

# Reporte Técnico de Actividades Práctico-Experimentales Nro. 001

## 1. Datos de Identificación del Estudiante y la Práctica

<b>Nombre del estudiante</b>	Alison Tapia
<b>Asignatura</b>	Teoría de la programación
<b>Ciclo</b>	1 A
<b>Unidad</b>	3
<b>Resultado de aprendizaje de la unidad</b>	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.
<b>Práctica Nro.</b>	001
<b>Tipo</b>	Individual
<b>Título de la Práctica</b>	Construcción de funciones y procedimientos en un lenguaje de programación.
<b>Nombre del Docente</b>	Lisette Geoconda López Faicán
<b>Fecha</b>	Jueves 8 de enero del 2026
<b>Horario</b>	10h30 – 13h30
<b>Lugar</b>	Aula física asignada al paralelo
<b>Tiempo planificado en el Sílabo</b>	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

### Materiales y reactivos:

- IDE de programación: Visual Studio Code u otro entorno compatible.
- Lenguaje de programación: C (según los contenidos de la unidad).

### 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

1. Se codificaron las funciones indicadas por la docente partiendo desde la más grande hasta la más pequeña.
2. Se realizó pruebas de escritorio para verificar su funcionamiento.
3. Se llamó las funciones desde la función principal main().
4. Se comprobó el funcionamiento de todo el programa con notas reales.

#### 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).
  - Calcular el promedio de la asignatura: promedio simple de acuerdo al número de unidades con la escala cualitativa.
    - 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:**

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

```

```
28 float calcularPromedioFinal(int nu){
29     float notaUnidad;
30     float notaSuma;
31     for (int i = 1; i <= nu; i++){ //Ingresar 3 veces
32         //Cálculo nota de cada unidad
33         printf("Unidad %i\n", i);
34         notaUnidad = calcularACD() + calcularAPE() + calcularAA() + calcularES();
35         notaSuma += notaUnidad;
36     }
37     notaSuma = notaSuma / nu; //Promedio final
38     return notaSuma;
39 }

```

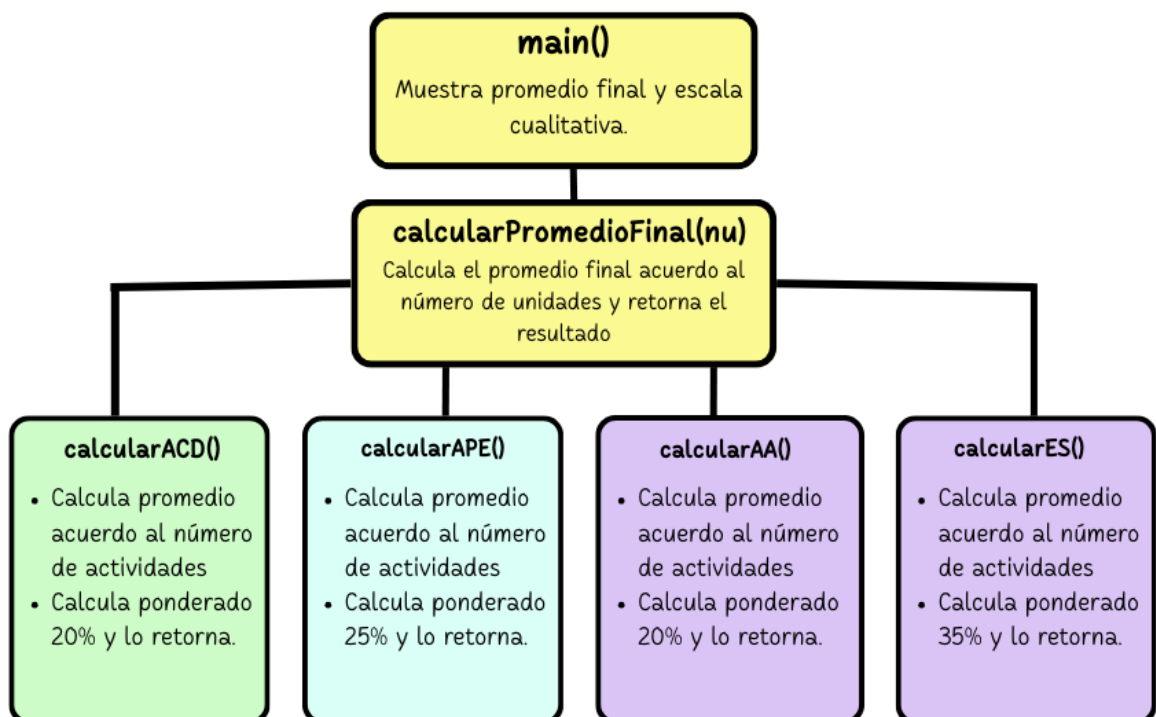
```
41 float calcularACD(){
42     int numeroActividades;
43     float notaActividad;
44     float notaAcumulativa = 0;
45     float notaPromedio;
46
47     printf("Ingrese el numero de actividades para ACD: ");
48     scanf("%i", &numeroActividades);
49
50     //Solicitar nota de cada actividad
51     for (int i = 1; i <= numeroActividades; i++){
52         printf("Ingrese la nota de la actividad %i: ", i);
53         scanf("%f", &notaActividad);
54
55         //Control rango de nota 0-10
56         while (notaActividad < 0 || notaActividad > 10)
57         {
58             printf("Ingrese una nota dentro del rango 0-10: ");
59             scanf("%f", &notaActividad);
60         }
61         notaAcumulativa += notaActividad;
62     }
63
64     notaPromedio = notaAcumulativa / numeroActividades;
65     notaPromedio = notaPromedio * 0.2; //Ponderado
66
67     return notaPromedio;
68 }
69 }
```

