

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

INTRODUCCION A LA PROGRAMACION Y COMPUTACION 2

CATEDRÁTICO: ING. FERNANDO JOSE PAZ

TUTOR ACADÉMICO: DENILSON FLORENTÍN DE LEÓN AGUILAR



Brayan Emanuel García

CARNÉ: 202300848

SECCIÓN: A

GUATEMALA, 4 DE SEPTIEMBRE DEL 2,024

ÍNDICE

Contenido

PROYECTO 1	0
ÍNDICE	1
OBJETIVOS DEL MANUAL	2
1. GENERAL	2
2. ESPECÍFICOS	2
INTRODUCCIÓN	3
INFORMACIÓN DEL SISTEMA	4
REQUISITOS DEL SISTEMA	5
FLUJO DE LAS FUNCIONALIDADES DEL SISTEMA.....	6
Menú inicial:	6
Cargar Archivo:.....	6
Procesar archivo:.....	7
Escribir Archivo de salida:.....	8
Datos del estudiante:	8
Generar Grafica:.....	9
Salir:	12

OBJETIVOS DEL MANUAL

1. GENERAL

- Instruir al usuario en el uso de las funcionalidades del sistema

2. ESPECÍFICOS

- Proporcionar orientación clara y concisa sobre el manejo de la interfaz de usuario basada en consola
- Facilitar la resolución de problemas comunes y ofrecer soluciones prácticas
- Informar acerca de los límites del sistema.

INTRODUCCIÓN

Este manual de usuario está diseñado para guiar a los usuarios en la utilización de nuestro sistema de optimización de control de accesos, una solución avanzada creada para abordar un problema NP-Hard asociado con la optimización de procesos e hilos en una red de computadoras. Los problemas NP-Hard, caracterizados por su alta complejidad y demanda de recursos, a menudo presentan desafíos significativos en términos de tiempo de procesamiento y memoria.

El objetivo de nuestro sistema es reducir las matrices de acceso, mejorando así la eficiencia en el uso de memoria. Además, el sistema incluye una herramienta de visualización basada en grafos, que permite observar de manera clara cómo varían las matrices a lo largo del proceso de optimización. Esta solución innovadora busca mitigar las limitaciones inherentes a la resolución de instancias grandes de problemas NP-Hard, facilitando una gestión más efectiva y eficiente de los recursos computacionales.

INFORMACIÓN DEL SISTEMA

El sistema de gestión de ventas de automóviles está mejorando el rendimiento de los diversos problemas que se pueden presentar al no optimizar el funcionamiento de accesos de sistemas a los servidores.. El sistema opera mediante una interfaz de usuario basada en consola, donde los usuarios pueden interactuar fácilmente con las diferentes funcionalidades del sistema.

Principales Características

1. **Cargar matriz:** Permite a los usuarios registrar un archivo dentro del programa en formato XML para el posterior análisis de la matriz que se encuentra en el archivo.
2. **Análisis del archivo:** Este apartado se encarga de reducir las matrices de acceso completas a unas resumidas
3. **Creación de salida:** Crea un archivo “.xml” que registra la matriz reducida para su uso en otros programas.
4. **Generación de Grafos:** El sistema permite la visualización de todas las matrices que tenemos almacenadas en el sistema actual y la creación de un grafo que nos permite visualizar de mejor manera los datos..
5. **Interfaz de Usuario Basada en Consola:** La interacción con el sistema se realiza a través de una serie de menús y opciones que se presentan en la consola, permitiendo una navegación intuitiva y directa.

REQUISITOS DEL SISTEMA

- Hardware:
 - 2GB De memoria Ram
 - Tarjeta de video
 - Procesador con al menos 1Ghz
 - 10 MB de espacio.
 - Teclado para interacción de consola
 - Pantalla
- Software:
 - SO: MacOS, Windows 10 y posteriores,
Linux
 - Python 3.8
 - Editor deCodigo

FLUJO DE LAS FUNCIONALIDADES DEL SISTEMA

Para correr el programa nos ubicamos en la carpeta donde tenemos el programa y escribimos: “python main.py” con esto empieza la ejecución.

