativo:

- Calcular nota del Aprendizaje en Contacto con el Docente (ACD): solicita nro de actividades, las notas para cada actividad, calcula promedio, y retorna el total ponderado sobre 2.0.
- Calcular nota del Aprendizaje Práctico Experimental (APE): solicita nro de actividades, las notas para cada actividad, calcula promedio, y retorna el total ponderado sobre 2.5.
- Calcular nota del Aprendizaje Autónomo (AA): solicita nro de actividades, las notas para cada actividad, calcula promedio, y retorna el total ponderado sobre 2.0.
- Calcular Evaluación sumativa (ES): solicita la nota de Aprendizaje Basado en Problemas y Portafolio Digital, calcular el ponderado (60% y 40%), y retornar el total ponderado sobre 3.5.
- Calcular promedio por unidad: ACD + APE + AA + ES (retorna el promedio para una unidad).
 - APROBADO: Si la nota final es mayor o igual a 7.
 - SUPLETORIO: Si la nota final es mayor o igual a 2.5 y menor a 7.
 - REPROBADO: Si la nota final es menor a 2.5.
- Las notas deben estar en el rango de 0.0 a 10.0
- Salida de Resultados: Imprimir la nota cuantitativa y cualitativa final del estudiante.



- Requerimiento del código: El programa debe estar bien estructurado, con comentarios y mensajes descriptivos que faciliten su uso.

• Código fuente del programa

En la función **main** se llama a la función **calcularPromedioFinal**, enviando como dato el número de unidades que tiene la asignatura. Esta función se encarga de calcular el promedio general del estudiante.

Al finalizar la ejecución del programa, la función **main** muestra en pantalla la nota final del estudiante y determina su estado académico, el cual puede ser Aprobado, Supletorio o Reprobado, según la escala establecida.

```
1  #include <stdio.h>
2
3  float calcularPromedioFinal(int nu);
4  float calcularACD();
5  float calcularAPE();
6  float calcularAA();
7  float calcularES();
8
9  int main(){
10    int NUMEROUNIDADES = 3;
11    float promedioFin;
12
13    promedioFin = calcularPromedioFinal(NUMEROUNIDADES);
14    printf("Su nota final de la asignatura es: %.2f\n", promedioFin);
15
16    if (promedioFin >= 7) {
17      printf("APROBADO\n");
18    } else if (promedioFin >= 2.5) {
19      printf("SUPLETORIO\n");
20    } else {
21      printf("REPROBADO\n");
22    }
23
24    return 0;
25 }
```

Dentro de la función **calcularPromedioFinal** se utiliza un ciclo para recorrer cada unidad de la asignatura. En cada iteración se llaman a las funciones **calcularACD**, **calcularAPE**, **calcularAA** y **calcularES**, las cuales calculan las notas de cada componente evaluativo. El resultado de la suma de estos componentes se guarda como la nota de la unidad y se acumula para obtener el promedio final de la asignatura.

```
26 //Calcular el promedio final de la asignatura
27 float calcularPromedioFinal(int nu){
28   float notaUnidad, notaSuma = 0;
29
30   for(int i = 1; i <= nu; i++){
31     printf("Unidad %i\n", i);
32     notaUnidad = calcularACD() + calcularAPE() + calcularAA() + calcularES();
33     notaSuma += notaUnidad;
34   }
35   return (notaSuma / nu);
36 }
```



La función **calcularACD** es llamada desde **calcularPromedioFinal**. En esta función se solicita el número de actividades del Aprendizaje en Contacto con el Docente, se ingresan las notas correspondientes, se valida que estén en el rango de 0 a 10 y se retorna la nota ponderada del ACD.

```
37 //ACD
38 float calcularACD(){
39     int numeroActividades;
40     float notaActividad, notaAcumulativa = 0;
41     printf("Ingrese el numero de actividades para ACD\n");
42     scanf("%i",&numeroActividades);
43
44     for(int i=1; i<=numeroActividades; i++){
45         do{
46             printf("Ingrese la nota de la actividad %i\n",i);
47             scanf("%f",&notaActividad);
48             if(notaActividad < 0 || notaActividad > 10){
49                 printf("Nota no valida. Debe estar entre 0 y 10.\n");
50             }
51         }while(notaActividad < 0 || notaActividad > 10);
52
53         notaAcumulativa += notaActividad;
54     }
55     return (notaAcumulativa / numeroActividades) * 0.2;
56 }
```

La función **calcularAPE** se utiliza para calcular la nota del Aprendizaje Práctico Experimental. Esta función solicita el número de actividades, permite ingresar las notas de cada una, valida los datos ingresados y retorna el valor ponderado correspondiente al APE.

```
57 //APE
58 float calcularAPE(){
59     int numeroActividades;
60     float notaActividad, notaAcumulativa = 0;
61
62     printf("Ingrese el numero de actividades para APE\n");
63     scanf("%i",&numeroActividades);
64
65     for(int i = 1; i <= numeroActividades; i++){
66         do{
67             printf("Ingrese la nota de la actividad %i\n",i);
68             scanf("%f",&notaActividad);
69             if(notaActividad < 0 || notaActividad > 10){
70                 printf("Nota no valida. Debe estar entre 0 y 10.\n");
71             }
72         }while(notaActividad < 0 || notaActividad > 10);
73
74         notaAcumulativa += notaActividad;
75     }
76     return (notaAcumulativa / numeroActividades) * 0.25;
77 }
```



