



**UNL**

Universidad  
Nacional  
de Loja  
1859

FEIRNNR - Carrera de Computación

---

# Reporte Técnico de Actividades Práctico-Experimentales Nro. 001

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

<b>Nombre del estudiante(s)</b>	Jeancarlos Fernando Aguirre Romero
<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>	Lissette Geoconda López Faicán
<b>Fecha</b>	Jueves 8 de enero del 2026 jueves 15 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



**UNL**

Universidad  
Nacional  
de Loja

FEIRNNR - Carrera de Computación

## 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

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



**UNL**

Universidad  
Nacional  
de Loja  
1859

FEIRNNR - Carrera de Computación

---

## 4. Procedimiento / Metodología Ejecutada

Para esta practica se siguieron los siguientes pasos:

1. Primero se analizaron los requerimientos iniciales: Se examinó el PDF guía del proyecto para identificar los componentes a calcular dentro del programa (ACD, APE, AA, ES) y sus respectivas ponderaciones sobre 10 puntos.
2. Luego se definieron las funciones iniciales, estructurando el programa de manera que cada proceso de cálculo fuera independiente uno del otro para su ejecución dinámica además se desarrolló la función principal para iterar a través de las 3 unidades del ciclo.
3. Se Implementación las respectivas estructuras dentro de las funciones incluyendo bucles como el while para asegurar que todas las notas ingresadas estuvieran en el rango obligatorio de 0.0 a 10.0.
4. Luego se realizaron los respectivos cálculos de ponderaciones, y para esto se integraron las fórmulas matemáticas para aplicar los porcentajes en sus respectivos valores (20% ACD, 25% APE, 20% AA y 35% ES) antes de retornar el subtotal de la unidad.
5. Se hizo la respectiva integración y cálculo del final, sumando sus resultados y calculando el promedio simple de la asignatura, además de la implementación de la categorización por notas para los estudiantes
6. Finalmente se hicieron las pruebas de funcionalidad, se ejecutó el programa en el IDE utilizando casos de prueba teóricos para verificar la precisión de los cálculos y la correcta asignación de la escala cualitativa final.



## 5. Resultados

- Función Main para la presentación de los resultados

```
int main(){
    // Configuración inicial del número de unidades del ciclo
    int NUMEROUNIDADES = 3;
    float promedioFin;

    // Llamada a la función principal que procesa todas las unidades
    promedioFin = calcularPromedioFinal (NUMEROUNIDADES);

    printf("\n Su nota final de la asignatura es %f", promedioFin);

    // Estructura condicional para categorizar el rendimiento en base a la nota del estudiante
    if (promedioFin >= 9){
        printf(" La cual se considera Excelente!");
    }

    else if (promedioFin >= 7 && promedioFin < 9){
        printf(" La cual se considera Buena");
    }

    else if (promedioFin >= 5 && promedioFin < 7){
        printf(" La cual se considera Regular");
    }

    else{
        printf(" La cual se considera Deficiente");
    }

    return 0;
}
```

La función main cumple dos funciones muy importantes y útiles dentro del programa las cuales son: la de definir la cantidad de unidades y la presentación de los datos ya sumados y asignarles una categorización por rendimiento



**UNL**

Universidad  
Nacional  
de Loja  
1859

FEIRNNR - Carrera de Computación

---

- Función para el Cálculo del Promedio Total

```
40 // Calculo del promedio general, sumando las notas ponderadas de cada unidad
41 float calcularPromedioFinal(int nu){
42     float notaUnidad;
43     float notaSuma = 0;
44     float PromedioFinal = 0;
45
46     for (int i = 1; i <= nu; i++){
47
48         printf("\n Unidad %i \n \n ", i);
49
50         // Sumatoria de los componentes de la unidad (ACD + APE + AA + ES)
51         notaUnidad = calcularACD() + calcularAPE() + calcularAA() + calcularES();
52         notaSuma = notaSuma + notaUnidad;
53
54         // Cálculo del promedio acumulado basado en el número de unidades procesadas
55         PromedioFinal = notaSuma / nu;
56     }
57
58     return PromedioFinal;
59 }
```

El propósito de esta función es de calcular y acumular los valores del programa es decir calcula los valores por unidad (ACD, APE, AA y ES) y los acumula por unidad, es decir los acumula por Unidades para después arrojar el valor final en la función main principal.

