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)
5	NOMBRE DE LA PRÁCTICA	Combinación de imágenes	2

1. INTRODUCCIÓN

En esta práctica retomaremos las clases `Pixel, Picture, SimplePicture y FileChooser` que fueron utilizadas en la práctica 4. Pero ahora utilizaremos ciclos para modificar las imágenes. Para ello también agregaremos un método a la clase `Picture` que nos facilitará el acceso a los pixeles de las imágenes.

2. OBJETIVO (COMPETENCIA)

Utilizar ciclos y sentencias de selección para desarrollar los métodos de una clase, mostrando una actitud creativa y ordenada.

Formuló	Revisó	Aprobó	Autorizó
Cecilia Curlango Rosas Maria Luisa González Ramírez	Gloria E. Chávez Valenzuela		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

Java utiliza los ciclos `for`, `while` y `do-while`.
La sintaxis en Java para el ciclo `for` es:

```
for(inicializacion;condicion;iteracion){
//sentencias
}
```

En la inicialización se declara e inicializa la variable que controla el ciclo, esta acción solo se realiza una vez. La condición indica cuantas veces se repetirá el ciclo, si la condición es verdadera se repite el ciclo y la iteración se realiza cada vez que se ejecuta el ciclo, puede ser incremento o decremento.

El ciclo `while` se repite siempre y cuando la condición sea verdadera.
La sintaxis del ciclo `while` es:

```
while(condicion)
{
      //sentencias
}
```

También existe el ciclo `do-while`. La principal diferencia del `do-while` contra el `while` es que el primero siempre se ejecuta por lo menos una vez.
La sintaxis del ciclo `do-while` es:

```
do{
//sentencias
}while(condicion);
```

Vamos a practicar el uso de los ciclos modificando los pixeles de una imagen.
La clase `Picture` tiene un método llamado `getPixels()`, para obtener un arreglo de pixeles de la imagen, podemos analizar el contenido de la imagen utilizando este arreglo.
El siguiente ejemplo ilustra el uso de las clases `Picture`, `Pixel` y `Color`, modificando el brillo de cada pixel dentro de una imagen. En el ejemplo del LISTADO 1 se selecciona una imagen con la clase `FileChooser`, después se imprime en consola el archivo seleccionado. Con el archivo se crea un objeto de la clase `Picture`, se leen todos los pixeles de la imagen almacenándolos en el `arregloPixel`. Para modificar cada pixel de la imagen se utiliza un ciclo `for` con el cual se obtiene el color de cada pixel y se obscurece. Esto lo logra invocando el método `darker()`, que pertenece a la clase `Color`. Por último se asigna el color modificado al pixel y cuando termina el ciclo se invoca el método `repaint()` de la clase `Picture` que mostrará la imagen modificada.

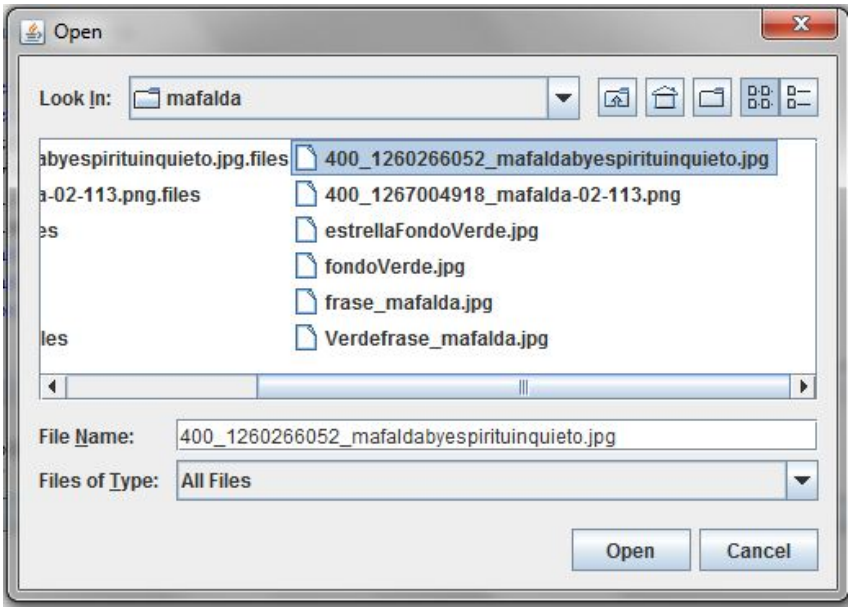
LISTADO 1

```
public static void main(String [] args){

    String archivo=FileChooser.pickAFile();
     System.out.println(archivo);
     Picture objetoImagen=new Picture(archivo);

     Pixel [] arregloPixel=objetoImagen.getPixels();
         for (int i = 0; i < arregloPixel.length; i++) {
         Color colorObscuro=arregloPixel[i].getColor().darker();
         arregloPixel[i].setColor(colorObscuro);
     }
     objetoImagen.repaint();
}
```

Cuando se ejecuta el código anterior, lo primero que sucede es que se abre una ventana para seleccionar el archivo de la imagen que se modificará, como se muestra en la siguiente figura:



En la imagen de la izquierda se muestra la imagen original y en la de la derecha se muestra la imagen modificada por el código del LISTADO 1.

4. PROCEDIMIENTO (DESCRIPCIÓN)

1.- Agregue un método a la clase `Picture` para acceder a los pixeles de una imagen a través de una matriz (en lugar de un arreglo). Cuando se ejecute este método, deberá regresar una matriz de pixeles. La firma del método es:

```
public Pixel[][] getPixelMatrix()
```

.

2.- Agregue un método en la clase Picture que realice un método para cambiar todos los pixeles de un color a otro color que usted prefiera.

```
public void cambiaColor(Picture imagen,Color viejo,Color nuevoColor)
```

3.- Realice un método para fusionar dos imágenes, utilice la foto con fondo verde, identifique el color de verde y cambie los pixeles verdes por otra imagen.

```
public void fusiona(Picture original,Picture fondo)
```

A) EQUIPO NECESARIO	MATERIAL

Computadoras con capacidad para ejecutar el entorno de desarrollo Netbeans. Paquete `misClases`. Una imagen tomada con un fondo de color verde y otra imagen que tenga las mismas dimensiones que la imagen de fondo verde.

7. REFERENCIAS

Netbeans
http://netbeans.org/downloads/
Java 6
http://www.oracle.com/technetwork/java/javase/downloads/index.html