# Reporte Técnico de Actividades Práctico-Experimentales Nro. 002

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

| | |
|---|---|
| **Nombre del estudiante** | Kiara Salomé Condoy Morocho |
| **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 04 de diciembre del 2025 |
| **Horario** | 10h30 – 13h30 |
| **Lugar** | Aula física asignada al paralelo. |
| **Tiempo planificado en el Sílabo** | 6 horas |

## 2. Objetivos 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 diagram 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

El desarrollo del programa se realizó en los siguientes pasos:

1. **Apertura del archivo base y análisis del problema:**
   Se abrió el archivo del APE1, donde previamente se había implementado la lógica de cálculo de nota final con estructuras condicionales. Este sirvió como base para incorporar nuevas funcionalidades.

   Además, se identificaron:
   - **Entradas:** notas de ACD1, ACD2, APE1, APE2, AA1, AA2, Portafolio y Evaluación Final; cantidad de estudiantes.
   - **Proceso:** validación de notas, cálculo de ponderados, suma de componentes, asignación de escala cualitativa, acumulación de notas finales.
   - **Salidas:** nota final individual, escala cualitativa, promedio general del grupo.

2. **Validación de entradas:**
   Se implementó la verificación individual de cada nota ingresada (ACD1, ACD2, APE1, APE2, AA1, AA2, Portafolio y Evaluación Final).
   - Si alguna nota estaba fuera del rango permitido (0 a 10), el programa mostraba un mensaje de error.
   - Solo la nota incorrecta se volvía a solicitar, sin afectar las demás.
3. **Ingreso de cantidad de estudiantes:**
   Se agregó una entrada para definir cuántos estudiantes se iban a procesar.
   - Se utilizó un bucle *for* para repetir el proceso de ingreso de notas, validación y cálculo de la nota final para cada estudiante.
4. **Cálculo del promedio general:**
   Se acumuló la nota final de cada estudiante en una variable suma.
   - Al finalizar el bucle, se calculó el promedio general dividiendo la suma entre el número total de estudiantes.
   - Este promedio se mostró como resultado final del programa.
5. **Validación a través de la ejecución:**
   Se definieron al menos tres casos de prueba con diferentes cantidades de estudiantes y combinaciones de notas válidas e inválidas.
   - Se verificó que el programa cumpliera con las condiciones de validación, repetición y cálculo correcto de resultados.
6. **Documentación**:
   Se elaboró el informe técnico en formato PDF, siguiendo la estructura establecida: portada, introducción, análisis del problema, diseño del algoritmo, codificación, pruebas de escritorio, resultados y conclusiones.

# 5. Resultados

1. **Contextualización del problema:**

   Basado en el ejercico del "Cálculo de la nota final de la Unidad 1 mediante estructuras secuenciales en C", se desea automatizar el proceso de cálculo para varios estudiantes utilizando estructuras repetitivas:

   ○ El programa debe permitir ingresar la cantidad total de estudiantes, y mediante un bucle, repetir el proceso de lectura de calificaciones y cálculo de la nota final.

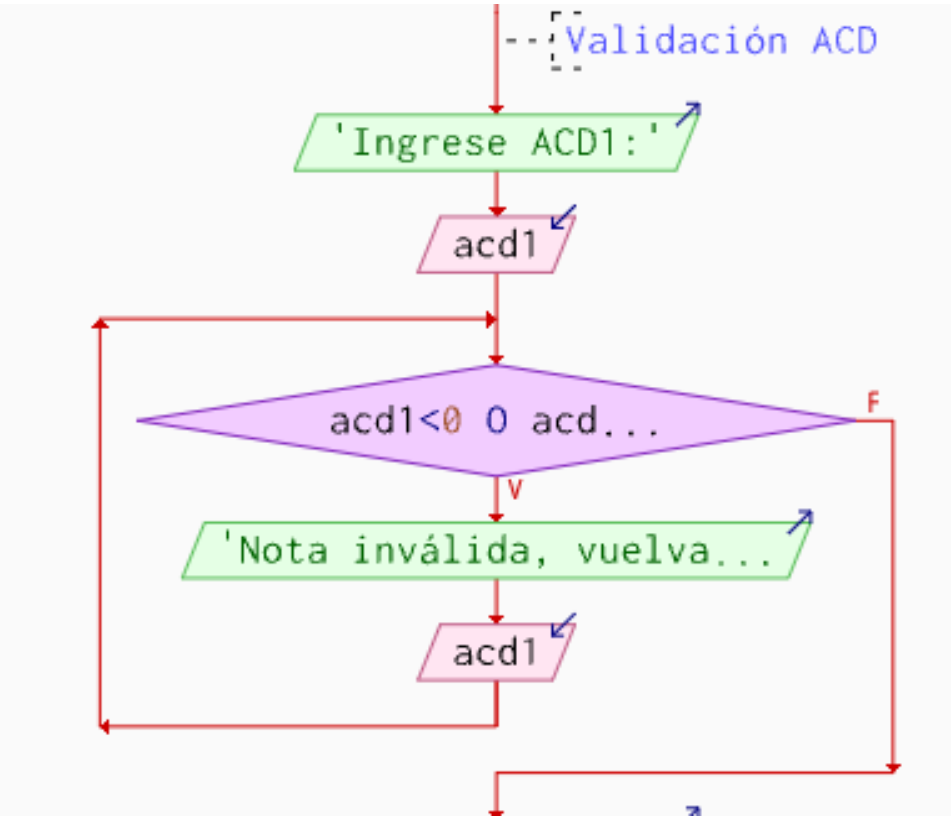
   ○ En cada repetición, el programa solicitará los valores de los componentes (ACD, APE, AA y ES), calculará la nota final y mostrará el resultado antes de pasar al siguiente estudiante.

