

Informe de Laboratorio 01

Tema: Git y GitHub

Nota

Estudiante	Escuela	Asignatura
Carlos Alberto Barriga Arenas cbarrigaa@ulasalle.edu.pe	Carrera Profesional de Ingeniería de Software	Fundamentos de Lenguaje de Programación Semestre: IV Código:

Laboratorio	Tema	Duración
01	Git y GitHub	26 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2025 - A	Del 11 Mayo 2025	Al 14 Mayo 2025

1. EXAMEN PARCIAL

- Implemente una solución utilizando programación secuencial y programación con Threads, para el caso de estudio de suma de los elementos de una matriz cuadrada. Ambos programas deben utilizar POO(Programación Orientada a Objetos), por lo tanto considere utilizar constructores, getters, setters, programa principal. El programa debe realizar una simulación generando matrices desde 1x1 hasta NxN, con números aleatorios entre [0 y 9] y hallar la sumatoria de los elementos.

2. Equipos, materiales y temas utilizados

- Sistema Operativo Windows x64 bytes
- notepad
- compiler c++ online
- Git 2.39.2.
- Cuenta en GitHub con el correo institucional.
- Msys2
- Git
- Gnuplot
- Git

3. URL de Repositorio Github

- URL del Repositorio GitHub para clonar o recuperar.
- `https://github.com/carlos20030401/Examen-Parcial-.git`

4. Clonar un Repositorio GitHub

4.1. Crear un repositorio GitHub privado

- Clona un repositorio remoto como un repositorio local, en el cual se puede hacer push. (actualizarlo)
- Los repositorios privados solicitarán el usuario github y el token.

Listing 1: Git clone

```
$ git clone https://github.com/carlos20030401/Examen-Parcial-.git
```

4.2. Sincronizar Repositorio Local con su Repositorio GitHub

- Permite descargar los cambios del repositorio remoto al directorio local. (sincronizar)
- Los repositorios privados solicitarán el usuario github y el token.

Listing 2: git pull

```
$ git pull
```

5. Matriz - Secuencial

5.1. Matriz - Secuencial

- Creamos `matrices-secuencial.cpp`

Listing 3: Creamos en lab02

```
$ notepad matrices-secuencial.cpp
$ git add matrices-secuencial.cpp
$ git commit -m " colocando libreriaas"
$ git push -u origin main
```

```
generarAleatorios();
}

void generarAleatorios() {
    for (int i = 0; i < size; ++i)
        for (int j = 0; j < size; ++j)
            datos[i][j] = rand() % 10; // Numeros randoms o aleatorios de 0 a 9
}
```

- El commit seria :84ec10baf9d40fe7ee92b7ee70312e18524e27cf

```
commit 84ec10baf9d40fe7ee92b7ee70312e18524e27cf
Author: Carlos Alberto Barriga Arenas <cbarrigaa@ulasalle.edu.pe>
Date: Tue May 13 08:33:30 2025 -0500

AÑADIMOS la funcion para poder tener numeros aleatorios del 0 al 9
```

Listing 4: Actualisamos matrices-secuencial.cpp

```
$ notepad matrices-secuencial.cpp
$ git add matrices-secuencial.cpp
$ git commit -m "AÑADIMOS la funcion para poder tener numeros aleatorios del 0 al 9"
$ git push -u origin main
```

```
generarAleatorios();
}
void generarAleatorios() {
    for (int i = 0; i < size; ++i)
        for (int j = 0; j < size; ++j)
            datos[i][j] = rand() % 10; // Numeros randoms o aleatorios de 0 a 9
}
```

- El commit seria : a4ae5b85ecb99aff0f7de4617d69bdae7aad8b0e

```
commit a4ae5b85ecb99aff0f7de4617d69bdae7aad8b0e
Author: Carlos Alberto Barriga Arenas <cbarrigaa@ulasalle.edu.pe>
Date: Tue May 13 09:13:03 2025 -0500

se elimino parte del codigo que creaba directorio secuencial
```

