

Universidad San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas
Introducción a la Programación y Computación 2



Inga. Claudia Liceth Rojas Morales
Ing. Marlon Antonio Pérez Türk
Ing. William Estuardo Escobar Argueta
Ing. Jose Manuel Ruiz Juarez
Ing. Edwin Estuardo Zapeta Gómez

Tutores de curso:

Oscar René Jordán Orellana, Juan Pablo Osuna de León
Astrid Edith Hernández González, Henry Adolfo Gálvez
César Dionicio Sazo Mayen, Oscar Alejandro Rodríguez Calderon
Edwar Everaldo Zacarias, Javier Estuardo Lima Abrego
Monica Raquel Calderon Muñoz, Marco Antonio López Grajeda

PROYECTO 2

OBJETIVO GENERAL

Desarrollar una solución integral que implemente **tipos de datos abstractos (TDA)** en base a archivos de entrada en formato **XML** bajo el concepto de programación orientada a objetos (**POO**).

OBJETIVOS ESPECÍFICOS

- **Implementar POO para** el desarrollo de la solución a través de lenguaje Python.
- Utilizar **estructuras de programación secuenciales, cíclicas y condicionales.**
- Utilizar **listas ordenadas y listas ortogonales** implementadas de forma original y **utilizando memoria dinámica.**
- Visualizar **TDA's por medio de la herramienta Graphviz.**
- Utilizar archivos **XML** como insumos para la lógica y comportamiento de la solución.
- Utilizar código **HTML** para **mostrar los resultados de la información obtenida** y trabajada.

Descripción General

La aplicación consiste en una forma de **representar imágenes** utilizando listas ortogonales y permitir realizar **operaciones sobre estas imágenes**.

En la figura 1 se puede observar un ejemplo de una imagen almacenada en una matriz de caracteres.

Nombre de la imagen

A	1	2	3	4	5	6	7	8	9	10
1										
2		*	*	*		*	*	*		
3			*			*	*	*		
4			*			*				
5		*	*	*		*				
6										
7		*	*	*		*	*	*		
8		*						*		
9		*	*	*		*	*	*		
10										

Figura 1 – Imagen contenida en una matriz ortogonal de 10 filas por 10 columnas

La figura 1 muestra una matriz denominada “A”, esta matriz es de 10 filas por 10 columnas y representa una imagen formada por el carácter “*” en los nodos que contienen información.

Se requiere poder gestionar “N” imágenes en una lista simple ordenada. Estas imágenes tendrán un nombre y una dimensión. La dimensión de la imagen se determina por la cantidad de filas “f” y la cantidad de columnas “c” de esta imagen.

Operaciones sobre una imagen

Estas operaciones se aplican sobre una imagen contenida en la lista ordenada de imágenes, modificando la imagen sobre la cual se aplican.

Las operaciones que se podrán realizar sobre una imagen son las siguientes:

1. Rotación horizontal de una imagen
2. Rotación vertical de una imagen
3. Transpuesta de una imagen
4. Limpiar zona de una imagen
5. Agregar línea horizontal a una imagen
6. Agregar línea vertical a una imagen
7. Agregar rectángulo
8. Agregar triángulo rectángulo

Ejemplos:

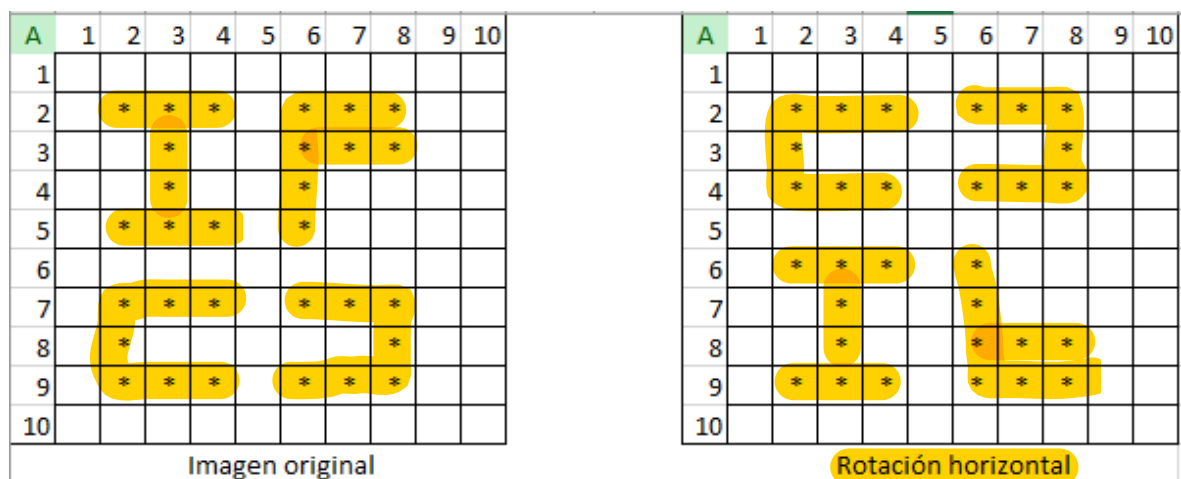


Figura 2 – Ejemplo de operación rotación horizontal

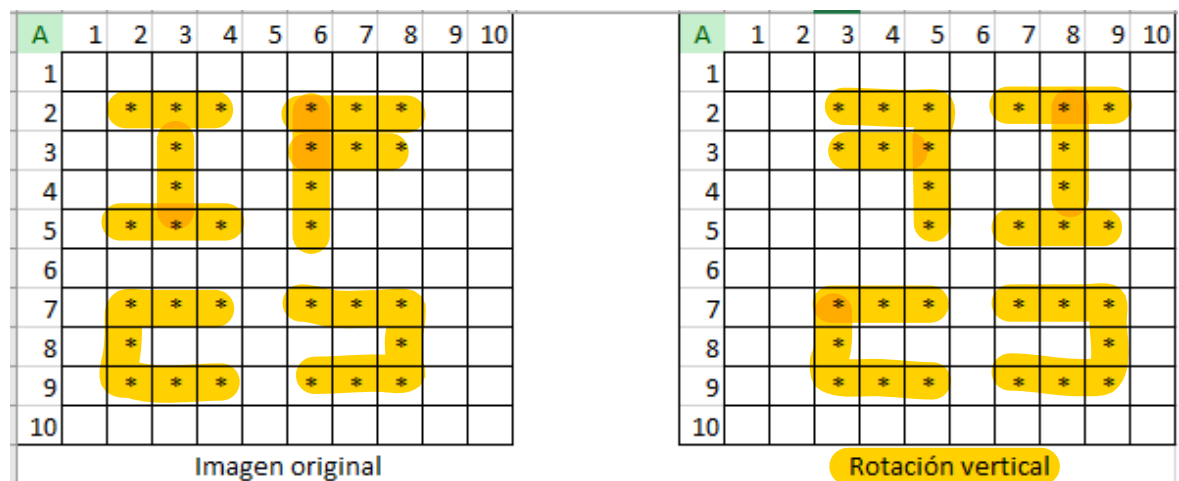


Figura 3 – Ejemplo de operación rotación vertical

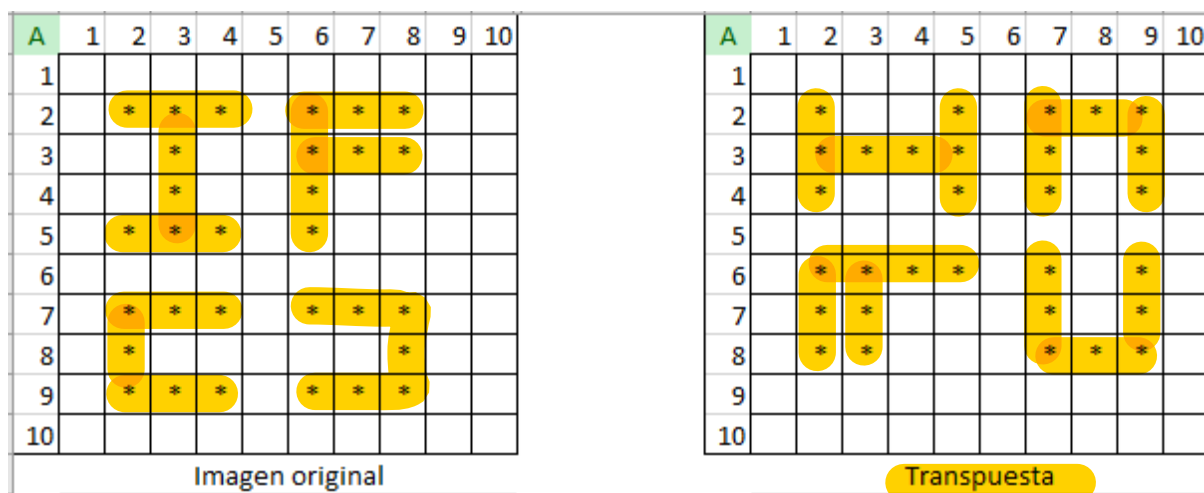


Figura 4 – Ejemplo de operación transpuesta

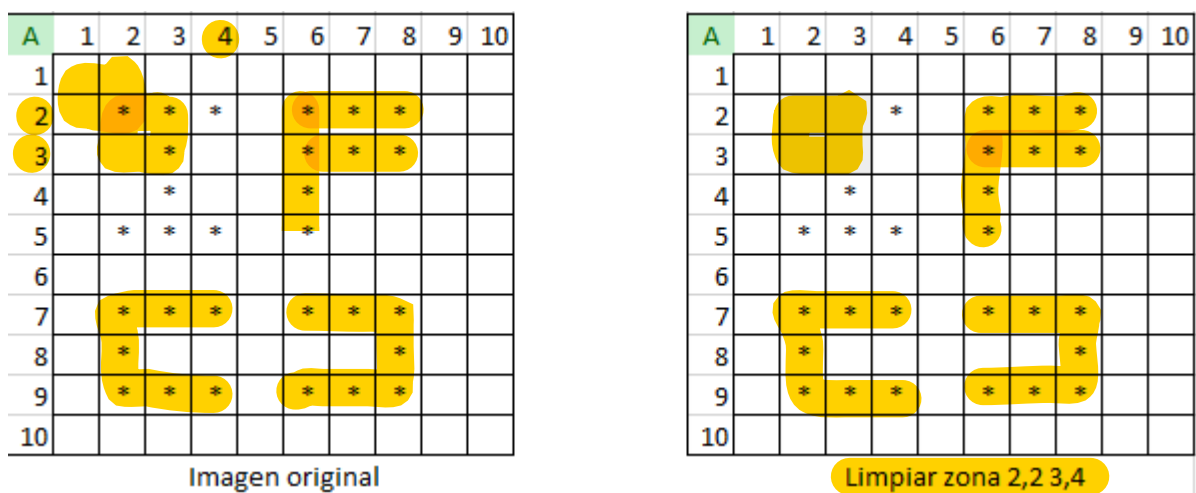


Figura 5 – Ejemplo de operación limpiar área desde la fila 2, columna 2; hasta la fila 3 columna 4

