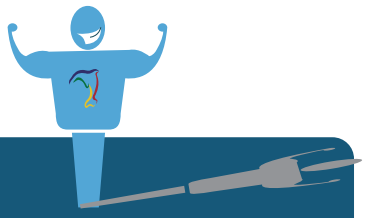
autentia

Avenida de Castilla,1 - Edificio Best Point - Oficina 21B
28830 San Fernando de Henares (Madrid)
tel./fax: +34 91 675 33 06
info@autentia.com - **www.autentia.com**

¿**Qué ofrece** Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
Ese apoyo que siempre quiso tener...

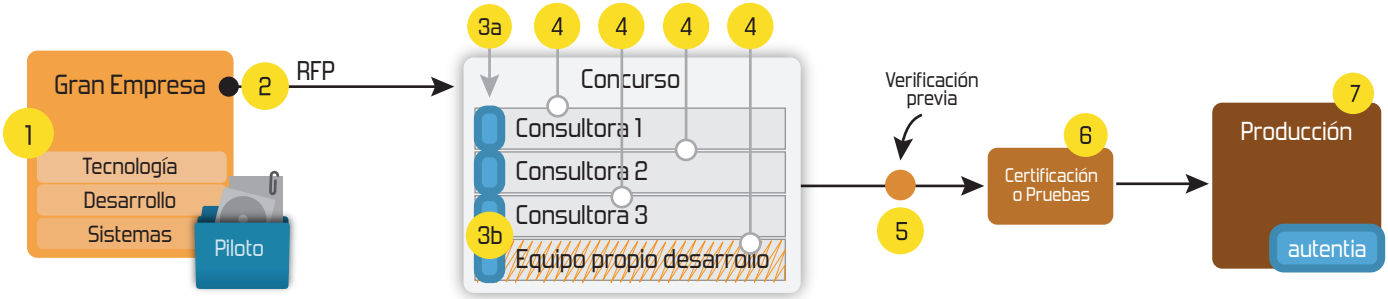
1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces, HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)
Gestor de contenidos (Alfresco)
Aplicaciones híbridas

Control de autenticación y acceso (Spring Security)
UDDI
Web Services
Rest Services
Social SSO
SSO (Cas)

JPA-Hibernate, MyBatis
Motor de búsqueda empresarial (Solr)
ETL (Talend)

Dirección de Proyectos Informáticos.
Metodologías ágiles
Patrones de diseño
TDD

Tareas programadas (Quartz)
Gestor documental (Alfresco)
Inversión de control (Spring)

BPM (jBPM o Bonita)
Generación de informes (JasperReport)
ESB (Open ESB)

Compartimos nuestro conocimiento en:
www.adictosaltrabajo.com

Para más información visítenos en:
www.autentia.com

Tutorial desarrollado por:
Alberto Carrasco Montenegro

Puedes encontrarme en Autentia
Somos expertos en Java/J2EE
Contacta en info@autentia.com

Descargar este documento en formato PDF pdfbox.pdf

En Autentia siempre estamos pendientes de las librerías de libre distribución que van surgiendo para facilitar y añadir mayor funcionalidad a nuestros desarrollos. Hoy queremos presentaros la librería java PDFBox, una herramienta muy interesante para el tratamiento de documentos en formato pdf.

1. ¿Qué es PDFBox?

PDFBox es una librería Java de código abierto que permite trabajar con documentos en formato pdf. Estas librerías permiten acciones relativas a la creación, manipulación y extracción del contenido de documentos en formato pdf.

Algunas de las funcionalidades concretas que ofrece esta librería son las siguientes:

- Extraer el texto contenido en archivos pdf.
- Unir ficheros pdf.
- Encriptación/desencriptación de documentos pdf.
- Crear un pdf a partir de un fichero de texto.
- Crear imágenes a partir de ficheros pdf.

Además, PDFBox incluye varias utilidades para línea de comandos que permiten realizar algunas de estas acciones. Este tutorial se centrará en la utilización de dichas utilidades, por lo que no son necesarios conocimientos profundos de Java. No será necesario realizar ningún programa en Java y solamente serán precisas nociones básicas sobre ejecución de programas Java en consola.