- Función para el Cálculo del ACD



UNL

Universidad  
Nacional  
de Loja  
1859

FEIRNNR - Carrera de Computación

```
61 // Función para calcular la nota referente al Aprendizaje en Contacto con el Docente [ACD] (Ponderado: 20%)
62 float calcularACD(){
63
64     int numeroActividades;
65     float notaActividad;
66     float notaAcumulativa=0;
67     float notaPromedio=0;
68
69     printf("Ingrese el numero de actividades para ACD: ");
70     scanf("%i", &numeroActividades);
71
72     for (int i = 1; i <= numeroActividades; i++)
73     {
74         int iNota = 0;
75
76         // Bucle de validación: asegura que la nota esté en el rango de 0 a 10
77         while (iNota < 1) {
78
79             printf("Ingrese la nota de la actividad ACD %i : ", i);
80             scanf("%f", &notaActividad);
81
82             if (notaActividad >= 0 && notaActividad <= 10){
83                 iNota++;
84             }
85             else{
86                 printf("La nota debe estar entre 0 y 10. Intente nuevamente.\n");
87             }
88         }
89         notaAcumulativa = notaActividad + notaAcumulativa;
90     }
91     // Se obtiene el promedio y su porcentaje equivalente
92     notaPromedio = (notaAcumulativa/numeroActividades);
93     notaPromedio = notaPromedio * 0.2;
94
95     printf("Nota ACD: %f \n", notaPromedio );
96     return notaPromedio;
97 }
```

El propósito de esta función es de calcular la cantidad actividades ingresadas por el usuario para el calculo del ACD, como se puede ver en la imagen hay algunas partes señaladas por comentarios las cuales cumplen una funcionan clave



- Función para el cálculo del APE

```
99 // Función para calcular la nota referente al Aprendizaje Práctico Experimental [APE] (Ponderado: 25%)
100 float calcularAPE(){
101
102     int numeroActividades;
103     float notaActividad;
104     float notaAcumulativa=0;
105     float notaPromedio=0;
106
107     printf("Ingrese el numero de actividades para APE: ");
108     scanf("%i", &numeroActividades);
109
110    for (int i = 1; i <= numeroActividades; i++){
111
112        int iNota = 0;
113
114        while (iNota < 1){
115
116            printf("Ingrese la nota de la actividad APE %i : ", i);
117            scanf("%f", &notaActividad);
118
119            if (notaActividad >= 0 && notaActividad <= 10){
120                iNota++;
121            }
122            else{
123                printf("La nota debe estar entre 0 y 10. Intente nuevamente.\n");
124            }
125        }
126        notaAcumulativa = notaActividad + notaAcumulativa;
127    }
128
129    // Se obtiene el promedio y su porcentaje equivalente
130    notaPromedio = (notaAcumulativa/numeroActividades);
131    notaPromedio = notaPromedio * 0.25;
132
133    printf("Nota APE: %f \n", notaPromedio);
134    return notaPromedio;
135 }
```

El propósito de esta función es de calcular la cantidad actividades ingresadas por el usuario para el cálculo del APE, como se puede ver en la imagen hay algunas partes señaladas por comentarios las cuales cumplen una función