   ○ Además, el programa debe validar que las notas ingresadas estén dentro del rango permitido (0 a 10). Si el usuario ingresa una nota fuera de este rango, el programa mostrará un mensaje de error y volverá a solicitar el dato hasta que sea correcto.

   ○ No se requiere guardar las notas; el programa únicamente procesará y mostrará el resultado individual en cada iteración.

2. **Esquema lógico simplificado:**

   Diagrama de flujo del bucle repetitivo para validar que la nota ingresada para el ACD 1 sea válida



## 3. Código fuente en lenguaje C:

```c
#include <stdio.h>

int main (){

    //Definición de variables
    float acd1, acd2, ape1, ape2, aa1, aa2, evaluacion, portafolio;
    float acdPonderado, apePonderado, aaPonderado, esPonderado, notaFinal, suma = 0, promedio;
    char *escala, *notaInvalida = "Nota invalida, vuelva a ingresar la calificacion (0-10)";
    int alumnos, i;

    //Datos de Entrada
    //Alumnos a calcular el promedio
    printf("De cuantos alumnos deseas calcular el promedio?\n");
    scanf("%i", &alumnos);

    //Cálculo de promedio individual
    for ( i = 1; i <= alumnos; i++)
    {
        printf("\nA continuacion, ingrese las notas (0-10) para los siguientes parametros del alumno %i:\n", i);

        printf("Ingrese las notas del Aprendizaje en Contacto con el Docente (ACD) 1 y 2 del alumno %i\n", i);
        scanf("%f, %f", &acd1, &acd2);
        while (acd1>10 || acd1<0 || acd2>10 || acd2<0){
            if(acd1>10 || acd1<0){
                printf(notaInvalida);
                printf(" del ACD1\n");
                scanf("%f", &acd1);
                }
            else if(acd2>10 || acd2<0){
                printf(notaInvalida);
                printf(" del ACD2\n");
                scanf("%f", &acd2);
            }
        }

        printf("Ingrese las notas del Aprendizaje Practico Experimental (APE) 1 y 2 del alumno %i\n", i);
        scanf("%f, %f", &ape1 ,&ape2);
        while(ape1>10 || ape1<0 ||ape2<0 || ape2>10){
            if(ape1>10 || ape1<0){
                printf(notaInvalida);
                printf(" del APE1\n");
                scanf("%f", &ape1);
            }
            else if(ape2>10 || ape2<0){
                printf(notaInvalida);
                printf(" del APE2\n");
                scanf("%f", &ape2);
            }
        }

        printf("Ingrese las notas del Aprendizaje Autonomo (AA) 1 y 2 del alumno %i\n", i);
        scanf("%f, %f", &aa1, &aa2);
        while (aa1>10 || aa1<0 || aa2>10 || aa2<0){
            if (aa1>10 || aa1<0){
                printf(notaInvalida);
                printf(" del AA1\n");
                scanf("%f", &aa1);
            }
            else if (aa2>10 || aa2<0){
                printf(notaInvalida);
                printf(" del AA2\n");
                scanf("%f", &aa2);
            }
        }

        printf("Ingrese la nota del Portafolio (Evaluacion Sumativa 1) del alumno %i\n", i);
        scanf("%f", &portafolio);
        while(portafolio>10 || portafolio<0){
            printf(notaInvalida);
            printf(" del portafolio\n");
            scanf("%f", &portafolio);
        }

        printf("Ingrese la nota de la Evaluacion Final (Evaluacion Sumativa 2) del alumno %i\n", i);
        scanf("%f", &evaluacion);
        while (evaluacion>10 || evaluacion<0){
            printf(notaInvalida);
            printf(" de la Evaluacion\n");
            scanf("%f", &evaluacion);
        }