2. Instalación y requisitos

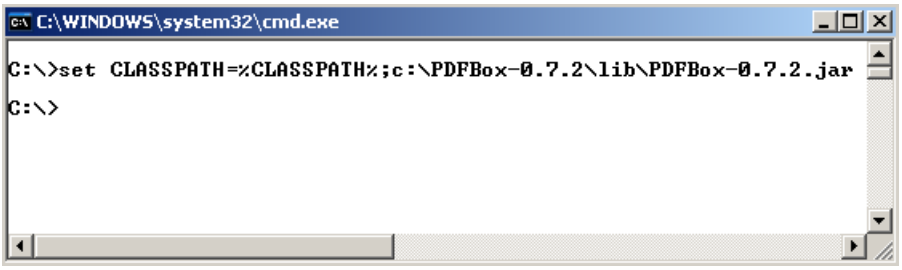
Los ejemplos que se estudiarán más adelante en este documento, se ejecutaron en un entorno *Windows XP* que disponía de la distribución Java *j2sdk-1.4.2.* Esta distribución puede obtenerse gratuitamente desde la web de *SUN* en el enlace:

http://java.sun.com/products/archive/j2se/1.4.2/index.html

La distribución de PDFBox utilizada en este tutorial es la contenida en el archivo *PDFBox-0.7.2.zip*, que se puede descargar gratuitamente desde el enlace:

http://sourceforge.net/project/showfiles.php?group_id=78314

Una vez descomprimido este archivo, será necesario que el archivo PDFBox-0.7.2/lib/**PDFBox-0.7.2.jar** sea visible en el classpath, para poder emplear las utilidades que se ilustrarán con ejemplos a continuación. Para ello, puede utilizarse el comando *set* de DOS añadiendo la ruta correspondiente a la variable de entorno *CLASSPATH.* Teclear desde la consola de símbolo del sistema lo siguiente:



Es recomendable disponer de algún visor de documentos pdf para seguir mejor la ilustración de los ejemplos que se expondrán a continuación. En este caso se utilizó el *Adobe Reader*, que puede obtenerse gratuitamente desde la dirección http://www.adobe.es/products/acrobat/readstep2.html

3. Acciones fundamentales

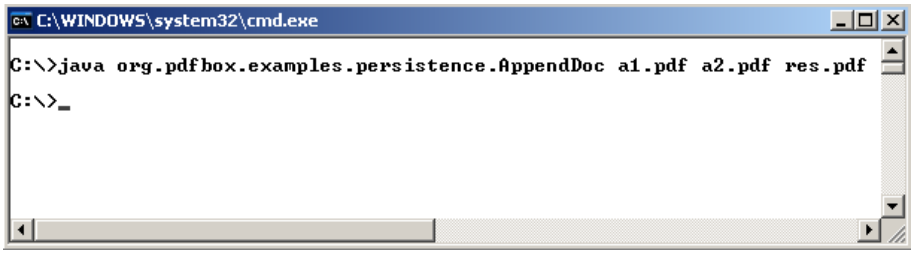
En este apartado se mostrarán ejemplos de algunas acciones fundamentales que pueden realizarse con la librería PDFBox. Como se comentó anteriormente, no será necesario realizar ningún programa, sino que se utilizarán las propias utilidades (*main* de clases Java) incluidas en esta librería.

3.1. Unir dos archivos pdf

El *main* incluido en la clase *org.pdfbox.examples.persistence.AppendDoc* permite unir dos documentos pdf en uno solo. Los argumentos de entrada que necesita este *main* son los siguientes:

- Primer documento pdf.
- Segundo documento pdf.
- Documento pdf donde se guardará el resultado de la unión de los dos anteriores.

De esta forma, puede ejecutarse dicho programa invocando la clase correspondiente con los argumentos que se acaban de precisar.



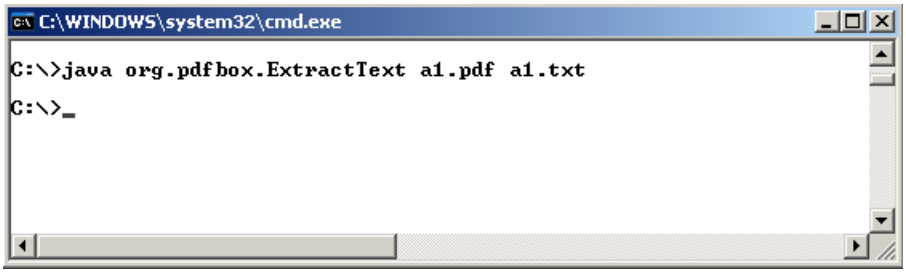
El resultado de esta ejecución es la unión de los documentos *a1.pdf* y *a2.pdf* en un documento resultado llamado *res.pdf*.