La figura No. 5 muestra la operación limpiar, esta operación necesita recibir dos coordenadas con la forma fila, columna que representarán el área a limpiar.

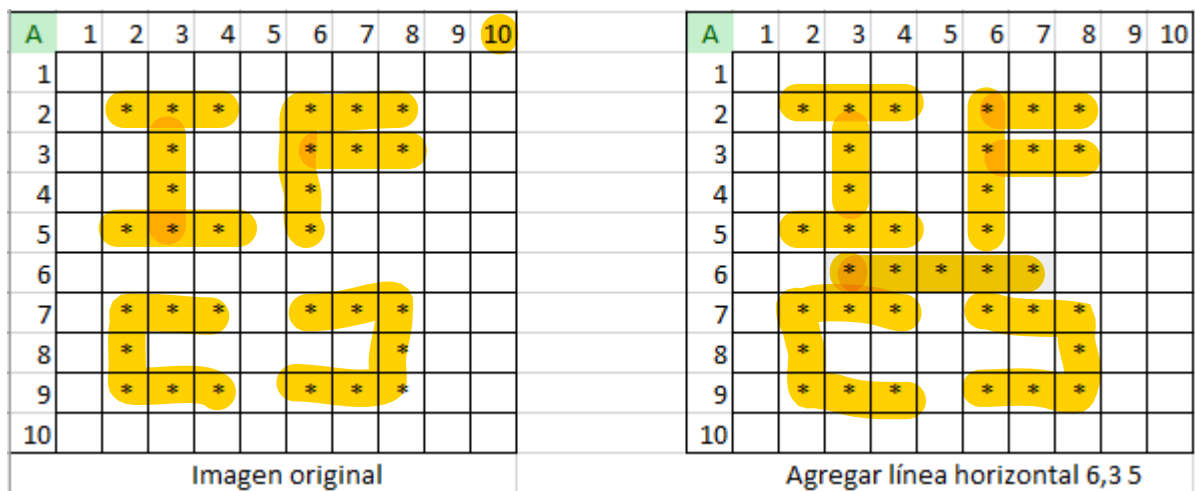


Figura 6 – Ejemplo de operación agregar línea horizontal en la fila 6, columna 3; 5 elementos

La figura No. 6 muestra la operación agregar línea horizontal, esta operación necesita recibir la coordenada inicial y la cantidad de elementos que conformarán la línea horizontal.