- Función para el cálculo del AA

```
137 // Función para calcular la nota referente al Aprendizaje Autonomo [AA] (Ponderado: 20%)
138 float calcularAA(){
139
140     int numeroActividades;
141     float notaActividad;
142     float notaAcumulativa=0;
143     float notaPromedio=0;
144
145     printf("Ingrese el numero de actividades para AA: ");
146     scanf("%i", &numeroActividades);
147
148     for (int i = 1; i <= numeroActividades; i++){
149
150         int iNota = 0;
151
152         while (iNota < 1) {
153
154             printf("Ingrese la nota de la actividad AA %i : ", i);
155             scanf("%f", &notaActividad);
156
157             if (notaActividad >= 0 && notaActividad <= 10){
158                 iNota++;
159             }
160             else{
161                 printf("La nota debe estar entre 0 y 10. Intente nuevamente.\n");
162             }
163         }
164         notaAcumulativa = notaActividad + notaAcumulativa;
165     }
166
167     // Se obtiene el promedio y su porcentaje equivalente
168     notaPromedio = (notaAcumulativa/numeroActividades);
169     notaPromedio = notaPromedio * 0.2;
170
171     printf("Nota AA: %f \n", notaPromedio);
172     return notaPromedio;
173 }
```

El propósito de esta función es de calcular la cantidad actividades ingresadas por el usuario para el cálculo del AA, como se puede ver en la imagen hay algunas partes señaladas por comentarios las cuales cumplen una función



- **Función para el cálculo del ES**

```
175 // Función para calcular las notas referentes a la Evaluación Sumativa (Ponderado: 35% del total de la unidad)
176 float calcularES(){
177
178     float portafolio;
179     float evaluacion;
180     float portafolioPorcentaje;
181     float evaluacionPorcentaje;
182     float notaAcumulativa=0;
183     float notaPromedio=0;
184     int iNota = 0;
185
186     while (iNota < 1) {
187
188         printf("Ingrese la nota de su portafolio digital: ");
189         scanf("%f", &portafolio);
190
191         if (portafolio >= 0 && portafolio <= 10){
192             iNota++;
193         }
194
195         else{
196             printf("La nota debe estar entre 0 y 10. Intente nuevamente.\n");
197         }
198     }
199
200     while (iNota < 2) {
201
202         printf("Ingrese la nota de su Evaluación formativa: ");
203         scanf("%f", &evaluacion);
204
205         if (evaluacion >= 0 && evaluacion <= 10){
206             iNota++;
207         }
208
209         else{
210             printf("La nota debe estar entre 0 y 10. Intente nuevamente.\n");
211         }
212     }
213
214     // Cálculo interno de la nota de la evaluación sumativa
215     portafolioPorcentaje = portafolio * 0.4;
216     evaluacionPorcentaje = evaluacion * 0.6;
217
218     notaAcumulativa = portafolioPorcentaje + evaluacionPorcentaje;
219
220     // Se obtiene el promedio y su porcentaje equivalente
221     notaPromedio = notaAcumulativa * 0.35;
222
223     printf("Nota ES: %f \n", notaPromedio);
224
225     return notaPromedio;
226 }
```

El propósito de esta función es de calcular las notas respectivas de la evaluación sumativa las cuales serán ingresadas por el usuario para su respectivo calculo cálculo, como se puede ver en la imagen hay algunas partes señaladas por comentarios las cuales cumplen una función en concreto

- **. Prueba de Escrito para Verificar la validez del programa**



**UNL**

Universidad  
Nacional  
de Loja  
1859

FEIRNNR - Carrera de Computación

Parámetro	Unidad 1	Unidad 2	Unidad 3
Notas ACD (20%)	9.0 - 8.0	7.0 - 7.0	10.0 - 9.0
Notas APE (25%)	8.0 - 8.0	6.0 - 8.0	9.0 - 9.0
Notas AA (20%)	9.0 - 10.0	8.0 - 8.0	8.0 - 10.0
Nota Portafolio (ES)	9.0	7.0	10.0
Nota Evaluación (ES)	8.0	7.0	9.0
<b>Subtotal Unidad</b>	<b>8.54</b>	<b>7.20</b>	<b>9.24</b>