- Este seria el codigo final

```
#include <ctime>
#include <chrono>
#include <fstream>
#include <direct.h> // Para crear directorios

using namespace std;
using namespace chrono;
@@ -22,7 +21,7 @@
void generarAleatorios() {
    for (int i = 0; i < size; ++i)
        for (int j = 0; j < size; ++j)
            datos[i][j] = rand() % 10; // Numeros aleatorios de 0 a 9
            datos[i][j] = rand() % 10; // Nmeros aleatorios de 0 a 9
}

int sumarElementos() {
@@ -34,44 +33,34 @@
}
};
```

```
bool crearDirectorio(const string& path) {
    return _mkdir(path.c_str()) == 0;
}

int main() {
    srand(time(0)); // Semilla para nmeros aleatorios
    int N;
    cout << "Ingrese el tamano maximo N de la matriz cuadrada: ";
    cin >> N;

    // Asegurate de que la carpeta 'secuencial' exista
    if (!crearDirectorio("secuencial")) {
        cerr << "No se pudo crear el directorio 'secuencial'.\n";
        return 1;
    }

    ofstream archivo("secuencial/secuencial.csv");
    ofstream archivo("secuencial.csv");
    if (!archivo.is_open()) {
        cerr << "No se pudo abrir secuencial.csv para escritura.\n";
        return 1;
    }

    // Escribe los encabezados en el archivo CSV
    archivo << "Tamano de matriz,Tiempo de ejecucion (ms)\n";

    for (int n = 1; n <= N; ++n) {
        auto inicio = high_resolution_clock::now();
        Matriz matriz(n);
        int suma = matriz.sumarElementos();
        auto fin = high_resolution_clock::now();

        duration<double, milli> duracion = fin - inicio;

        archivo << n << "," << duracion.count() << endl;
        cout << "Matriz " << n << "x" << n << ": suma = " << suma
            << ", tiempo = " << duracion.count() << " ms\n";
    }

    archivo.close();
    return 0;
}
```

```
C:\Users\Usuario\carlos20030401\Examen-Parcial-\secuencial>matriz-secuencial
Ingrese el tamano maximo N de la matriz cuadrada: 5
Matriz 1x1: suma = 5, tiempo = 0 ms
Matriz 2x2: suma = 26, tiempo = 0 ms
Matriz 3x3: suma = 42, tiempo = 0 ms
Matriz 4x4: suma = 72, tiempo = 0 ms
Matriz 5x5: suma = 105, tiempo = 0 ms
```

