

---

## Proyecto 2

---

**201532643 – Carlos Jezeh Gedeoni Toscano Palacios**

### Resumen

Se realizó un programa para modificar imágenes, el programa tiene una interfaz gráfica y se realizó en el lenguaje de programación Python, este se realizó con la finalidad de utilizar programación orientada a objetos.

La OOP es muy importante en este proyecto ya que gracias a este tipo de paradigma el proyecto esta mucho mas organizado, se puede entender más fácil y se puede implementar código con un mejor control, así como también se pueden realizar operaciones que sin el paradigma no se podrían hacer, eh aquí la importancia de este paradigma para la programación ya que muchas veces facilita la implementación de código, así como que es más organizado

También se implementó código para el manejo de archivos como XML, HTML e imágenes, la importancia de saber como implementar estos archivos a un programa es importante ya que la mayoría de los programas actualmente utiliza imágenes y tiene enlaces a páginas web.

### Palabras Clave

Paradigma

Imágenes

OOP

### Resume

*A program was made to modify images, the program has a graphical interface and was made in the Python programming language, this was done in order to use object-oriented programming.*

*The OOP is very important in this project because thanks to this type of paradigm the project is much more organized, you can understand easier and you can implement code with better control, as well as you can perform operations that without the paradigm could not be done, eh here the importance of this paradigm for programming since it often facilitates the implementation of code, as well as being more organized*

*Also implemented code for handling files such as XML, HTML and images, the importance of knowing how to implement these files to a program is important since most programs currently use images and have links to web pages. implementing these files to a program is important since most programs currently use images and have links to web pages*

### Keywords

Paradigm

Images

OOP

## Introducción

En el siguiente ensayo se explicará cómo funciona el proyecto realizado tanto como para una persona que no sabe de programación como para un desarrollador

Se tendrá en cuenta explicar cómo funciona el programa sus funcionalidades, así como poder acaparar todo lo que tiene el programa

Se mostrará como esta estructura el programa en base a su diagrama de clases y todas las funciones y métodos que contiene para su funcionamiento, así como también las librerías que se usaron

## Desarrollo del tema

### Lógica del Programa

Se importo la librería tkinter para poder hacer uso de todas sus extensiones para crear la interfaz gráfica dicha interfaz tiene una pantalla principal que cuenta con cuatro botones y un panel.

Esta ventana esta dentro de una clase ya que de esta forma es más fácil implementar y tener el control de sus componentes, los componentes se crean y se agregan a la ventana y se le indican sus propiedades y la posición donde uno los quiere

El primer botón es el de “Cargar Archivo” al presionar este se abre una pequeña ventana que contiene dos botones uno para cargar una imagen y el otro para cargar dos imágenes.

Esto se pudo hacer gracias a la propiedad `TopLevel` de tkinter ya que nos deja mostrar una nueva ventana después de la venta principal todas las demás ventanas que contiene el programa se muestran de la misma forma con `TopLevel`

Al presionar el botón “Cargar una imagen” este nos muestra una nueva ventana que puede navegar en los archivos de nuestra computadora, esto se logra

gracias importar la librería `filedialog` de tkinter, se crea una variable que contendrá la ruta del archivo que queremos utilizar y con esa ruta podemos leer el archivo deseado en este caso es un archivo XML, que contiene un la imagen esta imagen está compuesta de guiones “-” que representan los espacios y asteriscos que formaran la imagen que mostraremos que vamos a mostrar más adelante.

Con la librería `elementTree` de XML se puede leer el archivo que se selecciono se puede navegar en el archivo con for anidados y se puede devolver el texto que estos contengan con `.text`

Cuando se obtuvo la cadena que representa la imagen se procedió a colocarla dentro de una lista ortogonal que representa una matriz esta lista ortogonal es creada con dos listas simplemente enlazadas y para ello se usaron tres clases, para ingresar la cadena se recorrió y se fue colocando en la posición como venían esto gracias a unos contadores que aumentan en cada ciclo de lo for anidados.

Con un método dentro de la clase matriz se puede realizar una cadena que va a convertir los datos de la matriz en una imagen esto con el lenguaje dot de graphviz que también esta importado en la página principal, el método nos devuelve una cadena con el lenguaje dot y se lo pasamos a un objeto de `Source`(de graphviz) que se utilizó para crear la imagen le pasamos la cadena el archivo donde lo queremos escribir en este caso tendrá extensión `.gv` y por ultimo una extensión `.png` por lo tanto nuestra imagen se mostrara de la siguiente forma: `image.gv.png` por ultimo se le coloca la función `render` a el objeto para que se pueda ver en la pantalla ya que sin este función no se actualizara y no se mostrara en la ventana

Para mostrar la imagen en la interfaz se utilizó la librería `PIL` que sirve para mostrar imágenes en tkinter, y se importa `image` e `imageTK`, se crea un objeto con estas librerías y con la función

photoimage se le pasa la ruta donde esta ubicada la imagen, luego se crea un componente de la librería tkinter y se le coloca el objeto que se creo como una imagen

