Proyecto: Maratón de Algoritmos C++

29/09/2025 GRUPO: DAVID MENDOZA, DEREK MERA Y MICHAEL TACO

INFORMÁTICA

Bloque 1: Fundamentos (400 ejercicios)

Análisis del problema

Descripción del problema: Elaborar un algoritmo que me permita ingresar la calificación anual de un estudiante y verificar si el estudiante aprobó reprobó o se encuentra supletorio, recordar que para aprobar el año lectivo el estudiante tiene que tener un promedio igual o mayor a 7, si el estudiante se encuentra en un rango de calificación de 6.99 a 4, y si el promedio es inferior a 4 el estudiante será reprobado.

Entradas y Salidas:

- o Entradas: Un número flotante n (con rango de 0 a 10).
- o Salidas: Un número flotante que representa el promedio anual del estudiante.

Diseño de la Solución

- Algoritmo Propuesto:
- 1. Pedir al usuario que ingrese un número flotante n.
- 2. Ingresar un número
- 3. Comprobar que sea un número dentro del rango de 0 a 10
- 4. Verificar si aprobó, se quedó supletorio o reprobó
- Estructuras de Datos: No se usarán estructuras de datos complejas, solo una variable entera para

la suma.

• Funciones Principales: Se usará solo la función main.

Código Fuente

'calificacion'.

1:

#include <iostream> // Incluye la biblioteca estandar de entrada y salida para usar std::cout y std::cin. int main() { // La función principal donde comienza la ejecución del programa. double calificacion; // Declara una variable de tipo double para almacenar la calificacion, permitiendo decimales.

```
// Inicia un bucle do-while para solicitar la calificación y validarla.
// El bucle se ejecutará al menos una vez y se repetirá si la calificación no es válida.
do {
    // Muestra un mensaje en la consola pidiendo al usuario que ingrese la calificación.
    std::cout << "Ingrese la calificacion del estudiante (de 0 a 10): ";
    std::cin >> calificacion; // Lee el valor ingresado por el usuario y lo guarda en la variable
```

```
// Verifica si la calificación ingresada está fuera del rango [0, 10]. if (calificacion < 0 || calificacion > 10) {
```

```
// Si la calificación es inválida, muestra un mensaje de error.
       std::cout << "Calificacion no valida. Por favor, ingrese un valor entre 0 y 10." << std::endl;
    }
  } while (calificacion < 0 || calificacion > 10); // La condicion del bucle: si la calificacion es menor que
0 o mayor que 10, el bucle se repite.
  // A partir de aqui, la calificación es válida y se procede a su evaluación.
  // Comienza la estructura de decision if-else if-else para determinar el estado del estudiante.
  if (calificacion >= 7.0 && calificacion <= 10.0) {
    // Si la calificación es 7 o más, y hasta 10, el estudiante aprueba.
    std::cout << "El estudiante APROBO la asignatura. ¡Felicidades! 🎉" << std::endl;
  } else if (calificacion >= 4.0 && calificacion < 7.0) {
    // Si la calificación es 4 o más, pero menor que 7, el estudiante va a supletorio.
    std::cout << "El estudiante se encuentra en SUPLETORIO. Esfuérzate por el siguiente examen. 📚"
<< std::endl;
  } else { // Esta es la condicion por defecto si las anteriores no se cumplen.
    // Si la calificación es menor que 4, el estudiante reprueba.
    std::cout << "El estudiante REPROBO la asignatura. A seguir intentando. 😔 " << std::endl;
 }
  return 0; // Indica que el programa finalizó con exito.
}
```

PRUEBA DE ESCRITORIO

Casos de prueba

STOIN

1

Output:

```
Ingrese la calificacion del estudiante (de 0 a 10): El estudiante REPROBO la asignatura. A seguir intentando. 😓
```



Resultados de la prueba:

El estudiante sacó una calificación de 1 por lo tanto reprobó.

Título: Determinar si un número es par o impar (valor fijo)

Análisis del Problema:

Queremos determinar si un número entero (ya definido dentro del código) es par o impar.

Para ello, se utiliza el operador módulo % para verificar el residuo de la división entre 2.

Entradas y Salidas:

Entrada: número (fijo, valor 190).

• Salida: texto que indica si el número es par o impar.

Diseño de la Solución:

- 1. Asignar un número fijo.
- 2. Verificar si es divisible por 2.
- 3. Mostrar el resultado correspondiente.

Código en C++:

```
#include <iostream>
using namespace std;

int main() {
   int numero = 190; // valor fijo para prueba
   cout << "Numero es: " << numero << "\n";
   if (numero % 2 == 0)
      cout << "El numero es par.";
   else
      cout << "El numero es impar.";
   return 0;
}</pre>
```

Pruebas:

- Entrada: 190 → Salida: "El número es par."
- Entrada: 133 → Salida: "El número es impar."

Contribución del equipo:

Ejercicio elaborado por [nombre del integrante]. Se probó con valores fijos para simplificar la comprobación.

Ejercicio 3

Título: Número aleatorio par o impar

Análisis del Problema:

El programa genera un número aleatorio entre 1 y 100, y determina si es par o impar.

Entradas y Salidas:

- Entrada: ninguna (el número se genera automáticamente).
- Salida: número aleatorio y si es par o impar.

Diseño de la Solución:

- 1. Generar número aleatorio con rand().
- 2. Evaluar si el número es divisible entre 2.
- 3. Mostrar el número y el resultado.

Código en C++:

```
#include <iostream>
#include <cstdlib>
#include <ctime>
using namespace std;

int main() {
    int numero;
    srand(time(0)); // semilla aleatoria

    numero = rand() % 100 + 1; // número entre 1 y 100
    cout << "El numero generado es: " << numero << "\n";

    if (numero % 2 == 0)
        cout << "El numero es par.\n";
    else
        cout << "El numero es impar.\n";
    return 0;
}</pre>
```

Pruebas:

- Ejemplo de salida: "El número generado es: 57. El número es impar."
- Otro ejemplo: "El número generado es: 80. El número es par."

