



Universidad San Ignacio de Loyola

Introducción a las Ciencias de la Computación

PROYECTO FINAL

Número de grupo: 10

Apellidos y nombres de los alumnos:

AVENDAÑO HUAMAN ESTEBAN JOEL

CAHUANA ESTRADA PIERO DIEGO

CHUQUIMANGO OTAZU CRISTIAN ANDRE

PAZ BRUFFAU FABRICIO JESUS

RIOS VITO DANIELA PAOLA

VILLAR GONIA FABIANA JAZMIN

Lima –Perú

2022

OBJETIVOS GENERALES:

Nos dispusimos a poner en práctica el uso de estructuras repetitivas, así como funciones para ejecutar acciones y retornar valores con o sin parámetros.

OBJETIVOS ESPECÍFICOS:

Para el caso 1 del proyecto, elaboramos un juego de azar con las indicaciones que nos fueron provistas. Para el caso 2 del proyecto, desarrollamos una calculadora científica que proporciona distintas operaciones al usuario. Para el caso 3 del proyecto, determinamos un producto de matrices cuadradas ($n \times n$).

CASO 1: Juego de adivinanzas

Para esta práctica de laboratorio, necesitaré usar bucles do-while y declaraciones if para construir un juego de adivinanzas. La computadora elegirá un número aleatorio entre 1 y 100 y el usuario deberá adivinar cuál es el número. Si el usuario adivina incorrectamente, la computadora indicará si la suposición del usuario fue demasiado alta o demasiado baja. Si el usuario adivina correctamente, la computadora informa cuántos intentos se necesitaron para obtener la respuesta correcta y luego pregunta al usuario si desea volver a jugar. Este programa requerirá cuatro bucles do-while:

El ciclo do-while principal (el más externo) contendrá casi todo el código del programa y seguirá ejecutándose una y otra vez hasta que el usuario indique que ya no desea jugar.

El bucle do-while del juego está anidado dentro del bucle do-while principal y seguirá ejecutándose una y otra vez hasta que el usuario adivine la respuesta correcta y gane el juego. Tan pronto como el usuario gane un juego, el programa saldrá de este bucle do-while.

El ciclo do-while de validación de entrada numérica está anidado dentro del ciclo do-while del juego y garantizará que el usuario ingrese un número entero válido para cada suposición.

El bucle do-while de validación de entrada no numérica se encuentra DESPUÉS y FUERA del bucle do-while del juego. Solo se ejecuta después de que se completa un juego y le pregunta al usuario si desea volver a jugar. Está anidado dentro del ciclo do-while primario y garantizará que el usuario ingrese un 'Y' / 'y' / 'N' / 'n' válido en respuesta a la pregunta "¿Le gustaría volver a jugar (Y/N)?" pregunta.

```
#include <iostream>

using namespace std;

int main()
{
    float i = 0, x = 0; // i = maquina, x = usuario

    int intentos = 0;

    string flag;

    bool valido;

    srand(time(NULL));

    do
    {
        //REINICIANDO...

        intentos = 0;

        valido = false;

        //GENERANDO EL NÚMERO ALEATORIO...

        i = 1 + rand() % (99);

        cout << "Hemos generado un numero..." << endl;
```

```
//PIDIENDO AL USUARIO QUE ADIVINE...
```

```
do
```

```
{
```

```
    cout << "Adivine el numero del 1 al 100" << endl;
```

```
do
```

```
{
```

```
    cin >> x;
```

```
    if (x > 100 || x < 1)
```

```
    {
```

```
        valido = false;
```

```
        cout << "Numero invalido, ingrese un numero del 1 al 100: " << endl;
```

```
    }
```

```
    else
```

```
    {
```

```
        valido = true;
```

```
    }
```

```
} while (valido == false);
```

```
intentos++; //Cuenta cada intento
```

```
if (x > i) //Si el usuario fue mayor
```

```
{
```

```

        cout << "Su suposicion fue muy alta" << endl;
    }
    else if (x < i) // Si el usuario fue menor
    {
        cout << "Su suposicion fue muy baja" << endl;
    }
} while (x != i);

//GANASTE...
cout << "*****" << endl;
cout << "Felicidades, acerto la adivinanza!" << endl;
cout << "Numero de intentos: " << intentos << endl;

valido = false;
do
{
    cout << "Le gustaria volver a jugar? (Y/N): ";
    cin >> flag;

    if (flag == "y" || flag == "Y" || flag == "n" || flag == "N")
    {
        valido = true;
    }
    else
    {