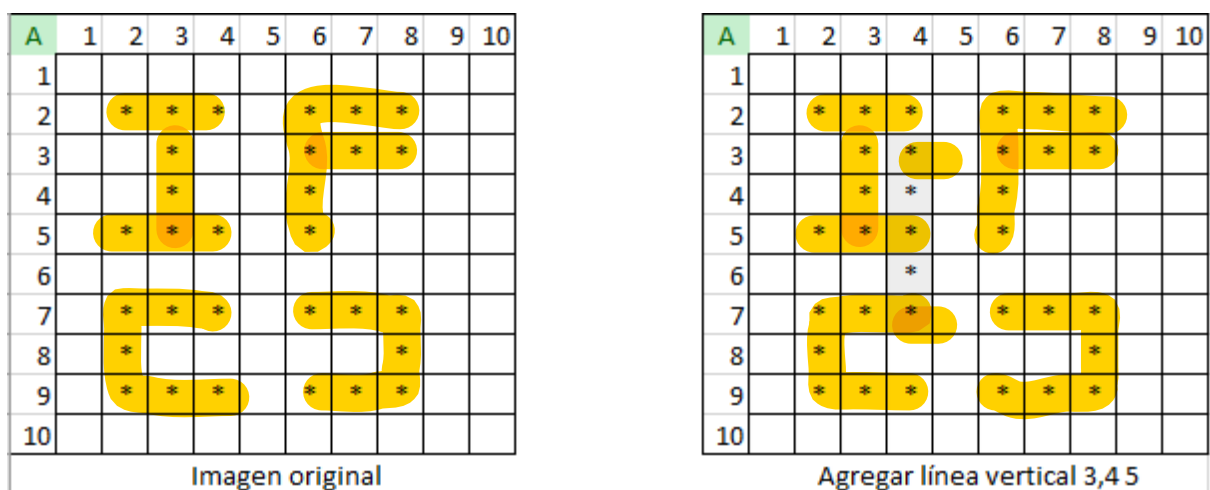


Figura 7 – Ejemplo de operación agregar línea vertical en la fila 3, columna 4; 5 elementos

La figura No. 7 muestra la operación agregar línea vertical, esta operación necesita recibir la coordenada inicial y la cantidad de elementos que conformarán la línea vertical.

A	1	2	3	4	5	6	7	8	9	10
1										
2		*	*	*		*	*	*		
3			*			*	*	*		
4			*			*				
5		*	*	*		*				
6										
7		*	*	*		*	*	*		
8		*						*		
9		*	*	*		*	*	*		
10										

Imagen original

A	1	2	3	4	5	6	7	8	9	10
1										
2		*	*	*		*	*	*		
3		*	*	*		*	*	*		
4		*	*	*		*	*	*		
5		*	*	*		*				
6										
7		*	*	*		*	*	*		
8		*						*		
9		*	*	*		*	*	*		
10										

Agregar rectángulo 2,2 4x3

Figura 8 – Ejemplo de operación **agregar rectángulo** en la fila 2, columna 2 con 4 filas de alto y 3 columnas de ancho

La figura No. 8 muestra la operación agregar rectángulo, esta operación necesita recibir la **coordenada inicial**, la cantidad de **filas** y la cantidad de **columnas** que formarán el rectángulo.

A	1	2	3	4	5	6	7	8	9	10
1										
2		*	*	*		*	*	*		
3			*			*	*	*		
4			*			*				
5		*	*	*		*				
6										
7		*	*	*		*	*	*		
8		*						*		
9		*	*	*		*	*	*		
10										

Imagen original

A	1	2	3	4	5	6	7	8	9	10
1										
2		*	*	*		*	*	*		
3		*	*	*		*	*	*		
4		*	*	*		*				
5		*	*	*	*	*				
6										
7		*	*	*		*	*	*		
8		*						*		
9		*	*	*		*	*	*		
10										

Agregar triángulo rectángulo 2,2 4

Figura 9 – Ejemplo de operación **agregar triángulo rectángulo** en la fila 2, columna 2 con una longitud de 4 filas x 4 columnas