5.2. Pseudocodigo

```
Inicio
  Sembrar la semilla de números aleatorios usando la hora actual
  Mostrar "Ingrese el tamaño máximo N de la matriz cuadrada: "
  Leer N

  Abrir archivo "secuencial.csv" para escritura
  Si el archivo no se puede abrir
    Mostrar "No se pudo abrir secuencial.csv para escritura"
    Terminar programa

  Escribir en el archivo: "Tamaño de matriz,Tiempo de ejecución (ms)"

  Para n desde 1 hasta N hacer:
    Registrar tiempo de inicio

    Crear objeto Matriz de tamaño n
    Dentro del constructor:
      Crear matriz n x n
      Llenar cada celda con número aleatorio entre 0 y 9

    Llamar método para sumar todos los elementos de la matriz

    Registrar tiempo de fin

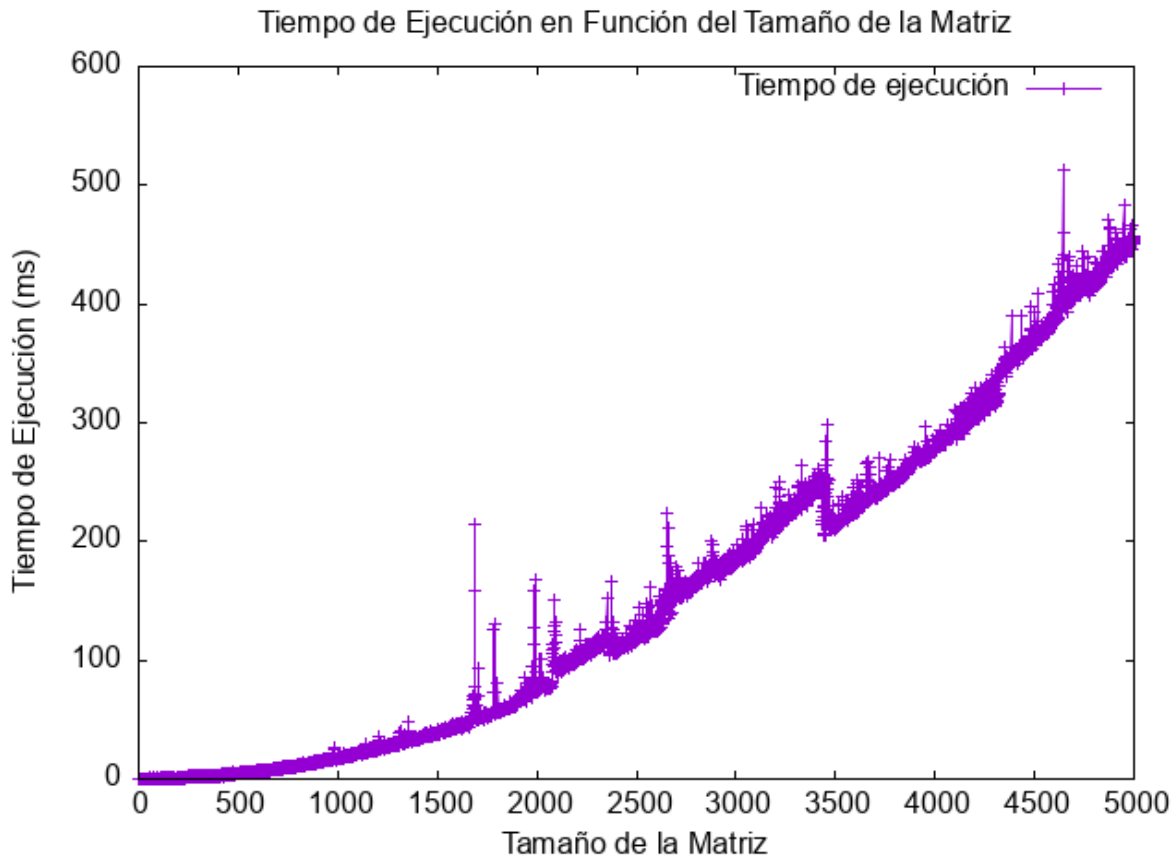
    Calcular duración en milisegundos (fin - inicio)

    Escribir en el archivo CSV: n, duración en milisegundos

    Mostrar en consola: tamaño de matriz, suma total, tiempo de ejecución

  Cerrar el archivo
Fin
```

- Al graficarlo



5.3. Matrices-Threads

- Creamos matrices-threads.cpp

Listing 5: Creamos en threads

```
$ notepad matrices-threads.cpp
$ git add matrices-threads.cpp
$ git commit -m " creamos lo que es en threads el .cpp"
$ git push -u origin main
```

```
#include <iostream>
#include <vector>
#include <thread>
#include <chrono>
#include <random>

using namespace std;
mutex mtx; // para proteger el acceso a la suma global

class Matriz {
private:
    int N;
```

```
vector<vector<int>> datos;  
public:  
    Matriz(int n) : N(n), datos(n, vector<int>(n)) {  
        random_device rd;  
        mt19937 gen(rd());  
        uniform_int_distribution<> dis(0, 9);  
        for (int i = 0; i < N; ++i)  
            for (int j = 0; j < N; ++j)  
                datos[i][j] = dis(gen);  
    }  
    int getValor(int i, int j) const {  
        return datos[i][j]  
    }
```

- El commit seria :5eef6f197e4847fc6e0745b57c1a57a6bf747bb0

```
commit 5eef6f197e4847fc6e0745b57c1a57a6bf747bb0  
Author: Carlos Alberto Barriga Arenas <cbarrigaa@ulasalle.edu.pe>  
Date: Tue May 13 16:19:03 2025 -0500  
  
este metodo nos permite consultar el valor en la posicion de la matriz
```

