



**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)
4	NOMBRE DE LA PRÁCTICA	Manipulación de Imágenes	2

### 1. INTRODUCCIÓN

En esta práctica vamos a modificar el color de las imágenes. Para ello veremos que las imágenes son arreglos de píxeles. Cada píxel (*picture element*) es un punto en una imagen. Cada punto está compuesto de tres colores rojo, verde y azul, siguiendo el modelo RGB (siglas en inglés de *red, green, blue*). La intensidad de cada uno de estos colores es lo que forma el color final del punto que perciben nuestros ojos. Esto se puede ver también en las imágenes de la televisión si nos acercamos mucho a la pantalla. En esta práctica modificaremos el color de unas imágenes cambiando el color de algunos de sus píxeles.

### 2. OBJETIVO (COMPETENCIA)

Modificar imágenes utilizando la clase `Pixel`, `Picture` y `FileChooser` y analizar el funcionamiento de la clase `SimplePicture` y su relación con la clase `Picture`.

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

La representación de imágenes con el lenguaje java es utilizando arreglos de pixeles. Cada pixel contiene 3 datos. Estos 3 datos representan el color del pixel. Se utiliza el modelo RGB (red,green,blue).



El modelo RGB se utiliza 3 bytes para representar el color; 1 byte por color. Por ejemplo si queremos representar el negro los valores de los 3 bytes deben ponerse a #000000 en hexadecimal. En la tabla se muestran ejemplos del color y el código en hexadecimal asociado.

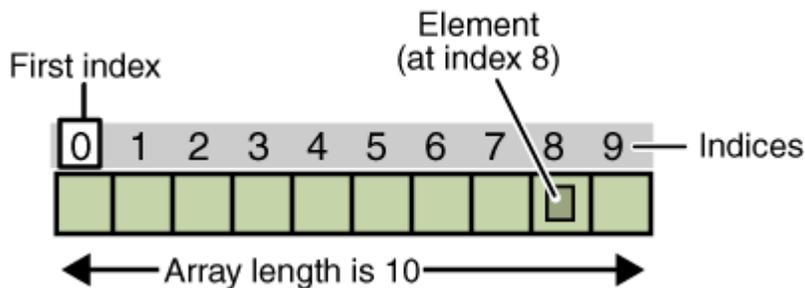
Color	Rojo	Verde	Azul	Hexadecimal
<b>Negro</b>	0	0	0	#000000
<b>Blanco</b>	255	255	255	#FFFFFF
<b>Rojo</b>	255	0	0	#FF0000
<b>Verde</b>	0	192	0	#00C000
<b>Azul</b>	0	0	255	#0000FF
<b>Amarillo</b>	255	255	0	#FFFF00

Las imágenes son arreglos de pixeles. Un arreglo es una secuencia de elementos del mismo tipo y cada elemento tiene un índice asociado.

Los arreglos en Java son objetos, tienen el atributo `length`, este atributo nos dice la longitud del arreglo. Además cuentan con todos los métodos que le fueron heredados de la clase `Object`. La sintaxis en Java para definir un arreglo es:

```
tipo [] nombreArreglo=new tipo[cantidadElementos];
```

El índice del primer elemento es el número 0 y del último es la cantidadElementos - 1.



Por ejemplo, si queremos crear un arreglo de tipo entero que se llame numeros y de 10 elementos debemos hacer lo siguiente:

```
int[] numeros=new int[10];
```

También podemos crear un arreglo y asignarle valores al mismo tiempo como se ve en el siguiente ejemplo.

```
double[] grados={80,90.25,88,92,94.5};
```

0      1      2      3      4

80	90.25	88	92	94.5
----	-------	----	----	------

Podemos crear una matriz que es un arreglo bidimensional. Se define como:

```
tipo [][] matriz=new tipo[x][y]
```

