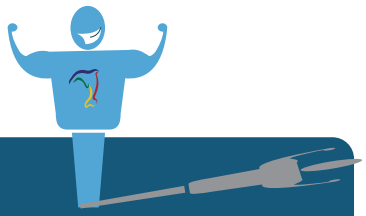
autentia

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...

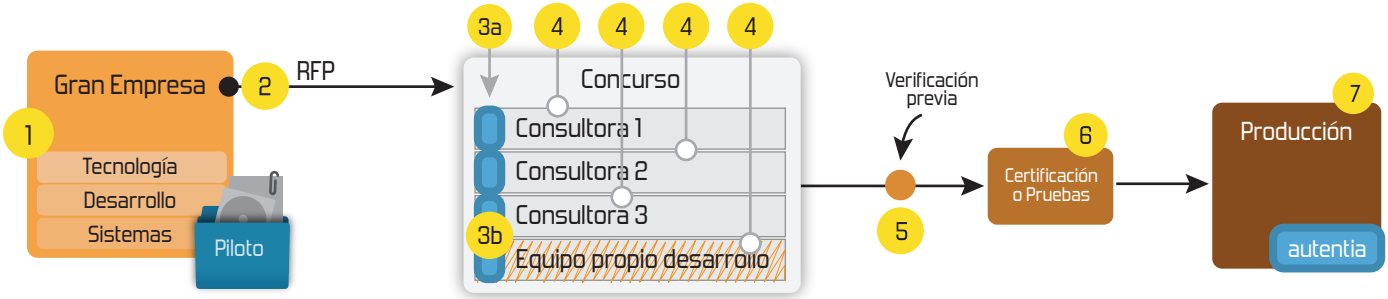
1. Desarrollo de componentes y proyectos a medida



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

adictos altrabajo Powered by autentia
Hosting Patrocinado por enREDados.com
enREDados

Tutorial desarrollado por: Francisco Javier Martínez Páez

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

autentia real business solutions

Descargar este documento en formato PDF POI.pdf

Firma en nuestro libro de Visitas

Free UML 2.0 Design Tool
Visually develop applications with Roundtrip model to code, ERD & DB
www.visual-paradigm.com

Implantación efectiva RUP
con técnicas y experiencia probadas en Latinoamérica
www.itera.com.mx

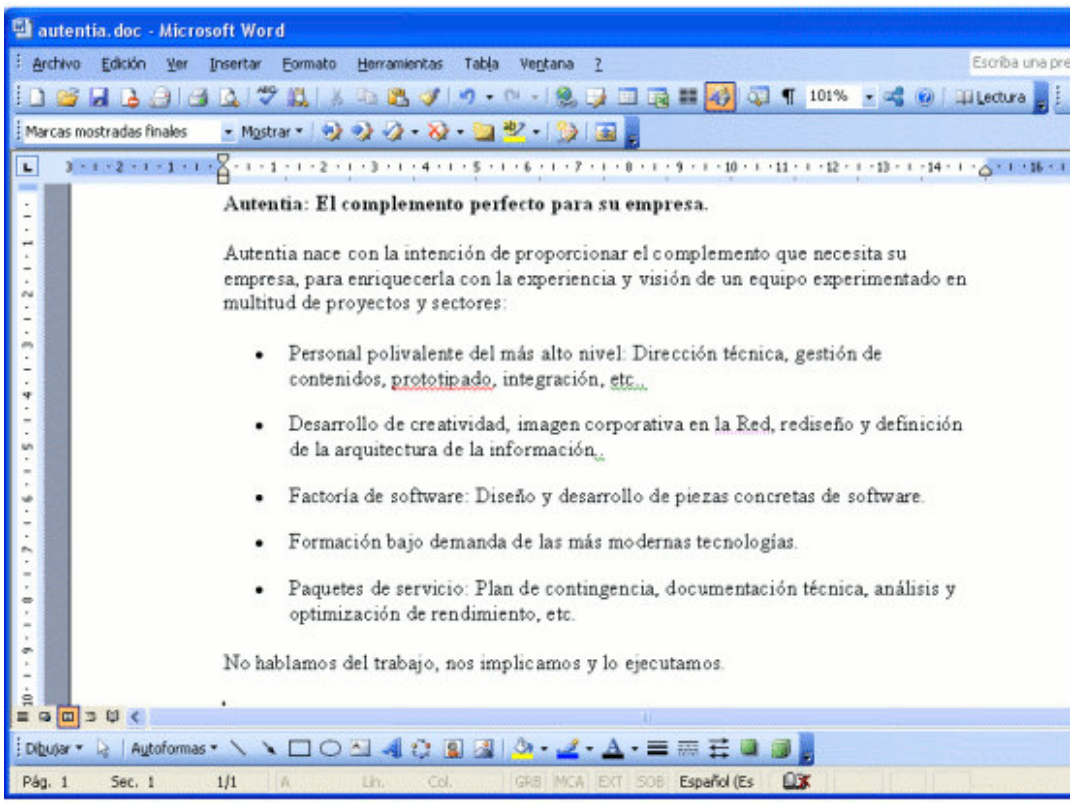
Softeng
Desarrollo soluciones web y gestión Consultoría informática Barcelona.
www.softeng.es

