



Reporte Técnico de Actividades Práctico-Experimentales Nro. 001

1. Datos de Identificación del Estudiante y la Práctica

Nombre del estudiante(s)	Jeancarlos Fernando Aguirre Romero
Asignatura	Teoría de la programación
Ciclo	1 A
Unidad	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.	00X1
Tipo	Individual
Título de la Práctica	Aplicación de estructuras condicionales en la resolución de problemas.
Nombre del Docente	Lisette Geoconda López Faicán
Fecha	Jueves 20 de noviembre del 2025
Horario	10h30 – 13h30
Lugar	Aula física asignada al paralelo.
Tiempo planificado en el Sílabo	1 hora



2. Objetivo(s) de la Práctica

- Comprender y aplicar las estructuras condicionales simples, dobles y múltiples en la resolución de problemas.
- Diseñar y codificar un algoritmo que utilice sentencias de decisión para analizar y clasificar información.
- Validar el funcionamiento del programa mediante la ejecución práctica.

3. Materiales, Reactivos, Equipos y Herramientas

- Herramientas de modelado de diagrama de flujo (PSeInt, Draw.io, Lucid chart, otros)
- 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.

4. Procedimiento / Metodología Ejecutada

Metodología de aprendizaje: aprendizaje basado en problemas.

- Presentación del objetivo de la práctica: Explicación de la relevancia de aplicar estructuras condicionales en la codificación de programas.
- Contextualización del problema: Basado en el ejercicio del “Cálculo de la nota final de la Unidad 1 mediante estructuras secuenciales en C”, mostrar el nivel de desempeño del promedio de la unidad 1, aplicando las siguientes condiciones:

El Problema Planteado para esta Actividad es el siguiente:

Un debe diseñar, programar y validar un programa en lenguaje C que realice el cálculo correcto de la nota final de la Unidad 1 y muestre los resultados con claridad.

1. El nombre del programa (archivo .c) y su compilación (archivo.exe) deben contener el nombre de evaluacionunidad1 escritos en minúsculas y sin espacios. Ejemplo: evaluacionunidad1.c y evaluacionunidad1.exe
2. Los nombres de las variables y los mensajes mostrados al usuario deben ser claros, descriptivos y coherentes con el propósito del programa, de modo que faciliten su comprensión.
3. La estructura del programa debe estar organizada y documentada mediante comentarios que separen las siguientes secciones:
 - // Definición de variables
 - // Datos de entrada
 - // Proceso
 - // Datos de salida
4. Utilizar únicamente estructuras secuenciales.
5. Las notas de cada actividad individual se califican sobre 10 puntos.
6. La nota final se calcula sobre 10 puntos, sumando los ponderados de cada componente (ACD, APE, AA y ES).
 - $ACD (2) + APE (2.5) + AA (2) + ES (3.5) = 10$ puntos.
7. Para cada componente (ACD, APE y AA), el programa debe calcular el promedio simple de las actividades y multiplicarlo por su ponderado.
8. La evaluación sumativa (ES) se calcula aplicando los pesos internos.
9. Finalmente, el programa debe mostrar:
 - La nota ponderada de cada componente (ACD, APE, AA y ES).
 - La nota final total de la Unidad 1.

//Este algoritmo se había realizado anteriormente como parte de la evolución final de la unidad número 1, sin embargo, se ha decidido mejorarlo a través de condicionales las cuales se le aplican al promedio final del algoritmo siguiendo las siguientes condiciones:

- Nota $\geq 9 \rightarrow$ "Excelente"
- Nota ≥ 7 y $< 9 \rightarrow$ "Bueno"
- Nota ≥ 5 y $< 7 \rightarrow$ "Regular"
- Nota $< 5 \rightarrow$ "Deficiente"



A partir de aquí se mostrará el proceso del desarrollo de esta mejora

La solución del problema se desarrolla mediante el diseño de un algoritmo y su posterior implementación en un lenguaje de programación, empleando estructuras condicionales. Para ello, seguir los siguientes pasos:

Pasos para la resolución de esta Práctica:

1. Lo primero que debemos hacer es reconocer los datos Involucrados dentro del algoritmo

- **Datos de Entrada:** algoritmos, lenguajeC, herramientas, instalacionLenguajes, construccionAlgoritmos, disenoAlgoritmosm, portafolioDigital, problema