3.2. Extraer el texto contenido en un documento pdf

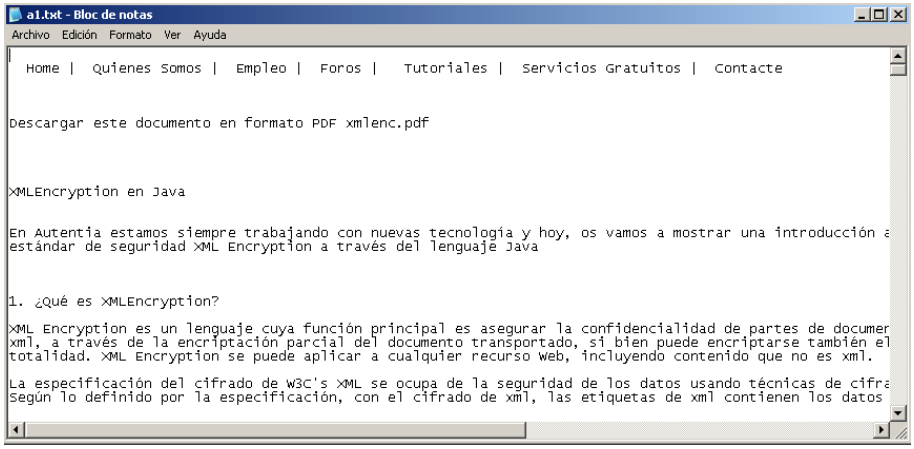
El *main* incluido en la clase *org.pdfbox.ExtractText* permite extraer el texto de un documento pdf en un documento de texto. Los argumentos de entrada que necesita este *main* son los siguientes:

- Documento pdf del que se desea extraer el texto.
- Documento de texto donde se escribirá dicho texto.

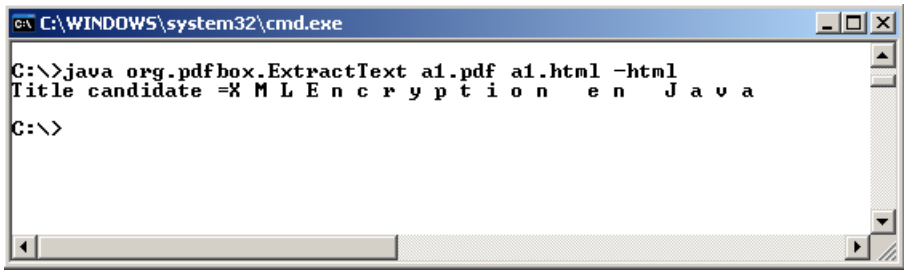
De esta forma, puede ejecutarse dicho programa invocando la clase correspondiente con los argumentos que se acaban de precisar.



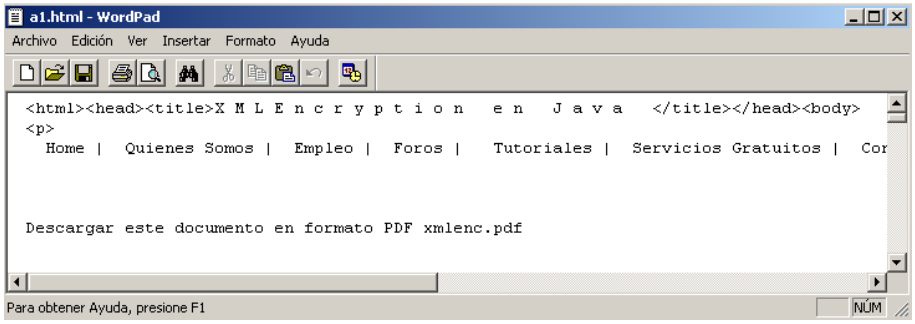
El resultado de esta ejecución es la extracción del texto contenido en el documento *a1.pdf* en el fichero *a1.txt*. A continuación se muestra una vista preliminar de ambos documentos.



Existen algunas opciones interesantes para ejecutar esta utilidad. Una de ellas consiste en la posibilidad de elegir extraer el texto de un documento pdf a un documento en formato html. Se muestra un ejemplo a continuación.



En este caso, el documento resultante (*a1.html*) tendría el siguiente aspecto:



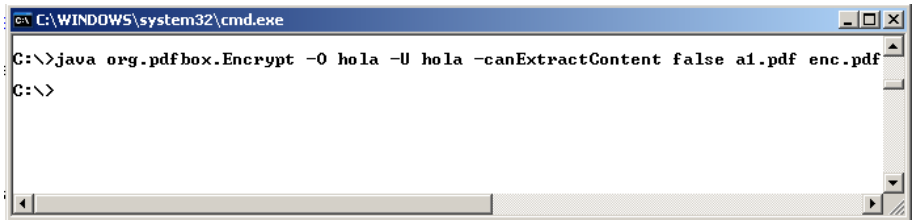
Pueden seleccionarse algunas otras opciones útiles como pueden ser elegir la primera página del documento pdf a partir de la que se empezará la extracción (*-startPage*), elegir la última página del documento pdf hasta la que se realizará la extracción (*-endPage*), elegir otros formatos de texto de salida para la extracción, tales como ISO-8859-1, UTF-16BE y similares (*-encoding*), introducir la contraseña si el documento pdf está protegido (*-password*), etc.