```

```
        valido = false;

        cout << "Solo se pueden colocar los caracteres (Y/N)" << endl;

    }

} while (valido == false);

} while (flag == "y" || flag == "Y");

cout << "*****" << endl;

cout << "Gracias por jugar";

}
```

CASO 2: CALCULADORA CIENTÍFICA

- **Acceso rápido a los números pi y e.**
- **Funciones de trigonometría.**
- **Funciones logarítmicas en base e y en base 10.**
- **Funciones exponenciales.**
- **Raíces cuadradas y superiores.**
- **Notaciones científicas.**
- **Aritmética de punto flotante.**
- **Cálculo binario, hexadecimal y octal.**

```
// Librerias
```

```
#include <iomanip>
```

```
#include <iostream>
```

```
#include <math.h>
```

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

```
#include <stdlib.h>
```

```
#include <string>
```

```
using namespace std;
```

```
// Variables Globales
```

```
int getSeleccion();
```

```
void showOpciones();
```

```
void getOperacion();
```

```
float getValor(string texto);
```

```
// Prototipos de funcion
```

```
float SumayResta();
```

```
float Multiplicacion();
```

```
float Division();  
float Potencia();  
float Logaritmo();  
void NotacionCientifica();  
void AritmeticaPuntoFlotante();  
float SistemaNumerico();  
float valorAbsoluto();  
float RaizCuadrada();
```

```
void showOptsTri();  
float FuncionesTri();  
int opcionTri();  
float resultadoTri(float num);
```

```
// CONVERSIONES
```

```
float Dec_Binario(int num) {  
    int dividendo = 0, cociente = 0;  
    float residuo = 0, k = 1, octal = 0;  
  
    cociente = num;  
  
    while (cociente >= 2) {  
        dividendo = cociente;  
        cociente = dividendo / 2;  
        residuo = (float)(dividendo % 2);  
        octal += residuo * k;  
        k = k * 10;  
    }  
    octal += cociente * k;
```



```

    return octal;
}

float Dec_Octal(int num) {
    int dividendo = 0, cociente = 0;
    float residuo = 0, k = 1, octal = 0;

    cociente = num;

    while (cociente >= 8) {
        dividendo = cociente;
        cociente = dividendo / 8;
        residuo = (float)(dividendo % 8);
        octal += residuo * k;
        k = k * 10;
    }
    octal += cociente * k;

    return octal;
}

float Dec_Hexa(int num) {
    int dividendo = 0, cociente = 0;
    float residuo = 0, k = 1, octal = 0;

    cociente = num;

    while (cociente >= 16) {
        dividendo = cociente;
        cociente = dividendo / 16;
    }

```

```

        residuo = (float)(dividendo % 16);
        octal += residuo * k;
        k = k * 10;
    }
    octal += cociente * k;

    return octal;
}

// Variables
int slt; // Seleccion del usuario
float resultado;

// Funcion principal
int main() { getOperacion(); }

// Validar que el usuario ingrese valores entre 1 y 13
int getSeleccion() {
    int valor = 0;
    bool valido = false;

    while (valido == false) {
        showOpciones();
        cin >> valor;

        if (valor > 13 || valor < 1 || typeid(valor) == typeid(string)) {
            valido = false;
            system("cls");
            cout << "VALOR INVALIDO (elija las opciones disponibles [1-13])" << endl;