- **Datos de Proceso y formulas:** certamen3, n

$$ACD = (\text{algoritmos} + \text{lenguajeC}) / 2 ;$$

$$AA = (\text{herramientas} + \text{instalacionLenguajes}) / 2 ;$$

$$APE = (\text{construccionAlgoritmos} + \text{disenoAlgoritmos}) / 2 ;$$

$$ES = (\text{portafolioDigital} * 0.4) + (\text{problema} * 0.6);$$

$$\text{porcentajeACD} = ACD/10 ;$$

$$\text{porcentajeTotalACD} = \text{porcentajeACD} * 2;$$

$$\text{porcentajeAA} = AA/10;$$

$$\text{porcentajeTotalAA} = \text{porcentajeAA} * 2;$$

$$\text{porcentajeAPE} = APE/10;$$

$$\text{porcentajeTotalAPE} = \text{porcentajeAPE} * 2.5;$$

$$\text{porcentajeES} = ES / 10;$$

$$\text{porcentajeTotalES} = \text{porcentajeES} * 3.5;$$

$$\text{notaFinal} = \text{porcentajeTotalACD} + \text{porcentajeTotalAA} + \text{porcentajeTotalAPE} + \text{porcentajeTotalES};$$

```
if (notaFinal >= 9)
```

```
{
```

```
    strcpy(escala, "Excelente!");
```

```
    getchar();
```

```
}else if (notaFinal >= 7 && notaFinal < 9)
```

```
{
```

```
    strcpy(escala, "Buena");
```

```
    getchar();
```

```
}else if (notaFinal >= 5 && notaFinal < 7)
{
    strcpy(escala, "Regular");
    getch();

}else
{
    strcpy(escala, "Deficiente");
    getch();
}
```

- **Datos de Salida:** porcentajeTotalACD, porcentajeTotalAA, porcentajeTotalAPE, porcentajeTotalES, escala
2. Una vez reconocidos los datos Utilizados dentro del código procederemos a realizar el respectivo código en PSeInt para obtener el nuestro Diagrama de flujo solicitado para esta Práctica.

(EL RESPECTIVO DIAGRAMA DE FLUJO SE MUESTRA EN LA SECCION 5. CORRESPONDIENTE A LOS RESULTADOS)

3. Para después de ello proceder a implementar estas mejoras y el respectivo código en el lenguaje de programación C y Hacer la respectiva comprobación con las notas obtenidas durante la unidad

Para ello procederemos a escribir las líneas de código correspondientes a las condicionales, solicitadas dentro del ejercicio:

(EL RESPECTIVO CODIGO EN EL LENGUAJE C SE MUESTRA EN LA SECCION 5. CORRESPONDIENTE A LOS RESULTADOS)

5. Resultados

- Diagrama de flujo.

Algoritmo evaluacionUnidad1

Definir ACD, algoritmo...

Definir AA, herramient...

Definir APE, construcc...

Definir EvS, portafoli...

Definir notaFinal Como...