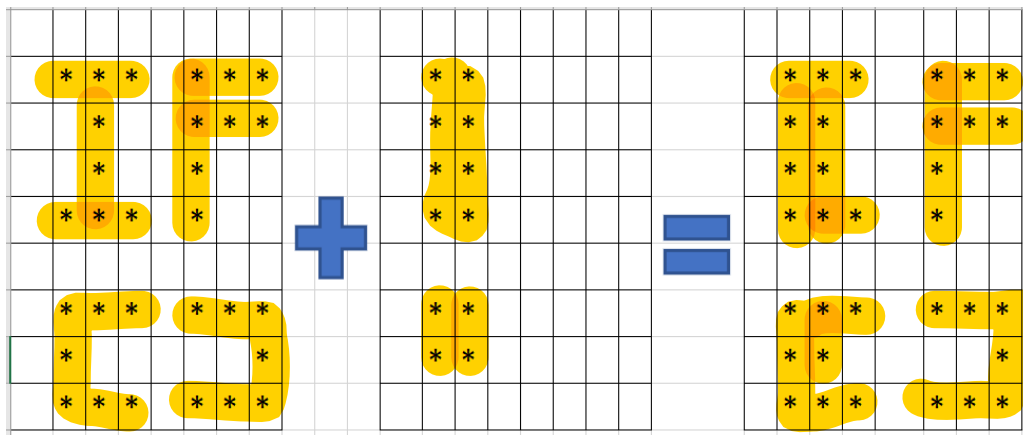
La figura No. 9 muestra la operación agregar triángulo rectángulo, esta operación necesita recibir la **coordenada inicial** y la **cantidad** que representa el número de filas y columnas que tendrá el rectángulo.

Operaciones sobre dos imágenes

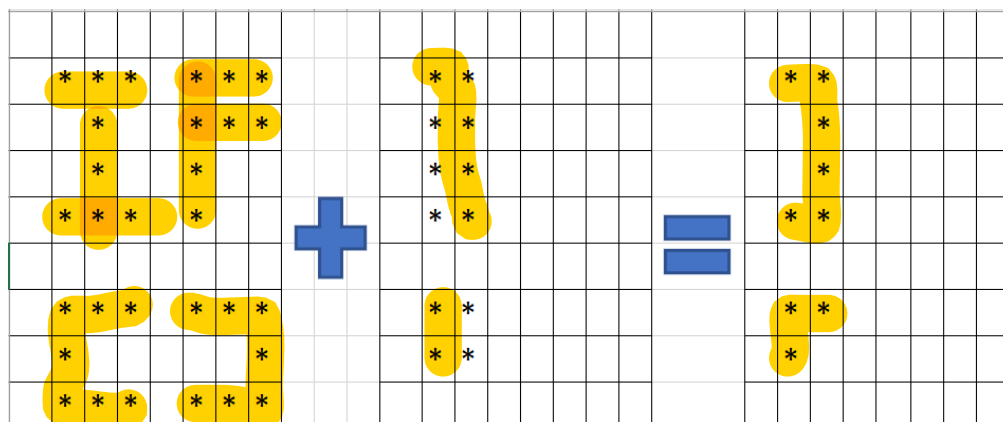
Estas operaciones se aplican sobre dos imágenes y producen como resultado una 3ª. Imagen, que podría ser una imagen existente previamente, o bien, una nueva imagen.

Las operaciones que se podrán realizar involucrando 2 imágenes son las siguientes.

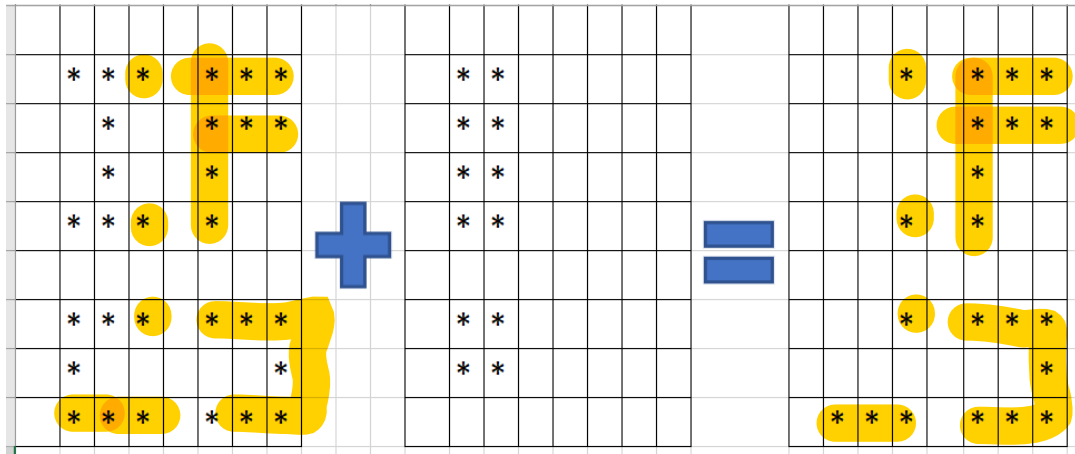
1. **Unión A, B:** Esta operación representa la unión de 2 imágenes (A, B), es decir, genera una nueva imagen que contendrá todos los elementos de las 2 imágenes ingresadas a la operación. El resultado podrá generar una nueva imagen, o bien, sustituir una imagen existente.



