

Universidad de Guadalajara

Centro Universitario de Ciencias Exactas e Ingenierías



Estructura de Datos

Actividad de Aprendizaje #01

Tipos de Dato primitivos y Tipos de Dato Estructurados

Alumna: Cervantes Araujo Maria Dolores

Código: 217782452

Fecha de Entrega: 27 enero de 2023



Autoevaluación			
Concepto	Si	No	Acumulación
Bajé el trabajo de internet o alguien me lo pasó (aunque sea de forma parcial)	-100 pts	0 pts	0
Incluí el código fuente en formato de texto (sólo si funciona cumpliendo todos los requerimientos)	+25pts	0 pts	25
Incluí las impresiones de pantalla (sólo si funciona cumpliendo todos los requerimientos)	+25pts	0 pts	25
Incluí una portada que identifica mi trabajo (nombre, código, materia, fecha, título)	+25 pts	0 pts	25
Incluí una descripción y conclusiones de mi trabajo	+25 pts	0 pts	25
Suma:			100

Introducción:

El programa cuenta con un mensaje de bienvenida, mostrando enseguida un menú con tres opciones para el usuario (*a. Tamaño y Rangos de los Tipos de Dato Primitivos*, *b. Ejemplo de uso de Tipo de dato Estructurado*, *c. Salir*), al ingresar a cualquiera de las dos primeras opciones le direcciona a otra función limpiando la pantalla y haciendo una pausa antes de mostrar los datos. Con la opción *a* el usuario puede visualizar una tabla donde se muestran los bits y el rango de cada tipo de dato primitivo que podemos manejar en el lenguaje respectivo; con la segunda opción al usuario se le solicita un número no menor que, 3 pero tampoco mayor que 10, esto para determinar un tamaño a la matriz y direccionarlo para que se puedan llenar las mismas, mostrando los resultados aleatoriamente en dos matrices y creando una tercera matriz donde suman las primeras dos matrices y una cuarta matriz de multiplicación; por último, la opción salir le da al usuario un mensaje de despedida y posteriormente cierra la ejecución del código.

Para esta actividad, fue necesario conocer el funcionamiento de otros recursos en la librería *random*, se abordó el problema analizando primero los requerimientos del programa, necesitando investigar un poco más de la cuenta para la sobrecarga de operadores con matrices.



Código Fuente:

Actividad1.h

```
#ifndef ACTIVIDAD1_H
#define ACTIVIDAD1_H
#include <limits>
#include <iostream>
#include <random>
#include <chrono>
#define filas 10
#define columnas 10

class actividad1 {
    float matriz[filas][columnas];
    float num;
public:
    actividad1();

    void tabla(void);
    void visualizacion(void);
    float getMatriz(int);
    void setMatriz(int);

    actividad1 operator+(actividad1&);
    actividad1 operator*(actividad1&);
};

#endif // ACTIVIDAD1_H
```