Leer un documento de WORD usando la librería POI de jakarta

Dentro del proyecto jakarta de apache, existe un proyecto muy interesante llamado POI, que pretende generar una serie de AP en java para poder trabajar con documentos en formato de Microsoft. La dirección es: http://jakarta.apache.org/poi/ .

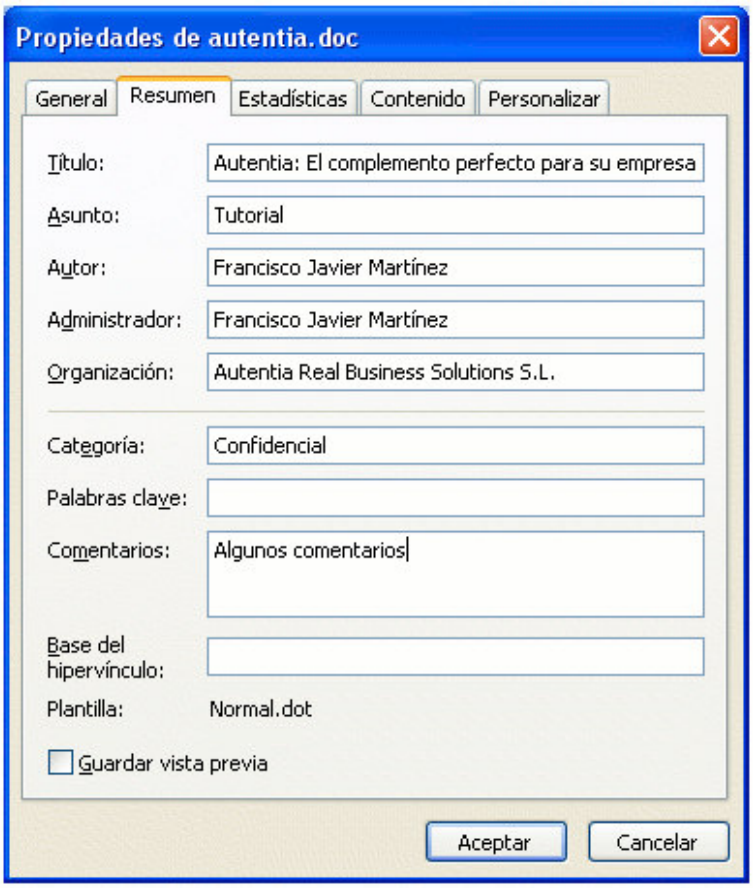
En este tutorial os voy mostrar un ejemplo sencillo para leer un documento de Word y sus características principales.

