Reporte Técnico de Actividades Práctico-Experimentales Nro. 002

1. Datos de Identificación del Estudiante y la Práctica

Nombre del estudiante(s)	Emerson Chamba Galarza
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

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).
- 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

Inicio

En esta etapa se presentó el objetivo general de la práctica, orientado a comprender y aplicar correctamente las estructuras repetitivas para resolver problemas de tipo iterativo.

Se explicó la importancia del uso de bucles (*for, while, do...while*) para automatizar procesos que requieren repetición controlada.

Posteriormente se contextualizó el problema, tomando como referencia el ejercicio previo del cálculo de la nota final de la Unidad 1 mediante estructuras secuenciales. En esta nueva versión se necesitó automatizar dicho proceso para múltiples estudiantes utilizando estructuras repetitivas. Para ello, se establecieron los siguientes requisitos:

- Ingresar la cantidad total de estudiantes y ejecutar un bucle para repetir el proceso de cálculo.
- Solicitar en cada iteración los valores de ACD, APE, AA y ES.
- Validar que las notas estén entre **0 y 10**, solicitando nuevos valores en caso de error.
- Mostrar la nota final y la clasificación antes de continuar con el siguiente estudiante.
- Procesar únicamente resultados individuales sin necesidad de almacenar datos.

Desarrollo

El desarrollo se centró en la creación de un **algoritmo estructurado** y su posterior codificación en lenguaje C. Las actividades se organizaron así:

1. **Análisis del problema**
    - **Entradas:** ACD1, ACD2, APE1, APE2, AA1, AA2, ES1, ES2 y número total de estudiantes.
    - **Proceso:** validación de notas, cálculo de ponderaciones, repetición del flujo por estudiante, clasificación final.
    - **Salida:** nota final por estudiante, clasificación cualitativa y promedio general del grupo.
2. **Diseño del algoritmo**

    Se elaboró un esquema lógico simplificado que detalla:
    - Lectura del número de estudiantes.
    - Uso del ciclo *for* para repetir el cálculo.
    - Validación mediante *while*.
    - Procesamiento de promedios y clasificación.

3. **Codificación en C**

    Se implementó el algoritmo utilizando:
    - Bucles *for* y *while*.

- Condicionales *if–else*.
- Variables acumuladoras para promedios grupales.
- Mensajes de retroalimentación y control de errores. *(El código completo se incluye en la sección correspondiente del informe)*.

4. Pruebas

Se ejecutó el programa en el IDE para verificar su funcionamiento con tres casos de prueba con resultados reales.

Las capturas mostradas en el informe evidencian:

- La correcta ejecución del bucle para múltiples estudiantes.
- El cálculo exacto de notas ponderadas.
- La clasificación automática basada en el resultado final.
- El promedio general del curso calculado al finalizar la iteración.

5. Resultados

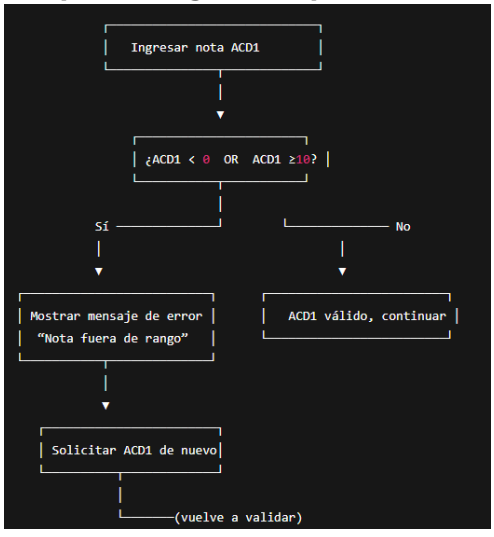
Contextualización del problema:

El ejercicio se basa en automatizar el proceso del "Cálculo de la nota final de la Unidad 1 mediante estructuras secuenciales en C", incorporando ahora estructuras repetitivas para procesar las notas de varios estudiantes.

Se requiere que el programa:

- Permita ingresar la cantidad total de estudiantes.
- Use un bucle para repetir el proceso de lectura de calificaciones y cálculo de la nota final para cada estudiante.
- Solicite los valores correspondientes a los componentes evaluativos: ACD, APE, AA y ES.
- Valide que todas las notas ingresadas estén en el rango permitido (0 a 10).
  En caso de error, debe mostrar un mensaje y solicitar nuevamente el dato.
- Calcule la nota final ponderada según los porcentajes establecidos.
- Muestre la nota final y clasificación de cada estudiante antes de continuar con el siguiente.
- No requiere almacenar notas; solo procesar y mostrar los resultados en cada iteración.