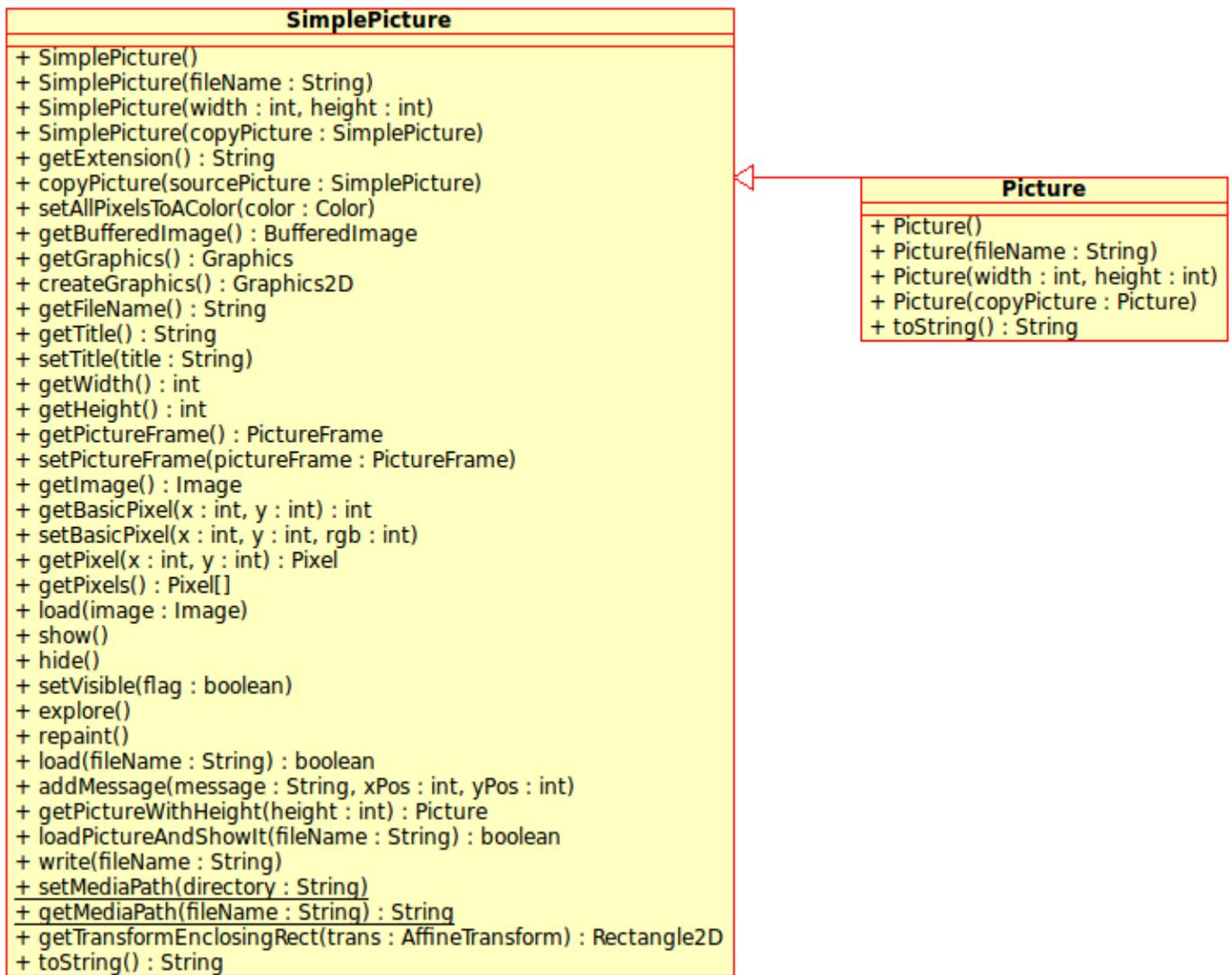
Con la instrucción anterior se crea una matriz de la cantidad de "x" renglones por "y" columnas

También se puede crear una matriz asignando los valores en la declaración, por ejemplo:

```
double[][] matriz={{1,2,3,4},{5,6},{7,8,9,10,11,12},{13}};
```

Se crea una matriz con el primer renglón de 4 elementos, el segundo de dos elementos, el tercer renglón con 6 renglones y el último renglón con un elemento.

