
Proyecto 2 IPC2

201801155 – Bryan Eduardo Caal Racanac

Resumen

Utilizando el lenguaje de programación python como base para el desarrollo de este proyecto, implementado estructuras básicas de la programación como programación secuencial, cíclicas y condicionales, y la técnica de programación denominada como programación orientada a objetos (POO), se desarrolló una solución integral que implementa tipos de datos abstractos y visualización de datos, bajo el concepto de programación orientada a objetos, por lo que se procedió a fusionar estos conocimientos en un solo proyecto, se realizó un programa con interfaz gráfica con la ayuda de la librería de python Tkinter para poder ser capaz de utilizar archivos con extensión XML como insumos para el desarrollo del proyecto efectuando este la lectura de estos archivos con una estructura diferente a la de los archivos comunes y Guardando los ratos de esta misma en una matriz ortogonal para realizar distintas acciones con estos datos como ordenar y comparar datos, generar archivos de salida con diferentes extensiones así como también poder visualizar TDA's por medio de la herramienta Graphviz

Palabras clave

POO, XML, Python, Graphviz, TDA

Abstract

Using the python programming language as the basis for the development of this project, implementing basic programming structures such as sequential, cyclic and conditional programming, and the programming technique called object-oriented programming (OOP), a comprehensive solution was developed that implements abstract data types and data visualization, under the concept of object-oriented programming, so that this knowledge was merged into a single project, a program with graphical interface was made with the help of the Tkinter python library to be able to use files with XML extension as inputs for the development of the project making this the reading of these files with a different structure than the common files and Saving the time of it in an orthogonal matrix to perform different actions with these data such as sorting and comparing data, generating output files with different extensions as well as being able to view ADD's using the Graphviz tool

Keywords

POO, XML, Python, Graphviz, TDA

Introducción

Actualmente la demanda de información es mucha y con el pasar de los años los formatos con los cuales la información puede ser proporcionada es igual de variada, debido a que no todos los formatos sirven para los mismo, como puede ser un documento informativo puede ser solo texto pero si necesitamos información dividida y agrupada por separado, el formato de texto común y corriente se vuelve muy complicado y tedioso de leer, con esta información tomada en cuenta se creó un programa el cual puede leer archivos de texto con la extensión .XML la cual consta de etiquetas padre que contienen etiquetas hijo que a su vez contienen información, con esta información podemos hacer diferentes procedimientos como agrupar la información parecida, o en este caso mostrar la información de manera gráfica para que el usuario tenga una mayor facilidad para poder interpretar la información que se le está presentando que si solo lee el archivo con extensión XML.

Desarrollo del tema

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 – Matriz Original

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:

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										

Rotación horizontal

Figura 2 - Rotación Horizontal

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										

Rotación vertical

Figura 3 – Rotación 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										

Transpuesta

Figura 4 – Transpuesta

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										

Limpiar zona 2,2 3,4

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.

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 línea horizontal 6,3 5

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.

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 línea vertical 3,4 5

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.

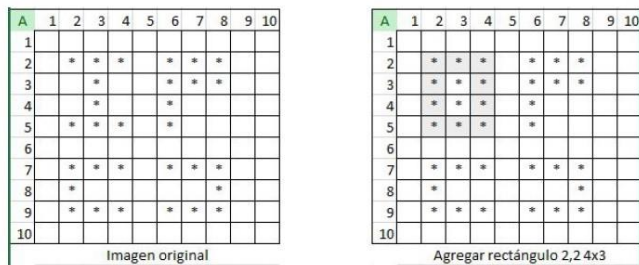


Figura 8 – Ejemplo de operación agregar rectángulo en la fila 2, columna 2 con 4 filas de alto y 3 columnas de ancho

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.

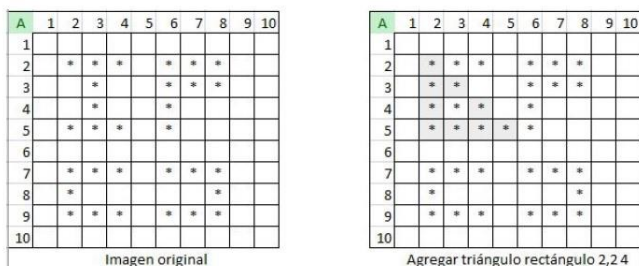


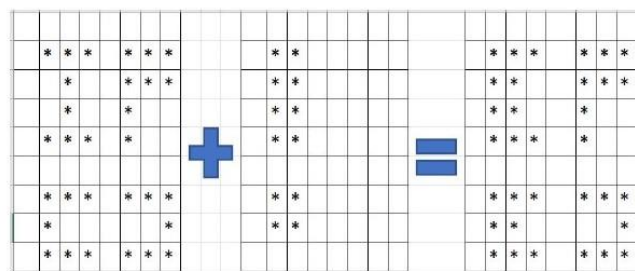
Figura 9 – Ejemplo de operación agregar triángulo rectángulo en la fila 2, columna 2 con una longitud de 4 filas x 4 columnas

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.