```
71 float calcularAPE(){
72     int numeroActividades;
73     float notaActividad;
74     float notaAcumulativa = 0;
75     float notaPromedio;
76
77     printf("Ingrese el numero de actividades para APE: ");
78     scanf("%i", &numeroActividades);
79
80     //Solicitar nota de cada actividad
81     for (int i = 1; i <= numeroActividades; i++){
82         printf("Ingrese la nota de la actividad %i: ", i);
83         scanf("%f", &notaActividad);
84
85         //Control rango de nota 0-10
86         while (notaActividad < 0 || notaActividad > 10)
87         {
88             printf("Ingrese una nota dentro del rango 0-10: ");
89             scanf("%f", &notaActividad);
90         }
91
92         notaAcumulativa += notaActividad;
93     }
94
95     notaPromedio = notaAcumulativa / numeroActividades;
96     notaPromedio = notaPromedio * 0.25; //Ponderado
97
98     return notaPromedio;
99 }
```

```
101 float calcularAA(){
102     int numeroActividades;
103     float notaActividad;
104     float notaAcumulativa = 0;
105     float notaPromedio;
106
107     printf("Ingrese el numero de actividades para AA: ");
108     scanf("%i", &numeroActividades);
109
110     //Solicitar nota de cada actividad
111     for (int i = 1; i <= numeroActividades; i++){
112         printf("Ingrese la nota de la actividad %i: ", i);
113         scanf("%f", &notaActividad);
114
115         //Control rango de nota 0-10
116         while (notaActividad < 0 || notaActividad > 10)
117         {
118             printf("Ingrese una nota dentro del rango 0-10: ");
119             scanf("%f", &notaActividad);
120         };
121
122         notaAcumulativa += notaActividad;
123     }
124
125     notaPromedio = notaAcumulativa / numeroActividades;
126     notaPromedio = notaPromedio * 0.2; //Ponderado
127
128     return notaPromedio;
129 }
```

```
131 float calcularES(){
132     float abp;
133     float portafolio;
134     float notaTotal;
135
136     printf("Ingrese la nota del Aprendizaje Basado en Problemas: ");
137     scanf("%f", &abp);
138     while (abp < 0 || abp > 10) //Control rango de nota 0-10
139     {
140         printf("Ingrese una nota dentro del rango 0-10: ");
141         scanf("%f", &abp);
142     }
143
144     printf("Ingrese la nota del Portafolio Digital: ");
145     scanf("%f", &portafolio);
146     while (portafolio < 0 || portafolio > 10) //Control rango de nota 0-10
147     {
148         printf("Ingrese una nota dentro del rango 0-10: ");
149         scanf("%f", &portafolio);
150     }
151
152     notaTotal = ((abp * 0.6) + (portafolio * 0.4)) * 0.35; //Ponderado
153
154     return notaTotal;
155 }
```

- Llamada de funciones:



- **Pruebas:**

**Unidad 1**

```
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: 75
Ingrese una nota dentro del rango 0-10: 7.5
Ingrese el numero de actividades para AA: 2
Ingrese la nota de la actividad 1: -1
Ingrese una nota dentro del rango 0-10: 10
Ingrese la nota de la actividad 2: 10
Ingrese la nota del Aprendizaje Basado en Problemas: 10
Ingrese la nota del Portafolio Digital: 9.5
```

**Unidad 2**

```
Ingrese el numero de actividades para ACD: 2
Ingrese la nota de la actividad 1: 8
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: 7.5
Ingrese la nota del Aprendizaje Basado en Problemas: 10
Ingrese la nota del Portafolio Digital: 9.5
```

**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 Aprendizaje Basado en Problemas: 10
Ingrese la nota del Portafolio Digital: 10
Su nota final de la asignatura es: 9.67
APROBADO
```

## 6. Preguntas de Control

- **¿Cuál es la diferencia entre una función y un procedimiento?**

La principal diferencia es que una función retorna un valor como resultado de su ejecución, mientras que un procedimiento no retorna ningún valor, sino que únicamente ejecuta una serie de instrucciones.

- **¿Qué ventajas aporta dividir un programa en funciones (modularidad)?**





---

Las ventajas son que facilita la lectura y comprensión del código, permite la reutilización del código, mejora el mantenimiento y ayuda a resolver el programa de una forma más eficiente y ordenada.

- **¿Qué se mejoraría del programa si se tuviera que usarlo para varios estudiantes?**

Se podría optimizar la entrada de datos solicitando la información de forma más concreta; por ejemplo, permitir ingresar todas las notas del ACD en una sola línea separadas por comas, lo que haría el programa más rápido y práctico.

## **7. Conclusiones**

Se pudo aplicar a profundidad los fundamentos de la programación modular en la construcción y uso de funciones y procedimientos, para resolver un problema real, por lo que se garantizó un código estructurado, reutilizable y correctamente documentado.

## **8. Recomendaciones**

Se recomienda utilizar comentarios para mejorar la legibilidad del código, así como aplicar la modularidad en programas complejos, ya que esta facilita la detección de errores y mejora la comprensión del programa.