Actividad1.cpp

```
#include "actividad1.h"

using namespace std;

actividad1::actividad1() {}

void actividad1::tabla(void) {
    cout <<"Tipo de dato    \t\t|Bits |\t Valor Má-nimo    |\tValor Má-ximo\t|"<<endl;
    cout <<"Carãcter con signo:\t\t    "<< CHAR_BIT <<"\t    "<< SCHAR_MIN <<"\t\t\t"
    "<< SCHAR_MAX << '\n';
    cout <<"Carãcter sin signo:\t\t    "<< CHAR_BIT <<"\t\t"<<WCHAR_MIN<<"\t\t    "<<
    UCHAR_MAX << '\n';
    cout <<"Entero con signo:\t\t    "<<sizeof(signed short int)*CHAR_BIT<<"\t    "<<
    INT_MIN <<"\t\t"<< INT_MAX << '\n';
    cout <<"Entero sin signo:\t\t    "<<sizeof(unsigned short
    int)*CHAR_BIT<<"\t\t"<<numeric_limits<unsigned short int>::min()<<"\t\t"<<UINT_MAX
    << '\n';
    cout <<"Entero Largo con signo:\t\t    "<<sizeof(signed long int)*CHAR_BIT<<"\t
    "<<LONG_MIN<<"\t\t"<< LONG_MAX << '\n';
```



```
    cout << "Entero Largo sin signo:\t\t" << sizeof(unsigned long
int)*CHAR_BIT << "\t\t" << numeric_limits<unsigned long int>::min() << "\t\t" << ULONG_MAX
<< '\n';
    cout << "Real de precisión simple:\t\t" << sizeof(float)*CHAR_BIT << "\t\t"
<< numeric_limits<float>::min() << "\t\t" << numeric_limits<float>::max() << '\n';
    cout << "Real de doble precisión:\t\t" << sizeof(double)*CHAR_BIT << "\t\t" <<
numeric_limits<double>::min() << "\t\t" << numeric_limits<double>::max() << '\n';
}

///FUNCIONES PARA LLENAR LA MATRIZ A y B
void actividad1::setMatriz(int f) {
    ///Crea los números aleatorios
    default_random_engine
generator(chrono::system_clock::now().time_since_epoch().count());
    uniform_real_distribution<float> rango(-100.0,100.0);

    for(int x=0; x<f; x++) {

        for(int y=0; y<f; y++) {
            matriz[x][y]=rango(generator);
        }
    }
}

float actividad1::getMatriz(int f) {
    for(int x=0; x<f; x++) {

        for(int y=0; y<f; y++) {
            cout<<matriz[x][y]<<" \t ";
        }
        cout<<endl;
    }
    return num;
}

///SOBRECARGA DE OPERADORES PARA REALIZAR LA SUMA EN MATRIZ C
actividad1 actividad1::operator+(actividad1 &matr1) {
    actividad1 matrizC;
    for (int x=0; x < filas; x++) {
        for (int y=0; y < columnas; y++) {
            matrizC.matriz[x][y] = this->matriz[x][y] + matr1.matriz[x][y];
        }
    }
    return matrizC;
}

///SOBRECARGA DE OPERADORES PARA REALIZAR LA MULTIPLICACIÓN EN MATRIZ D
actividad1 actividad1::operator*(actividad1 &matr2) {
    actividad1 matrizD;
    for (int x=0; x < filas; x++) {
        for (int y=0; y < columnas; y++) {
```



```
matrizD.matriz[x][y] = this->matriz[x][y] * matriz2.matriz[x][y];  
}  
}  
return matrizD;  
}
```

///**FUNCIÓN PRINCIPAL**

```
void actividad1::visualizacion(void) {  
    int f, x, y;  
    actividad1 matrizA;  
    actividad1 matrizB;  
    actividad1 matrizC;  
    actividad1 matrizD;  
  
    cout<<"Ingresa un valor entre 3-10: "<<endl;  
    cin>>f;  
  
    if(f>=3&&f<=10) {  
        system("cls");  
        fflush(stdin);  
  
        cout<<"Matriz A: \n";  
        matrizA.setMatriz(f);  
        matrizA.getMatriz(f);  
  
        cout<<"\nMatriz B: \n";  
        matrizB.setMatriz(f);  
        matrizB.getMatriz(f);  
  
        cout<<"\nMatriz C: "<<endl;  
        matrizC = matrizA + matrizB;  
        for(x=0; x<f; x++) {  
            for(y=0; y<f; y++) {  
                cout<<matrizC.matriz[x][y]<<" \t ";  
  
            }  
            cout<<endl;  
        }  
  
        cout<<"\nMatriz D: "<<endl;  
        matrizD = matrizA * matrizB;  
        for(x=0; x<f; x++) {  
            for(y=0; y<f; y++) {  
                cout<<matrizD.matriz[x][y]<<" \t ";  
  
            }  
            cout<<endl;  
        }  
    }  
    else {  
        cout<<"Error No.444. Favor de digitar lo que se te pide"<<endl;  
    }  
}
```



Main.cpp

```
#include <iostream>
#include "actividad1.h"

using namespace std;

int main() {
    setlocale(LC_ALL, "");
    actividad1 tablaM;
    actividad1 opcionb;

    char i;
    int x=0;
    while(x==0) {
        system("cls");
        cout<<"BIENVENIDO :D"<<endl;
        cout<<"Selecciona una opción del menú:"<<endl;
        cout<<"a. Tamaño y Rangos de los Tipos de Dato Primitivos"<<endl;
        cout<<"b. Ejemplo de Uso de Tipo de Dato Estructurado"<<endl;
        cout<<"c. Salir "<<endl;
        cin>>i;

        switch (i) {
            case 'a':
                system("cls");
                tablaM.tabla();
                cout<<"\n";
                system("PAUSE()");
                break;
            case 'b':
                system("cls");
                fflush(stdin);
                opcionb.visualizacion();
                cout<<"\n";
                system("PAUSE()");
                break;
            case 'c':
                getchar(); system("cls");
                cout<<"HASTA LA PRÓXIMA!! :D"<<endl;
                x+=1;
                break;
            default:
                cout<<"Digita un valor dentro del rango, por favor\n"<<endl;
                system("PAUSE()");
                system("cls");
        }
    }
    return 0;
}
```