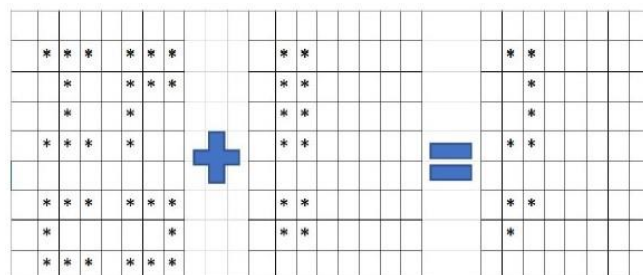
Operaciones sobre dos imágenes

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.

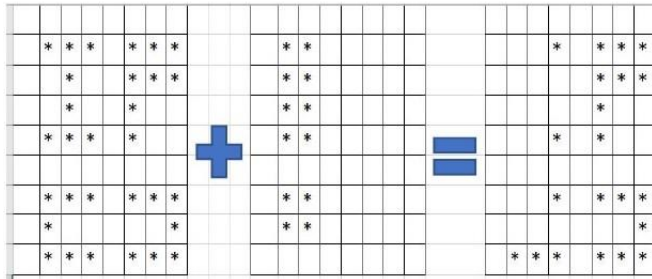
Unión A, B: Esta operación representa la unión de 2 imágenes (A, B), es decir, genera una nueva imagen que contendrá todos los elementos de las 2 imágenes ingresadas a la operación. El resultado podrá generar una nueva imagen, o bien, sustituir una imagen existente.



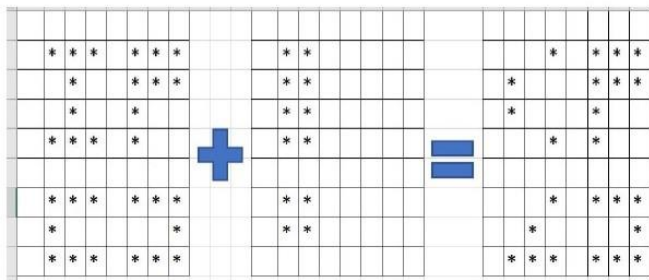
Intersección A,B: Esta operación representa la intersección de 2 imágenes (A,B), es decir, genera una nueva imagen que contendrá todos los elementos comunes en ambas imágenes. El resultado podrá generar una nueva imagen, o bien, sustituir una imagen existente.



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.



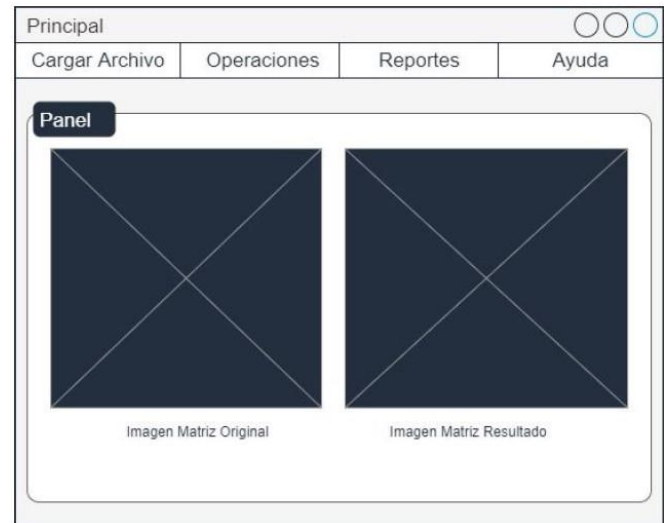
Diferencia simétrica A,B: Esta operación representa la diferencia simétrica entre 2 imágenes (A,B), es decir, genera como resultado una imagen con todos los elementos de la imagen A que no pertenezcan a la imagen B y todos los elementos de la imagen B que no pertenezcan a la imagen A.



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.

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.



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.

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.

Conclusiones

Para poder ordenar grandes cantidades de datos en diferentes formatos se aconseja manejar lo que son las estructuras de datos mucho más avanzadas, implementación de TDA's y tener en consideración todas las excepciones que se puedan ejecutar al momento de leer archivos con grandes cantidades de datos, como podemos ver la forma tradicional de guardar los datos en listas requiere de mucha memoria y no siempre es lo más óptimo debido a que cuando se tienen demasiados datos el recorrer toda la lista de principio a fin podría tardar mucho, el uso de listas enlazadas facilita un poco esta lectura de archivos y un almacenamiento mucho más ordenado, bajando así el rendimiento y los requerimiento que se le exigen a la maquina al momento de correr un programa.

Referencias bibliográficas

Antonakos, James L. and Mansfield, Kenneth C., Jr. *Practical Data Structures Using C/C++* (1999). Prentice-Hall. ISBN 0-13-280843-9, pp. 165–190

Collins, William J. *Data Structures and the Java Collections Framework* (2002,2005) New York, NY: McGraw Hill. ISBN 0-07-282379-8, pp. 239–303

Cormen, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L.; Stein, Clifford *Introductions to Algorithms* (2003). MIT Press. ISBN 0-262-03293-7, pp. 205–213, 501–505