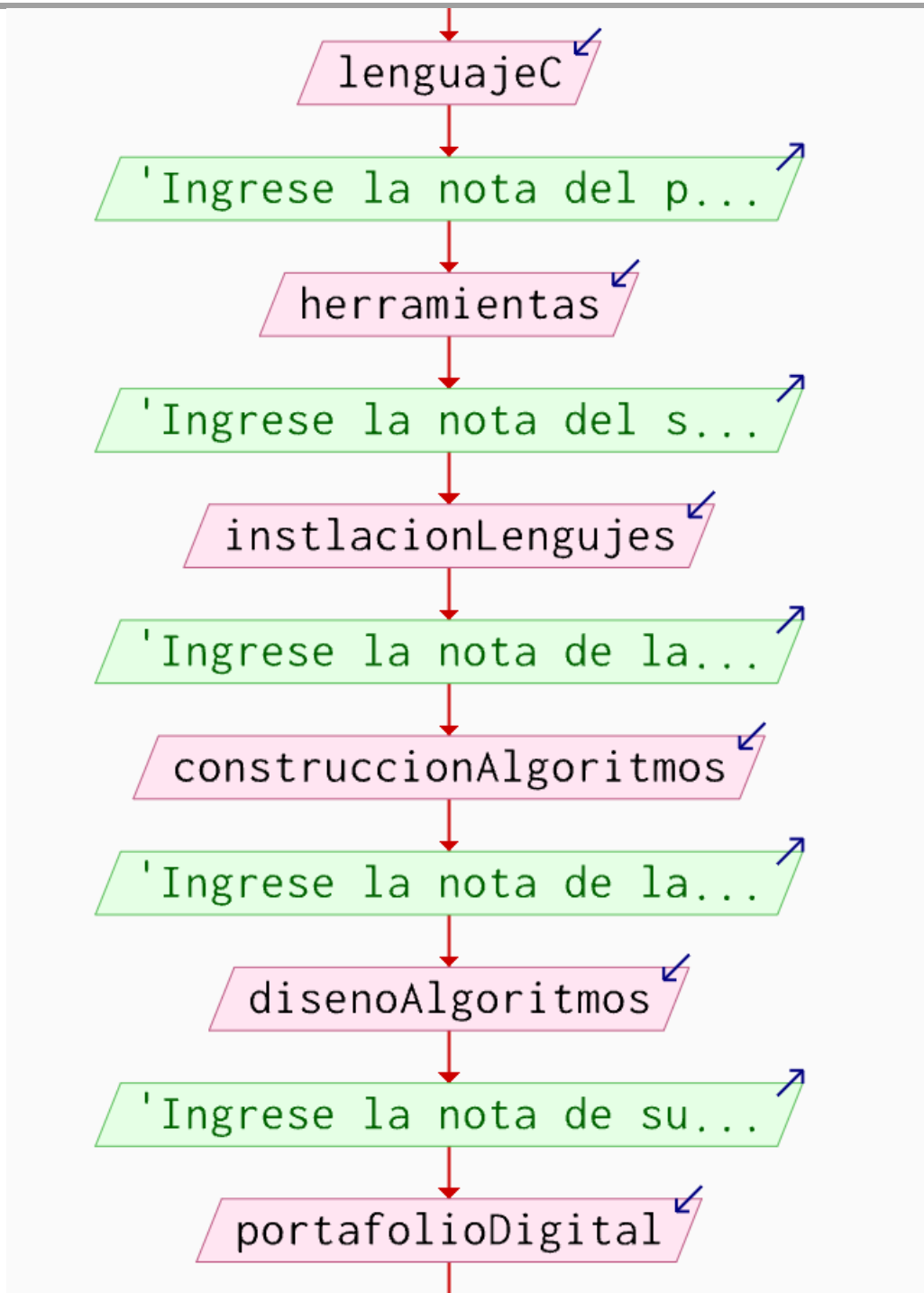
Definir escala Como Ca...