**** Nota 1:** No separar todos los archivos que trae la carpeta, de esto depende la funcionalidad.

Menú inicial:

Se muestra un menú en el cual se puede escribir la opción que se quiera. 1,2,3,4, 5 o 6. Aunque al inicio se recomienda que se seleccione la opción 1 ya que esta cargara el archivo con la matriz a analizar, se hace escribiendo en consola el numero de la opción.

```
PS D:\GITHUB\IPC2_Proyecto1_202300848> python .\main.py
1. Cargar archivo
2. Procesar archivo
3. Escribir archivo salida
4. Datos del estudiante
5. Generar grafica
6. Salir
Seleccione una opción: 1
```

Cargar Archivo:

Si seleccionas la opción uno, tendrás que escribir el nombre de tu archivo .xml que contiene la matriz. Tomar en cuenta que debe de estar en la misma carpeta que el archivo main, así que es recomendable pasar el archivo .xml a la carpeta.

```
Seleccione una opción: 1
Ingrese la ruta del archivo XML: entrada.xml
Archivo cargado exitosamente.
```

El archivo .xml de entrada debe de tener la siguiente estructura:

```
entrada.xml
1  <matrices>
2    <matriz nombre="Ejemplo" n="5" m="4">
3      <dato x="1" y="1">2</dato>
4      <dato x="1" y="2">3</dato>
5      <dato x="1" y="3">0</dato>
6      <dato x="1" y="4">4</dato>
7      <dato x="2" y="1">0</dato>
8      <dato x="2" y="2">0</dato>
9      <dato x="2" y="3">6</dato>
10     <dato x="2" y="4">3</dato>
11     <dato x="3" y="1">3</dato>
12     <dato x="3" y="2">4</dato>
13     <dato x="3" y="3">0</dato>
14     <dato x="3" y="4">2</dato>
15     <dato x="4" y="1">1</dato>
16     <dato x="4" y="2">0</dato>
17     <dato x="4" y="3">1</dato>
18     <dato x="4" y="4">5</dato>
19     <dato x="5" y="1">0</dato>
20     <dato x="5" y="2">0</dato>
21     <dato x="5" y="3">3</dato>
22     <dato x="5" y="4">1</dato>
23   </matriz>
24 </matrices>
```

Do you want to install the extension from tomoki1

Procesar archivo:

Luego de cargar el archivo. Regresamos al menú principal e ingresamos el numero 2. Esto hará la reducción de la matriz y nos mostrará en terminal el resultado.

```
6. Salir
Seleccione una opción: 2
Fila reducida 0: 5 -> 7 -> 0 -> 6
Fila reducida 1: 0 -> 0 -> 9 -> 4
Fila reducida 2: 1 -> 0 -> 1 -> 5
1
Matriz Reducida Ejemplo_reducida:
Fila 1: [5, 7, 0, 6]
Fila 2: [0, 0, 9, 4]
Fila 3: [1, 0, 1, 5]
Archivo procesado exitosamente.
```


Escribir Archivo de salida:

Continuando ingresamos el numero 3. Lo que nos permitirá crear un .xml con la información de la matriz reducida. Para guardarla tenemos que escribir el nombre del archivo y su extensión. Puede ser la que sea, pero se recomienda que sea “.xml”. El archivo se guardara en la misma carpeta con el nombre que nosotros elegimos.

```
1. Cargar archivo
2. Procesar archivo
3. Escribir archivo salida
4. Datos del estudiante
5. Generar grafica
6. Salir
Seleccione una opción: 3
Ingrese la ruta para el archivo de salida XML: matrizRedSalida.xml
1
Fila 1 de la matriz 'Ejemplo_reducida': [5, 7, 0, 6]
Escribiendo dato en posición (1, 1): 5
Escribiendo dato en posición (1, 2): 7
Escribiendo dato en posición (1, 3): 0
Escribiendo dato en posición (1, 4): 6
Fila 2 de la matriz 'Ejemplo_reducida': [0, 0, 9, 4]
Escribiendo dato en posición (2, 1): 0
Escribiendo dato en posición (2, 2): 0
Escribiendo dato en posición (2, 3): 9
Escribiendo dato en posición (2, 4): 4
Fila 3 de la matriz 'Ejemplo_reducida': [1, 0, 1, 5]
Escribiendo dato en posición (3, 1): 1
Escribiendo dato en posición (3, 2): 0
Escribiendo dato en posición (3, 3): 1
Escribiendo dato en posición (3, 4): 5
Archivo de salida escrito exitosamente.
```

Ejemplo de como el archivo de salida es generado.

```
<?xml version='1.0' encoding='utf-8'?>
<matrices><matriz nombre="Ejemplo_reducida" n="3" m="4"><dato x="1" y="1">5</dato><dato x="1" y="2">7</dato>
```

****Nota 2:** aunque aparezca así, por como fue estructurado aunque es difícil de leer para un humano, para una maquina no hay diferencias.