3.3. Encriptación/Desencriptación de documentos pdf

El *main* incluido en la clase *org.pdfbox.Encrypt* permite proteger el acceso y modificación de un documento pdf. Los argumentos básicos de entrada que necesita este *main* son los siguientes:

- Opciones de protección. Esto incluye la especificación de la contraseña de acceso, longitud de la clave de cifrado y especificación de la protección que desea otorgarse al documento (frente a extracción de texto, modificaciones, etc).
- Documento pdf que se desea encriptar.
- Documento pdf resultante encriptado.

Por ejemplo, podría protegerse un documento pdf con una ejecución similar a esta:

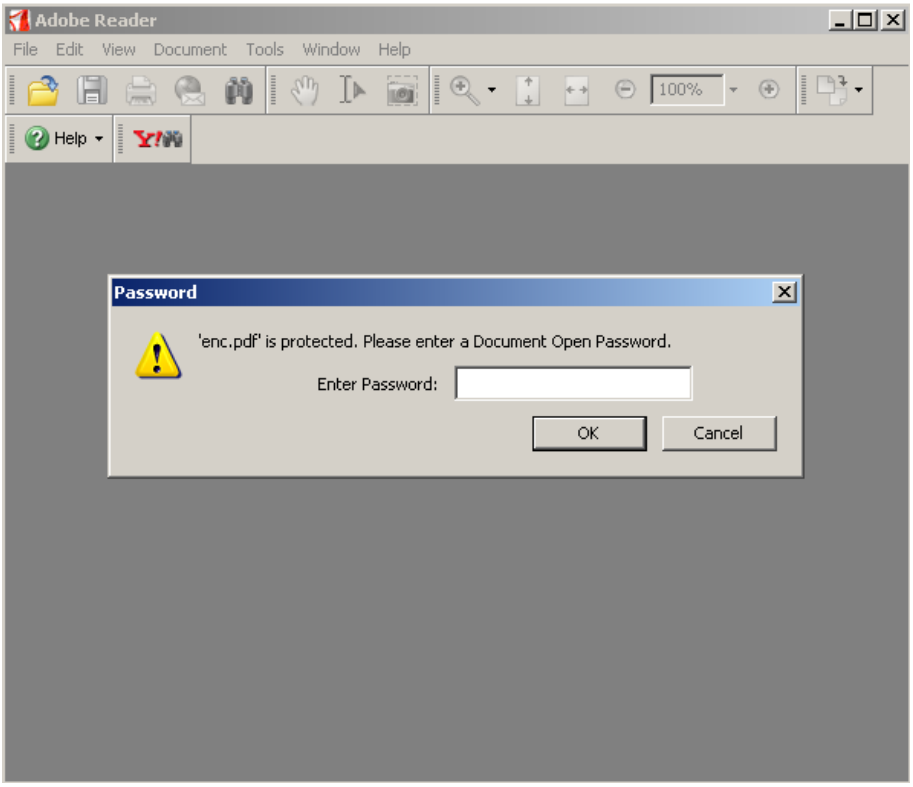


Con el parámetro *-O* se asigna la contraseña de propietario del documento. Esta será necesaria en caso de que quiera desprotegerse el documento posteriormente.

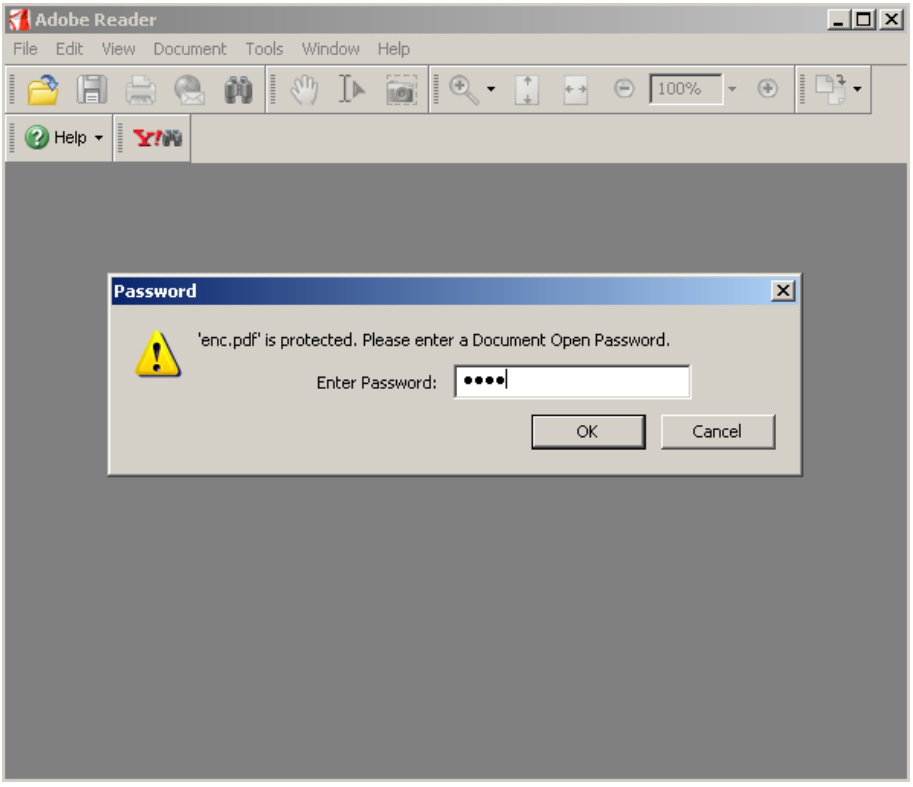
Con el parámetro *-U* se indica la contraseña de usuario que se ligará al documento encriptado. Esta será necesaria para acceder al documento normalmente a través de un visor de pdf's.