En esta práctica utilizaremos las siguientes clases SimplePicture, Picture, Pixel, Color y FileChooser. Observe las siguientes clases:



La figura anterior es el diagrama de las clases SimplePicture y Picture. La flecha que las une indica que Picture hereda los atributos de la clase SimplePicture. Esto significa que los objetos Picture tienen todos los métodos de Picture y los métodos de SimplePicture.

En el diagrama de clases podemos ver rápidamente los nombres de los métodos, los argumentos que requieren los métodos y el tipo de datos que regresan. Por ejemplo en la clase SimplePicture, encontramos el método `getPixel(x : int, y : int) : Pixel`. Esto nos indica que el método recibe dos argumentos de tipo `int` que son las coordenadas en X y Y del pixel que queremos recuperar. También nos indica que este método regresará un objeto de tipo `Pixel`, que será el pixel que se encuentra en las coordenadas que especificamos.

<b>Pixel</b>
+ Pixel(picture : DigitalPicture, x : int, y : int)
+ getX() : int
+ getY() : int
+ getAlpha() : int
+ getRed() : int
+ <u>getRed(value : int) : int</u>
+ getGreen() : int
+ <u>getGreen(value : int) : int</u>
+ getBlue() : int
+ <u>getBlue(value : int) : int</u>
+ getColor() : Color
+ setColor(newColor : Color)
+ updatePicture(alpha : int, red : int, green : int, blue : int)
+ setRed(value : int)
+ setGreen(value : int)
+ setBlue(value : int)
+ setAlpha(value : int)
+ colorDistance(testColor : Color) : double
+ <u>colorDistance(color1 : Color, color2 : Color) : double</u>
+ getAverage() : double
+ toString() : String

La figura anterior muestra el diagrama de la clase Pixel. Note que algunos de los métodos se encuentran subrayados. Esto indica que son métodos estáticos. Un método estático es el que se invoca sobre la clase.

A continuación se muestra el diagrama de la clase FileChooser. Esta clase la utilizaremos en nuestros programas para seleccionar fácilmente, por medio de una interfaz gráfica los archivos con los que queremos trabajar.

<b>FileChooser</b>
+ <u>pickAFile() : String</u>
+ <u>getMediaPath(fileName : String) : String</u>
+ <u>getMediaDirectory() : String</u>
+ <u>setMediaPath(directory : String)</u>

Cuando cambiemos los colores de nuestras imágenes, necesitaremos una forma de especificarle a Java el color que queremos. Esto lo podemos hacer por medio de la clase Color que pertenece al paquete java.awt que es parte de los paquetes incluidos con Java.

En la liga: <http://download.oracle.com/javase/6/docs/api/java/awt/Color.html> podemos consultar sus constructores, atributos y métodos. Ahí veremos que esta clase cuenta con algunos colores predefinidos a los cuales podemos acceder con enunciados como: Color.red. Para los colores que no están predefinidos, podemos especificar el valor de sus componentes RGB en uno de los constructores de la clase.