```

```

    }

    else {

        valido = true;

    }

}

return valor;

}

// Mostrar menu de opciones

void showOpciones() {

    cout << "Seleccione la operacion que desee realizar: " << endl << endl;
    cout << "Suma y Resta.....[1]" << endl;
    cout << "Multiplicacion.....[2]" << endl;
    cout << "Division.....[3]" << endl;
    cout << "Potencia.....[4]" << endl;
    cout << "Raices cuadradas y superiores.....[5]" << endl;
    cout << "Logaritmo.....[6]" << endl;
    cout << "Notacion Cientifica.....[7]" << endl;
    cout << "Aritmetica de punto flotante.....[8]" << endl;
    cout << "Sistemas de numeracion.....[9]" << endl;
    cout << "Funciones de trigonometria.....[10]" << endl;
    cout << "Valor absoluto.....[11]" << endl;
    cout << "Instrucciones.....[12]" << endl;

}

// Casos

void getOperacion() {

    switch (getSeleccion()) {

```

case 1:

```
resultado = SumayResta();  
cout << "El resultado es: " << resultado << endl;  
break;
```

case 2:

```
resultado = Multiplicacion();  
cout << "El resultado es: " << resultado << endl;  
break;
```

case 3:

```
resultado = Division();  
cout << "El resultado es: " << resultado << endl;  
break;
```

case 4:

```
resultado = Potencia();  
cout << "El resultado es: " << resultado << endl;  
break;
```

case 5:

```
resultado = RaizCuadrada();  
cout << "El resultado es: " << resultado << endl;  
break;
```

case 6:

```
resultado = Logaritmo();  
cout << "El resultado es: " << resultado << endl;  
break;
```

case 7:

```
NotacionCientifica();
```

```
break;
```

case 8:

```
AritmeticaPuntoFlotante();
```

```
cout << "El resultado es: " << resultado << endl;
```

```
break;
```

case 9:

```
resultado = SistemaNumerico();
```

```
cout << "El resultado es: " << resultado << endl;
```

```
break;
```

case 10:

```
resultado = FuncionesTri();
```

```
cout << "El resultado es: " << resultado << endl;
```

```
break;
```

case 11:

```
resultado = valorAbsoluto();
```

```
cout << "El resultado es: " << resultado << endl;
```

```
break;
```

case 12:

```
cout << "INDICACIONES IMPORTANTES" << endl;
```

```
cout << "- escriba "
```

```
"pi"
```

```

    " o "

    "e"

    " para usar el numero pi y el numero neperiano respectivamente"

    << endl;

    cout << "- las funciones trigonometricas se calculan en radianes" << endl;

    break;
}

bool valido = false;
string flag;
do {
    cout << "Desea volver a hacer una operacion (S/N)?" << endl;
    cin >> flag;
    if (flag == "s" || flag == "S") {
        valido = true;
        getOperacion();
    }
    else if (flag == "n" || flag == "N") {
        valido = false;
        cout << "*****: " << endl;
    }

} while (valido);
}

// Valores pi y e
float getValor(string texto) {

```

```
float valor = 0;
if (texto == "pi") {
    valor = 3.1416f;
}
else if (texto == "e") {
    valor = 2.718f;
}
else {
    valor = stof(texto);
}
return valor;
}
```

```
float SumayResta() {
    int cantidad = 0;
    float suma = 0;
    string sumando = "0", signo = "+";
    bool valido = false;

    cout << "Ingrese cantidad de numeros a operar: ";
    do {
        cin >> cantidad;
        if (cantidad > 1) {
            valido = true;
        }
        else {
            valido = false;
            cout << "Porfavor ingrese una cantidad mayor a 1:" << endl;
        }
    }
```

```
} while (!valido);
```

```
valido = false; // resetando...
```

```
for (int i = 0; i < cantidad; i++) {
```

```
    valido = false; // reseteando el valor...
```

```
    while (valido == false) {
```

```
        cout << "Ingrese el numero " << i + 1 << " : ";
```

```
        cin >> sumando;
```

```
        if (getValor(sumando) >= 0 || getValor(sumando) <= 0) {
```

```
            valido = true;
```

```
        }
```

```
        else {
```

```
            valido = false;
```

```
            cout << "Dato invalido, ingrese un numero" << endl;
```

```
        }
```

```
    }
```

```
    if (signo == "+" || i == 0) {
```

```
        suma += getValor(sumando);
```

```
    }
```

```
    else if (signo == "-") {
```

```
        suma -= getValor(sumando);
```

```
    }
```

```
    if (i != cantidad - 1)
```

```
        valido = false; // resetemos este valor para la siguiente validación...
```



```

while (valido == false) {
    cout << "Desea sumar o restar?, escriba (+/-): ";
    cin >> signo;

    if (signo == "+" || signo == "-") {
        valido = true;
    }
    else {
        valido = false;
        cout << "Dato invalido, ingrese signos (+/-)" << endl;
    }
}

return suma;
}