Vamos primero a hacer lo más complicado, abrir el word y generarnos un documento:



Vamos también a modificar las propiedades del documento:

Menú : Archivo > Propiedades:



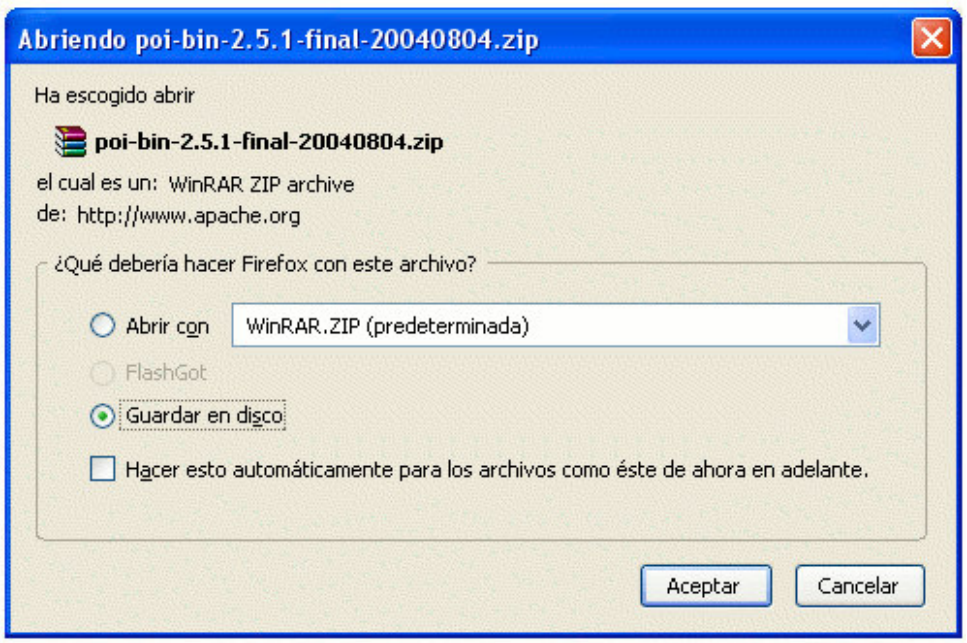
Guardamos el documento con el nombre de **autentia.doc**

Una vez que hemos hecho lo más dificil, vamos a lo sencillo.

Nos descargamos la librería POI:

http://www.apache.org/dist/jakarta/poi/release/bin/

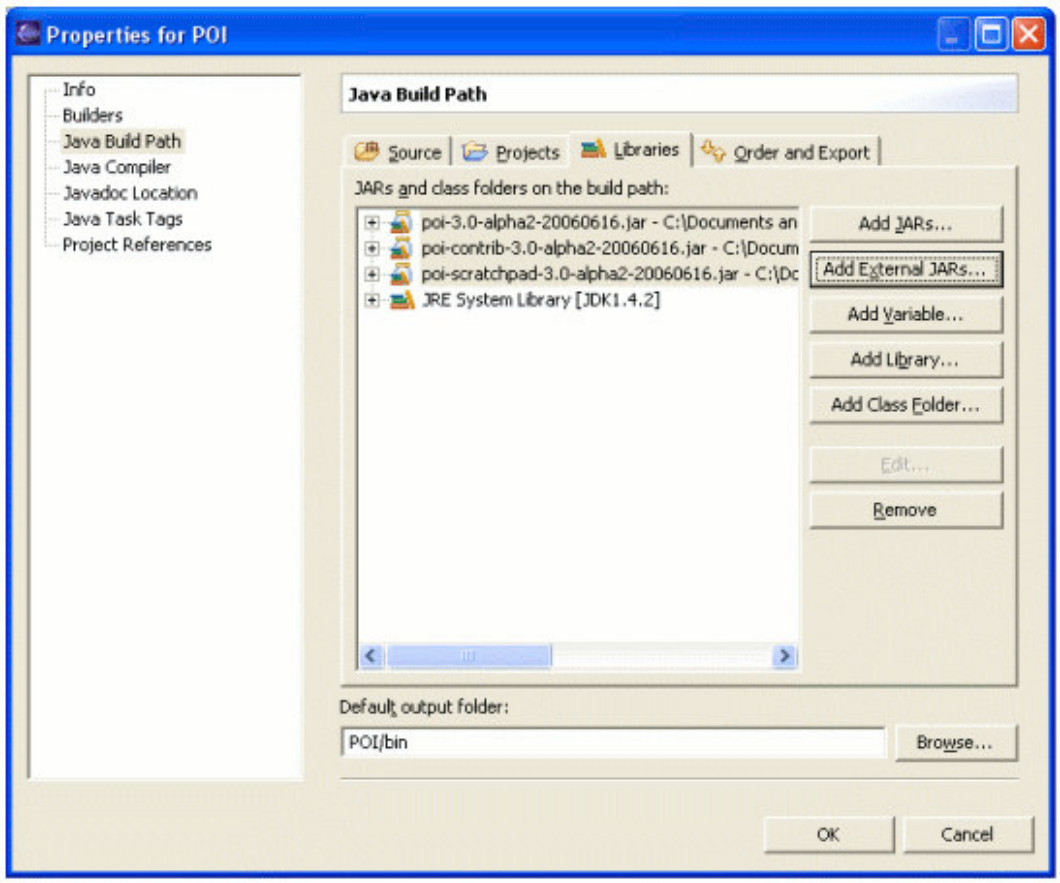
Nos descargamos la versión 2.5.1:



