

Trabajo Práctico 1

Ejercicio 1:

- a) y b) Utilizar fórmula de distancia focal y magnificación
- c) Considerar cuál es la distancia mínima que debe poder ser representada

Ejercicio 2: Considerar si se puede representar esa situación con la cantidad de bits especificada y si no alcanza, incrementar la cantidad de bits hasta que alcance.

Ejercicio 3: ...

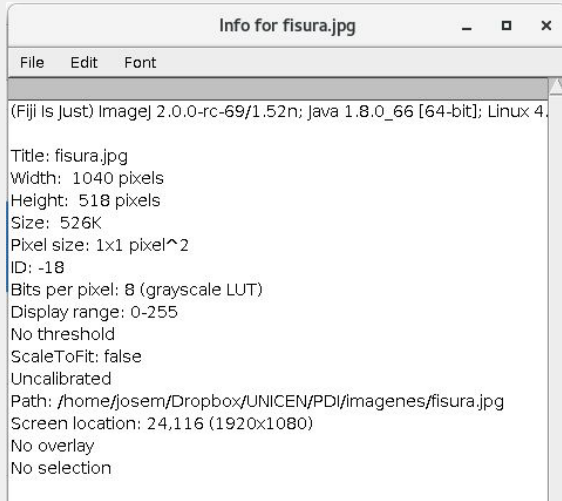
Ejercicio 4: Ver ejemplos en Octave, ImageJ, C++ (EasBMP y otros), Delphi, Visual C++, Python con Scikit-Image.

Octave: Apertura y muestra de propiedades

```
>> ci = imread('carotida.png');  
>> imshow(ci);  
>> imfinfo('carotida.png');
```

Ejercicio 4:

ImageJ



Python con Scikit-image

```
import skimage.io as io
w = io.imread('lena.bmp')
w.dtype
w.shape
```

Java

```
package pdi;
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.IOException;
import javax.imageio.ImageIO;

public class PDI {
    public static void main(String[] args) {
        BufferedImage img;
        try {
            img = ImageIO.read(new File("lenaRGB.bmp"));
            int j = img.getType();
            System.out.println(j);
        } catch (IOException e) {
        }
    }
}
```

Ejercicio 5:

Java

```
import java.io.File;
import java.io.IOException;
import java.awt.image.BufferedImage;
import javax.imageio.ImageIO;
public class CrearImagen
{
    public static void main(String args[])throws IOException
    {
        int width = 640, height = 480;
        BufferedImage img = null;
        img = new BufferedImage(width, height, BufferedImage.TYPE_INT_ARGB);
        File f = null;
        for (int y = 0; y < height; y++)
        {
            for (int x = 0; x < width; x++)
            {
                int a = (int)(Math.random()*256);
                int r = (int)(Math.random()*256);
                int g = (int)(Math.random()*256);
                int b = (int)(Math.random()*256);
                int p = (a<<24) | (r<<16) | (g<<8) | b;
                img.setRGB(x, y, p);
            }
        }
    }
}
```

Continúa Ejercicio 5 Java

```
try
{
    f = new File("random.png");
    ImageIO.write(img, "png", f);
}
catch(IOException e)
{
    System.out.println("Error: " + e);
}
}
```

Plugin ImageJ

Continúa Ejercicio 7 Plugin de ImageJ

```
import ij.*;
import ij.process.*;
import ij.gui.*;
import java.awt.*;
import ij.plugin.filter.*;

public class Cambiar_Color
implements PlugInFilter {
    ImagePlus imp;

    public int setup(String arg,
ImagePlus imp) {
        this.imp = imp;
        return DOES_ALL;
    }
}
```

```
public void run(ImageProcessor ip) {
    int w = ip.getWidth();
    int h = ip.getHeight();
    for (int u = 0; u < w; u++) {
        for (int v = 0; v < h; v++) {
            int p = ip.getPixel(u, v);
            if ((p > 0) && (p <= 32))
                ip.putPixel(u, v,0);
            if ((p > 32) && (p <= 64))
                ip.putPixel(u, v,32);
            ....
        }
    }
}
```