UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

INTRODUCCION A LA PROGRAMACION Y COMPUTACION 2

CATEDRÁTICO: FERNANDO JOSE PAZ

TUTOR ACADÉMICO: DENILSON FLORENTIN DE LEON AGUILAR



**MANUAL TÉCNICO**

NOMBRE: BRAYAN EMANUEL GARCIA

CARNÉ: 202300848

SECCIÓN: P

GUATEMALA, 4 DE SEPTIEMBRE DEL 2,024

# ÍNDICE

Contenido

[ÍNDICE 1](#_Toc175787659)

[INTRODUCCIÓN 2](#_Toc175787660)

[OBJETIVOS 3](#_Toc175787661)

[1. GENERAL 3](#_Toc175787662)

[2. ESPECÍFICOS 3](#_Toc175787663)

[ALCANCES DEL SISTEMA 3](#_Toc175787664)

[ESPECIFICACIÓN TÉCNICA 4](#_Toc175787665)

[REQUISITOS DE HARDWARE 4](#_Toc175787666)

[REQUISITOS DE SOFTWARE 4](#_Toc175787667)

[LÓGICA DEL PROGRAMA 4](#_Toc175787668)

# INTRODUCCIÓN

El presente proyecto tiene como objetivo desarrollar un sistema de análisis y procesamiento de matrices a partir de archivos XML. Este sistema está diseñado para leer matrices desde archivos XML, analizar patrones, reducir matrices según ciertos criterios, y finalmente generar un grafo que represente la estructura de las matrices. Los resultados se guardan en un archivo XML y se presentan visualmente mediante un grafo. El sistema está desarrollado en Python utilizando módulos estándar.

# OBJETIVOS

## GENERAL

* Desarrollar un sistema que lea, procese y reduzca matrices desde archivos XML, y genere un grafo para representar la estructura de las matrices.

## ESPECÍFICOS

* Implementar un lector XML que pueda extraer matrices desde archivos XML.
* Desarrollar un analizador de matrices que identifique patrones y reduzca matrices según criterios predefinidos.
* Crear una clase para representar y mostrar grafo basado en las matrices procesadas.
* Implementar un escritor XML para guardar matrices reducidas en un archivo XML.
* Desarrollar un menú de usuario que permita interactuar con el sistema para cargar archivos, procesar datos y generar reportes.

# ALCANCES DEL SISTEMA

El sistema permite:

* Leer matrices desde archivos XML.
* Analizar patrones en matrices y reducirlas según criterios específicos.
* Generar un grafo visual que represente la estructura de las matrices.
* Escribir las matrices reducidas en un archivo XML.

El sistema no permite:

* Modificar la estructura de los archivos XML de entrada.
* Generar gráficos complejos que no se basen en la estructura de las matrices.

# ESPECIFICACIÓN TÉCNICA

## REQUISITOS DE HARDWARE

* + 2 Gb de memoria Ram
  + 500MB de almacenamiento interno
  + Procesador con 1Ghz de velocidad
  + Tarjeta de video integrada
  + Pantalla

## REQUISITOS DE SOFTWARE

* **Sistema Operativo**:
  + **Windows 7/8/10/11** o superior.
  + **Linux**: Cualquier distribución moderna (Ubuntu, Fedora, etc.).
  + **macOS**: Versiones recientes.
* **Entorno de Desarrollo**:
  + **Python 3.8** o superior: El sistema está desarrollado en Python, por lo que se requiere la instalación de este lenguaje.
  + **Editor de Código/IDE**: Se recomienda el uso de un editor de texto como **Visual Studio Code**, **PyCharm**, o **Sublime Text** para desarrollar, ejecutar y modificar el código.
  + **Bibliotecas**: No se requieren bibliotecas adicionales de Python, ya que el sistema solo utiliza módulos estándar como csv.