Descomprimimos el fichero en algún lugar de nuestro equipo.

Abrimos el eclipse y creamos un proyecto java nuevo y le llamamos POI.

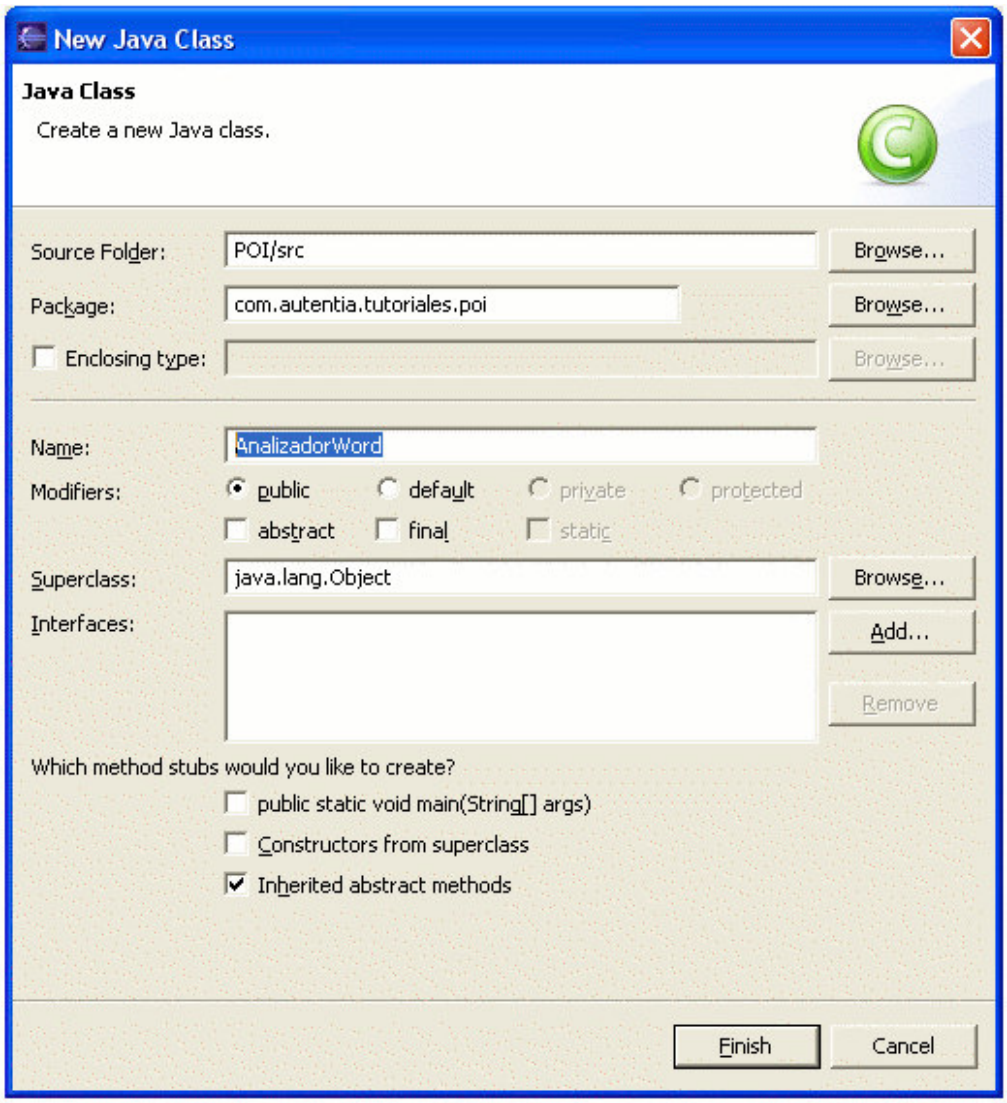
Recordad que debemos añadir las librerías descargadas a nuestro proyecto para compilar:



Nuestro objetivo será analizar el documento de word creado, mostrando algunas de las características del mismo, así como el número de caracteres y el texto del mismo.

Nos creamos una clase que nos servirá de base: AnalizadorWord.

En esta clase almacenaremos las características del documento que queremos analizar:




```
public class AnalizadorWord {

   private int numeroCaracteres;
   private int numeroPalabras;
   private int numeroPaginas;
   private String titulo;
   private String autor;
   private String comentarios;
   private String texto;
   private String nombreFichero;

   public AnalizadorWord(String nombreFichero) {
      this.nombreFichero = this.nombreFichero;
   }

}
```

Recordar generar también los métodos accesores.

Nos creamos ahora una clase que nos servirá para obtener las características estándar de un documento word. Para ello implementaremos el interfaz *org.apache.poi.poifs.eventfilesystem.POIFSReaderListener.*

Le añadiremos un método para pasarle nuestra clase AnalizadorWord.

```
import org.apache.poi.poifs.eventfilesystem.POIFSReaderEvent;
import org.apache.poi.poifs.eventfilesystem.POIFSReaderListener;

public class StandardReaderListener implements POIFSReaderListener {

   private AnalizadorWord datos = null;
```

```
    public void setDatos(AnalizadorWord datos) {
        this.datos = datos;
    }

    public void processPOIFSReaderEvent(POIFSReaderEvent event) {

    }

}
```

Vamos a rellenar el método que debemos implementar del interfaz:

```
    public void processPOIFSReaderEvent(POIFSReaderEvent event) {

    // Clase que almacena las características estándar de un documento.
    SummaryInformation si = null;

    try {
        si = (SummaryInformation)PropertySetFactory.create(event.getStream());
    } catch (Exception ex) {
        throw new RuntimeException ("Property set stream \"" +
            event.getPath() + event.getName() + "\": " + ex);
    }

/* Recogemos los datos que nos interesan y los almacenamos en la clase AnalizadorWord.*/

    datos.setTitulo(si.getTitle());
    datos.setAutor(si.getAuthor());
    datos.setComentarios(si.getComments());
    datos.setNumeroCaracteres(si.getCharCount());
    datos.setNumeroPalabras(si.getWordCount());
    datos.setNumeroPaginas(si.getPageCount());

  /* Usamos la clase WordExtractor para obtener el texto del documento */
  WordExtractor we = null;
  try {
      FileInputStream fis = new FileInputStream(datos.getNombreFichero());
      we = new WordExtractor(fis);
      fis.close();
  } catch (Exception e1) {
      System.out.println("Error->"+e1.toString());
  }
  datos.setTexto(we.getText());
  }
```

Ya tenemos generado el código necesario para obtener los datos buscados. Ahora sólo debemos usarlo.