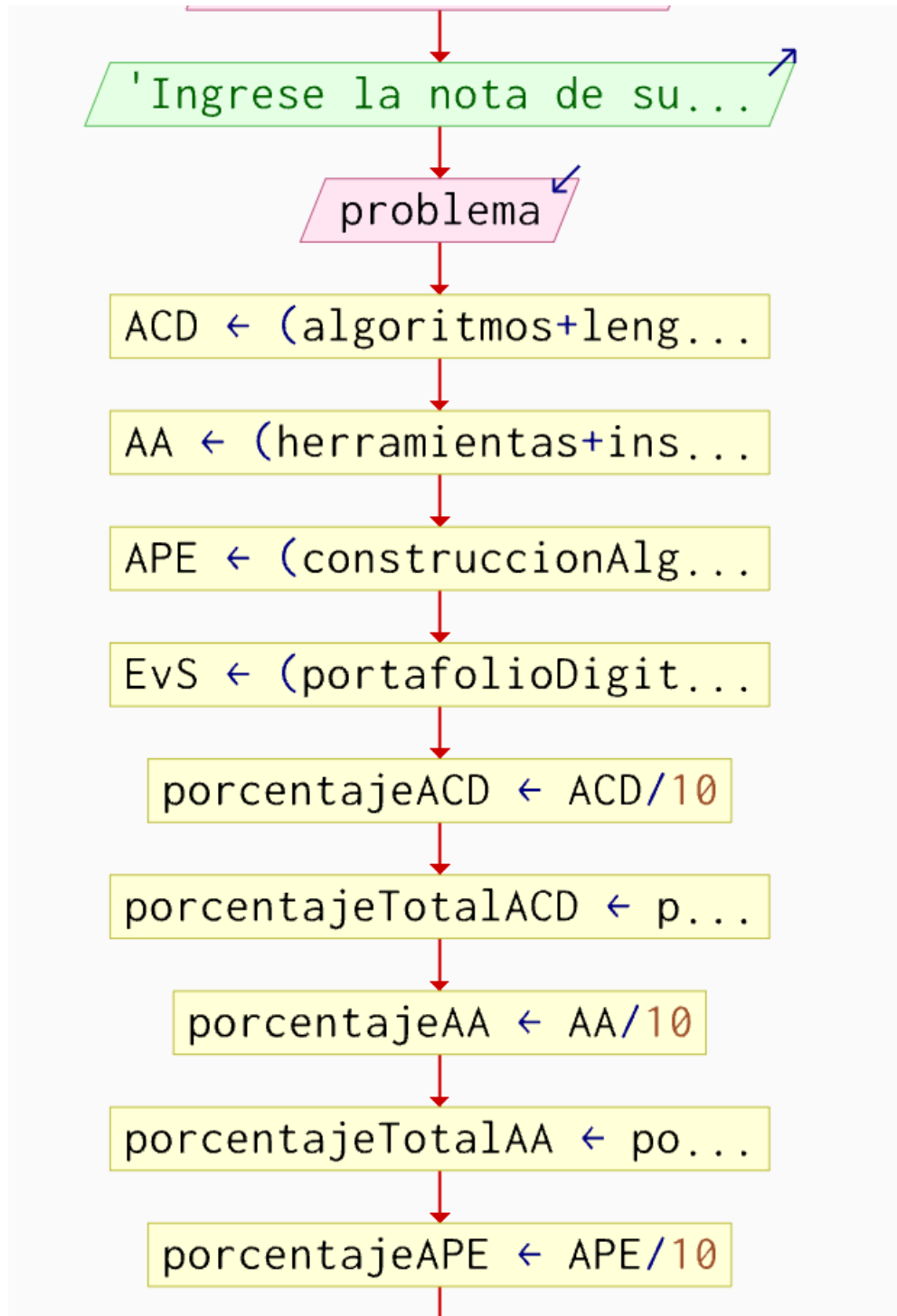
'Ingrese la nota de la...