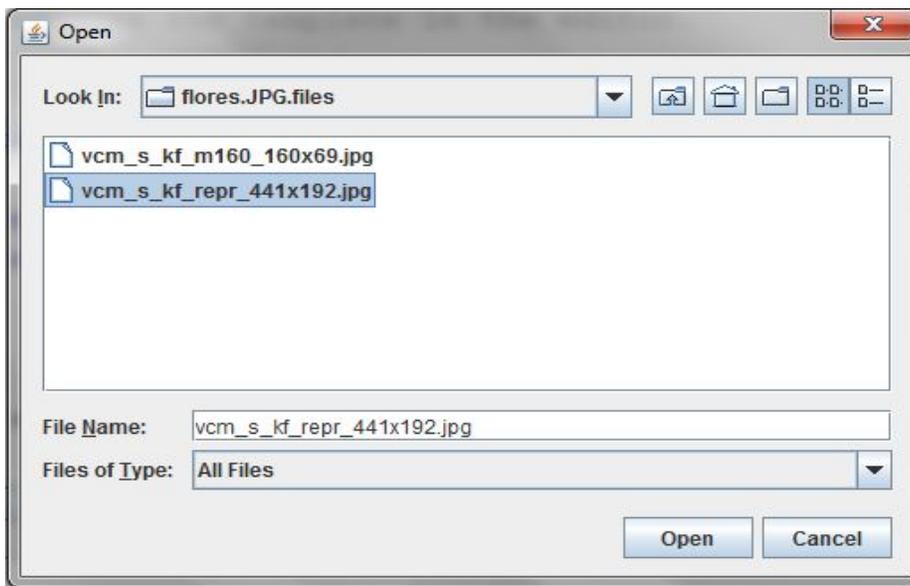
Pongamos en práctica las clases antes mencionadas.

Considere el código siguiente:

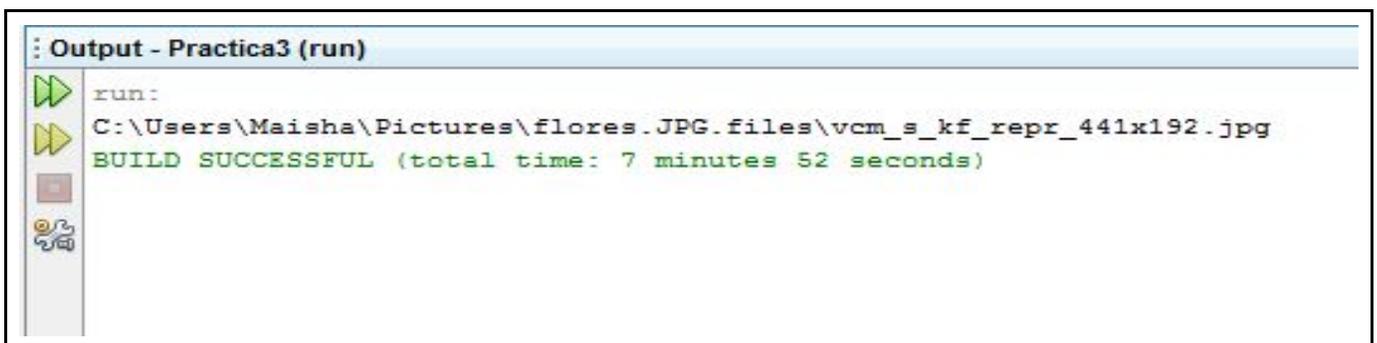
#### LISTADO 1

```
public static void main(String [] args){
    String archivo=FileChooser.pickAFile(); //seleccionamos un archivo de imagen
    System.out.println(archivo);          //se muestra en consola la informacion
    Picture objetoImagen=new Picture(archivo); //creamos un objeto Picture con
//el archivo seleccionado
    objetoImagen.show(); //abre una ventana con la imagen seleccionada
}
```