Vamos a volver a nuestra clase AnalizadorWord y le vamos a crear el método analizar:

```
    public void analizar() throws Exception {
        POIFSReader r1 = new POIFSReader();

        FileInputStream fis=null;

        // Registramos el listener para después analizar la información.

        StandardReaderListener stdReader = new StandardReaderListener();

        stdReader.setDatos(this);

        r1.registerListener(stdReader, "\005SummaryInformation");

        try {

        // Forzamos la lectura del documento.
            fis = new FileInputStream(this.nombreFichero);
        r1.read(fis);
        } catch (Exception e) {
            System.out.println("Error->" + e.toString());
        } finally {
            if(fis!=null)
                fis.close();
        }

    }
```

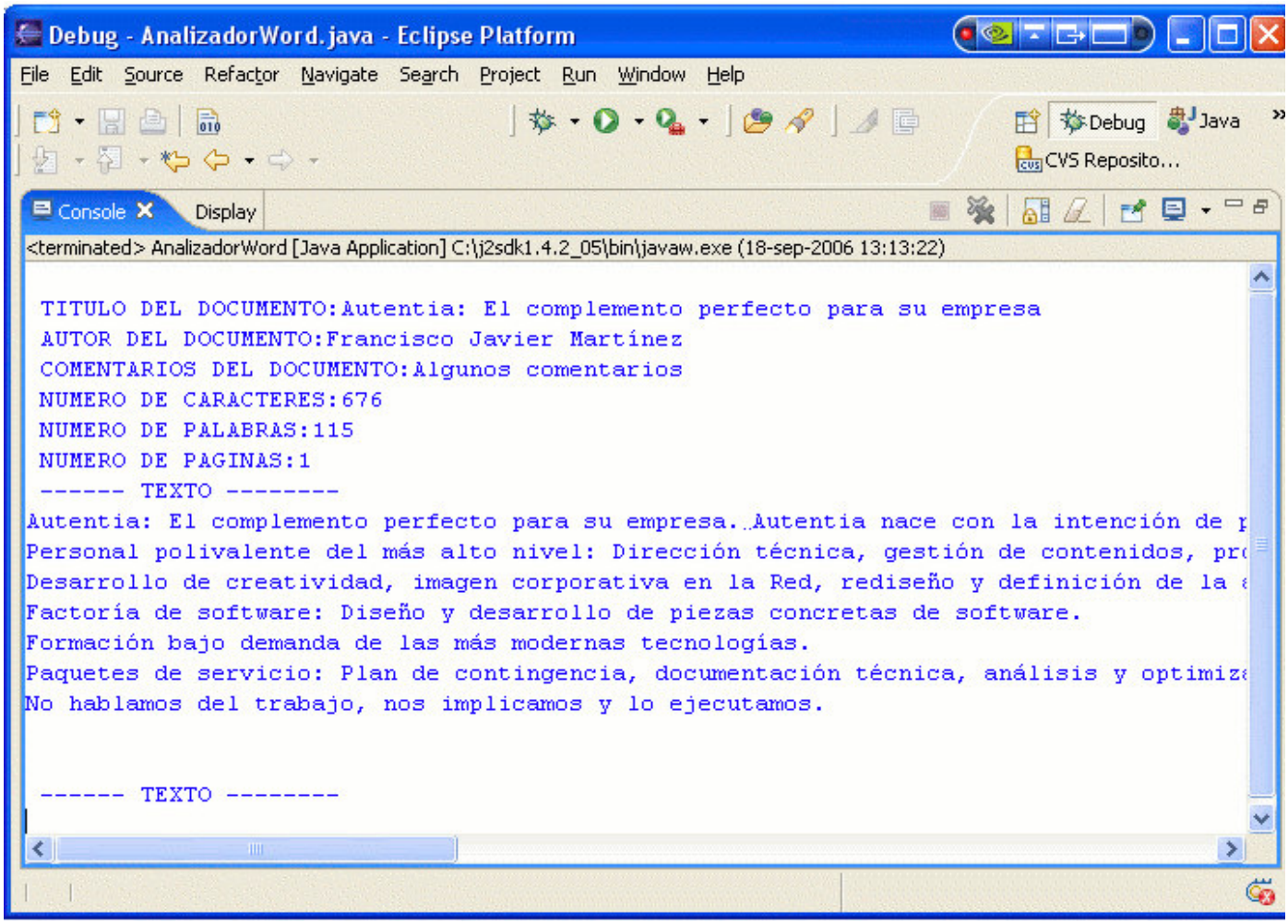
Añadiremos también el método toString() a esta clase para mostrar los datos obtenidos:

```
public String toString() {
   StringBuffer sb = new StringBuffer("");

 sb.append("\n TITULO DEL DOCUMENTO:"+getTitulo());
 sb.append("\n AUTOR DEL DOCUMENTO:"+getAutor());
 sb.append("\n COMENTARIOS DEL DOCUMENTO:"+getComentarios());
sb.append("\n NUMERO DE CARACTERES:"+getNumeroCaracteres());
sb.append("\n NUMERO DE PALABRAS:"+getNumeroPalabras());
sb.append("\n NUMERO DE PAGINAS:"+getNumeroPaginas());
sb.append("\n ------ TEXTO --------");
sb.append("\n");
sb.append(getTexto());
sb.append("\n ------ TEXTO --------");
return sb.toString();

}
```