Datos del estudiante:

Muestra la información del autor del programa.

```
Archivo de salida escrito exitosamente.
1. Cargar archivo
2. Procesar archivo
3. Escribir archivo salida
2. Procesar archivo
3. Escribir archivo salida
3. Escribir archivo salida
4. Datos del estudiante
5. Generar grafica
6. Salir
Seleccione una opción: 4
Brayan Emanuel Garcia
```

Generar Grafica:

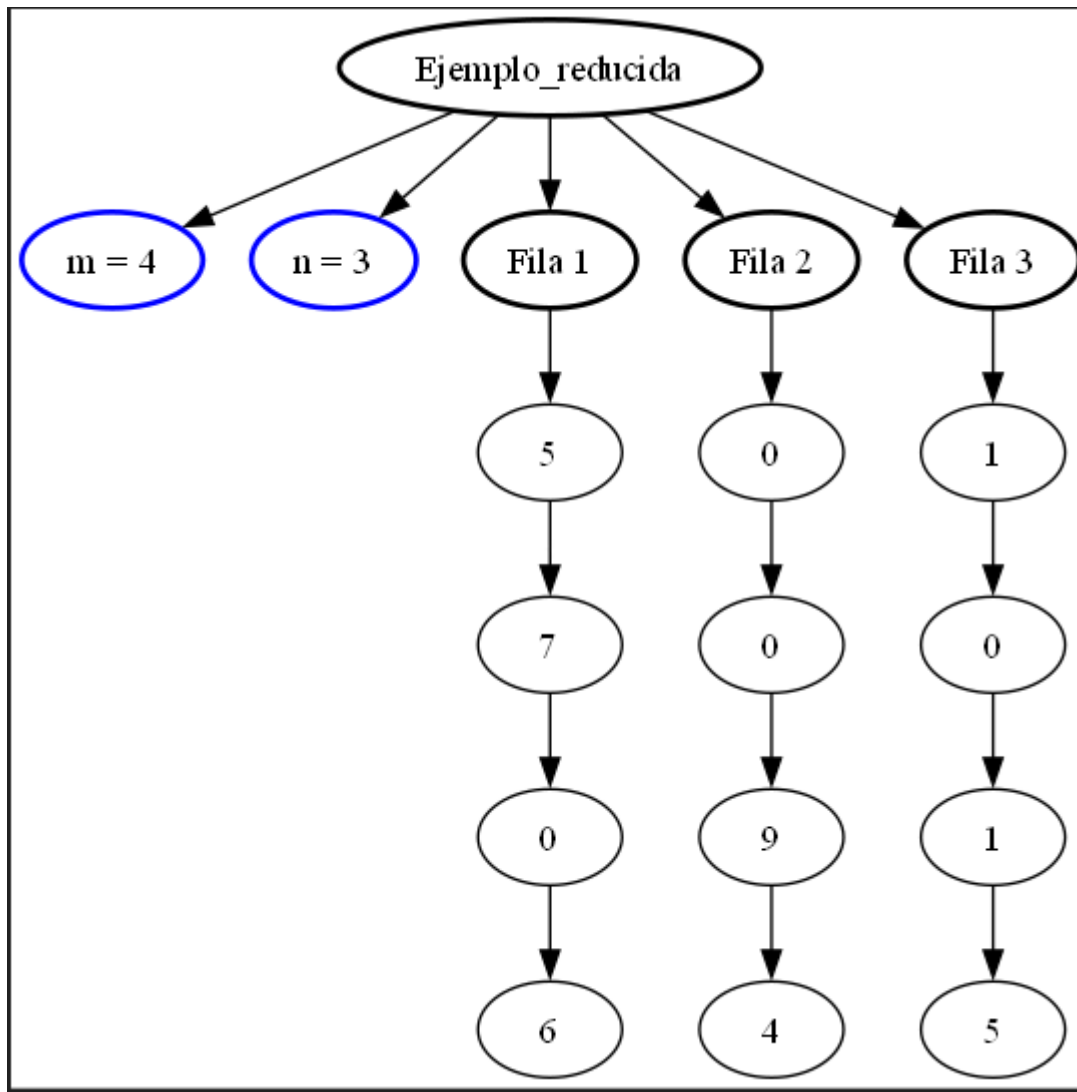
Ingresando el numero 5 a la consola nos abrirá un menu, con las matrices que tengamos disponibles para crear el grafo, Sino hemos hecho el proceso de archivo(opción 2) solo nos aparecerá para imprimir o crear grafo la matriz sacada del xml de entrada. Si no esta el archivo de entrada, nos mostrara un mensaje de que no hay matrices para trabajar.

```
1. Cargar archivo
2. Procesar archivo
3. Escribir archivo salida
4. Datos del estudiante
5. Generar grafica
6. Salir
Seleccione una opción: 5
Empezando a generar grafo
1
MATRICES DISPONIBLES:
1. Ejemplo (Original)
2. Ejemplo_reducida (Reducida)
Seleccione la matriz a graficar: █
```