```

```c
        //Cálculo de Ponderados
        acdPonderado = ((acd1+acd2)/2)*0.2;
        apePonderado = ((ape1+ape2)/2)*0.25;
        aaPonderado = ((aa1+aa2)/2)*0.2;
        esPonderado = ((evaluacion*0.6) + (portafolio*0.4))*0.35;
        notaFinal = acdPonderado + apePonderado + aaPonderado + esPonderado;

        //Evaluación Cualitativa
        if (notaFinal>=9){
            escala = "Excelente";
        } else if (notaFinal>=7 && notaFinal<9){
            escala = "Bueno";
        } else if (notaFinal>= 5 && notaFinal<7){
            escala = "Regular";
        } else if (notaFinal<5){
            escala = "Deficiente";
        }

        //Datos de Salida Individual
        printf("\nSu nota cuantitativa es:\n");
        printf("La nota ponderada del APE es de %.2f\n", apePonderado);
        printf("La nota ponderada del ACD es de %.2f\n", acdPonderado);
        printf("La nota ponderada del AA es de %.2f\n", aaPonderado);
        printf("La nota ponderada del ES es de %.2f\n", esPonderado);
        printf("La Nota Final de la Unidad 1 del estudiante es de %.2f/10\n\n", notaFinal);
        printf("La nota cualitativa del alumno %i es: %s\n\n",i, escala);

        suma = notaFinal + suma;
    }
    //Promedio General
    if(alumnos>=2){
        promedio = suma/alumnos;
        printf("El promedio fnal entre los %i alumnos es de %.2f", alumnos, promedio);
    }
    return 0;
}
```

## 4. Pruebas:

```
PS C:\Users\ASUS\Downloads\UNL\1erCiclo\Teoria de la Programación\Unidad 2\Ejercicios C> ./U2APE2.exe
De cuantos alumnos deseas calcular el promedio?
3

A continuacion, ingrese las notas (0-10) para los siguientes parametros del alumno 1:
Ingrese las notas del Aprendizaje en Contacto con el Docente (ACD) 1 y 2 del alumno 1
10,80
Nota invalida, vuelva a ingresar la calificacion (0-10) del ACD2
10
Ingrese las notas del Aprendizaje Practico Experimental (APE) 1 y 2 del alumno 1
-8,10
Nota invalida, vuelva a ingresar la calificacion (0-10) del APE1
10
Ingrese las notas del Aprendizaje Autonomo (AA) 1 y 2 del alumno 1
10,10
Ingrese la nota del Portafolio (Evaluacion Sumativa 1) del alumno 1
10
Ingrese la nota de la Evaluacion Final (Evaluacion Sumativa 2) del alumno 1
10

La nota cuantitativa del alumno 1 es:
La nota ponderada del APE es de 2.50
La nota ponderada del ACD es de 2.00
La nota ponderada del AA es de 2.00
La nota ponderada del ES es de 3.50
La Nota Final de la Unidad 1 del estudiante es de 10.00/10

La nota cualitativa del alumno 1 es: Excelente
```

```
A continuacion, ingrese las notas (0-10) para los siguientes parametros del alumno 2:
Ingrese las notas del Aprendizaje en Contacto con el Docente (ACD) 1 y 2 del alumno 2
10,10
Ingrese las notas del Aprendizaje Practico Experimental (APE) 1 y 2 del alumno 2
9.5,9
Ingrese las notas del Aprendizaje Autonomo (AA) 1 y 2 del alumno 2
9.5,9.25
Ingrese la nota del Portafolio (Evaluacion Sumativa 1) del alumno 2
80
Nota invalida, vuelva a ingresar la calificacion (0-10) del portafolio
8
Ingrese la nota de la Evaluacion Final (Evaluacion Sumativa 2) del alumno 2
-5
Nota invalida, vuelva a ingresar la calificacion (0-10) de la Evaluacion
9
```

```
La nota cuantitativa del alumno 2 es:
La nota ponderada del APE es de 2.31
La nota ponderada del ACD es de 2.00
La nota ponderada del AA es de 1.88
La nota ponderada del ES es de 3.01
La Nota Final de la Unidad 1 del estudiante es de 9.20/10

La nota cualitativa del alumno 2 es: Excelente


A continuacion, ingrese las notas (0-10) para los siguientes parametros del alumno 3:
Ingrese las notas del Aprendizaje en Contacto con el Docente (ACD) 1 y 2 del alumno 3
10,10
Ingrese las notas del Aprendizaje Practico Experimental (APE) 1 y 2 del alumno 3
10,10
Ingrese las notas del Aprendizaje Autonomo (AA) 1 y 2 del alumno 3
10,10
Ingrese la nota del Portafolio (Evaluacion Sumativa 1) del alumno 3
9.75
Ingrese la nota de la Evaluacion Final (Evaluacion Sumativa 2) del alumno 3
10

La nota cuantitativa del alumno 3 es:
La nota ponderada del APE es de 2.50
La nota ponderada del ACD es de 2.00
La nota ponderada del AA es de 2.00
La nota ponderada del ES es de 3.46
La Nota Final de la Unidad 1 del estudiante es de 9.97/10

La nota cualitativa del alumno 3 es: Excelente

El promedio fnal entre los 3 alumnos es de 9.72
```