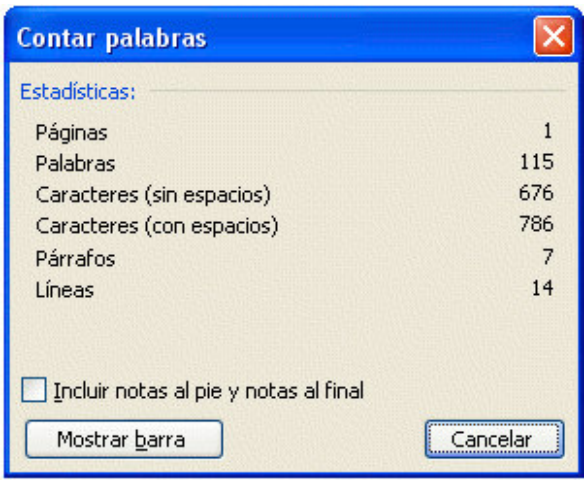
Y por último, invocamos el análisis. Creamos un método main en la clase AnalizadorWord:

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

String filename = "C:\\autentia.doc";
AnalizadorWord analizador = new AnalizadorWord(filename);
try {

analizador.analizar();

} catch (Exception e2) {
System.out.println("Error"+e2.toString());
}

System.out.println(analizador);

}
```

Ejecutamos el código y mostramos el resultado en la consola:



Mostramos la información que nos muestra word:



Espero que os haya servido para una pequeña iniciación en la librería. Podemos usarla no sólo para analizar documento sino para crearlos también, tanto documentos Word como Excel.

Si necesitas ayuda: http://www.autentia.com

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

Autentia S.L. Somos expertos en:
J2EE, Struts, JSF, C++, OOP, UML, UP, Patrones de diseño ..
y muchas otras cosas

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**
Crear pdfs a partir de páginas HTML	Os mostramos como agrupar y convertir páginas HTML que os interesen en documentos PDF a través de herramientas gratuitas
Integración de Access y Word	Os mostramos lo últiles que pueden ser algunas de las herramientas ofimáticas de Microsoft: Word y Access (y más aún integradas) a la hora de generar la documentación necesaria en un proyecto.
Generación de Informes pdf con DataVision	Os mostramos como simplificar la creación de informes un múltiples formatos, incluyendo pdf, con una fantástica herramienta visual (DataVision). Os mostramos como integrarla con Servlets
Struts Jakarta	Cuando se ha trabajado creando aplicaciones Java poco a poco se va viendo la necesidad de normalizar los desarrollo. Uno de los Framework (entornos) más extendidos es Struts
Librería PDFBOX de Java	En este tutorial os mostramos como utilizar algunas de las utilidades de línea de comandos que incorpora la librería Java PDFBOX, para manejar documentos en formato pdf

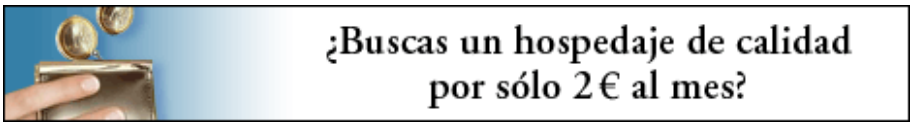
Nota: Los tutoriales mostrados en este Web tienen como objetivo la difusión del conocimiento.

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.

Patrocinados por enredados.com .... Hosting en Castellano con soporte Java/J2EE



www.AdictosAlTrabajo.com Opimizado 800X600