Contribución del equipo:

Ejercicio programado por [nombre del integrante]. Se validó con varios resultados aleatorios.

Ejercicio 4

Título: Cálculo del área de un círculo

Análisis del Problema:

Dado el radio de un círculo, se calcula su área aplicando la fórmula:

 $A=\pi \times radio2A = \pi \times radio^2A = \pi \times radio^2A$

Entradas y Salidas:

- Entrada: radio (número real positivo).
- Salida: área del círculo.

Diseño de la Solución:

- 1. Solicitar el radio.
- 2. Verificar que sea positivo.
- 3. Calcular área usando M_PI y pow().
- 4. Mostrar resultado.

```
#include <iostream>
#include <cmath>
using namespace std;
int main() {
```

```
double radio;
cout << "Ingrese el radio del circulo: ";
cin >> radio;

if (radio <= 0) {
    cout << "Error: El radio debe ser positivo." << endl;
    return 1;
}

double area = M_PI * pow(radio, 2);
cout << "El area del circulo es: " << area << endl;
return 0;
}</pre>
```

- Radio = $5 \rightarrow \text{Área} = 78.54$
- Radio = 2 → Área = 12.57

Contribución del equipo:

Desarrollado por [nombre del integrante]. Se usó la librería <cmath> para el cálculo matemático.

Ejercicio 5

Título: Conversor de temperatura

Análisis del Problema:

Convierte una temperatura de Celsius a Fahrenheit o viceversa, según la opción seleccionada.

Entradas y Salidas:

- Entrada: opción (1 o 2) y valor de temperatura.
- Salida: temperatura convertida.

Diseño de la Solución:

- 1. Mostrar menú con opciones.
- 2. Leer opción y temperatura.
- 3. Aplicar fórmula de conversión.
- 4. Mostrar resultado o advertencia si la opción es inválida.

```
#include <iostream>
using namespace std;
int main() {
    int opcion;
    double temp, resultado;
    cout << "Conversor de Temperatura" << endl;</pre>
    cout << "1. Celsius a Fahrenheit" << endl;</pre>
    cout << "2. Fahrenheit a Celsius" << endl;</pre>
    cout << "Seleccione una opcion: ";</pre>
    cin >> opcion;
    cout << "Ingrese la temperatura: ";</pre>
    cin >> temp;
    if (opcion == 1) {
        resultado = (temp * 9.0/5.0) + 32;
        cout << temp << " °C = " << resultado << " °F" << endl;</pre>
    } else if (opcion == 2) {
        resultado = (temp - 32) * 5.0/9.0;
        cout << temp << " °F = " << resultado << " °C" << endl;</pre>
    } else {
        cout << "Opcion invalida" << endl;</pre>
    }
    return 0;
}
```

- Opción 1, 0°C → 32°F
- Opción 2, 212°F → 100°C
- Opción 3 → "Opción inválida"

Contribución del equipo:

Realizado por [nombre del integrante]. Se probó en ambos modos de conversión.

Ejercicio 6

Título: Tabla de multiplicar

Análisis del Problema:

Se ingresa un número entero y se muestra su tabla de multiplicar del 1 al 10.

Entradas y Salidas:

- Entrada: número entero.
- Salida: tabla de multiplicar correspondiente.

Diseño de la Solución:

- 1. Solicitar número.
- 2. Usar un ciclo for de 1 a 10.
- 3. Calcular y mostrar cada producto.

```
#include <iostream>
using namespace std;
int main() {
   int numero;
```

```
cout << "Introduce un número para generar su tabla de
multiplicar: ";
    cin >> numero;

    for (int i = 1; i <= 10; ++i) {
        cout << numero << " x " << i << " = " << numero * i << endl;
    }

    return 0;
}</pre>
```

- Entrada: 5 → muestra 5×1 hasta 5×10.
- Entrada: 9 → muestra 9×1 hasta 9×10.

Contribución del equipo:

Ejercicio dirigido por [nombre del integrante]. Se validó el uso correcto del bucle for.

Ejercicio 7

(Idéntico al anterior, para prueba duplicada o validación)

Análisis del Problema:

Igual que el Ejercicio 6, genera una tabla de multiplicar usando un ciclo for.

Código en C++:

(idéntico al Ejercicio 6)

Contribución del equipo:

Ejercicio duplicado para reforzar la práctica y validar resultados consistentes.

Ejercicio 8:

Título: Factorial de un número

Análisis del Problema:

El programa calcula el factorial de un número ingresado por el usuario.

Entradas y Salidas:

- Entrada: número entero.
- Salida: factorial del número.

Diseño de la Solución:

- 1. Leer el número.
- 2. Verificar si es negativo.
- 3. Si es positivo, multiplicar desde 1 hasta el número.
- 4. Mostrar el resultado.

```
#include <iostream>
using namespace std;
int main() {
    int numero;
    long long factorial = 1;
    cout << "Introduce un número para calcular su factorial: ";</pre>
    cin >> numero;
    if (numero < 0) {</pre>
        cout << "El factorial no existe para números negativos."</pre>
<< endl:
    } else {
        for (int i = 1; i <= numero; ++i) {
             factorial *= i;
        cout << "El factorial de " << numero << " es: " <<</pre>
factorial << endl;</pre>
    }
```

```
return 0;
}
```

• Entrada: 5 → 120

• Entrada: $0 \rightarrow 1$

 $\bullet \quad \text{Entrada: -3} \rightarrow \text{"El factorial no existe para números negativos."}$