## 6. Preguntas de Control

- **¿En qué se diferencia una estructura repetitiva de una condicional?**

  La diferencia entre una ***estructura condicional*** y una ***estructura repetitiva*** radica en la forma en que utilizan la condición y el propósito de su ejecución. Una estructura condicional evalúa una condición para decidir si un bloque de instrucciones se ejecuta una sola vez o no; es decir, sirve para tomar decisiones en el flujo del programa.

  En cambio, una estructura repetitiva también depende de una condición, pero su objetivo es repetir un bloque de instrucciones varias veces, ya sea un número específico de iteraciones o mientras la condición se mantenga verdadera. En resumen, la condicional controla la ejecución única de un bloque según una condición, mientras que la repetitiva controla la ejecución continua o múltiple de un bloque hasta que la condición deje de cumplirse.

- **¿Qué diferencia existe entre las estructuras for, while y do...while en cuanto a su funcionamiento y uso?**

  La estructura ***while*** (mientras) comprueba la condición al inicio del ciclo, por lo que si esta es verdadera se ejecutan las instrucciones y si es falsa no se entra al bucle; se utiliza cuando no se sabe cuántas veces se repetirá la acción, ya que depende de los datos o instrucciones.

  En cambio, la estructura ***do...while*** (repetir) evalúa la condición al final, lo que garantiza que el bloque de instrucciones se ejecute al menos una vez, incluso si la condición resulta falsa en la primera comprobación; se emplea cuando se necesita asegurar una ejecución inicial.

  Finalmente, la estructura ***for*** (para) está diseñada para repetir un conjunto de instrucciones un número específico de veces, controlando automáticamente las iteraciones mediante un contador que requiere un valor inicial, una condición de fin y un incremento o decremento; se usa cuando se conoce de antemano la cantidad exacta de repeticiones necesarias.

- **¿Por qué es importante incluir validaciones dentro de un programa cuando se solicitan datos al usuario?**

  Es importante incluir validaciones dentro de un programa cuando se solicitan datos al usuario porque estas permiten verificar que la información ingresada cumpla con los datos esperados. Sin validaciones, el usuario muy comúnmente introduce datos incorrectos, incompletos o en un formato no válido, lo que provoca errores en la ejecución y fallos en el funcionamiento del programa.

## 7. Conclusión

Se concluye, con el presente trabajo se logró comprender y aplicar de manera efectiva las estructuras repetitivas en la resolución del problema planteado, lo que permitió diseñar y codificar un algoritmo capaz de utilizar bucles para abordar situaciones de tipo iterativo. Asimismo, se validó el correcto funcionamiento del programa mediante su ejecución práctica, garantizando que las instrucciones implementadas cumplan con los objetivos propuestos y que la solución desarrollada sea confiable y eficiente.

## 8. Recomendación

Se recomienda que en las pruebas de escritorio se incluya una validación específica para los casos en que el usuario ingrese caracteres no permitidos, como letras o signos especiales. Se confirmó esta necesidad al realizar pruebas con este tipo de entradas y se evidenció que el programa continuaba su ejecución sin control y no funcionaba correctamente. Implementar dichas validaciones permitirá garantizar un manejo adecuado de los datos.