

```

package leerpdf;
import java.awt.Rectangle;
import java.awt.print.PageFormat;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintStream;
import java.io.PrintWriter;
import java.util.List;
import org.apache.pdfbox.pdmodel.PDDocument;
import org.apache.pdfbox.pdmodel.PDPage;
import org.apache.pdfbox.util.PDFTextStripperByArea;
public class LeerPdf {
    public static void main(String[] args) {
        LeerPdf leerPDF =new LeerPdf();
        leerPDF.lecturaPDF();
    }

    public void lecturaPDF(){
        String ln = System.getProperty("line.separator");
        File dir = new File("d:\\carpeta");//CREO UN OBJETO CON TODOS LOS
ARCHIVOS QUE CONTIENE LA CARPETA QUE CONTIENE LOS PDFS.
        String[] ficheros = dir.list();//ARREGLO QUE ALMACENARÁ TODOS LOS NOMBRES
DE LOS ARCHIVOS QUE ESTAN DENTRO DEL OBJETO.

        if (ficheros == null)//EXCEPCION
            System.out.println("No hay archivos en la carpeta especificada");
        else {
            for (int x=0;x<ficheros.length;x++){//RECORREMOS EL ARREGLO CON LOS
NOMBRES DE ARCHIVO
                String ruta=new String();//VARIABLE QUE DETERMINARA LA RUTA DEL
ARCHIVO A LEER.
                ruta=("d:\\carpeta\\"+ficheros[x]); //SE ALMACENA LA RUTA DEL ARCHIVO
A LEER.

                try {
                    PDDocument pd = PDDocument.load(ruta); //CARGAR EL PDF
                    List l = pd.getDocumentCatalog().getAllPages();//NUMERO LAS
PAGINAS DEL ARCHIVO
                    Object[] obj = l.toArray();//METO EN UN OBJETO LA LISTA DE
PAGINAS PARA MANIPULARLA
                    PDPage page = (PDPage) obj[0];//PAGE ES LA PAGINA 1 DE LA QUE
CONSTA EL ARCHIVO
                    PageFormat pageFormat = pd.getPageFormat(0);//PROPIEDADES DE LA
PAGINA (FORMATO)
                    Double d1 = new Double(pageFormat.getHeight());//ALTO
                    Double d2 = new Double(pageFormat.getWidth());//ANCHO
                    int width = d1.intValue();//ANCHO
                    int eigh=1024;//ALTO

                    PDFTextStripperByArea stripper
= new PDFTextStripperByArea();//COMPONENTE PARA ACCESO AL TEXTO
                    Rectangle rect = new Rectangle(0, 0, width, eigh);//DEFNIR
AREA DONDE SE BUSCARA EL TEXTO
                    stripper.addRegion("areal", rect);//REGISTRAMOS LA REGION CON
UN NOMBRE

                    stripper.extractRegions(page);//EXTRAE TEXTO DEL AREA

                    String contenido = new String();//CONTENIDO = A LO QUE CONTENGA
EL AREA O REGION
                    contenido=(rect+stripper.getTextForRegion("areal"));

```

```

        File archivo=new File(ficheros[x]+".txt");//CREAMOS ARCHIVO CON
NOMBRE ORIGINAL PERO EN TXT
        BufferedWriter writer
= new BufferedWriter(new FileWriter(archivo));//CREAMOS EL ESCRITOR
        writer.write(ruta);//IMPRIMIMOS LA RUTA
        writer.write(contenido);//IMPRIMIMOS EL CONTENIDO
        writer.close();//CERRAMOS EL ESCRITOR

        pd.close();//CERRAMOS OBJETO ACROBAT
    } catch (IOException e) {
        if(e.toString()!=null){
            File archivo=new File("dañado_"+ficheros[x]+".txt");//SEPARA
LOS DAÑADOS
        }
        System.out.println("Archivo dañado "+ficheros[x]);// INDICA EN
CONSOLA CUALES SON LOS DAÑADOS
        e.printStackTrace();
    } //CATCH
} //FOR
} //ELSE
} //LECTURAPDF()
} //CLASS

```