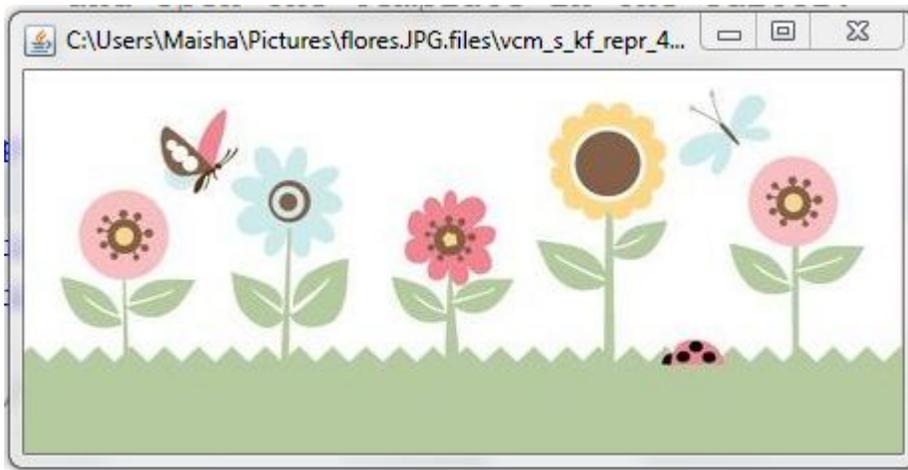
Con el listado 1 comprobamos el funcionamiento de la clase FileChooser y Picture. Primero se selecciona un archivo, que en este caso debe ser una imagen, invocando el método estático pickAFile() este método nos regresa una cadena con el nombre del archivo que contiene la imagen. El resultado de ejecutar esa línea se muestra en la siguiente figura



Enseguida mostramos la cadena el resultado en la consola es:



Enseguida creamos el objeto de la clase Picture e invocamos el método show(). Este método muestra una ventana con la imagen seleccionada como podemos ver en la siguiente imagen



Utilizando las clases Pixel y Color podemos modificar el color de un pixel dentro de la imagen. Debemos incluir el siguiente código en el listado 1

#### LISTADO 2

```
Pixel pix1=objetoImagen.getPixel(0,0);  
pix1.setColor(Color.red);  
objetoImagen.repaint();
```

El resultado de ejecutar las líneas anteriores se muestra en la imagen, la flecha indica el punto.

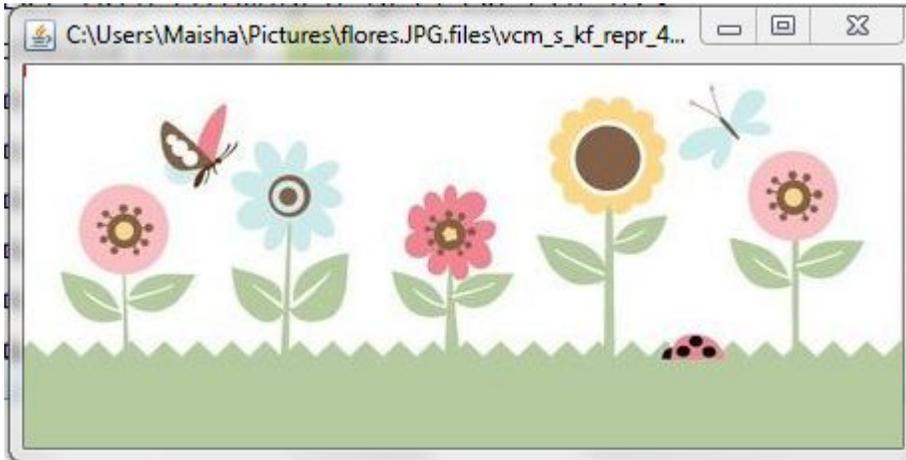


Podemos dibujar una línea de otro color dentro de una imagen, para hacerlo debemos incluir hacer los que se muestra en el siguiente código.

#### LISTADO 3

```
objetoImagen.getPixel(0, 1).setColor(Color.red);  
objetoImagen.getPixel(0, 2).setColor(Color.red);  
objetoImagen.getPixel(0, 3).setColor(Color.red);  
objetoImagen.getPixel(0, 4).setColor(Color.red);  
objetoImagen.getPixel(0, 5).setColor(Color.red);  
objetoImagen.repaint();
```

El resultado es:



Como se puede apreciar se muestra una pequeña línea roja vertical en la esquina superior izquierda de la imagen. A continuación se muestra con mayor detalle la línea roja.