La imagen aparece debajo de los botones que se crearon

Por ultimo se usa la función destroy en la ventana que esta abierta para que esta desaparezca

El botón operaciones nos llevara una nueva ventana donde tendremos varios botones que en los que podremos agregar una fila asteriscos, una columna de asteriscos o borrar el contenido de la imagen entre otras operaciones que se realizaron las cuales están implementadas de una forma similar

Al presionar en cualquier operación nos llevara a una nueva ventana en la cual tendremos que ingresar las coordenadas o las filas y columnas en donde queremos ver el nuevo elemento en la imagen esto se realiza con componente Entry y al colocarle un botón en su comando se puede obtener lo que contiene con .get

Lo que está en los Entry se le pasa como parámetros a métodos que están en la clase matriz ya que como ahí esta guardada los datos de la imagen es más fácil para su manipulación, estos métodos de las operaciones recorren la matriz y van cambiando los valores que tiene la imagen que en este caso son asteriscos y guiones

Se llama al método y luego se manda a llamar al método que nos devuelve la cadena con dot y se realiza el mismo proceso que se realizo para cargar la primera imagen siempre es indispensable colocar el método render en la imagen ya que sin esto no se mostrara ningún cambio en la ventana principal

El botón reporte nos llevara a el navegador que tengamos por defecto y nos mostrara una pagina web local

Esto se realizó importando la librería webbrowser que puede abrir archivos HTML en nuestro navegador

Para crear el archivo HTML se utiliza un método y tres listas que contienen la información de los diferentes reportes

El método contiene una cadena que contiene el lenguaje HTML y se le va concatenando cadenas de las listas, que se obtienen con un método que también crea una cadena con lenguaje HTML solo que no completo solo la parte que nos interesa las tres listas contienen este método y a la hora de obtener la cadena se le concatena a la cadena del método y crea solo una cadena luego se escribe un archivo HTML con open y se abre con un objeto de la librería webbrowser

Las listas mencionadas anteriormente se llenan cada vez que se carga o se realiza un cambio en una imagen estas contienen los errores, las operaciones que se realizan y las matrices cargadas en el programa

El ultimo botón Ayuda nos mostrará una ventana donde tendremos dos botones que uno de información sobre que creador del proyecto y otro donde se podrá acceder a este documento

## Clases y métodos

### Clase window:

Esta clase contiene todos los componentes que se muestran en las diferentes ventanas que tienen el programa sus métodos van relacionados con los botones ya que estos le dan su funcionalidad.

#### \_\_init\_\_()

El método constructor y el que inicializa la ventana con sus diferentes atributos

### **ventanaCargar()**

Muestra una nueva ventana con sus componentes cuando presionamos el botón Cargar

### **Cargar1()**

Se encarga de leer el archivo XML y de cargar la imagen original en la ventana, se activa cuando se presiona el botón “Cargar 1 imagen”

### **Hora()**

Devuelve la hora actual del sistema, sirve para colocar la hora en los reportes

### **Ventana\_lineH(), ventana\_limpiar(), ventana\_lineV, ventana\_rectangulo, ventana\_triangulo()**

Muestran una ventana con varias opciones que tenemos que llenar para que pueda pasarle esos datos como parámetros a la matriz y pueda colocar hacer el cambio deseado en la imagen, también contiene dentro un método que va de la mano del botón aceptar para que se pueda realizar el cambio

### **Reporte()**

Contiene una cadena con lenguaje HTML y también recolecta cadenas de las listas simples que se usan para el reporte y luego abre un archivo HTML en el navegador

### **Ayuda()**

Muestra una nueva ventana con dos botones al presionar el botón información nos mostrará un cuadro de dialogo que contendrá la información, y al presionar documentación nos abrirá este documento

### **Clase lista\_Simple, Lista\_matriz, lista\_error, lista\_ope**

En estas clases se guarda la información que nosotros queramos en forma de lista estas listas contiene dentro de ellas otras clases que son nodos donde se guarda la información todas las listas tienen en común los siguientes métodos

#### **\_\_init\_\_()**

Constructor de la clase inicializa los valores

#### **Insertar()**

Recorre la lista analizando si la lista está vacía y si no lo esta ingresa los datos que se pasan como parámetro hasta el final de la lista

#### **Recorrer()**

Recorre la lista e imprime el contenido de la lista en consola

#### **Cadena\_html()**

Recorre la lista y crea una cadena que se puede pasar para que se use como se desee

#### **Clase Matriz:**

Esta clase contiene es producto de otras dos listas enlazadas y tiene la finalidad de contener los datos de la imagen que se mostrara en la ventana

#### **Insertar()**

Método que inserta los datos en la matriz

#### **imprimirColumna()**

método que imprime en consola los datos que contiene la matriz

`cadena_grap()`

método que devuelve una cadena con el lenguaje dot de graphviz

`línea_vertical()`

crea una línea vertical de asteriscos en la matriz con los parámetros que se nos pasan