En este menu se vuelve a ingresar el numero de archivo que nos va a imprimir o generar el grafo. Al seleccionar la opción, se nos crearan dos archivos, una imagen “.png” u uno sin extensión.

```
1. Ejemplo (Original)
2. Ejemplo_reducida (Reducida)
Seleccione la matriz a graficar: 2
Grafo de la matriz 'Ejemplo_reducida' generado y guardado con éxito.
```

El archivo .png nos mostrara el grafo directamente.



** Nota 3, m y n representan el numero de columnas y filas respectivamente. Ambos archivos creados tienen el mismo nombre, solo cambia la extensión, el nombre es: “grafo_matriz.png” y “grafo_matriz”.

El otro archivo contendrá el código para poder usarlo en otros lugares.

```
grafo_matriz
1  digraph {
2      rankdir=TB
3      Ejemplo_reducida [shape=ellipse style=bold]
4      m [label="m = 4" color=blue shape=ellipse style=bold]
5      n [label="n = 3" color=blue shape=ellipse style=bold]
6      Ejemplo_reducida -> m
7      Ejemplo_reducida -> n
8      "Fila 1" [label="Fila 1" shape=ellipse style=bold]
9      Ejemplo_reducida -> "Fila 1"
10     "Fila 1_dato_1" [label=5 shape=ellipse]
11     "Fila 1" -> "Fila 1_dato_1"
12     "Fila 1_dato_2" [label=7 shape=ellipse]
13     "Fila 1_dato_1" -> "Fila 1_dato_2"
14     "Fila 1_dato_3" [label=0 shape=ellipse]
15     "Fila 1_dato_2" -> "Fila 1_dato_3"
16     "Fila 1_dato_4" [label=6 shape=ellipse]
17     "Fila 1_dato_3" -> "Fila 1_dato_4"
18     "Fila 2" [label="Fila 2" shape=ellipse style=bold]
19     Ejemplo_reducida -> "Fila 2"
20     "Fila 2_dato_1" [label=0 shape=ellipse]
21     "Fila 2" -> "Fila 2_dato_1"
22     "Fila 2_dato_2" [label=0 shape=ellipse]
23     "Fila 2_dato_1" -> "Fila 2_dato_2"
24     "Fila 2_dato_3" [label=9 shape=ellipse]
25     "Fila 2_dato_2" -> "Fila 2_dato_3"
26     "Fila 2_dato_4" [label=4 shape=ellipse]
27     "Fila 2_dato_3" -> "Fila 2_dato_4"
28     "Fila 3" [label="Fila 3" shape=ellipse style=bold]
29     Ejemplo_reducida -> "Fila 3"
30     "Fila 3_dato_1" [label=1 shape=ellipse]
31     "Fila 3" -> "Fila 3_dato_1"
32     "Fila 3_dato_2" [label=0 shape=ellipse]
33     "Fila 3_dato_1" -> "Fila 3_dato_2"
34     "Fila 3_dato_3" [label=1 shape=ellipse]
35     "Fila 3_dato_2" -> "Fila 3_dato_3"
36     "Fila 3_dato_4" [label=5 shape=ellipse]
37     "Fila 3_dato_3" -> "Fila 3_dato_4"
38     {
39         rank=same
40         m
41         n
42     }
43 }
44
```

Salir:

Esta ultima opción a la cual entramos y termina el programa de manera segura.

```
Seleccione una opción: 6
PS D:\GITHUB\IPC2_Proyecto1_202300848>
```

****Nota 4:** Solo se puede trabajar un archivo a la vez y cada archivo tiene una matriz.