Listing 6: Actualisamos matrices-threads.cpp

```
$ notepad matrices-threads.cpp  
$ git add matrices-threads.cpp  
$ git commit -m "este metodo nos permite consultar el valor en la posicion de la  
matriz"  
$ git push -u origin main
```

```
#include <iostream>  
#include <vector>  
#include <thread>  
#include <chrono>  
#include <random>  
  
class Matriz {  
private:  
    int N;  
    vector<vector<int>> datos;  
public:  
    Matriz(int n) : N(n), datos(n, vector<int>(n)) {  
        random_device rd;  
        mt19937 gen(rd());  
        uniform_int_distribution<> dis(0, 9);  
        for (int i = 0; i < N; ++i)  
            for (int j = 0; j < N; ++j)  
                datos[i][j] = dis(gen);  
    }  
  
    int getValor(int i, int j) const {  
        return datos[i][j];  
    }  
  
    int getSize() const {
```

```
        return N;
    }
};

void sumarFila(const Matriz& matriz, int filaInicio, int filaFin, long long& sumaParcial) {
    long long localSuma = 0;
    for (int i = filaInicio; i < filaFin; ++i) {
        for (int j = 0; j < matriz.getSize(); ++j) {
            localSuma += matriz.getValor(i, j);
        }
    }
    lock_guard<mutex> lock(mtx);
    sumaParcial += localSuma;
}

int main() {
    int N;
    cout << "Ingrese el valor de N: ";
    cin >> N;
    ofstream archivo("threads.csv");
    archivo << "Tamao de matriz,Tiempo de ejecucion (ms)\n";

    for (int n = 1; n <= N; ++n) {
        Matriz matriz(n);
        long long sumaTotal = 0;
        int numThreads = thread::hardware_concurrency();
        vector<thread> threads
        int filasPorHilo = / numThreads;
        int resto = n % numThreads;
        auto inicio = chrono::high_resolution_clock::now();
        auto inicio = chrono::high_resolution_clock::now();
        int inicioFila = 0;
        for (int i = 0; i < numThreads; ++i) {
            int finFila = inicioFila + filasPorHilo + (i < resto ? 1 : 0);
            threads.emplace_back(sumarFila, ref(matriz), inicioFila, finFila, ref(sumaTotal));
            inicioFila = finFila;
        }
    }
}
```

- El commit seria : de8fd54c03b94a8b718b6962999239a842d3a4e0

```
commit de8fd54c03b94a8b718b6962999239a842d3a4e0
Author: Carlos Alberto Barriga Arenas <cbarrigaa@ulasalle.edu.pe>
Date: Tue May 13 16:29:19 2025 -0500

    sumamos todos los hilos threads que pues se hizo de forma paralela
```