`línea_horizontal()`

crea una línea horizontal de asteriscos en la matriz con los parámetros que se nos pasan

`agregar_rectangulo()`

crea un rectángulo de asteriscos en la matriz con los parámetros que se nos pasan

`limpiar_espacio()`

agrega guiones en la matriz en las posiciones que se nos pasan como parámetros

`Agregar_triangulo()`

Crea un triángulo en la matriz en las posiciones que se nos pasan como parámetros

Se logra aprender mas cuando se usan diferentes archivos en un programa ya que es algo nuevo con lo que uno puede ver como se comportan los objetos dentro de una clase

## Referencias bibliográficas

C. J. Date, (1991). *An introduction to Database Systems*. Addison-Wesley Publishing Company, Inc.

## Conclusiones

En este proyecto se utilizo mucho el paradigma de programación orientado a objetos y uno como desarrollador se puede dar cuenta que el paradigma es al principio un poco difícil de entender, pero cuanto mas uno usa este paradigma se da cuenta que uno puede llevar mucho mejor orden en su código, así como es mas entendible

## Anexos

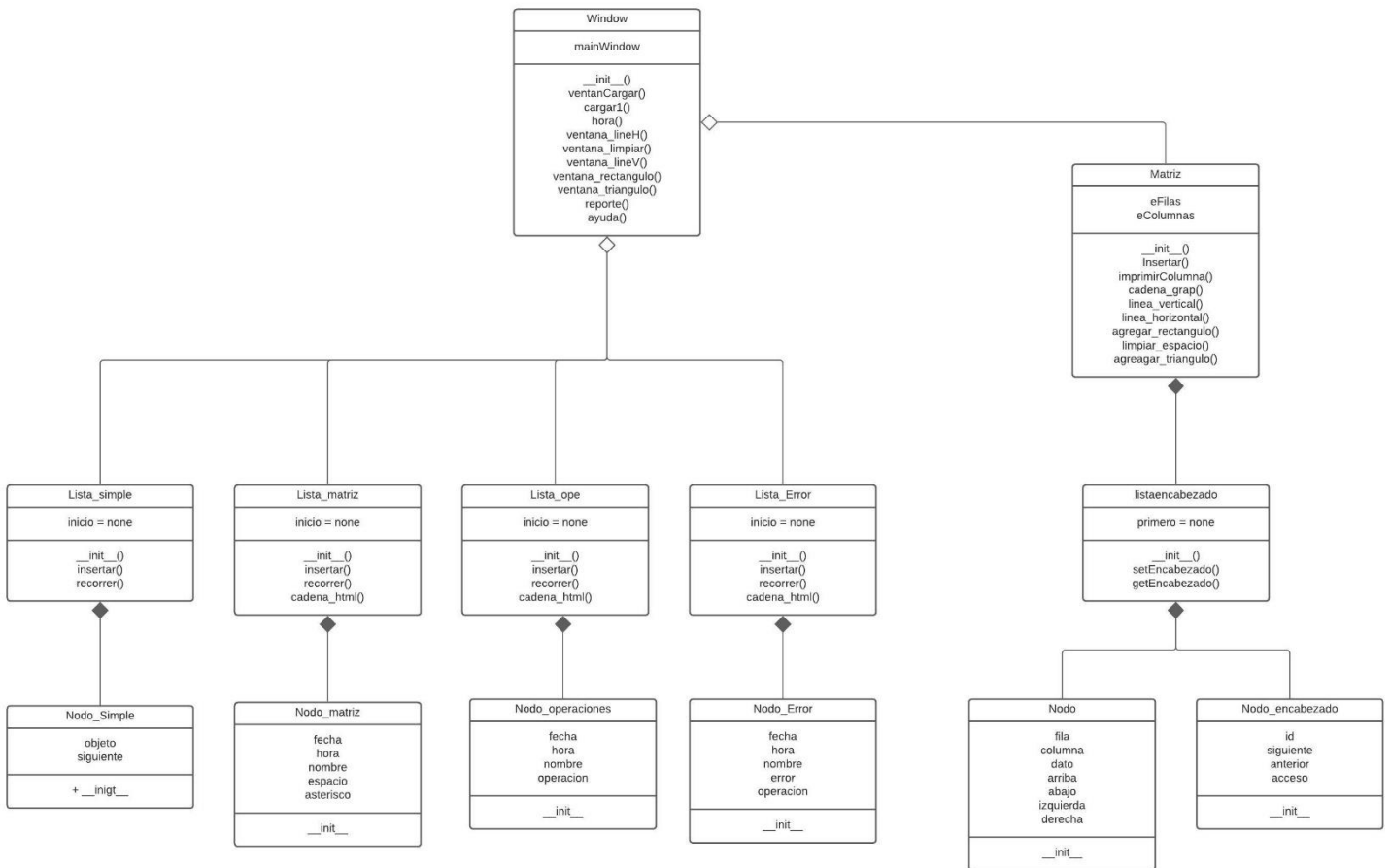


Figure 1 Diagrama de Clases