



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)**

Formato para prácticas de laboratorio

CARRERA	PLAN DE ESTUDIO	CLAVE DE UNIDAD DE APRENDIZAJE	NOMBRE DE LA UNIDAD DE APRENDIZAJE
Ingeniero en Computación	2009-2	12099	Programación Orientada a Objetos

PRÁCTICA No.	LABORATORIO DE	Programación Orientada a Objetos	DURACIÓN (HORAS)
6	NOMBRE DE LA PRÁCTICA	Manipulación avanzada de imágenes.	2

1. INTRODUCCIÓN

En esta práctica vamos a seguir trabajando con imágenes, utilizando las clases Picture, Color, FileChooser y Pixel. Las acciones que realizaremos serán las de rotar una imagen, hacer una imagen en espejo y reducir el tamaño de la imagen. Para poder realizar estas acciones debemos contar con el método que realizaron en la practica 5 acerca de tomar los pixeles de una imagen pero como matriz.

2. OBJETIVO (COMPETENCIA)

Utilizar matrices y arreglos de pixeles manipulando las imágenes por medio de ciclos y matrices, con imaginación y perseverancia.

Formuló Cecilia Curlango Rosas Maria Luisa González Ramírez	Revisó Gloria E. Chávez Valenzuela	Aprobó	Autorizó M.C. Maximiliano de las Fuentes Lara
Nombre y Firma del Maestro	Nombre y Firma del Responsable de Programa Educativo	Nombre y Firma del Responsable de Gestión de Calidad	Nombre y Firma del Director de la Facultad

3. FUNDAMENTO

Como vimos en la clase pasada, utilizando un arreglo podemos modificar una imagen. Ahora utilizaremos el método que crearon en la practica 5 para manejar una imagen con una matriz de pixeles.

Para rotar una imagen debe considerar las dimensiones de la imagen original. Por ejemplo, en la matriz de la izquierda se muestran los valores originales, que proceso debemos seguir para que la matriz se modifique y los datos queden como en la matriz de la derecha.

	0	1
0	3	4
1	5	6

	0	1
0	4	6
1	3	5

En el caso anterior es una matriz cuadrada, pero ¿qué pasaría con una matriz de diferente dimensión en renglones y columnas?

En esta práctica retomaremos el método `getPixelMatrix()`. Los ciclos necesarios para realizar la rotación a la izquierda es:

LISTADO 1

```
public static void main(String[] args) {
    Picture pic = new Picture(FileChooser.pickAFile());
    Pixel[][] matrix = pic.getPixelMatrix();
    Picture picRotada = new Picture(pic.getHeight(), pic.getWidth());
    Pixel[][] resultado = picRotada.getPixelMatrix();

    for (int i = 0; i < pic.getWidth(); i++) {
        for (int j = 0; j < pic.getHeight(); j++) {
            resultado[j][pic.getWidth() - 1 - i].setColor(matrix[i][j].getColor());
        }
    }
    picRotada.repaint();
}
```

También podemos modificar el tamaño de las imágenes. Se pueden hacer pequeñas o grandes. El proceso para hacerlas pequeñas es que en lugar de pasar todos los pixeles de una imagen a otra, debemos pasar cada 2 pixeles. Enseguida se muestra una imagen de cómo hacer que la imagen representada por la matriz de la izquierda se reduzca de tal forma que el resultado es la matriz de la derecha.

3	4	5	6
7	8	9	10
11	12	13	14
15	16	17	18

3	5
7	9
11	13
15	17

El código para realizar esta acción esta en el listado 2

LISTADO 2

```
public static void main(String[] args) {
    Picture pic = new Picture(FileChooser.pickAFile());
    Pixel[][] matrix = pic.getPixelMatrix();
    Picture picRotada = new Picture(pic.getWidth(), pic.getHeight());
    Pixel[][] resultado = picRotada.getPixelMatrix();
    int x = 0;

    for (int i = 0; i < pic.getWidth(); i += 2, x++) {
        int y = 0;
        for (int j = 0; j < pic.getHeight(); j += 2, y++) {
            resultado[x][y].setColor(matrix[i][j].getColor());
        }
    }
    picRotada.repaint();
}
```

Otro proceso que podemos realizar con las imágenes el de hacer imágenes en espejo. Debemos copiar el pixel en la posición (0,0) a la ultima posición (ancho-1,0), debemos copiar el pixel del primer renglón segunda columna (1,0) en la posición (ancho-2,0), este proceso se repetirá hasta llegar al punto medio. El punto medio es el punto en que se verá reflejada la imagen. El punto medio es el ancho de la imagen/2. Por ejemplo si el ancho es de 2 el punto medio será $2/2 = 1$. Para el caso de que el ancho es impar el proceso es similar.

	0	1
0	1	2
1	3	4

	0	1
0	1	1
1	3	3

En caso de que el ancho=3 el punto medio es $3/1=1$.

	0	1	2
0	1	2	3
1	4	5	6

	0	1	2
0	1	2	1
1	4	5	4

El código para ver la imagen en espejo de forma vertical se muestra en el listado 3

LISTADO 3

```
public static void main(String []args){  
  
    Picture pic = new Picture(FileChooser.pickAFile());  
    Pixel[][] matrix = PruebaRotar.getPixelMatrix(pic);  
    int puntoMedio=pic.getWidth()/2;  
    for (int i= 0; i < pic.getHeight(); i++) {  
        for (int j= 0; j < puntoMedio; j++) {  
            matrix[pic.getWidth()-1-i][j].setColor(matrix[i][j].getColor());  
        }  
    }  
    pic.repaint();  
}
```

4. PROCEDIMIENTO (DESCRIPCIÓN)

Compruebe que los códigos anteriores funcionan.

Modifique la clase Picture para que contenga los siguientes métodos.

1. Realice un método que rote una imagen 90° a la izquierda.

```
public void rotarIzq();
```

2. Realice un método que rote una imagen 90° a la derecha.

```
public void rotarDer();
```

3. Realice un método que regrese una imagen más pequeña que la imagen original.

```
public Pixel[][] reduce();
```

4. Realice un método que genere otra imagen en espejo horizontal de la original.

```
public void espejoHorizontal();
```

A) EQUIPO NECESARIO	MATERIAL
---------------------	----------

Computadoras con capacidad para ejecutar el entorno de desarrollo Netbeans

7. REFERENCIAS

Netbeans

<http://netbeans.org/downloads/>

Java 6

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>