- Este seria el codigo final

```
#include <iostream>
#include <vector>
#include <thread>
#include <chrono>
#include <fstream>
#include <mutex>
```



```
#include <random>

using namespace std;
mutex mtx; // para proteger el acceso a la suma global

class Matriz {
private:
    int N;
    vector<vector<int>> datos;
public:
    Matriz(int n) : N(n), datos(n, vector<int>(n)) {
        random_device rd;
        mt19937 gen(rd());
        uniform_int_distribution<> dis(0, 9);
        for (int i = 0; i < N; ++i)
            for (int j = 0; j < N; ++j)
                datos[i][j] = dis(gen);
    }

    int getValor(int i, int j) const {
        return datos[i][j];
    }

    int getSize() const {
        return N;
    }
};

void sumarFila(const Matriz& matriz, int filaInicio, int filaFin, long long& sumaParcial) {
    long long localSuma = 0;
    for (int i = filaInicio; i < filaFin; ++i) {
        for (int j = 0; j < matriz.getSize(); ++j) {
            localSuma += matriz.getValor(i, j);
        }
    }
    lock_guard<mutex> lock(mtx);
    sumaParcial += localSuma;
}

int main() {
    int N;
    cout << "Ingrese el tamaño máximo N de la matriz cuadrada: ";
    cin >> N;

    ofstream archivo("threads.csv");
    archivo << "Tamaño de matriz,Tiempo de ejecución (ms),Suma total\n";

    for (int n = 1; n <= N; ++n) {
        Matriz matriz(n);
        long long sumaTotal = 0;

        int numThreads = thread::hardware_concurrency();
        numThreads = numThreads > n ? n : numThreads;
        numThreads = numThreads > 0 ? numThreads : 4;

        vector<thread> threads;
```

```
int filasPorHilo = n / numThreads;
int resto = n % numThreads;

auto inicio = chrono::high_resolution_clock::now();

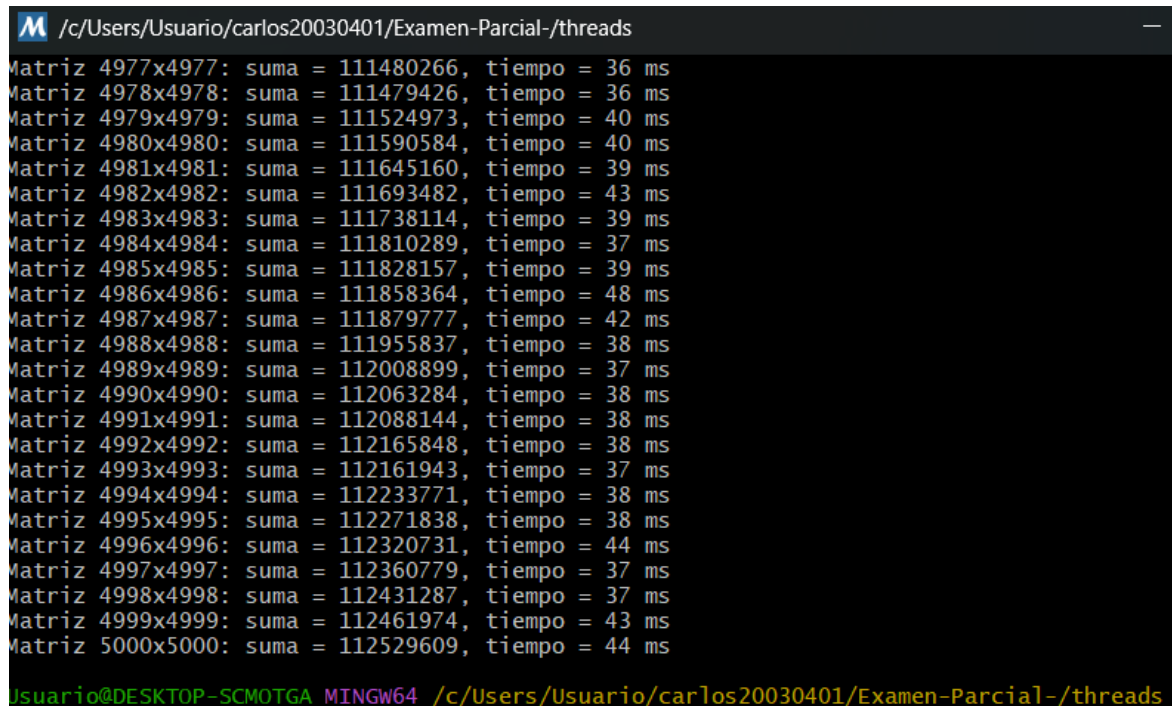
int inicioFila = 0;
for (int i = 0; i < numThreads; ++i) {
    int finFila = inicioFila + filasPorHilo + (i < resto ? 1 : 0);
    threads.emplace_back(sumarFila, ref(matriz), inicioFila, finFila, ref(sumaTotal));
    inicioFila = finFila;
}

for (auto& t : threads) {
    t.join();
}

auto fin = chrono::high_resolution_clock::now();
chrono::duration<double, milli> duracion = fin - inicio;

archivo << n << "," << duracion.count() << "," << sumaTotal << "\n";
cout << "Matriz " << n << "x" << n << ": suma = " << sumaTotal
    << ", tiempo = " << static_cast<int>(duracion.count()) << " ms" << endl;
}

archivo.close();
return 0;
}
```



```
M /c/Users/Usuario/carlos20030401/Examen-Parcial-/threads
Matriz 4977x4977: suma = 111480266, tiempo = 36 ms
Matriz 4978x4978: suma = 111479426, tiempo = 36 ms
Matriz 4979x4979: suma = 111524973, tiempo = 40 ms
Matriz 4980x4980: suma = 111590584, tiempo = 40 ms
Matriz 4981x4981: suma = 111645160, tiempo = 39 ms
Matriz 4982x4982: suma = 111693482, tiempo = 43 ms
Matriz 4983x4983: suma = 111738114, tiempo = 39 ms
Matriz 4984x4984: suma = 111810289, tiempo = 37 ms
Matriz 4985x4985: suma = 111828157, tiempo = 39 ms
Matriz 4986x4986: suma = 111858364, tiempo = 48 ms
Matriz 4987x4987: suma = 111879777, tiempo = 42 ms
Matriz 4988x4988: suma = 111955837, tiempo = 38 ms
Matriz 4989x4989: suma = 112008899, tiempo = 37 ms
Matriz 4990x4990: suma = 112063284, tiempo = 38 ms
Matriz 4991x4991: suma = 112088144, tiempo = 38 ms
Matriz 4992x4992: suma = 112165848, tiempo = 38 ms
Matriz 4993x4993: suma = 112161943, tiempo = 37 ms
Matriz 4994x4994: suma = 112233771, tiempo = 38 ms
Matriz 4995x4995: suma = 112271838, tiempo = 38 ms
Matriz 4996x4996: suma = 112320731, tiempo = 44 ms
Matriz 4997x4997: suma = 112360779, tiempo = 37 ms
Matriz 4998x4998: suma = 112431287, tiempo = 37 ms
Matriz 4999x4999: suma = 112461974, tiempo = 43 ms
Matriz 5000x5000: suma = 112529609, tiempo = 44 ms

Usuario@DESKTOP-SCMOTGA MINGW64 /c/Users/Usuario/carlos20030401/Examen-Parcial-/threads
```