# LÓGICA DEL PROGRAMA

**LÓGICA DEL PROGRAMA**

**Clases**

**EscritorXML**

* **Atributos:**
  + matrices\_reducidas: ListaEnlazada
* **Métodos:**
  + escribir(archivo\_salida)
    - Escribe las matrices reducidas en un archivo XML de salida.

**LectorXML**

* **Atributos:**
  + archivo: str
  + matrices: ListaEnlazada
* **Métodos:**
  + leer()
    - Lee y carga matrices desde el archivo XML.
  + obtener\_matrices(): ListaEnlazada
    - Retorna la lista de matrices leídas.

**AnalizadorMatrices**

* **Atributos:**
  + matrices: ListaEnlazada
  + matrices\_reducidas: ListaEnlazada
* **Métodos:**
  + analizar\_patrones(matriz: Matriz)
    - Analiza patrones en una matriz específica.
  + reducir\_matriz(matriz: Matriz)
    - Reduce la matriz según criterios predefinidos.
  + procesar()
    - Procesa todas las matrices, aplicando análisis y reducción.
  + obtener\_matrices\_reducidas(): ListaEnlazada
    - Retorna la lista de matrices reducidas.

**Grafo**

* **Atributos:**
  + matriz: Matriz
  + grafo: Digraph
* **Métodos:**
  + crear\_nodos\_matriz()
    - Crea nodos en el grafo basado en la matriz.
  + mostrar\_grafo()
    - Muestra el grafo visualmente.

**Matriz**

* **Atributos:**
  + nombre: str
  + n: int
  + m: int
  + filas: ListaEnlazada
* **Métodos:**
  + agregar\_dato(x: int, y: int, valor: int)
    - Agrega un valor a la posición (x, y) en la matriz.
  + obtener\_dato(x: int, y: int): int
    - Obtiene el valor en la posición (x, y) en la matriz.
  + mostrar\_matriz()
    - Muestra la matriz en formato legible.

**Nodo**

* **Atributos:**
  + dato: any
  + siguiente: Nodo

**ListaEnlazada**

* **Atributos:**
  + cabeza: Nodo
  + tamaño: int
* **Métodos:**
  + agregar(dato: any)
    - Agrega un dato al final de la lista.
  + obtener(indice: int): any
    - Obtiene el dato en la posición especificada.
  + eliminar(indice: int)
    - Elimina el dato en la posición especificada.
  + insertar(indice: int, dato: any)
    - Inserta un dato en la posición especificada.
  + mostrar(): str
    - Muestra todos los datos de la lista.
  + \_\_len\_\_(): int
    - Retorna el tamaño de la lista.

**Menu**

* **Atributos:**
  + lector\_xml: LectorXML
  + analizador\_matrices: AnalizadorMatrices
  + escritor\_xml: EscritorXML
* **Métodos:**
  + cargar\_archivo(ruta: str)
    - Carga el archivo XML desde la ruta especificada.
  + procesar\_archivo()
    - Procesa el archivo XML cargado.
  + escribir\_archivo\_salida(ruta\_salida: str)
    - Escribe el archivo de salida con las matrices reducidas.
  + generar\_grafo()
    - Genera y muestra el grafo basado en las matrices procesadas.

**Flujo de Ejecución**

1. **Carga del Archivo:**
   * El usuario utiliza la opción cargar\_archivo del Menu para especificar la ruta del archivo XML.
   * LectorXML lee el archivo XML y carga las matrices en la lista matrices.
2. **Procesamiento de Matrices:**
   * Menu invoca el método procesar\_archivo, que a su vez llama a AnalizadorMatrices.
   * AnalizadorMatrices analiza cada matriz para identificar patrones y las reduce según criterios establecidos.
3. **Generación del Grafo:**
   * Menu llama al método generar\_grafo.
   * Grafo crea nodos y muestra el grafo basado en la matriz procesada.
4. **Escritura del Archivo de Salida:**
   * Menu utiliza escribir\_archivo\_salida para guardar las matrices reducidas en un archivo XML mediante EscritorXML.