



**Universidad Autónoma de Zacatecas**

**Unidad Académica de Ingeniería Eléctrica**

**Programa Académico de Ingeniería de Software.**

**Nombre de la Práctica**    Arreglos multidimensionales.

**Numero de Práctica**        11

**Nombre de la carrera**       Ingeniería de Software

**Nombre de la materia**       Lab. Estructuras de Datos

**Nombre del alumno**        Jesús Manuel Juárez Pasillas

**Nombre del docente**        Aldonso Becerra Sánchez

**Fecha:** 11/09/2021

## Práctica 11: Arreglos multidimensionales.

### Introducción:

Los arreglos de dos dimensiones nos permiten representar imágenes, ya que las imágenes están compuestas por píxeles, estos píxeles tienen un color y están en una posición específica de la imagen. Sabiendo esto los píxeles se almacenarán en un arreglo 2D el cual tendrá las mismas dimensiones que la imagen, y con esto poder realizar operaciones sobre los píxeles extraídos y con esto modificar la imagen.

### Desarrollo:

Para esta práctica se hicieron dos clases, Pixel e Imagen. La clase Pixel es la que almacenará los valores de los colores primarios y el valor de alfa, además la clase Imagen almacena la imagen con la que va estar trabajando, una tabla 2D en la que se almacenarán datos de tipo Pixel, y una imagen vacía en la que se va estar almacenando los datos de la tabla y la imagen que va crear.

#### Clase Pixel:

- *Obtiene el valor del color rojo del pixel.*  
`public int getRojo()`
- *Obtiene el valor del color verde del pixel.*  
`public int getVerde()`
- *Obtiene el valor del color azul del pixel.*  
`public int getAzul()`
- *Obtiene el valor del color alfa del pixel.*  
`public int getAlfa()`

#### Clase Imagen:

- *Rellena la matriz con los datos de la imagen.*  
`private void rellenarMatriz()`
- *Introduce los datos de la tabla a la imagen vacía (imagen2).*  
`private void grabarImagen()`
- *Guarda la imagen en el paquete imágenes, con el nombre pasado como parámetro.*  
`public boolean guardarImagen(String nombre)`

- *Le aplica un filtro a la imagen.*  
`public void escalaGris()`
- *Modifica el brillo, valores positivos aumenta el brillo, valores negativos lo disminuye.*  
`public void brillo(int nivel)`
- *Invierte la imagen horizontalmente.*  
`public void invertirH()`
- *Invierte la imagen verticalmente.*  
`public void invertirV()`
- *Gira la imagen 90, 180 o 270 grados, según se indique con el argumento.*  
`public void girar(String giro)`
- *Aplica la traspuesta a la imagen.*  
`public void aplicarTraspuesta()`

**Nota:** Toda la documentación esta agregada en la carpeta “doc” dentro de la carpeta del proyecto (“edylab\_2021\_11/doc”).

### Capturas del programa funcionando:

**Pruebaimagen:** Todas las imágenes resultantes se almacenaron dentro de la carpeta imágenes (Ejecute la prueba para ver estos resultados).

Imagen original:



Escala de grises:



Mas brillo:



Menos brillo:



Invertir horizontalmente:



Invertir verticalmente:



Girar 90 grados:





Gira 180 grados:



Girar 270 grados:



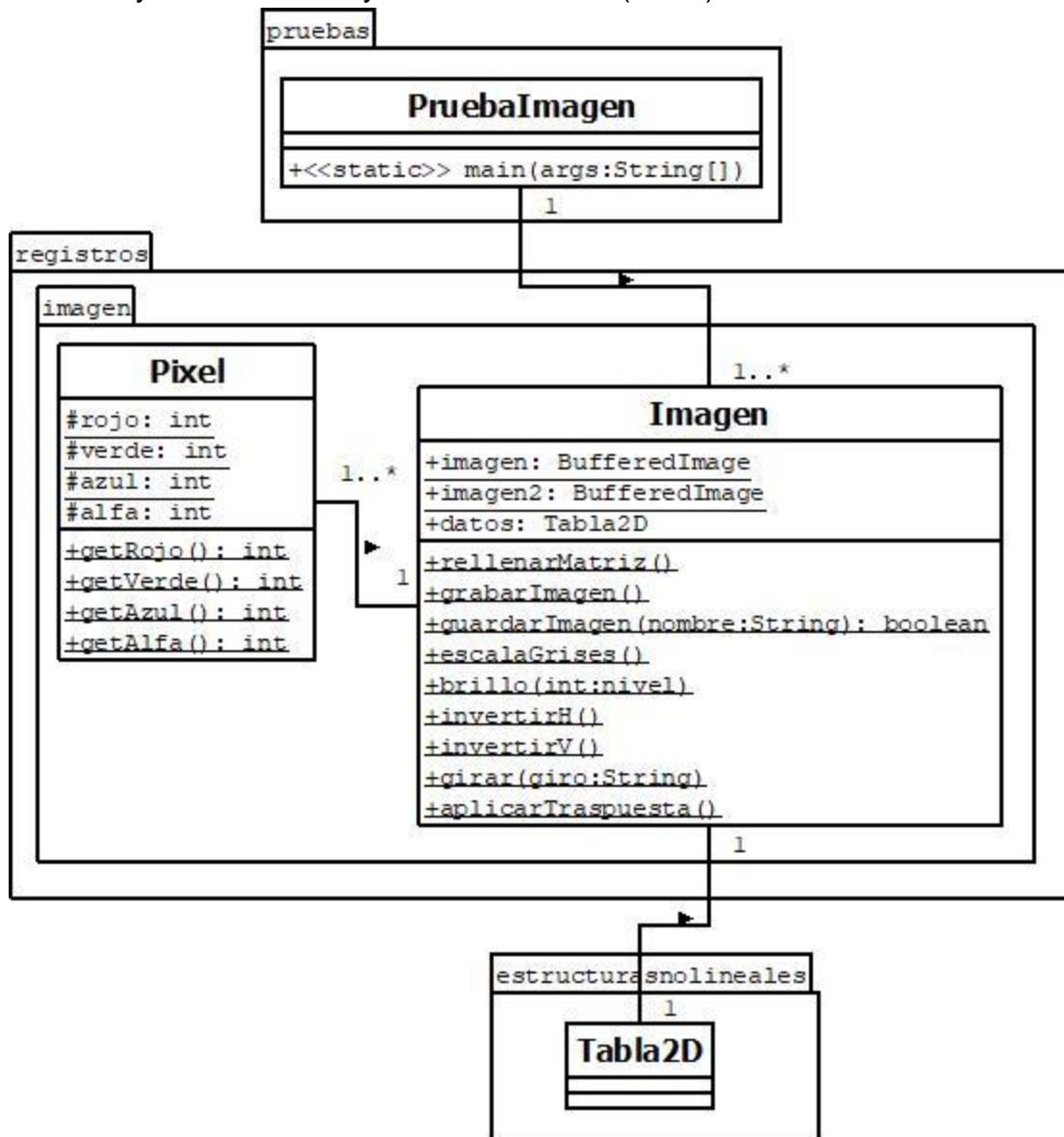
Aplicar traspuesta:



Al aplicar la traspuesta obtenemos el mismo resultado que girar 90 grados, ya que esto hace lo mismo que es cambiar las filas por las columnas.

## Código agregado:

Todas las clases que se ven en el diagrama son nuevas a excepción de Tabla2D, y los métodos y atributos subrayados son nuevos (todos).





## Pre-evaluación:

Pre-Evaluación para prácticas de Laboratorio de Estructuras de Datos	PRE-EVALUACIÓN DEL ALUMNO
CUMPLE CON LA FUNCIONALIDAD SOLICITADA.	No
DISPONE DE CÓDIGO AUTO-DOCUMENTADO.	Sí
DISPONE DE CÓDIGO DOCUMENTADO A NIVEL DE CLASE Y MÉTODO.	Sí
DISPONE DE INDENTACIÓN CORRECTA.	Sí
CUMPLE LA POO.	Sí
DISPONE DE UNA FORMA FÁCIL DE UTILIZAR EL PROGRAMA PARA EL USUARIO.	Sí
DISPONE DE UN REPORTE CON FORMATO IDC.	Sí
LA INFORMACIÓN DEL REPORTE ESTÁ LIBRE DE ERRORES DE ORTOGRAFÍA.	Sí
SE ENTREGÓ EN TIEMPO Y FORMA LA PRÁCTICA.	No
INCLUYE LA DOCUMENTACIÓN GENERADA CON JAVADOC.	Sí
INCLUYE EL CÓDIGO AGREGADO EN FORMATO UML.	Sí
INCLUYE LAS CAPTURAS DE PANTALLA DEL PROGRAMA FUNCIONANDO.	Sí
LA PRÁCTICA ESTÁ TOTALMENTE REALIZADA (ESPECIFIQUE EL PORCENTAJE COMPLETADO).	70%
Observaciones:	

## Conclusión:

Los arreglos multidimensionales nos permiten resolver una gran cantidad de problemáticas por su fácil manejo de los datos y su gran cantidad de espacios donde se pueden almacenar los datos. Esto puede ser muy bueno o malo dependiendo el caso, ya que puede que haya demasiados espacios nulos o puede que todos los espacios sean utilizados. En este caso todos los espacios eran utilizados ya que la tabla se creaba con las mismas dimensiones de la imagen y esto nos permitía modificar cada pixel por separado para al final tener la imagen modificada.