**Suma de las 3 Unidades:**  $8.54 + 7.20 + 9.24 = 24.98$

**Promedio Final:**  $24.98 / 3 = 8.33$

- Ejecución del programa para su verificación con la prueba de escritorio

```
PS C:\Users\USUARIO\Documents\Programacion\Lenguaje C\Unidad 3 TD\calculoAsignatura> ./main.exe
```

```
Unidad 1
```

```
Ingrese el numero de actividades para ACD: 2
Ingrese la nota de la actividad ACD 1 :9
Ingrese la nota de la actividad ACD 2 :8
Nota ACD: 1.700000
Ingrese el numero de actividades para APE: 2
Ingrese la nota de la actividad APE 1 :8
Ingrese la nota de la actividad APE 2 :8
Nota APE: 2.000000
Ingrese el numero de actividades para AA: 2
Ingrese la nota de la actividad AA 1 :9
Ingrese la nota de la actividad AA 2 :10
Nota AA: 1.900000
Ingrese la nota de su portafolio digital: 9
Ingrese la nota de su Evaluaci|n formativa: 8
Nota ES: 2.940000
```



**UNL**

Universidad  
Nacional  
de Loja  
1859

FEIRNNR - Carrera de Computación

#### Unidad 2

```
Ingrese el numero de actividades para ACD: 2
Ingrese la nota de la actividad ACD 1 :7
Ingrese la nota de la actividad ACD 2 :7
Nota ACD: 1.400000
Ingrese el numero de actividades para APE: 2
Ingrese la nota de la actividad APE 1 :6
Ingrese la nota de la actividad APE 2 :8
Nota APE: 1.750000
Ingrese el numero de actividades para AA: 2
Ingrese la nota de la actividad AA 1 :8
Ingrese la nota de la actividad AA 2 :8
Nota AA: 1.600000
Ingrese la nota de su portafolio digital: 7
Ingrese la nota de su Evaluaci|n formativa: 7
Nota ES: 2.450000
```

#### Unidad 3

```
Ingrese el numero de actividades para ACD: 2
Ingrese la nota de la actividad ACD 1 :10
Ingrese la nota de la actividad ACD 2 :9
Nota ACD: 1.900000
Ingrese el numero de actividades para APE: 2
Ingrese la nota de la actividad APE 1 :9
Ingrese la nota de la actividad APE 2 :9
Nota APE: 2.250000
Ingrese el numero de actividades para AA: 2
Ingrese la nota de la actividad AA 1 :8
Ingrese la nota de la actividad AA 2 :10
Nota AA: 1.800000
Ingrese la nota de su portafolio digital: 10
Ingrese la nota de su Evaluaci|n formativa: 9
Nota ES: 3.290000
```

Su nota final de la asignatura es 8.326667 La cual se considera Buena  
PS C:\Users\USUARIO\Documents\Programacion\Lenguaje C\Unidad 3 TD\calculoAsignatura> □

## 6. Preguntas de Control

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

La diferencia principal es que una función realiza un cálculo y devuelve un resultado específico al programa usando “return”, como las funciones del código que entregan una nota decimal. Por el contrario, un procedimiento ejecuta una serie de instrucciones como mostrar un texto sin devolver ningún valor, utilizando generalmente el tipo “void”.

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



La modularidad permite reutilizar código y facilita el mantenimiento, ya que permite corregir errores en un módulo específico sin afectar al resto del sistema. Además, mejora la legibilidad del programa, permitiendo que el bloque principal (es decir el main) sea más sencillo al delegar tareas complejas a funciones con nombres descriptivos.

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

Para gestionar múltiples alumnos, se deberían implementar estructuras y arreglos para organizar los datos de forma masiva. También sería fundamental añadir la lectura y escritura de archivos para guardar las notas permanentemente y crear un menú interactivo que facilite la navegación entre los registros de cada estudiante registrado

## 7. Conclusiones

El desarrollo de esta práctica permitió aplicar con éxito los fundamentos de la programación modular, logrando resolver el problema del cálculo de notas mediante la construcción de funciones específicas para cada componente evaluativo (ACD, APE, AA y ES). A través de esta estructura, se comprobó que la división del código en módulos independientes no solo garantiza un programa más organizado y reutilizable, sino que también facilita la observación de errores al aislar la lógica de cada cálculo. Además, la implementación de validaciones mediante bucles aseguró que todos los datos ingresados se mantuvieran en el rango obligatorio de 0.0 a 10.0, cumpliendo así con los controles de calidad exigidos en el procedimiento. Finalmente, el programa automatiza eficazmente la entrega de resultados tanto cuantitativos como cualitativos, permitiendo identificar rápidamente el estado académico del estudiante según la escala de aprobación