```

```

float Multiplicacion() {
    int cantidad = 0;
    float producto = 1;
    string multiplo = "0";
    bool valido = false;

    cout << "Ingrese cantidad de numeros a multiplicar: ";
    do {
        cin >> cantidad;
        if (cantidad > 1) {
            valido = true;

```

```

    }

    else {

        valido = false;

        cout << "Porfavor ingrese una cantidad mayor a 1:" << endl;

    }

} while (!valido);


for (int i = 0; i < cantidad; i++) {

    cout << "Ingrese multiplo " << i + 1 << " : ";

    cin >> multiplo;


    producto = producto * getValor(multiplo);

}


return producto;

}


float Division() {

    int cantidad = 0;

    float division = 1;

    string divisor = "0";

    bool valido = false;


    cout << "Ingrese cantidad de numeros a dividir: ";

    do {

        cin >> cantidad;

        if (cantidad > 1) {

            valido = true;

        }

    }

```

```

    else {
        valido = false;
        cout << "Porfavor ingrese una cantidad mayor a 1:" << endl;
    }
} while (!valido);

for (int i = 0; i < cantidad; i++) {
    cout << "Ingrese divisor " << i + 1 << " : ";
    cin >> divisor;

    if (i == 0)
        division = getValor(divisor);
    else
        division = division / getValor(divisor);
}

return division;
}

float Potencia() {
    float base, exponente, potencia = 1;

    cout << "Ingrese el numero base: ";
    cin >> base;
    cout << "Ingrese el numero exponente: ";
    cin >> exponente;

    potencia = pow(base, exponente);
    return potencia;
}

```

```
}
```

```
float RaizCuadrada() {  
    float n = 0, m = 0, raiz = 0;  
    cout << "Ingresar radicando:";  
    cin >> n;  
    cout << "Ingresar el numero indice: ";  
    cin >> m;  
  
    raiz = pow(n, (1 / m));  
    return raiz;  
}
```

```
float Logaritmo() {  
    float base = 0, argumento = 0;  
    float logaritmo = 0;  
    bool valido = false;  
  
    do {  
        cout << "Ingresar la base de logaritmo diferente a 0:";  
        cin >> base;  
        if (base == 0)  
            valido = false;  
        else  
            valido = true;  
    } while (!valido);  
  
    cout << "Ingresar argumento:";  
    cin >> argumento;
```

```
logaritmo = (log(argumento) / log(base));
```

```
cout << "log" << base << "(" << argumento << ")"  
    << "=" << logaritmo << endl;
```

```
return logaritmo;
```

```
}
```

```
void NotacionCientifica() {
```

```
    float numero;
```

```
    cout << "Ingrese un numero grande para calcular su notacion cientifica: ";
```

```
    cin >> numero;
```

```
    cout << "Su notacion cientifica es: ";
```

```
    cout << numero;
```

```
}
```

```
void AritmeticaPuntoFlotante() {
```

```
    double x;
```

```
    cout << "Ingresar numero:" << endl;
```

```
    cin >> x;
```

```
    cout << setprecision(4);
```

```
    cout << "float:";
```

```
    cout << x << endl;
```

```
    cout << setprecision(8);
```

```
    cout << "double:";
```

```
    cout << x << endl;
```

```
    cout << setprecision(16);  
    cout << "long double:";  
    cout << x << endl;  
}
```

```
float SistemaNumerico() {  
    int opcion1 = 1;  
    float old_num = 0,  
        new_num = 0; // old es el ingresado y new es la transformacion  
    bool valido = false;  
  
    // CUAL ES ESE NUMERO...?  
  
    cout << "Que numero desea transformar?: ";  
    cin >> old_num;  
  
    cout << "Binario.....[1] " << endl;  
    cout << "Octal.....[2] " << endl;  
    cout << "Hexadecimal..[3]" << endl;  
    do {  
        cout << "A que sistema numerico lo quiere pasar?: " << endl;  
        cin >> opcion1;  
  
        if (opcion1 > 3 || opcion1 < 1) {  
            valido = false;  
            cout << "Dato nvalido, por favor seleccion una de las opciones "  
                "disponibles";  
        }  
    }  
    else
```

```

        valido = true;

    } while (!valido);

    switch (opcion1) {
    case 1: // A binario...
        new_num = Dec_Binario(old_num);
        break;

    case 2: // A octal...
        new_num = Dec_Octal(old_num);
        break;

    case 3: // A hexadecimal...
        new_num = Dec_Hexa(old_num);
        break;
    }
    return new_num;
}