Esquema lógico simplificado:



Código fuente en lenguaje C:

```c
#include <stdio.h>

int main() {
     // Entrada

    float acd1, acd2, ape1, ape2, aa1, aa2, es1, es2;

    float nota_final_unidad1;

    float promedio_acd, promedio_ape, promedio_aa;

    float nota_es_sin_pond;

    char *clasificacion;

    int i, estudiantes;

    float suma_nota_finales = 0.0;
    float  promedio_nota_finales = 0.0;

    printf("Ingrese el numero de estudiantes: ");
    scanf("%i", &estudiantes);

    for (i = 0; i < estudiantes; i++) {
        printf("\n--- Ingrese Nota del Estudiante %i ---\n", i + 1);


    printf("\n--- Calcular la Nota Final de la Unidad 1 del Estudiante %i ---\n", i + 1);

    printf("Ingrese la nota de ACD 1: ");
    scanf("%f", &acd1);
    while (acd1 < 0.0 || acd1 >= 10.0) {
        printf("Nota fuera de rango, Ingrese nuevamente ACD 1: ");
        scanf("%f", &acd1);
    }

    printf("Ingrese la nota de ACD 2: ");
    scanf("%f", &acd2);
    while (acd2 < 0.0 || acd2 >= 10.0) {
        printf("Nota fuera de rango, Ingrese nuevamente ACD 2: ");
        scanf("%f", &acd2);
    }
    printf("Ingrese la nota de APE 1: ");
    scanf("%f", &ape1);
    while (ape1 < 0.0 || ape1 >= 10.0) {
```

```
    printf("Nota fuera de rango, Ingrese nuevamente APE 1: ");
    scanf("%f", &ape1);
}
printf("Ingrese la nota de APE 2: ");
scanf("%f", &ape2);
while (ape2 < 0.0 || ape2 >= 10.0) {
    printf("Nota fuera de rango, Ingrese nuevamente APE 2: ");
    scanf("%f", &ape2);
}
printf("Ingrese la nota de AA 1: ");
scanf("%f", &aa1);
while (aa1 < 0.0 || aa1 >= 10.0) {
    printf("Nota fuera de rango, Ingrese nuevamente AA 1: ");
    scanf("%f", &aa1);
}
printf("Ingrese la nota de AA 2: ");
scanf("%f", &aa2);
while (aa2 < 0.0 || aa2 >= 10.0) {
    printf("Nota fuera de rango, Ingrese nuevamente AA 2: ");
    scanf("%f", &aa2);
}


printf("Ingrese la nota de ES 1: ");
scanf("%f", &es1);
while (es1 < 0.0 || es1 >= 10.0) {
    printf("Nota fuera de rango, Ingrese nuevamente ES 1: ");
    scanf("%f", &es1);
}

printf("Ingrese la nota de ES 2: ");
scanf("%f", &es2);
while (es2 < 0.0 || es2 >= 10.0) {
    printf("Nota fuera de rango, Ingrese nuevamente ES 2: ");
    scanf("%f", &es2);

}
// Promedios

promedio_acd = (acd1 + acd2)/2 * 0.2;
promedio_ape = (ape1 + ape2)/2 * 0.25;
promedio_aa = (aa1 + aa2)/2 * 0.2;


nota_es_sin_pond =( (es1 * 0.4) + (es2 * 0.6) ) * 0.35;

nota_final_unidad1 = promedio_acd + promedio_ape + promedio_aa + nota_es_sin_pond;
```

```
    //Acumular Suma
    suma_nota_finales += nota_final_unidad1;


    // Clasificacion de Nota Final
    if (nota_final_unidad1 >= 9.0) {
        clasificacion = "Excelente";
        } else if (nota_final_unidad1 >= 7.0 && nota_final_unidad1 < 9.0) {
        clasificacion = "Bueno";
        } else if (nota_final_unidad1 >= 5.0 && nota_final_unidad1 < 7.0) {
        clasificacion = "Regular";
        } else {
        clasificacion = "Deficiente";
    }


    // Salida
    printf("\nResultados de la Unidad 1:\n");

    printf("Nota Ponderada ACD: %.2f\n", promedio_acd);
    printf("Nota Ponderada APE: %.2f\n", promedio_ape);
    printf("Nota Ponderada AA: %.2f\n", promedio_aa);
    printf("\nLa nota final de la Unidad 1 es: %.2f\n", nota_final_unidad1);
    printf("Clasificacion: %s\n", clasificacion);

    }
    // Promedio entre estudiantes
    if (estudiantes > 0) {
        promedio_nota_finales = suma_nota_finales / estudiantes;
    }

    printf("\nPromedio General de los %i Estudiantes es: %.2f\n", estudiantes,
promedio_nota_finales);

    return 0;
}
```