Con el parámetro *-canExtractContent* se indica si se permite el acceso para extraer el contenido del documento (en este caso, se deniega con *false*). Por defecto, éste y el resto de permisos que pueden modificarse se encuentran puestos a *true*.

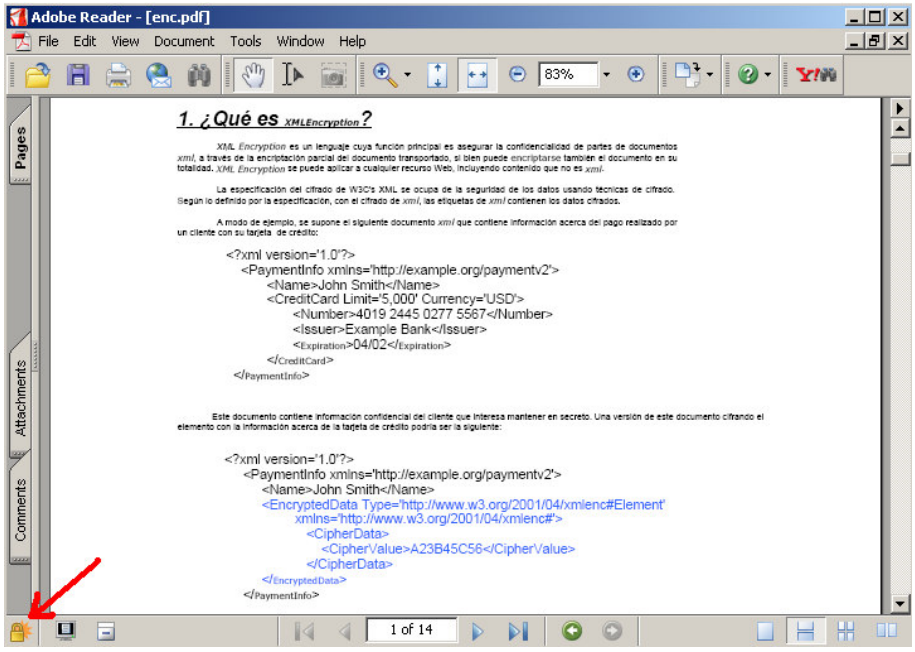
Si se intenta abrir el documento *enc.pdf* con un visor de pdf's, se solicitará en primer lugar la contraseña de acceso.

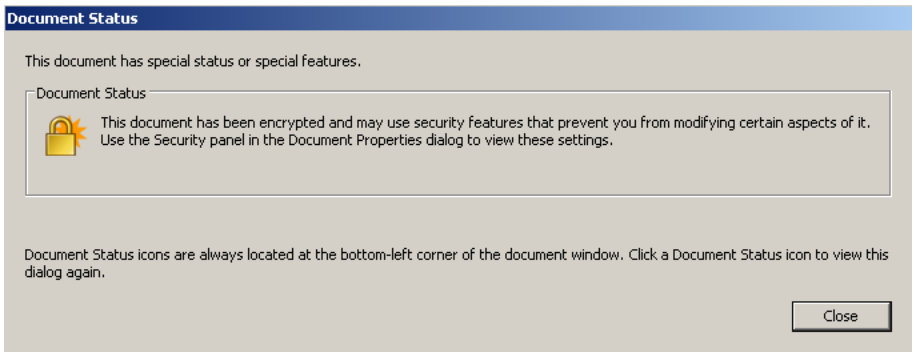


Si se dispone de la contraseña de usuario podrá accederse al documento.



En este caso, puede observarse que el visor de documentos pdf informa de que el documento está encriptado de forma que algunas acciones sobre el documento están protegidas.



