



Centro Universitario de Ciencias Exactas e Ingenierías (CUCEI)

Actividad de aprendizaje 1:
Tipos de Datos Primitivos y Tipos de Datos Estructurados

Víctor Agustín Díaz Méndez
Ingeniería en Informática

Estructura de Datos I (Sección D12)
Profesor: Dr. Gutierrez Hernandez Alfredo



Para el proyecto decidí crear dos clases independientes del main del programa, una para mostrar el menú (por muy sencilla que sea al tener un solo método) y otra para las matrices junto con sus operaciones.

Para el menú decidí utilizar un switch para las opciones, y do-while para que se repitiera según la condición: "eleccion!='c' && eleccion!='C'". Dentro del switch no coloque el caso para la opción "c", ya que esta siempre se revisara para terminar el ciclo.

En la clase TiposDatos hice el método "mostrarTiposDatos()", que servia para mostrar la tabla con la información de los distintos tipos de datos básicos. Para mostrar su tamaño en bits, utilicé el método "sizeof()" y lo multiplique por 8 ya que obtiene los valores en bytes. Para mostrar los valores máximos y mínimos de los diferentes tipos de datos utilicé las constantes de la librería "climits" y tambien las de "cfloat"(este caso en particular para las variables de tipo flotante porque no están definidas en la librería "climits").

Así se realizó la impresión de pantalla de una de las filas:

```
"cout << "Caracter con signo          "<< sizeof(char)*8 << "    "    <<
SCHAR_MIN<< "                "<<SCHAR_MAX << endl;"
```

Para la clase matriz añadí cuatro funciones, declare dos variables, un arreglo con tres matrices de tipo "double" y una variable para guardar el tamaño de la matriz :

```
"    public:
        Matriz();
        virtual ~Matriz();
        void crearMatrices();

    protected:
    private:
        double matriz[3][10][10];
        int tam;
        void llenarMatrices();
        void mostrarMatrices();
        void multiplicarMatrices();"
```



El metodo “crearMatrices()” crea las matrices con su respectivo tamaño y ademas las llena con valores aleatorios, también llama a las funciones “multiplicarMatrices()” y “eliminarMatrices()”.

```
“void Matriz::crearMatrices()
{
    int cont, subCont;
    bool repetir=true;
    do
    {
        cout << "Define el tamaño de las matrices (de 3 a 10)" << endl;
        cin >> tam;
        cin.ignore();

        if(3<=tam && tam<=10)
        {
            for (cont=0; cont < tam; cont++)
            {
                for(subCont=0; subCont < tam; subCont++)
                {
                    matriz[0][cont][subCont]=(1+rand()%(10001-1))*0.01;
                    matriz[1][cont][subCont]=(1+rand()%(10001-1))*0.01;
                }
            }
            multiplicarMatrices();
            mostrarMatrices();
            repetir=false;
        }
        else
        {
            cout << "El numero que ingreso no es valido";
        }
    }
    while(repetir);
}”
```

Para el desarrollo de la subrutina multiplicar matrices adapter el proceso matemático para realizar esta operación a código en C++.

```
“void Matriz::multiplicarMatrices()
{
    int cont, subCont, tercerCont,mat=0;
    float suma;
    cout << "\t Matriz: " << mat+1 << endl;
    for (cont=0; cont < tam; cont++)
    {
        cout << "\t";
        for(subCont=0; subCont < tam; subCont++)
        {
            for(tercerCont=0; tercerCont < tam; tercerCont++)
```



```
        {  
suma+=matriz[0][subCont][tercerCont]*matriz[1][tercerCont][cont];  
        }  
        matriz[2][subCont][cont]=suma;  
        suma=0;  
    }  
    cout << endl;  
}  
cout << endl;  
mat++;  
}  
}"
```

El me todo “mostrarMatrices()” hace la función de mostrar las matrices, junto con el resultado de su multiplicación, usando ciclos para recorrer las posiciones del arreglo y cambiar entre una matriz y otra.

```
“void Matriz::mostrarMatrices()  
{  
    int cont, subCont, mat=0;  
    do  
    {  
        cout << “\t MatrizZ: ” << mat+1 << endl;  
        for (cont=0; cont < tam; cont++){  
            cout << “\t”;  
            for(subCont=0; subCont < tam; subCont++){  
                cout << floorf((matriz[mat][cont][subCont]*100))/100 << “ ”;  
            }  
            cout << endl;  
        }  
        if(mat==0) {  
            cout << “\n\t    *” << endl;  
        }  
        else  
        {  
            if(mat==1)  
            {  
                cout << “\n\t    =” << endl;  
            }  
        }  
        cout << endl;  
        mat++;  
    }  
    while(mat<3);  
}"
```

Universidad de Guadalajara
CUCEI