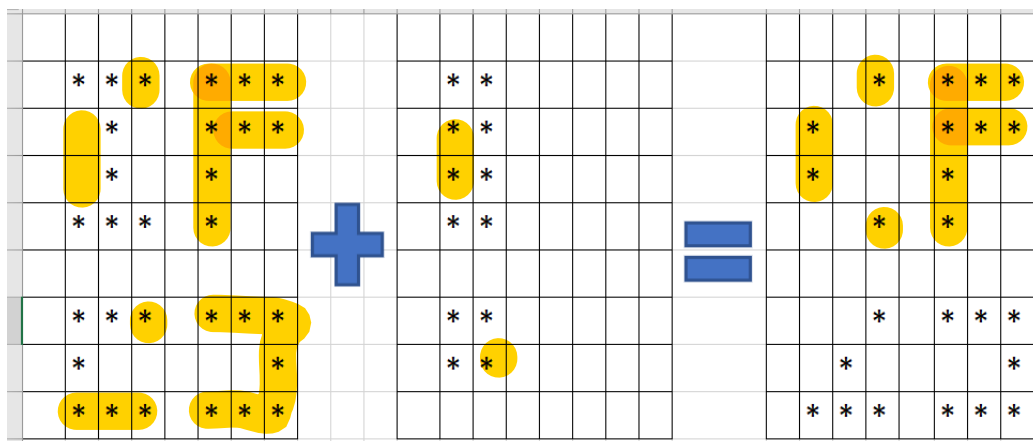
2. **Intersección A,B:** Esta operación representa la intersección de 2 imágenes (A,B), es decir, genera una nueva imagen que contendrá todos los elementos comunes en ambas imágenes. El resultado podrá generar una nueva imagen, o bien, sustituir una imagen existente.



3. **Diferencia A,B:** Esta operación representa la diferencia entre 2 imágenes (A,B), es decir, genera como resultado una imagen con todos los elementos de la imagen A que no pertenezcan a la imagen B.



4. **Diferencia simétrica A,B:** Esta operación representa la diferencia simétrica entre 2 imágenes (A,B), es decir, genera como resultado una imagen con todos los elementos de la imagen A que no pertenezcan a la imagen B y todos los elementos de la imagen B que no pertenezcan a la imagen A.



Finalmente, es necesario que la solución provea una salida que muestre el estado actual de cualquier imagen, así como la cantidad de elementos internos que maneja, de acuerdo con la lógica de manejo óptimo de nodos en una lista ortogonal.

Interfaz Gráfica

La solución del proyecto será a través de una aplicación con interfaz gráfica. A continuación, se describe los componentes de la aplicación.

Ventana Principal

La ventana principal por la cual se desarrollará todo el flujo del proyecto, la imagen No. 14 es una representación visual recomendada. Debe contener mínimo botones.