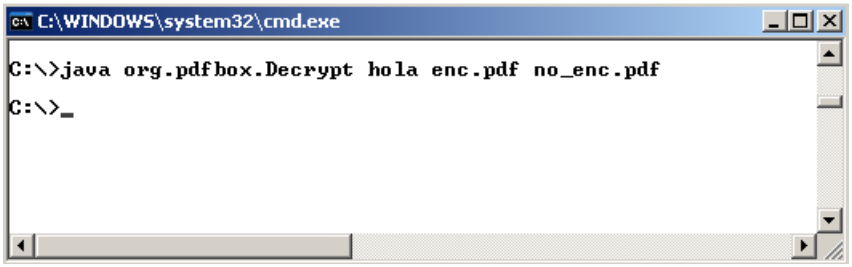


De hecho, puede comprobarse en este caso que no es posible realizar una extracción de texto desde el propio visor (con *copy&paste*).

Para desencriptar el documento, se utiliza el *main* incluido en la clase *org.pdfbox.Decrypt*. Los argumentos básicos de entrada que necesita este *main* son los siguientes:

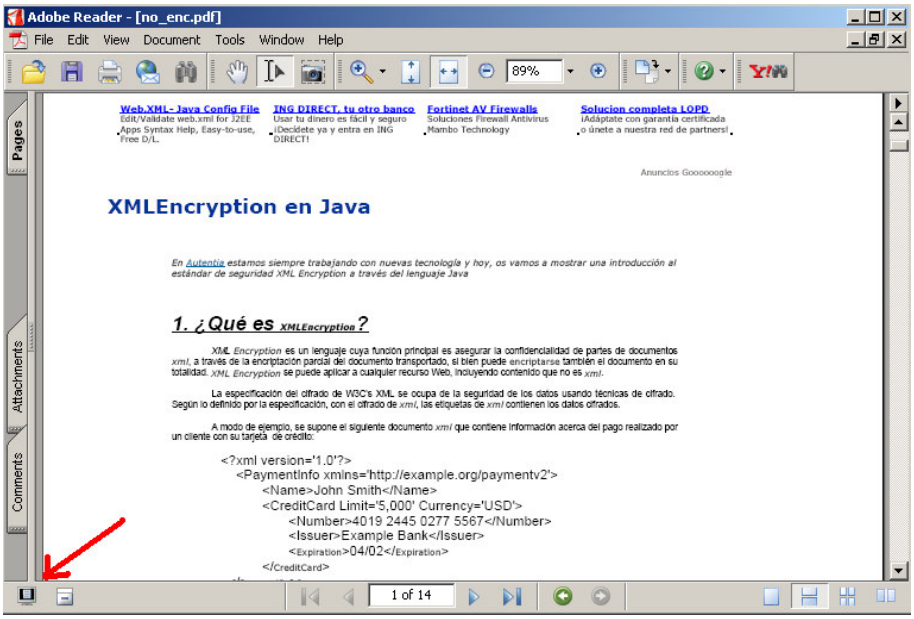
- Contraseña de propietario del documento.
- Documento pdf que se desea desencriptar.
- Documento pdf resultante desencriptado.

El documento encriptado *enc.pdf* podría desencriptarse del siguiente modo:



Se recuerda que *hola* es la contraseña de propietario especificada en la fase de encriptado realizada anteriormente. El documento desencriptado se guardará en *no_enc.pdf* .

Si se abre el fichero *no_enc.pdf* con un visor de pdf's, podrá comprobarse que ya no es necesario introducir una contraseña de acceso, y que ya no aparece el icono informando de que el documento esta protegido.