```

```

float FuncionesTri() {
    float num, result;
    cout << "Ingresar el numero:";
    cin >> num;
    cout << endl;
    result = resultadoTri(num);
    return result;
}

int opcionTri() {

```

```

int valor = 0;

bool valido = false;

while (valido == false) {

    showOptsTri();

    cin >> valor;

    if (valor > 3 || valor < 1 || typeid(valor) == typeid(string)) {

        valido = false;

        system("cls");

        cout << "VALOR INVALIDO (elija las opciones disponibles)" << endl;

    }

    else {

        valido = true;

    }

}

return valor;
}

void showOptsTri() {

    cout << "MENU DE OPCIONES:" << endl;

    cout << "Convertir a seno:[1]" << endl;

    cout << "Convertir a coseno:[2]" << endl;

    cout << "Convertir a tangente:[3]" << endl;

    cout << "Elige una opcion:" << endl;

}

float resultadoTri(float num) {

    float result{};

    switch (opcionTri()) {

    case 1:

```



```

        result = sin(num * 3.14159 / 180);
        break;
case 2:
        result = cos(num * 3.14159 / 180);
        break;
case 3:
        result = tan(num * 3.14159 / 180);
        break;
    }
    return result;
}

```

```

float valorAbsoluto() {
    float n, m;
    cout << "Ingrese el numero a hallar su V.A: ";
    cin >> n;
    if (n < 0) {
        m = n * -1;
    }
    else if (n > 0) {
        m = n * 1;
    }
    return m;
}

```

CASO 03: Construya la multiplicación de matrices para el caso $n \times n$

```
#include<iostream>

using namespace std;

const int FIL=100; //Establecer un límite en las filas
const int COL=100; //Establecer un límite en las columnas


//Prototipos

void entradaDatos(float MATRIZ[FIL][COL],int fil,int col);

void multiplicacionMatrices(float A[FIL][COL],int FILa,int COLa,float B[FIL][COL],int FILb,int COLb,float P[FIL][COL] );

void imprimirMatriz(float MATRIZ[FIL][COL],int fil,int col);


int main ()
{
    float A[FIL][COL];
    float B[FIL][COL];
    float P[FIL][COL];
    int n,filA,colA,filB,colB;

    cout<<"Ingresar el orden de la matriz:";
    cin>>n;
    cout<<"\n";
    cout<<"MATRIZ MULTIPLICANDO:"<<"\n";
    cout<<"Como la matriz es cuadrada, su orden es "<<n<<" x "<<n<<"\n";
    filA=n;
    colA=n;
    entradaDatos(A,filA,colA);
    cout<<"\n";
    cout<<"MATRIZ MULTIPLICADOR:"<<"\n";
    cout<<"Para que se puedan multiplicar, la segunda matriz debe tener el mismo orden: "<<n<<" x "<<n<<"\n";
```

```

    filB=n;
    colB=n;
    entradaDatos(B,filB,colB);
    multiplicacionMatrices(A,filA,colA,B,filB,colB,P);
    cout<<"\n";
    cout<<"MATRIZ MULTIPLICANDO:"<<"\n";
    imprimirMatriz(A,filA,colA);
    cout<<"\n";
    cout<<"MATRIZ MULTIPLICADOR:"<<"\n";
    imprimirMatriz(B,filB,colB);
    cout<<"\n";
    cout<<"PRODUCTO DE MATRICES:"<<"\n";
    imprimirMatriz(P,filA,colB);

}

void entradaDatos(float MATRIZ[FIL][COL],int fil,int col)
{
    for(int i=0;i<fil;i++)
    {
        for(int j=0;j<col;j++)
        {
            cout<<"Ingresar el dato("<<i<<")("<<j<<":";
            cin>>MATRIZ[i][j];
        }
    }
}