## 8. Recomendaciones

Para futuros proyectos de programación modular, se recomienda planificar la estructura de las funciones y sus retornos antes de codificar, asegurando que cada módulo cumpla una tarea única. Es fundamental implementar validaciones de entrada robustas mediante bucles para prevenir errores de usuario y garantizar la integridad de los datos procesados. Asimismo, es aconsejable mantener un código limpio y documentado con comentarios descriptivos, lo que facilita enormemente el mantenimiento y la colaboración técnica.

## 9. Bibliografía / Referencias

- [1] M. Goin, Caminando junto al Lenguaje C. Río Negro, Argentina: Editorial UNRN, 2022. [Online]. Available:



**UNL**

Universidad  
Nacional  
de Loja  
1859

FEIRNNR - Carrera de Computación

---

[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](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>

## 10. Anexos

El código Completo de la practica

```
C main.c > @ calcularPromedioFinal(int)
1  #include <stdio.h>
2
3 // Llamado a las funciones inferiores para su correcta ejecucion
4 float calcularACD();
5 float calcularAPE();
6 float calcularAA();
7 float calcularES();
8 float calcularPromedioFinal(int nu);
9
10 int main(){
11     // Configuración inicial del número de unidades del ciclo
12     int NUMEROUNIDADES = 3;
13     float promediofin;
14
15     // Llamada a la función principal que procesa todas las unidades
16     promedioFin = calcularPromedioFinal (NUMEROUNIDADES);
17
18     printf("\n Su nota final de la asignatura es %f", promedioFin);
19
20     // Estructura condicional para categorizar el rendimiento en base a la nota del estudiante
21     if (promedioFin >= 9){
22         printf(" La cual se considera Excelente!");
23     }
24
25     else if (promedioFin >= 7 && promedioFin < 9){
26         printf(" La cual se considera Buena");
27     }
28
29     else if (promedioFin >= 5 && promedioFin < 7){
30         printf(" La cual se considera Regular");
31     }
32
33     else{
34         printf(" La cual se considera Deficiente");
35     }
36
37 :
```



UNL

Universidad  
Nacional  
de Loja

1859

FEIRNNR - Carrera de Computación

```
69 printf("Ingrese el numero de actividades para ACD: ");
70 scanf("%i", &numeroActividades);
71
72 for (int i = 1; i <= numeroActividades; i++)
73 {
74     int iNota = 0;
75
76     // Bucle de validación: asegura que la nota esté en el rango de 0 a 10
77     while (iNota < 1) {
78
79         printf("Ingrese la nota de la actividad ACD %i : ", i);
80         scanf("%f", &notaActividad);
81
82         if (notaActividad >= 0 && notaActividad <= 10){
83             iNota++;
84         }
85
86         else{
87             printf("La nota debe estar entre 0 y 10. Intente nuevamente.\n");
88         }
89     }
90
91     notaAcumulativa = notaActividad + notaAcumulativa;
92 }
93
94 // Se obtiene el promedio y su porcentaje equivalente
95 notaPromedio = (notaAcumulativa/numeroActividades);
96 notaPromedio = notaPromedio * 0.2;
97
98 printf("Nota ACD: %f \n", notaPromedio );
99
100 return notaPromedio;
101 }
102 return 0;
103 }
104
105 // Calculo del promedio general, sumando las notas ponderadas de cada unidad
106 float calcularPromedioFinal(int nu){
107     float notaUnidad;
108     float notaSuma = 0;
109     float PromedioFinal = 0;
110
111     for (int i = 1; i <= nu; i++){
112
113         printf("\n Unidad %i \n \n ", i);
114
115         // Sumatoria de los componentes de la unidad (ACD + APE + AA + ES)
116         notaUnidad = calcularACD() + calcularAPE() + calcularAA() + calcularES();
117         notaSuma = notaSuma + notaUnidad;
118
119         // Cálculo del promedio acumulado basado en el número de unidades procesadas
120         PromedioFinal = notaSuma / nu;
121     }
122
123     return PromedioFinal;
124 }
125
126 // Función para calcular la nota referente al Aprendizaje en Contacto con el Docente [ACD] (Ponderado: 20%)
127 float calcularACD(){
128
129     int numeroActividades;
130     float notaActividad;
131     float notaAcumulativa=0;
132     float notaPromedio=0;
133 }
```