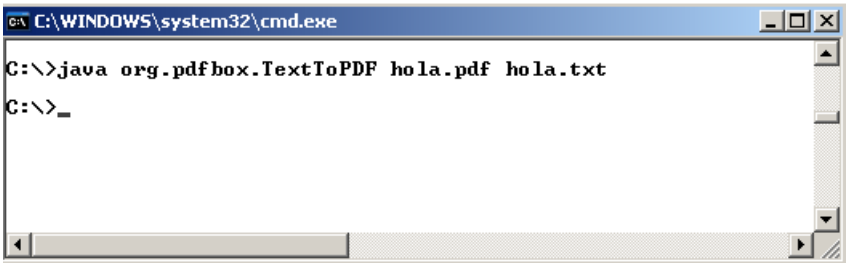


3.4. Crear un pdf a partir de un fichero de texto

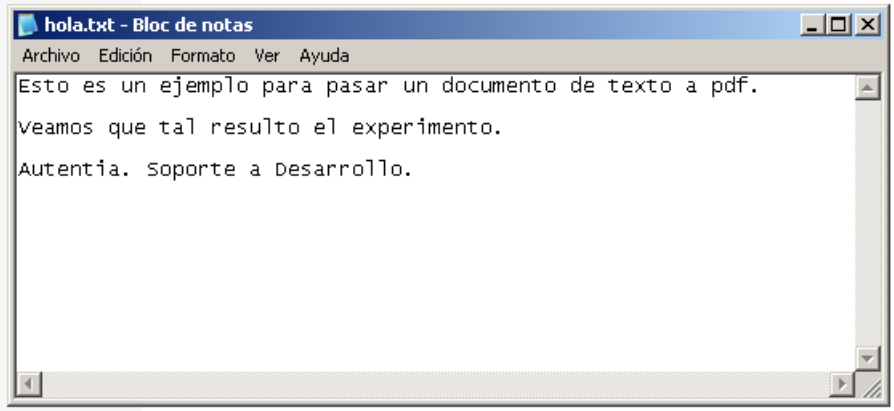
El *main* incluido en la clase *org.pdfbox.TextToPDF* permite crear un documento en pdf a partir de un documento de texto. Los argumentos de entrada que necesita este *main* son los siguientes:

- Documento pdf que se creará.
- Documento de texto a partir del que se creará el documento pdf mencionado.

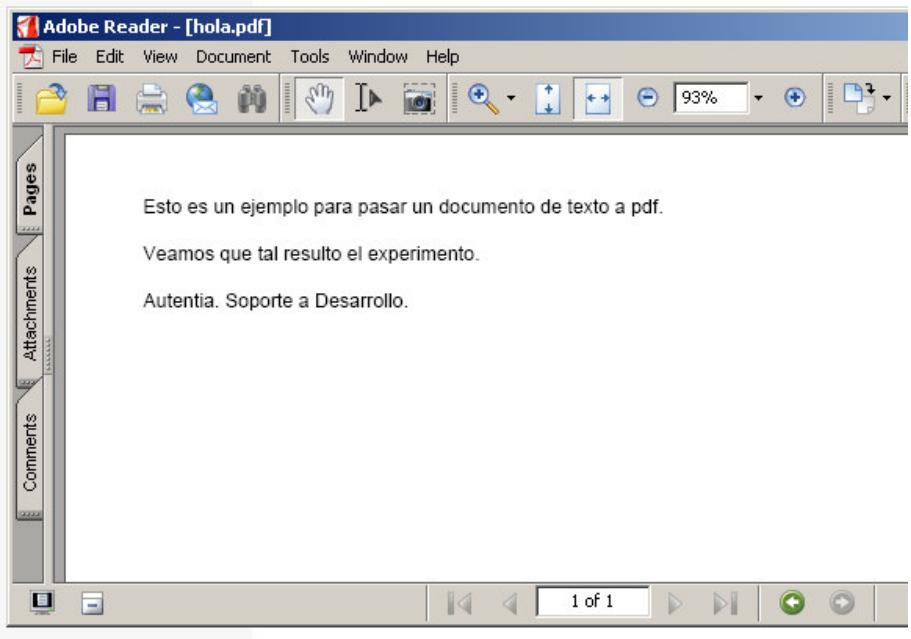
De esta forma, puede ejecutarse dicho programa invocando la clase correspondiente con los argumentos que se acaban de precisar.



El documento de texto original *hola.txt* tiene el siguiente aspecto:



El documento pdf generado, por tanto, tendrá el siguiente aspecto:



Puede seleccionarse si se desea el tipo de fuente que tendrá el texto en el documento pdf que se creará (*-standardFont*).

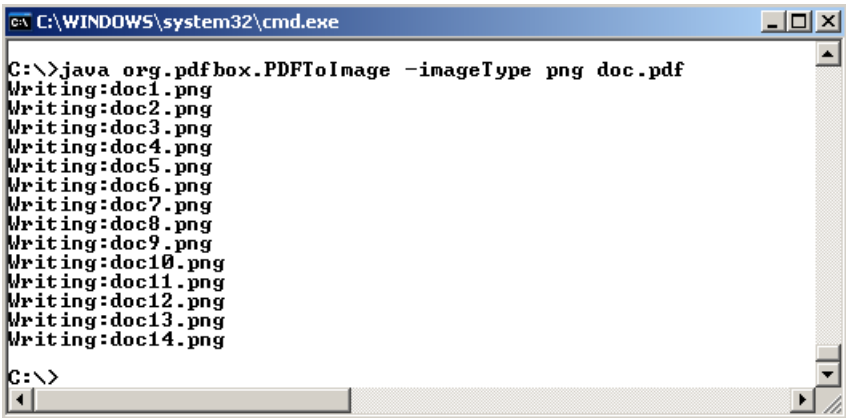
3.5. Crear imágenes a partir de un fichero pdf

El *main* incluido en la clase *org.pdfbox.PDFToImage* permite pasar a imágenes el contenido de un documento en pdf. Los argumentos de entrada que necesita este *main* son los siguientes:

- Opciones. Pueden especificarse diversas opciones como el formato de las imágenes (*-imageType*), página desde o hasta la cual se generarán imágenes (*-startPage, -endPage*), etc.
- Documento pdf a partir del que se generarán imágenes.