- Aca cabe aclarar que se uso Msys2 por problemas de versiones en cmd y g++

5.4. Pseudocódigo

```
Inicio
  Mostrar "Ingrese el tamaño máximo N de la matriz cuadrada:"
  Leer N

  Abrir archivo "threads.csv" para escritura
  Escribir en el archivo: "Tamaño de matriz,Tiempo de ejecución (ms),Suma total"

  Para n desde 1 hasta N hacer:
    Crear matriz de tamaño n x n con valores aleatorios entre 0 y 9

    Inicializar variable sumaTotal en 0

    Obtener número de hilos disponibles en el sistema
    Si número de hilos > n, usar n hilos
    Si número de hilos no válido (0), usar 4 hilos por defecto

    Calcular cuántas filas manejará cada hilo: filasPorHilo = n / numThreads
    Calcular el resto de filas sobrantes: resto = n % numThreads

    Registrar el tiempo de inicio

    Para cada hilo i desde 0 hasta numThreads - 1 hacer:
      Calcular filaInicio y filaFin para ese hilo
      Crear hilo que sume los valores desde filaInicio hasta filaFin
      El hilo suma todos los valores en su rango y suma el resultado a sumaTotal
      (protegido con mutex para evitar condiciones de carrera)

    Esperar a que todos los hilos terminen (join)

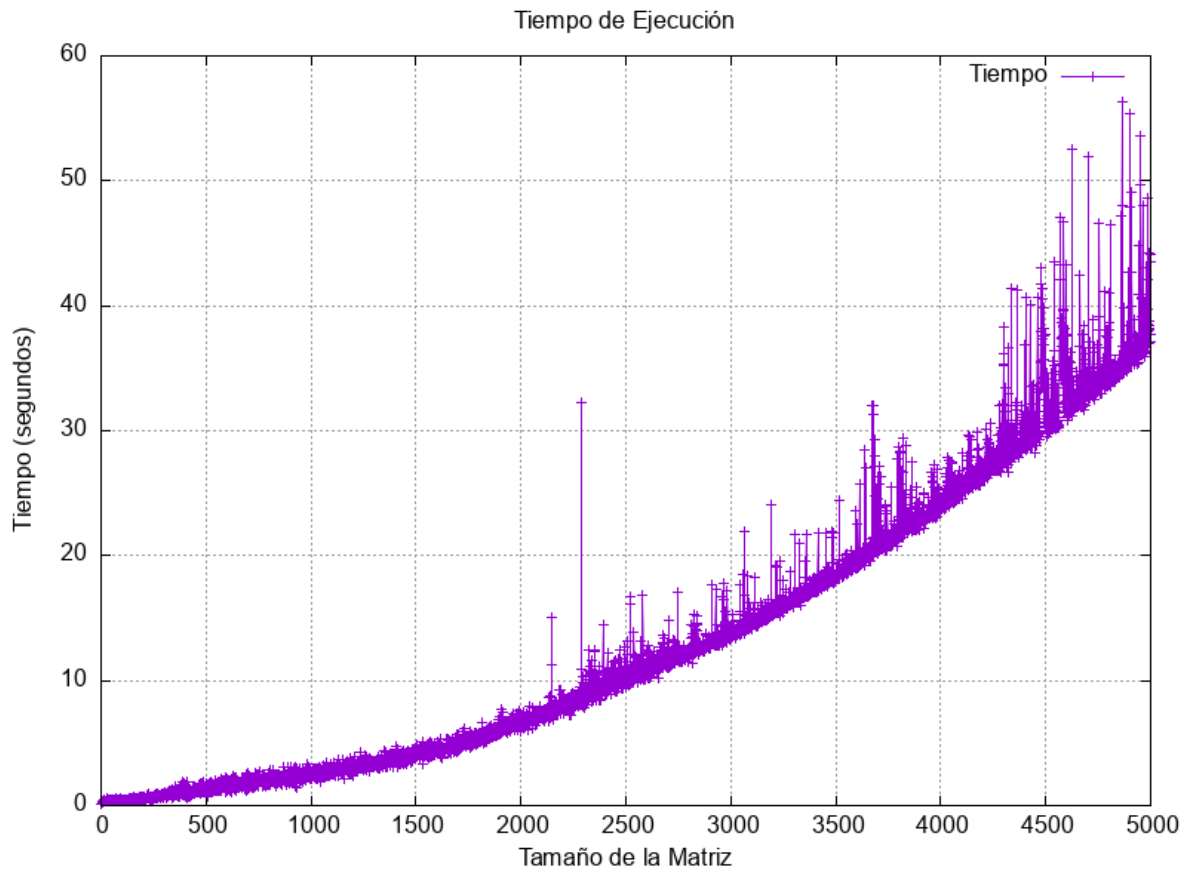
    Registrar el tiempo de fin

    Calcular duración en milisegundos = fin - inicio

    Escribir en el archivo: tamaño de matriz, duración, suma total
    Mostrar en consola: tamaño de matriz, suma total, duración

  Cerrar el archivo
Fin
```