UNL

Universidad  
Nacional  
de Loja

1859

## FEIRNNR - Carrera de Computación

```
102
103 // Función para calcular la nota referente al Aprendizaje Practico Experimental [APE] (Ponderado: 25%)
104 float calcularAPE(){
105
106     int numeroActividades;
107     float notaActividad;
108     float notaAcumulativa=0;
109     float notaPromedio=0;
110
111     printf("Ingrese el numero de actividades para APE: ");
112     scanf("%i", &numeroActividades);
113
114     for (int i = 1; i <= numeroActividades; i++){
115
116         int iNota = 0;
117
118         while (iNota < 1){
119
120             printf("Ingrese la nota de la actividad APE %i : ", i);
121             scanf("%f", &notaActividad);
122
123             if (notaActividad >= 0 && notaActividad <= 10){
124                 iNota++;
125             }
126
127             else{
128                 printf("La nota debe estar entre 0 y 10. Intente nuevamente.\n");
129             }
130         }
131
132         notaAcumulativa = notaActividad + notaAcumulativa;
133     }
134
135 // Se obtiene el promedio y su porcentaje equivalente
136 notaPromedio = (notaAcumulativa/numeroActividades);
137 notaPromedio = notaPromedio * 0.25;
138
139 printf("Nota APE: %f \n", notaPromedio);
140
141 return notaPromedio;
142 }
143
144 // Función para calcular la nota referente al Aprendizaje Autonomo [AA] (Ponderado: 20%)
145 float calcularAA(){
146
147     int numeroActividades;
148     float notaActividad;
149     float notaAcumulativa=0;
150     float notaPromedio=0;
151
152     printf("Ingrese el numero de actividades para AA: ");
153     scanf("%i", &numeroActividades);
154
155     for (int i = 1; i <= numeroActividades; i++){
156
157         int iNota = 0;
158
159         while (iNota < 1) {
160
161             printf("Ingrese la nota de la actividad AA %i : ", i);
162             scanf("%f", &notaActividad);
163
164             if (notaActividad >= 0 && notaActividad <= 10){
165                 iNota++;
```