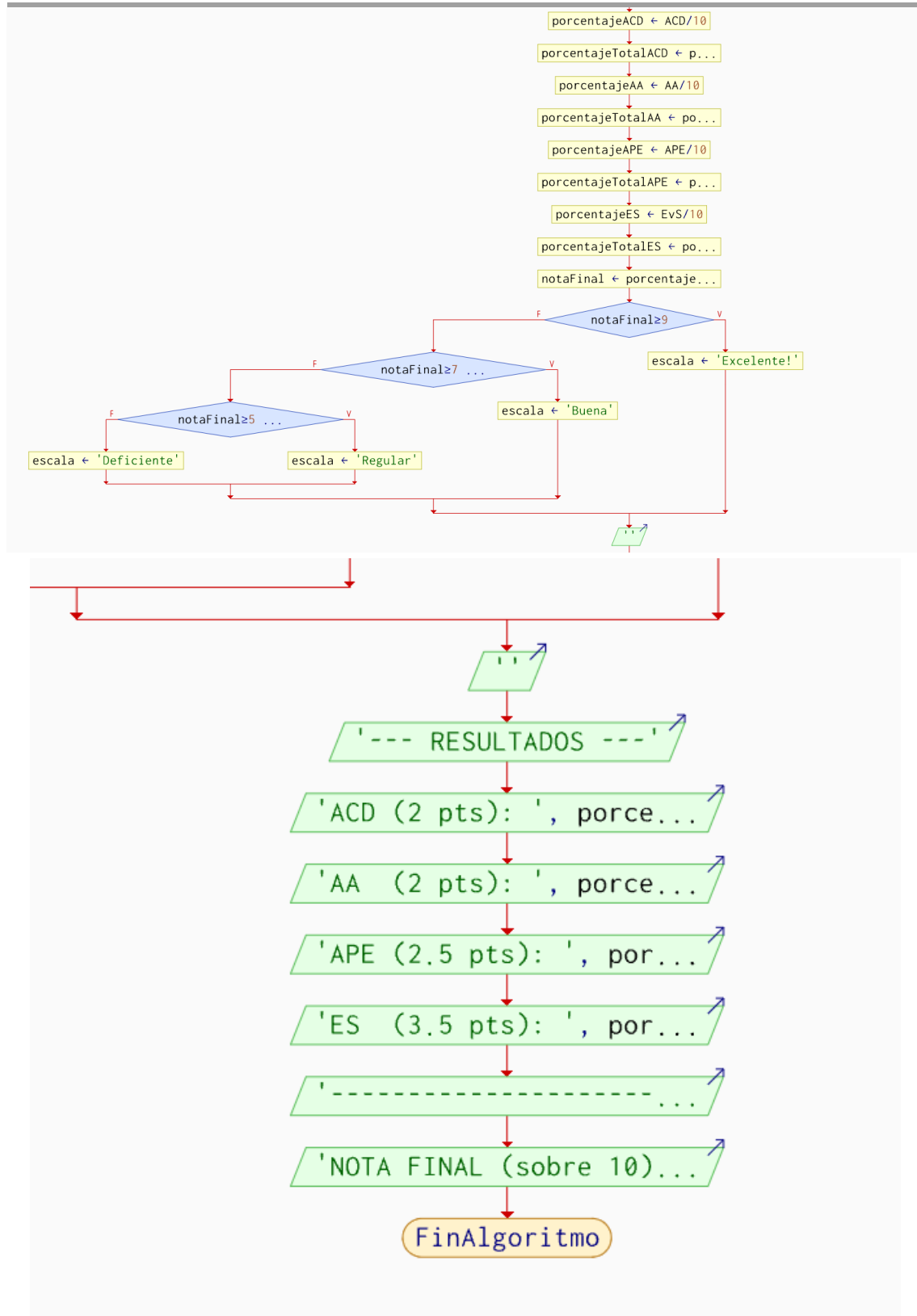
algoritmos

'Ingrese la nota de la...

lenguajeC







- Código fuente en lenguaje C.



```
#include <stdio.h>
#include <string.h>

int main() {

    float ACD, algoritmos, lenguajeC, porcentajeACD, porcentajeTotalACD ;

    float AA, herramientas, instalacionLenguajes, porcentajeAA,
    porcentajeTotalAA;

    float APE, construccionAlgoritmos, disenoAlgoritmos, porcentajeAPE,
    porcentajeTotalAPE;

    float ES, portafolioDigital, problema, porcentajeES,
    porcentajeTotalES, portafolioDigitalTotal, problemaTotal ;

    float notaFinal;

    char escala[30];

    printf("Ingrese la nota de la primera actividad ACD referente a los
    algoritmos en Pseudocodigo : \n");
    scanf("%f", &algoritmos);

    printf("Ingrese la nota de la segunda actividad ACD referente a la
    programacion en C : \n");
    scanf("%f", &lenguajeC);

    printf("Ingrese la nota del primer deber AA referente a las
    herramientas digitales para Pseudocodigo : \n");
    scanf("%f", &herramientas);

    printf("Ingrese la nota del segundo deber AA referente a la Instalacion
    de Lenguajes de Programacion : \n");
    scanf("%f", &instalacionLenguajes);

    printf("Ingrese la nota de la primera practica APE referente a la
    construccion de algoritmos con estructura secuencial : \n");
    scanf("%f", &construccionAlgoritmos);

    printf("Ingrese la nota de la segunda practica APE referente al diseño
    de algoritmos con estructura secuencial a la construccion del programa:
    \n");
    scanf("%f", &disenoAlgoritmos);

    printf("Ingrese la nota de su portafolio digital como ES: \n");
    scanf("%f", &portafolioDigital);
```

```
printf("Ingrese la nota de su evaluacion sumativa como ES: \n");
scanf("%f", &problema);

ACD = (algoritmos + lenguajeC) / 2 ;

AA = (herramientas + instalacionLenguajes) / 2 ;

APE = (construccionAlgoritmos + disenoAlgoritmos) / 2 ;

ES = (portafolioDigital * 0.4) + (problema * 0.6);

porcentajeACD = ACD/10 ;
porcentajeTotalACD = porcentajeACD * 2;

porcentajeAA = AA/10;
porcentajeTotalAA = porcentajeAA * 2;

porcentajeAPE = APE/10;
porcentajeTotalAPE = porcentajeAPE * 2.5;

porcentajeES = ES / 10;
porcentajeTotalES = porcentajeES * 3.5;

notaFinal = porcentajeTotalACD + porcentajeTotalAA +
porcentajeTotalAPE + porcentajeTotalES;

if (notaFinal >= 9)
{
    strcpy(escala, "Excelente!");
    getchar();

}else if (notaFinal >= 7 && notaFinal < 9)
{
    strcpy(escala, "Buena");
    getchar();

}else if (notaFinal >= 5 && notaFinal < 7)
{
    strcpy(escala, "Regular");
    getchar();

}else
{
    strcpy(escala, "Deficiente");
    getchar();
}
```

```
printf("\n--- RESULTADOS ---\n");
printf("ACD (2 pts): %f\n", porcentajeTotalACD);
printf("AA (2 pts): %f\n", porcentajeTotalAA);
printf("APE (2.5 pts): %f\n", porcentajeTotalAPE);
printf("ES (3.5 pts): %f\n", porcentajeTotalES);
printf("-----\n");
printf("NOTA FINAL (sobre 10): %f la cual se considera %s \n",
notaFinal, escala);

return 0;
}
```