- Al graficarlo



- Se probó con 5000 matrices demoro aprox. unos 15 min

6. Calificación

	Indicador	Excelente	Adecuado	Mínimo	Insuficiente
5	Guía en LATEX del framework	El entregable cumple con todos los pasos de la guía base entregada y es reproducible para obtener el producto.	El entregable cumple con las consideraciones necesarias. Pero pudieron considerarse más detalles.	El entregable no cumple con todas las consideraciones necesarias.	No hay entregable. Carece de las consideraciones o está incompleto.
3	Puntualidad y responsabilidad	Asistencia en punto el día de la presentación y con toda la predisposición a presentar el trabajo.	Asistencia el día de la presentación sin considerar los necesario para que todo se desarrolle de forma correcta.	Asistencia la presentación tarde y con poca predisposición para presentar el trabajo.	No asiste o llega muy tarde a la presentación o no muestra predisposición a presentar el trabajo.
3	Manejo del tiempo	Utiliza el tiempo de manera eficiente.	Ovia algunos detalles, pero se ha abarcado todo lo necesario.	Necesita más del tiempo y no puede concluir con su presentación.	No presenta su trabajo.
5	Obtención del producto final	Muestra el producto final y explica cada uno de sus componentes utilizando la guía paso a paso.	Muestra el producto final y explica lo necesario de componentes obviando la guía paso a paso.	Producto final parcialmente terminado con explicación mínima.	No hay producto final.
4	Majeo del escenario palabra y seguridad	Expone manejando el escenario preocupándose para que todos entiendan con una voz adecuada para ello.	Hay observaciones en el manejo del escenario, pero es aceptable.	Carece de entusiasmo para manejar el escenario.	No se presenta.
20		17-20	14-16	8-13	0-7

7. Referencias

- <https://github.com/>
- <https://www.onlinegdb.com/fork/trSsLe7w1>
- <https://www.msys2.org/#sponsors>
- <https://www.mingw-w64.org/downloads/>
- <https://azulschool.net/todos-los-grupos/grupo-de-c/forum/topic/generar-numeros-aleatorios-en-c/>
- <https://www.w3resource.com/cpp-exercises/vector/cpp-vector-exercise-3.php>