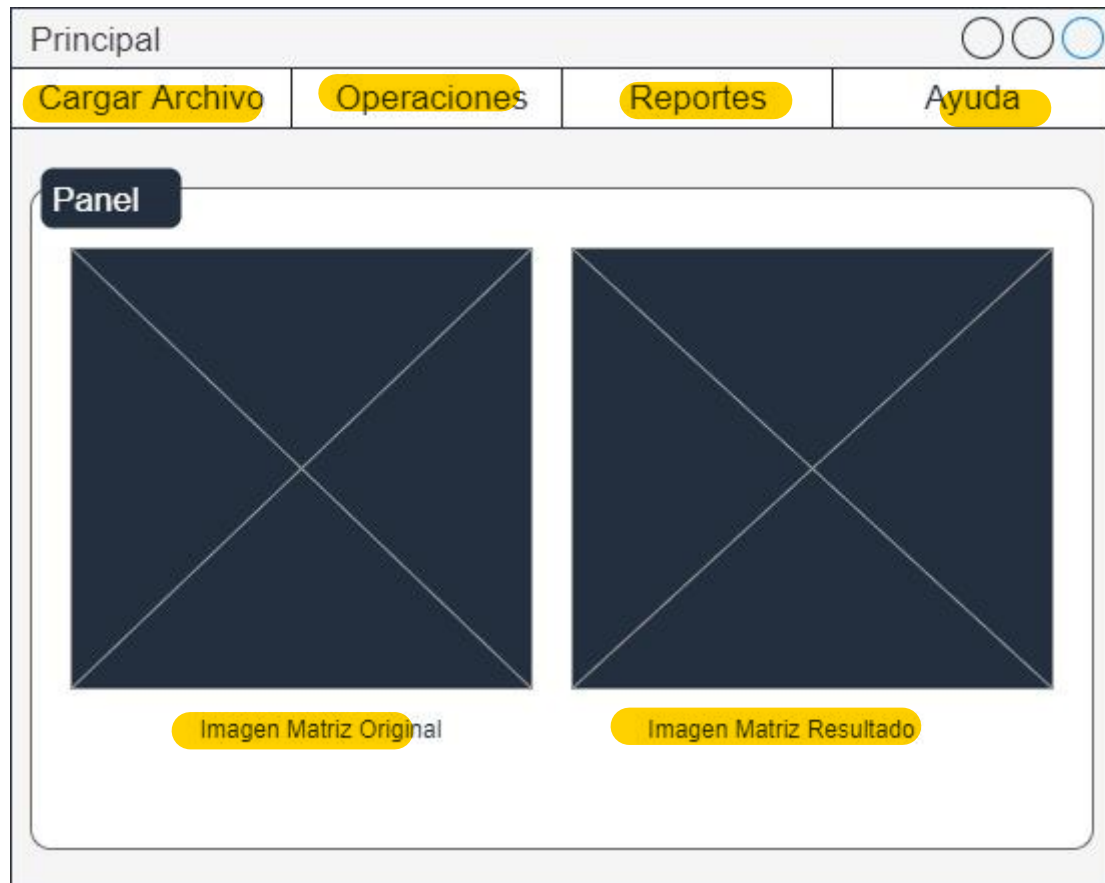


Figura No. 14 - Ventana Principal

Componentes:

- **Barra de menú:** un menú con el que se pueden realizar las siguientes funciones:
 - **Cargar Archivo:** Se desplegará una pantalla para gestionar la carga de los archivos de entrada con extensión .xml con una o más matrices. Se especifica más adelante la estructura de estos archivos.
 - **Operaciones:** Se desplegará una pantalla para gestionar las distintas operaciones descritas en la descripción general, permitiendo ingresar la información necesaria para realizar la operación. Al ejecutar la operación seleccionada, deberá desplegar la pantalla principal y presentar las imágenes originales involucradas en la operación y su resultado, identificando claramente los nombres de las imágenes.

- **Reportes:** se desplegará el reporte con extensión html para visualizar una bitácora con todas las operaciones gestionadas por el programa. Más adelante se describe el formato de este reporte.
- **Ayuda:** desplegará 2 opciones, una para visualizar información del estudiante y otra para visualizar la documentación del programa.
- **Panel:** en este panel se visualizará el resultado de la última operación identificando claramente la imagen o imágenes originales involucradas y la imagen resultante.

Archivos de Entrada

Los archivos de entrada, contendrán la definición de las imágenes iniciales/originales.

Los archivos de entrada consistirán en archivos con extensión y estructura xml en el cual se limitará a utilizar únicamente las etiquetas:

- ❖ **matrices:** este será necesario para la lectura inicial del archivo, ya que será la etiqueta padre de todo.
- ❖ **matriz:** esta etiqueta será la que indica que una nueva matriz será creada para su respectivo análisis y únicamente puede estar dentro de la etiqueta **matrices** y puede tener los siguientes componentes hijos:
 - **nombre:** este contendrá el identificador de la matriz leída (se deberá validar la existencia de matrices con el mismo nombre, para mantener la consistencia de los datos).
 - **fila:** será el número de filas que tendrá la matriz, si la imagen cargada no coincide con este valor se considerará un error.
 - **columna:** será el número de columnas que tendrá la matriz, si la imagen cargada no coincide con este valor se considerará un error.
 - **imagen:** esta etiqueta únicamente podrá estar dentro de la etiqueta **matriz** y contendrá los valores respectivos a cada celda de la matriz, consiste en usar cadenas de caracteres; donde el carácter "*" representa una celda con datos y el cambio de fila en la matriz se identificará mediante el carácter salto de línea (\n). Además, se utilizará el carácter "-" para representar los espacios en blanco.

Ver ejemplo más adelante.

Reporte en HTML

Se debe crear un reporte en HTML en el cual se mostrarán algunos datos acerca del funcionamiento del programa en forma de logs.