La función **calcularAA** calcula la nota del Aprendizaje Autónomo. En ella se solicita el número de actividades, se ingresan las notas respectivas, se validan los valores ingresados y se retorna la nota ponderada correspondiente.

```
78 //AA
79 float calcularAA(){
80     int numeroActividades;
81     float notaActividad, notaAcumulativa = 0;
82
83     printf("Ingrese el numero de cactividades para AA\n");
84     scanf("%i",&numeroActividades);
85
86     for(int i = 1; i <= numeroActividades; i++){
87         do{
88             printf("Ingrese la nota de la actividad %i\n",i);
89             scanf("%f",&notaActividad);
90             if(notaActividad < 0 || notaActividad > 10){
91                 printf("Nota no valida. Debe estar entre 0 y 10.\n");
92             }
93         }while(notaActividad < 0 || notaActividad > 10);
94
95         notaAcumulativa += notaActividad;
96     }
97     return (notaAcumulativa / numeroActividades) * 0.2;
98 }
```

Finalmente, la función **calcularES** solicita la nota del **Portafolio Digital** y la nota de la **Evaluación**. Luego, aplica la ponderación del 60% y 40% respectivamente, y retorna la nota final de la Evaluación Sumativa.

```
99 //ES
100 float calcularES(){
101     float notaE, notaP;
102
103     printf("Ingrese la nota del Portafolio Digital\n");
104     scanf("%f",&notaP);
105
106     while(notaP < 0 || notaP > 10){
107         printf("Notas no validas, ingrese nuevamente su calificacion del Portafolio Digital\n");
108         scanf("%f", &notaP);
109     }
110
111     printf("Ingrese su calificacion de la Evalucion\n");
112     scanf("%f", &notaE);
113
114     while(notaE < 0 || notaE > 10){
115         printf("Notas no validas, ingrese nuevamente su calificacion de la Evalucion\n");
116         scanf("%f", &notaE);
117     }
118     return ((notaE * 0.6) + (notaP * 0.4)) * 0.35;
119 }
```

- **Pruebas, ejecución de un caso de prueba**

```
Unidad 1
Ingrese el numero de actividades para ACD: 2
Ingrese la nota de la actividad 1: 8
Ingrese la nota de la actividad 2: 8
Ingrese el numero de actividades para APE: 2
Ingrese la nota de la actividad 1: 10
Ingrese la nota de la actividad 2: 6.5
Ingrese el numero de actividades para AA: 2
Ingrese la nota de la actividad 1: 9.5
Ingrese la nota de la actividad 2: 8.5
Ingrese la nota del Portafolio Digital: 9
Ingrese su calificacion de la Evalucion: 10
Unidad 2
Ingrese el numero de actividades para ACD: 2
Ingrese la nota de la actividad 1: 5
Ingrese la nota de la actividad 2: 7
Ingrese el numero de actividades para APE: 2
Ingrese la nota de la actividad 1: 10
Ingrese la nota de la actividad 2: 9.5
```



```
Ingrese el numero de actividades para AA: 2
Ingrese la nota de la actividad 1: 9.5
Ingrese la nota de la actividad 2: 6.5
Ingrese la nota del Portafolio Digital: 9.5
Ingrese su calificacion de la Evaluacion: 10
Unidad 3
Ingrese el numero de actividades para ACD: 2
Ingrese la nota de la actividad 1: 10
Ingrese la nota de la actividad 2: 10
Ingrese el numero de actividades para APE: 2
Ingrese la nota de la actividad 1: 10
Ingrese la nota de la actividad 2: 10
Ingrese el numero de actividades para AA: 2
Ingrese la nota de la actividad 1: 10
Ingrese la nota de la actividad 2: 10
Ingrese la nota del Portafolio Digital: 10
Ingrese su calificacion de la Evaluacion: 10
Su nota final de la asignatura es: 9.16
APROBADO
```

6. Preguntas de Control

a. ¿Cuál es la diferencia entre una función y un procedimiento?

La diferencia principal es que una función devuelve un valor al finalizar su ejecución, mientras que un procedimiento no lo hace, ejecuta instrucciones sin retornar un valor.

b. ¿Qué ventajas aporta dividir un programa en funciones (modularidad)?

Las ventajas que nos aporta dividir un programa en funciones es que nos facilita la organización del código, mejora la legibilidad, permite la reutilización de funciones y simplifica el mantenimiento del programa.

c. ¿Qué se mejoraría del programa si se tuviera que usarlo para varios estudiantes?

Se podría mejorar implementando estructuras para almacenar los datos de varios estudiantes, lo que permitiría calcular las notas de todos de manera automática, ahorrando tiempo y evitando repetir el proceso para cada estudiante.

7. Conclusiones

Esta práctica permitió crear un programa en lenguaje C que calcula correctamente el promedio final de un estudiante. Además, se reforzó el uso de funciones, ciclos y la validación de datos, lo que permitió desarrollar un programa más ordenado y fácil de comprender. Comprendiendo la importancia de estructurar correctamente un programa para obtener resultados correctos y confiables.