UNL

Universidad  
Nacional  
de Loja

1859

FEIRNNR - Carrera de Computación

```
166 }  
167 }  
168 else{  
169     printf("La nota debe estar entre 0 y 10. Intente nuevamente.\n");  
170 }  
171 }  
172 notaAcumulativa = notaActividad + notaAcumulativa;  
173 }  
174  
175 // Se obtiene el promedio y su porcentaje equivalente  
176 notaPromedio = (notaAcumulativa/numeroActividades);  
177 notaPromedio = notaPromedio * 0.2;  
178  
179 printf("Nota AA: %f \n", notaPromedio);  
180  
181 return notaPromedio;  
182 }  
183  
184 // Función para calcular las notas referentes a la Evaluación Sumativa (Ponderado: 35% del total de la unidad)  
185 float calcularES(){  
186  
187     float portafolio;  
188     float evaluacion;  
189     float portafolioPorcentaje;  
190     float evaluacionPorcentaje;  
191     float notaAcumulativa=0;  
192     float notaPromedio=0;  
193     int iNota = 0;  
194  
195     while (iNota < 1) {  
196  
197         printf("Ingrese la nota de su portafolio digital: ");  
198         scanf("%f", &portafolio);  
199  
200         if (portafolio >= 0 && portafolio <= 10){  
201             iNota++;  
202         }  
203  
204         else{  
205             printf("La nota debe estar entre 0 y 10. Intente nuevamente.\n");  
206         }  
207     }  
208  
209     while (iNota < 2) {  
210  
211         printf("Ingrese la nota de su Evaluación formativa: ");  
212         scanf("%f", &evaluacion);  
213  
214         if (evaluacion >= 0 && evaluacion <= 10){  
215             iNota++;  
216         }  
217  
218         else{  
219             printf("La nota debe estar entre 0 y 10. Intente nuevamente.\n");  
220         }  
221     }  
222  
223  
224 // Cálculo interno de la nota de la evaluación sumativa  
225 portafolioPorcentaje = portafolio * 0.4;  
226 evaluacionPorcentaje = evaluacion * 0.6;  
227  
228 notaAcumulativa = portafolioPorcentaje + evaluacionPorcentaje;  
229  
230 // Se obtiene el promedio y su porcentaje equivalente  
231 notaPromedio = notaAcumulativa * 0.35;  
232  
233 printf("Nota ES: %f \n", notaPromedio);  
234  
235 return notaPromedio;  
236 }
```



**UNL**

Universidad  
Nacional  
de Loja

FEIRNNR - Carrera de Computación

El programa ejecutado:

```
PS C:\Users\USUARIO\Documents\Programacion\Lenguaje C\Unidad 3 TD\calculoAsignatura> ./main.exe
```

```
Unidad 1
```

```
Ingrese el numero de actividades para ACD: 2
Ingrese la nota de la actividad ACD 1 :9
Ingrese la nota de la actividad ACD 2 :8
Nota ACD: 1.700000
Ingrese el numero de actividades para APE: 2
Ingrese la nota de la actividad APE 1 :8
Ingrese la nota de la actividad APE 2 :8
Nota APE: 2.000000
Ingrese el numero de actividades para AA: 2
Ingrese la nota de la actividad AA 1 :9
Ingrese la nota de la actividad AA 2 :10
Nota AA: 1.900000
Ingrese la nota de su portafolio digital: 9
Ingrese la nota de su Evaluaci|n formativa: 8
Nota ES: 2.940000
```

```
Unidad 2
```

```
Ingrese el numero de actividades para ACD: 2
Ingrese la nota de la actividad ACD 1 :7
Ingrese la nota de la actividad ACD 2 :7
Nota ACD: 1.400000
Ingrese el numero de actividades para APE: 2
Ingrese la nota de la actividad APE 1 :6
Ingrese la nota de la actividad APE 2 :8
Nota APE: 1.750000
Ingrese el numero de actividades para AA: 2
Ingrese la nota de la actividad AA 1 :8
Ingrese la nota de la actividad AA 2 :8
Nota AA: 1.600000
Ingrese la nota de su portafolio digital: 7
Ingrese la nota de su Evaluaci|n formativa: 7
Nota ES: 2.450000
```



### Unidad 3

```
Ingrese el numero de actividades para ACD: 2
Ingrese la nota de la actividad ACD 1 :10
Ingrese la nota de la actividad ACD 2 :9
Nota ACD: 1.900000
Ingrese el numero de actividades para APE: 2
Ingrese la nota de la actividad APE 1 :9
Ingrese la nota de la actividad APE 2 :9
Nota APE: 2.250000
Ingrese el numero de actividades para AA: 2
Ingrese la nota de la actividad AA 1 :8
Ingrese la nota de la actividad AA 2 :10
Nota AA: 1.800000
Ingrese la nota de su portafolio digital: 10
Ingrese la nota de su Evaluaci|n formativa: 9
Nota ES: 3.290000
```

Su nota final de la asignatura es 8.326667 La cual se considera Buena  
PS C:\Users\USUARIO\Documents\Programacion\Lenguaje C\Unidad 3 TD\calculoAsignatura> []