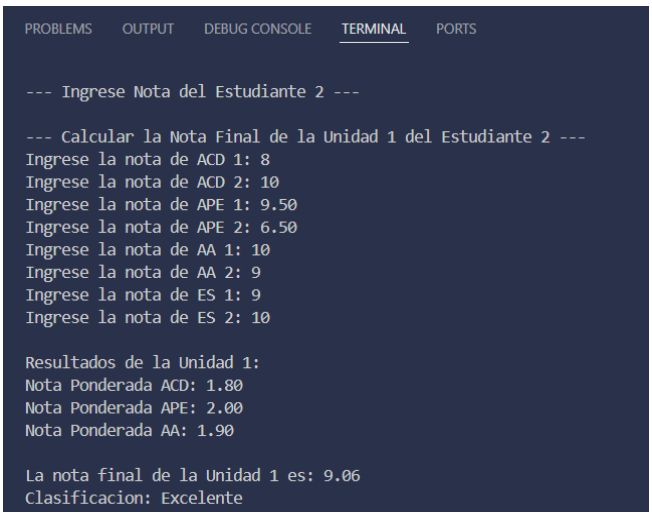
Pruebas:

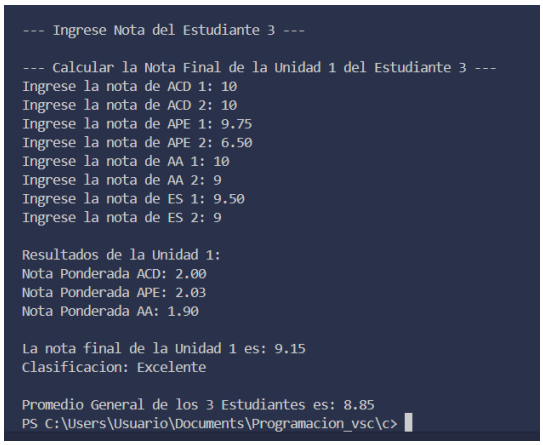
Estudiante	Entradas (ACD1, ACD2, APE1, APE2, AA1, AA2, Portafolio, Examen)	Cálculos Internos (según ponderaciones del código)	Nota Final	Clasificación
Roy	7, 10, 9.25, 6, 9.50, 9, 7, 9	ACD = 1.70 APE = 1.91 AA = 1.85 ES = 2.87	**8.33**	**Bueno**
Emerson	8, 10, 9.50, 6.50, 10, 9, 9, 10	ACD = 1.80 APE = 2.00 AA = 1.90 ES = 3.36	**9.06**	**Excelente**
Arlette	10, 10, 9.75, 6.50, 10, 9, 9.5, 9	ACD = 2.00 APE = 2.03 AA = 1.90 ES = 3.22	**9.15**	**Excelente**

Promedio General de los 3 estudiantes:

8.33 + 9.06 + 9.15 / 3 = **8.85**

Ejecución en C





6. Preguntas de Control

- **¿En qué se diferencia una estructura repetitiva de una condicional?**
  La estructura condicional (if) se usa para tomar decisiones y ejecuta un bloque de código una sola vez si se cumple la condición. La estructura repetitiva (for, while) se usa para automatizar tareas y ejecuta un bloque de código múltiples veces mientras la condición se mantenga verdadera.

- **¿Qué diferencia existe entre las estructuras for, while y do...while en cuanto a su funcionamiento y uso?**
  1. El bucle **for** se usa cuando se conoce el número exacto de repeticiones (bucle por contador, ej. número de estudiantes).
  2. El bucle **while** se usa cuando el número de repeticiones es indeterminado y la condición se evalúa al inicio.
  3. El bucle **do...while** se usa para garantizar que el código se ejecute al menos una vez antes de verificar la condición

- **¿Por qué es importante incluir validaciones dentro de un programa cuando se solicitan datos al usuario?**
  Es crucial para asegurar la integridad de los datos (ej. que la nota esté entre 0 y 10), prevenir errores de cálculo y hacer que el programa sea robusto,

forzando al usuario a ingresar información correcta antes de continuar con el procesamiento.

7. Conclusiones

La implementación exitosa de estructuras repetitivas (for y while) y condicionales (if/else) en el código C permitió automatizar eficientemente el cálculo de la nota final para múltiples estudiantes. El uso del bucle for gestionó la iteración sobre la cantidad total de estudiantes, mientras que el bucle do...while (o while anidado) fue esencial para validar robustamente que todas las notas ingresadas por el usuario se mantuvieran dentro del rango permitido (0 a 10), cumpliendo así con el objetivo de la práctica y garantizando la precisión de los resultados individuales y del promedio general.

8. Recomendaciones

Se recomienda extender la funcionalidad del programa para **almacenar y procesar los datos** de los estudiantes (nombre, notas individuales y nota final) utilizando **arrays o estructuras de datos** dentro de la próxima unidad. Esto permitiría la generación de reportes más complejos, como listados ordenados por calificación o el cálculo de la desviación estándar, añadiendo valor al sistema más allá del simple procesamiento iterativo.

9. Bibliografía / Referencias

[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-lenguajec?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