```

```
void multiplicacionMatrices(float A[FIL][COL],int FILa,int COLa,float B[FIL][COL],int FILb,int COLb,float P[FIL][COL] )
```

```
{
    for(int i=0;i<FILa;i++)
    {
        for(int j=0;j<COLb;j++)
        {
            P[i][j]=0;
            for(int k=0;k<COLa;k++)
            {
                P[i][j]+= A[i][k] * B[k][j]; //P[i][j]=P[i][j]+A[i][k]*B[k][j]
            }
        }
    }
}
```

```
void imprimirMatriz(float MATRIZ[FIL][COL],int fil,int col)
```

```
{
    for(int i=0;i<fil;i++)
    {
        for(int j=0;j<col;j++)
        {
            cout<<MATRIZ[i][j]<<"\t";
        }
        cout<<"\n";
    }
}
```

CONCLUSIONES: En conclusión, por medio de estructuras condicionales, repetitivas y funciones en conjunto con distintos tipos y usos de variables, somos capaces de determinar un algoritmo de comportamiento complejo. Este trabajo en concreto, es de mucha ayuda pues nos permite reforzar y aplicar nuestros conocimientos en C++ para la solución de estos casos.

GLOSARIO DE TÉRMINOS:

- **#include<stdlib.h>**

Archivo de cabecera estándar que tiene las funciones útiles para hacer búsqueda y ordenamiento de datos de cualquier tipo.

- **#include<time.h>**

Formatea y manipula la hora y fecha del sistema.

- **#include <iomanip>**

Por medio de la función setprecision() podemos establecer la precisión de puntos flotantes.

- **#include <iostream>**

Sirve para poder tener acceso a los dispositivos estándar de entrada y/o salida.

- **#include <math.h>**

Está diseñado para realizar operaciones matemáticas básicas. Nos permite utilizar funciones de potenciación (pow()) y funciones trigonométricas (sen(), cos(), tan()).

- **#include <stdio.h>**

Se utiliza para definir y modificar ficheros.

- **#include <string>**

Nos permite utilizar las variables de tipo “string” y todas sus cualidades.

- **Parámetros**

Son valores que solicita una función cada vez que es invocada, por lo siempre habremos de transmitir valores.

- **Variable global**

Se define fuera de cualquier función, normalmente al principio del programa y cualquier función puede acceder a dichas variables para poder leer y escribir en ellas.

● **Variable local**

Su ámbito se restringe a la función que la ha declarado y solo se podrá manipular en dicha sección.

● **Bibliotecas de c++**

Colección de clases y funciones escritas en el núcleo del lenguaje.

● **Función**

Bloque de código encargado de realizar una determinada operación. Puede definir parámetros de entrada opcionalmente que permiten a los llamadores pasar argumentos por la función, asimismo también puede devolver un valor como salida

● **Prototipos de función**

Permiten declarar una función al inicio del programa (valor a retornar, nombre de la función y lista de argumentos) seguida de punto y coma.

● **Entrada de datos**

Recoge datos externos para incorporarlos en el programa.

● **Proceso**

Es la parte del algoritmo en la que se llevan a cabo las acciones del mismo. Ya sean cálculos, invocación de funciones, cambios de valores, comportamiento a partir de estructuras condicionales, etc.

● **Salida de datos**

Es la parte del algoritmo en la cual se imprimen (muestran) los datos determinados previamente.

LINK DEL VIDEO:

<https://youtu.be/6bozmz1o61Y>