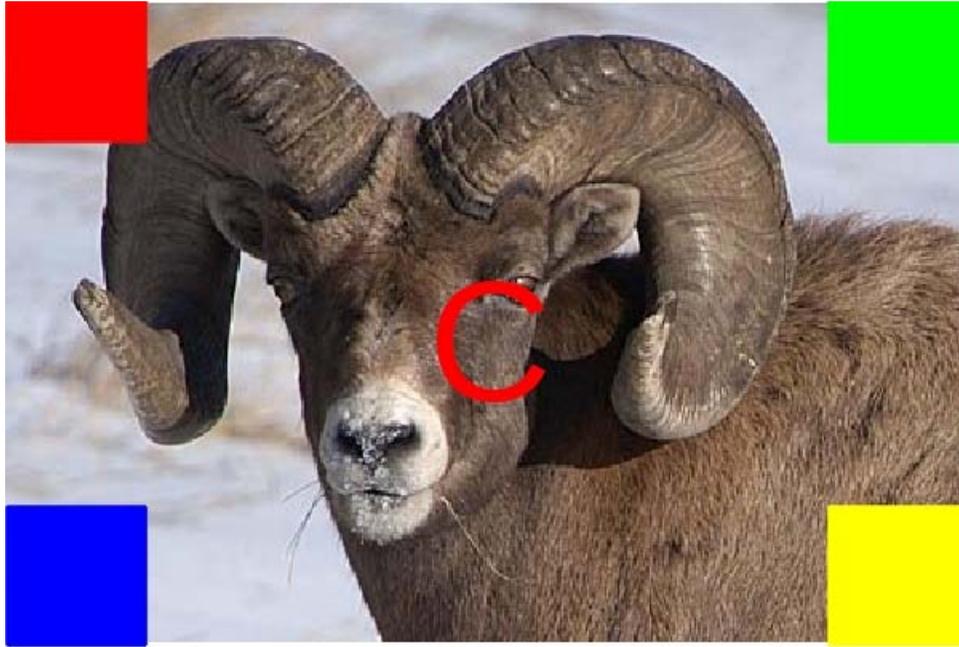


#### 4. PROCEDIMIENTO (DESCRIPCIÓN)

En esta práctica modificará los colores de una parte de una imagen. Para ello es recomendable utilizar una cuadrícula para representar a la imagen como una matriz y poder ubicar con mayor facilidad las coordenadas de los píxeles que se modificarán.

1. Buscar y descargar una imagen en la Web que se encuentre en formato JPG.
2. Determine cuantos píxeles mide la imagen a lo ancho y a lo largo. Esto le permitirá ubicar las coordenadas de los píxeles que cambiará de color en los siguientes pasos.
3. Cambiar el color de los píxeles de las esquinas de la imagen.
  - a. Cambiar el color de un área de 20 x 20 píxeles a color rojo en la esquina superior izquierda.
  - b. Cambiar a verde el color de un área de 20x20 en la esquina superior derecha.
  - c. Cambiar a color azul un área de 20 x 20 en la esquina inferior izquierda.
  - d. Cambiar un área de 20x20 a color amarillo en la esquina inferior derecha.
4. En el centro de la imagen, dibujar la primer letra de su nombre utilizando su color favorito.

La siguiente figura muestra un ejemplo de cómo pudiera quedar la imagen después de la transformación.



<b>A) EQUIPO NECESARIO</b>	<b>MATERIAL</b>
----------------------------	-----------------

Computadoras con capacidad para ejecutar el entorno de desarrollo Netbeans. Paquete misClases

<b>7. REFERENCIAS</b>
-----------------------

Netbeans

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

Java 6

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

Información sobre el modelo de color RGB

[http://es.wikipedia.org/wiki/Modelo\\_de\\_color\\_RGB](http://es.wikipedia.org/wiki/Modelo_de_color_RGB)