Impresiones de Pantalla:

```
"C:\Users\cerva\Escritorio\F.Prog\POO C++\Actividad1\bin\Debug\Actividad1.exe"

BIENVENIDO :D
Selecciona una opción del menú:
a. Tamaño y Rangos de los Tipos de Dato Primitivos
b. Ejemplo de Uso de Tipo de Dato Estructurado
c. Salir
```

```
"C:\Users\cerva\Escritorio\F.Prog\POO C++\Actividad1\bin\Debug\Actividad1.exe"

Tipo de dato      |Bits| Valor Mínimo | Valor Máximo |
Carácter con signo:      8   -128         127
Carácter sin signo:      8    0          255
Entero con signo:       16  -2147483648  2147483647
Entero sin signo:       16    0          4294967295
Entero Largo con signo:  32  -2147483648  2147483647
Entero Largo sin signo:  32    0          4294967295
Real de precisión simple: 32   1.17549e-38 3.40282e+38
Real de doble precisión: 64   2.22507e-308 1.79769e+308

Presione una tecla para continuar . . .
```

```
"C:\Users\cerva\Escritorio\F.Prog\POO C++\Actividad1\bin\Debug\Actividad1.exe"

BIENVENIDO :D
Selecciona una opción del menú:
a. Tamaño y Rangos de los Tipos de Dato Primitivos
b. Ejemplo de Uso de Tipo de Dato Estructurado
c. Salir
b
```

```
"C:\Users\cerva\Escritorio\F.Prog\POO C++\Actividad1\bin\Debug\Actividad1.exe"

Ingresa un valor entre 3-10:
4_
```



Universidad de Guadalajara
Centro Universitario de Ciencias Exactas e Ingenierías
ICOM – Ingeniería en Computación
Módulo Estructura de Datos



```
"C:\Users\cerva\Escritorio\F.Prog\POO C++\Actividad1\bin\Debug\Actividad1.exe"

Matriz A:
-9.94384      73.8818      -68.3903      -35.1165
-3.62917      4.59715      64.3881      -28.9709
85.3096       -1.80506     62.3498      -87.154
3.25475       -97.4848     -27.7056     -48.7249

Matriz B:
89.6116       -98.3528     -16.3115     52.3519
78.4018       98.6339     -59.6137     71.9492
50.1284       -92.5036     91.7881     82.5103
-49.3351      24.2525     11.7389     96.4085

Matriz C:
79.6677       -24.471      -84.7018     17.2354
74.7726       103.231     4.77437     42.9782
135.438       -94.3087    154.138     -4.64367
-46.0804      -73.2323    -15.9667     47.6836

Matriz D:
-891.083      -7266.49     1115.55     -1838.42
-284.533      453.435     -3838.42    -2084.44
4276.43       166.975     5722.97     -7191.1
-160.573      -2364.25    -325.235    -4697.5

Presione una tecla para continuar . . .
```

```
"C:\Users\cerva\Escritorio\F.Prog\POO C++\Actividad1\bin\Debug\Actividad1.exe"

BIENVENIDO :D
Selecciona una opción del menú:
a. Tamaño y Rangos de los Tipos de Dato Primitivos
b. Ejemplo de Uso de Tipo de Dato Estructurado
c. Salir
b
```

```
"C:\Users\cerva\Escritorio\F.Prog\POO C++\Actividad1\bin\Debug\Actividad1.exe"

Ingresa un valor entre 3-10:
6
```




Universidad de Guadalajara
Centro Universitario de Ciencias Exactas e Ingenierías
ICOM – Ingeniería en Computación
Módulo Estructura de Datos



"C:\Users\cerva\Escritorio\F.Prog\POO C++\Actividad1\bin\Debug\Actividad1.exe"

Matriz A:

66.9085	-68.6832	42.2443	0.539192	62.3165	-45.9693
-5.27914	73.5419	19.0348	-81.9037	44.3472	-56.0656
-94.3869	39.2914	-29.2811	73.0197	41.4392	69.3369
-55.4338	-75.4032	98.8452	90.706	95.3962	-76.1745
-64.7184	77.1919	-35.6242	63.3681	27.4079	44.3785
69.0744	-66.6478	49.6843	44.1908	-84.6707	-59.9092

Matriz B:

-57.2781	-73.5794	-48.7836	93.7808	-26.5977	-26.7996
-21.3715	9.78071	-15.7626	77.5375	-26.7349	66.7913
-39.2314	37.0247	74.5799	64.4204	-85.8601	-49.8822
29.5287	88.2546	95.2863	75.99	-36.3426	-10.6786
-74.6419	94.0996	-68.5291	31.7207	-70.2641	70.6078
-94.8803	-53.6124	-63.8981	64.2054	99.9618	57.2422

Matriz C:

9.63037	-142.263	-6.53927	94.32	35.7189	-72.7689
-26.6506	83.3226	3.27219	-4.36619	17.6123	10.7257
-133.618	76.3161	45.2988	137.44	-44.4208	19.4546
-25.9051	12.8514	194.131	166.696	59.0536	-86.8531
-139.36	171.291	-104.153	95.0888	-42.8563	114.986
-25.8059	-120.26	-14.2138	108.396	15.2911	-2.66697

Matriz D:

-3832.39	5053.66	-2060.83	50.5659	-1657.47	1231.96
112.823	719.292	-300.039	-6350.61	-1185.62	-3744.69
3702.94	1454.75	-2183.78	4703.96	-3557.98	-3458.68
-1636.89	-6654.68	9418.58	6892.75	-3466.95	813.434
4830.71	7263.72	2441.3	2010.08	-1925.79	3133.47
-6553.8	3573.15	-3174.73	2837.29	-8463.83	-3429.33

Presione una tecla para continuar . . .

"C:\Users\cerva\Escritorio\F.Prog\POO C++\Actividad1\bin\Debug\Actividad1.exe"

Ingresa un valor entre 3-10:

11

Error No.444. Favor de digitar lo que se te pide

Presione una tecla para continuar . . .



```
"C:\Users\cerva\Escritorio\F.Prog\POO C++\Actividad1\bin\Debug\Actividad1.exe"

BIENVENIDO :D
Selecciona una opción del menú:
a. Tamaño y Rangos de los Tipos de Dato Primitivos
b. Ejemplo de Uso de Tipo de Dato Estructurado
c. Salir
e
Digita un valor dentro del rango, por favor

Presione una tecla para continuar . . .
```

```
"C:\Users\cerva\Escritorio\F.Prog\POO C++\Actividad1\bin\Debug\Actividad1.exe"

BIENVENIDO :D
Selecciona una opción del menú:
a. Tamaño y Rangos de los Tipos de Dato Primitivos
b. Ejemplo de Uso de Tipo de Dato Estructurado
c. Salir
c
```

```
"C:\Users\cerva\Escritorio\F.Prog\POO C++\Actividad1\bin\Debug\Actividad1.exe"

HASTA LA PRÓXIMA!! :D

Process returned 0 (0x0)   execution time : 163.695 s
Press any key to continue.
```



Resumen Personal

En primera instancia, una vez analizado el problema y comprendido como programaríamos la solución, se crearon los métodos y atributos necesarios para el funcionamiento satisfactorio del código y sobre la marcha se fueron ajustando valores e incluso agregando más para la función de las matrices; ya que, solicitaba más especificaciones, siendo esta la parte que más me causo conflicto al momento de llenar las matrices y posteriormente para que me generara diferentes números aleatorios con cada ejecución del código, optando por separar y modificar la manera en la que estábamos creando el método, para hacerlo desde otra función aparte, a manera que se logaran los resultados esperados, haciendo no solo óptima la codificación, sino también que visualmente se viera estilizado y organizado para su comprensión total.

En lo personal, la actividad si bien no era compleja era entretenida, en especial para la *opción b* del menú que fue donde más complicaciones tuve; en un principio tuve problemas para mostrar diferentes números aleatorios con cada ejecución del código, dado que había puesto todo el código de las matrices en un mismo método, después opte por separar el llenado para *matriz A* y *matriz B* en una función aparte para mandarlas llamar en el método que ya tenía anteriormente, y en el *engine* utilizar la librería *Chrono*, de esta manera solucione ese error, generando los números adecuados y no grandes exponenciales.

Para realizar la sobrecarga no fue complejo, ya que me guie de los trabajos que tenía del semestre pasado, solo que al momento de apuntar a lo que tenía la matriz objeto, fue donde me causo ciertos conflictos, dado que no direccionaba bien en el apuntador de la matriz objeto, sino que creaba una nueva instancia de objeto dentro del constructor, haciendo que arrojara valores nulos al momento de llamarlo con el *get* y *set*; ya solo en cada iteración limpiaba el buffer en el programa. Con el programa pude recordar conocimientos del semestre pasado para poder aplicarlos en el problema, pero había detalles que si me costaron poquito más de lo normal; para la organización del código opte por definir el menú en el archivo *main.cpp* y los métodos solo mandarlos a llamar según la selección del usuario, comentando cada sección para no perderme entre las líneas del código.

El manejo de sobrecargas de operadores con matrices es interesante porque si bien no cambia la sintaxis, si cambia la manera de aplicarlo a comparación de utilizar otros tipos de datos o métodos, en los que no es tan necesario utilizar iteraciones, además de utilizar nuevas maneras de generar ciertos métodos.