Al procesar un archivo de entrada de matrices se debe crear una entrada en el archivo HTML en el cual se especifique el nombre de la matriz, número de espacios vacíos y llenos junto con la hora y fecha en que se procesa.

Ejemplo:

'09/03/2021 - 12:00:00 - NombreMatriz - Espacios llenos: # - Espacios vacíos: #'.

En el caso de las operaciones que se realicen sobre matrices, deberá mostrar la hora y fecha en que se procesa, el tipo de operación y que matrices están involucradas.

Ejemplo:

'09/03/2021 - 12:00:00 - TipoOperación - Matriz (o matrices) usadas: Nombre (o nombre1, nombre2)'.

En caso de que una operación genere un error, éste deberá quedar descrito en este reporte indicando fecha y hora, y una descripción del error.

Ejemplo:

'09/03/2021 - 12:00:00 - Error: Descripción. Identificar operación y matrices involucradas.'

El orden en que vengan los datos quedará a criterio del estudiante pero se debe cuidar que vengan todos los datos anteriormente descritos. De igual manera la presentación de los datos en el HTML queda a criterio del estudiante.

CONSIDERACIONES .

Debe utilizarse versionamiento para el desarrollo del proyecto. Se utilizará la plataforma **GitHub** en la cual se debe crear un repositorio en el que se gestionará el proyecto. Se deben realizar 4 releases o versiones del proyecto (se recomienda realizar una por semana del tiempo disponible). Se deberá agregar a su respectivo auxiliar como colaborador del repositorio. El último release será el release final y se deberá de realizar antes de entregar el proyecto en la fecha estipulada.

Para la realización de la interfaz gráfica queda a discreción del estudiante que librería utilizar.

DOCUMENTACIÓN

Para que el proyecto sea calificado, el estudiante deberá entregar la documentación utilizando el formato de ensayo definido para el curso. En el caso del proyecto, el ensayo puede tener un mínimo de 4 y un máximo de 7 páginas de contenido, este máximo no incluye los apéndices o anexos donde se pueden mostrar modelos y diseños utilizados para construir la solución. Este informe debe expresar con claridad el diseño de objetos ideado para resolver este proyecto por lo que debe expresar el diagrama de clases y los diagramas de actividades de los algoritmos más importantes.

RESTRICCIONES

- Solo se permitirá la utilización de los IDEs discutidos en el laboratorio.
- Se deberá utilizar una lista ortogonal para el manejo de las matrices. Esta lista ortogonal debe ser creada completamente por el estudiante mediante clases.
- Uso obligatorio de programación orientada a objetos (POO).
- El nombre del repositorio debe de ser **IPC2_Proyecto2_#Carnet**.

- El estudiante debe entregar la documentación solicitada para poder optar a la calificación.
- Los archivos de entrada no podrán modificarse.
- Se calificará la versión del cuarto release. Los cambios realizados después de ese release no se tomarán en cuenta.
- Para dudas concernientes al proyecto se utilizarán los foros en UEDI de manera que todos los estudiantes puedan ver las preguntas y las posteriores respuestas.
- **COPIAS TOTALES O PARCIALES SERÁN REPORTADOS A LA ESCUELA Y OBTENDRÁN NOTA DE 0 PUNTOS.**
- **NO HABRÁ PRÓRROGA.**

ENTREGA

- La entrega será el **5 de abril** a más tardar a las 11:59 pm.
- La entrega será por medio de la UEDI.
- **La documentación debe estar subida en el repositorio en una carpeta separada.**
- Para entregar el proyecto en UEDI se deberá subir un archivo de texto con el link del repositorio.

ANEXOS:

Archivo de entrada:

Ejemplo de Entrada para definir la matrices:

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <matrices>
3    <matriz>
4      <nombre>Matriz_1</nombre>
5      <filas>10</filas>
6      <columnas>10</columnas>
7      <imagen>
8        -----
9        .***.***.
10       .-*-.***.
11       .-*-.*-.
12       .***.*-.
13       -----
14       .***.***.
15       .*-.*-.
16       .***.***.
17       -----
18     </imagen>
19   </matriz>
20   <nombre>Matriz_2</nombre>
21   <filas>8</filas>
22   <columnas>8</columnas>
23   <imagen>
24     -----
25     .*_*_*_*
26     .-*-*_*
27     .*_*_*_*
28     -----
29     .***_*_*
30     .-*_*_*
31     .***_*_*
32   </imagen>
33 </matriz>
34 </matrices>
```