- Pruebas.

```
PS C:\Users\USUARIO\Documents\Programacion\Lenguaje C> gcc MEJORADO_evaluacionunidad1.c -o MEJORADO_evaluacionunidad1
PS C:\Users\USUARIO\Documents\Programacion\Lenguaje C> ./MEJORADO_evaluacionunidad1.exe
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 dise o 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): 1.700000
AA (2 pts): 2.000000
APE (2.5 pts): 2.437500
ES (3.5 pts): 2.415000
-----
NOTA FINAL (sobre 10): 8.552500 la cual se considera Buena
PS C:\Users\USUARIO\Documents\Programacion\Lenguaje C> 
```

6. Preguntas de Control

-  Qu  es una condici n y cu l es su funci n dentro de una estructura condicional en programaci n?

Una condici n es una expresi n l gica que puede evaluarse como verdadera o falsa. Y su funci n dentro de una estructura condicional es permitir que el programa tome decisiones y ejecute instrucciones espec ficas seg n el valor dado, controlando as  el flujo del algoritmo.

-  Qu  diferencia existe entre una estructura condicional simple, doble y m ltiple?

Una estructura condicional simple ejecuta una acci n  nicamente cuando la condici n es verdadera. Una estructura condicional doble (if-else) permite ejecutar una acci n cuando la condici n es

verdadera y otra distinta cuando es falsa. Una estructura condicional múltiple permite evaluar varias condiciones en orden, ejecutando la primera que resulte verdadera.

- ¿Qué es una estructura condicional anidada y en qué casos resulta útil emplearla dentro de un programa?

Es una estructura en la que una sentencia condicional se encuentra dentro de otra. Resulta útil cuando se requiere tomar decisiones más complejas o cuando el resultado de una condición depende del resultado de otra, permitiendo evaluar diversos escenarios de manera jerárquica.

7. Conclusiones

La práctica permitió comprender de manera más clara cómo aplicar estructuras condicionales para complementar un algoritmo previamente elaborado con estructuras secuenciales. Al integrar decisiones basadas en el promedio final, se fortaleció la capacidad para analizar y clasificar información dentro de un programa en C, cumpliendo así con los objetivos de diseño, codificación y validación establecidos en la guía. Además, se reafirmó la importancia de organizar el código, usar variables descriptivas y seguir un método ordenado para resolver problemas mediante programación estructurada.

8. Recomendaciones

Se recomienda continuar practicando con ejercicios que combinen estructuras secuenciales y condicionales, incrementando progresivamente la complejidad de los problemas. Esto ayudará a consolidar el dominio de la lógica de programación y permitirá desarrollar algoritmos más robustos, claros y eficientes, aplicables tanto en actividades académicas como en situaciones reales de programación.

9. Bibliografía

[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>