PDFBox generará una imagen para cada página del documento.

De esta forma, puede ejecutarse dicho programa invocando la clase correspondiente con los argumentos que se acaban de precisar.



En este caso se ha escogido el formato *png* para las imágenes, y se generó una imagen para cada página de las catorce que contiene el documento pdf original, como puede observarse en las trazas de ejecución.

4. Conclusión

Como ha podido comprobarse, la librería PDFBox puede ser una herramienta muy útil y sencilla para manejar documentos en formato pdf. No son siquiera necesarias nociones de programación en Java para manipular documentos pdf, tal como se ha visto. Además puede obtenerse gratuitamente, con lo que constituye una opción muy interesante para cualquiera que precise realizar operaciones básicas o no tan básicas sobre documentos pdf, ya sean desarrolladores de software o usuarios con ligeros conocimientos de Java, tal como se comentó anteriormente.

En Autentia trabajamos constantemente con este tipo de librerías, cómo podéis comprobar en muchos otros tutoriales de esta web, lo que supone una garantía de calidad en nuestros servicios de soporte a desarrollo.

5. Fuentes

- PDFBox. Página oficial del proyecto.
  http://www.pdfbox.com
- *SUN*. Página oficial.
  http://java.sun.com
- Página de Adobe.
  http://www.adobe.es

Puedes opinar sobre este tutorial aquí

Recuerda

que el personal de Autentia te regala la mayoría del conocimiento aquí compartido (Ver todos los tutoriales)

¿Nos vas a tener en cuenta cuando necesites consultoría o formación en tu empresa?

¿Vas a ser tan generoso con nosotros como lo tratamos de ser con vosotros?

info@autentia.com

Somos pocos, somos buenos, estamos motivados y nos gusta lo que hacemos .....
Autentia = Soporte a Desarrollo & Formación

Nuevo servicio de notificaciones

Si deseas que te enviemos un correo electrónico cuando introduzcamos nuevos tutoriales, inserta tu dirección de correo en el siguiente formulario.

Subscribirse a Novedades	
e-mail	
	Enviar

Otros Tutoriales Recomendados (También ver todos)

Nombre Corto	Descripción
Soporte de Asserts en Java 1.4.x	Os mostramos como utilizar los asserts en Java (disponibles a partir de la versión 1.4)
Construir un Servidor Web en Java	En este tutorial os enseñamos los principios de las aplicaciones multi-hilo a través de la creación de un servidor web básico en Java. Podremos ver en un ejemplo real el uso de sockets, threads, excepciones, etc.
Documentar código Java con JavaDoc	Os mostramos como utilizar los comentarios y etiquetas de JavaDoc para documentar programas Java.
Mensajes multi-idioma en Java	Os mostramos como aprovechar las caracteristicas mutilenguaje de Java, usando las clases: Locate, ResourceBundle, MessageFormat, etc. Fundamental para un correcto diseño ...
Upload de ficheros en Java	Os mostramos como enviar ficheros a un servidor Web y manipularlos en un servlet en el servidor, gracias a APIs de apache
Programa de dibujo en Java con NetBeans	En este tutorial os enseñamos a manejar el entorno de desarrollo NetBeans a través de la creación de una aplicación gráfica que sea capaz de pintar lineas de un modo persistente (a repintados). Es un buen ejemplo de gestión de eventos gráficos .

Decompilar Java	Os mostramos como recuperar el fuente de vuestro código a partir de los ficheros compilados .class
Java en tu movil con J2ME	Os enseñamos como construir una aplicación Java capaz de correr en tu Movil gracias a J2ME
Técnicas básicas y poco comentadas en Java	Os mostramos como realizar algunas cosas simples en Java: Formateo de decimales y enteros, gestión de preferencias y comparación entre objetos de nuevas clases
Gráficas en Java con JFreeChart	Os mostramos como generar gráficas profesionales, en aplicaciones y servlets, en Java con la librería gratuita JFreeChart

Nota: Los tutoriales mostrados en este Web tienen como objetivo la difusión del conocimiento.

Los contenidos y comentarios de los tutoriales son responsabilidad de sus respectivos autores.

En algún caso se puede hacer referencia a marcas o nombres cuya propiedad y derechos es de sus respectivos dueños. Si algún afectado desea que incorporemos alguna reseña específica, no tiene más que solicitarlo.

Si alguien encuentra algún problema con la información publicada en este Web, rogamos que informe al administrador rcanales@adictosaltrabajo.com para su resolución.

www.AdictosAlTrabajo.com Opimizado 800X600