

¿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

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

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

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



[Home](#) | [Quienes Somos](#) | [Empleo](#) | [Tutoriales](#) | [Contacte](#)

<p>Tutorial desarrollado por: Francisco Javier Martínez Páez</p> <p>Puedes encontrarme en Autentia Somos expertos en Java/J2EE Contacta en info@autentia.com</p>	 <p>autentia real business solutions</p>
---	---

Descargar este documento en formato PDF [CSV.pdf](#)

[Firma en nuestro libro de Visitas](#)

[Java Struts o J2EE](#)

¿experiencia en Java Struts o J2EE? Mejora tu carrera. Unete a nosotros
www.soitsa-intesys.com/emple...

[Regression testing java](#)

More than 1600 satisfied customers! Download Product Evaluation.
www.radview.com

[Integrar SOA, WebServices](#)

Sus datos 3270/5250 en J2EE & Java Integrar CICS/IMS con BEA, CRM
www.scort.com

[Master en java](#)

Plataforma especializada en Masters Online, a Distancia y Presencial.
www.emagister.com

[Anuncios Google](#)

[Anunciarse en este s](#)

TRABAJAR CON FICHEROS CSV

Los ejemplos de este tutorial están hechos con el siguiente entorno de desarrollo:

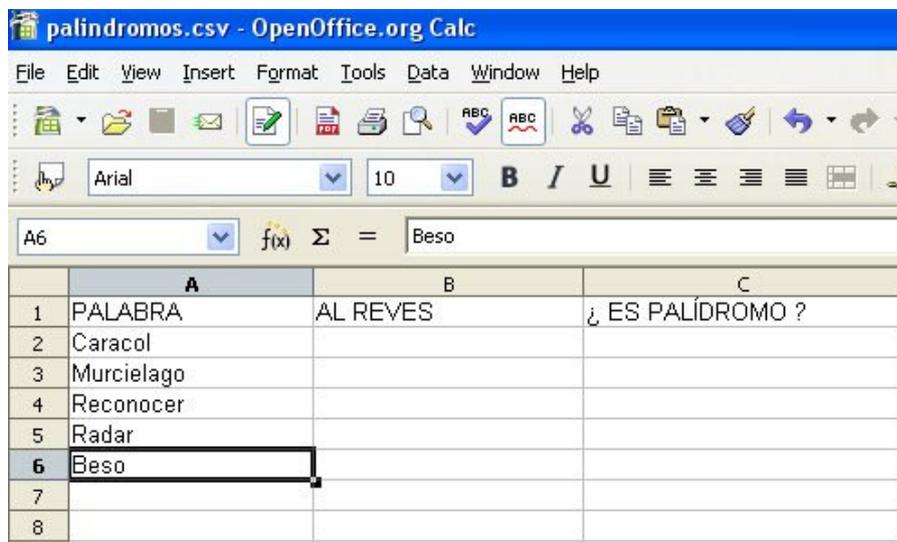
- Jboss Eclipse IDE Milestone 5.
- JDK 1.5

Muchas veces necesitamos trabajar con documentos en forma de filas y columnas, como fuentes de datos. Podemos para ello usar documentos excel y utilizar alguna librería como POI para procesarlos. Sin embargo, a veces no es necesario complicar tanto las cosas y podemos usar formatos de ficheros mas sencillos como los ficheros CSV (comma separated values). En la Wikipedia dan esta definición de CSV:

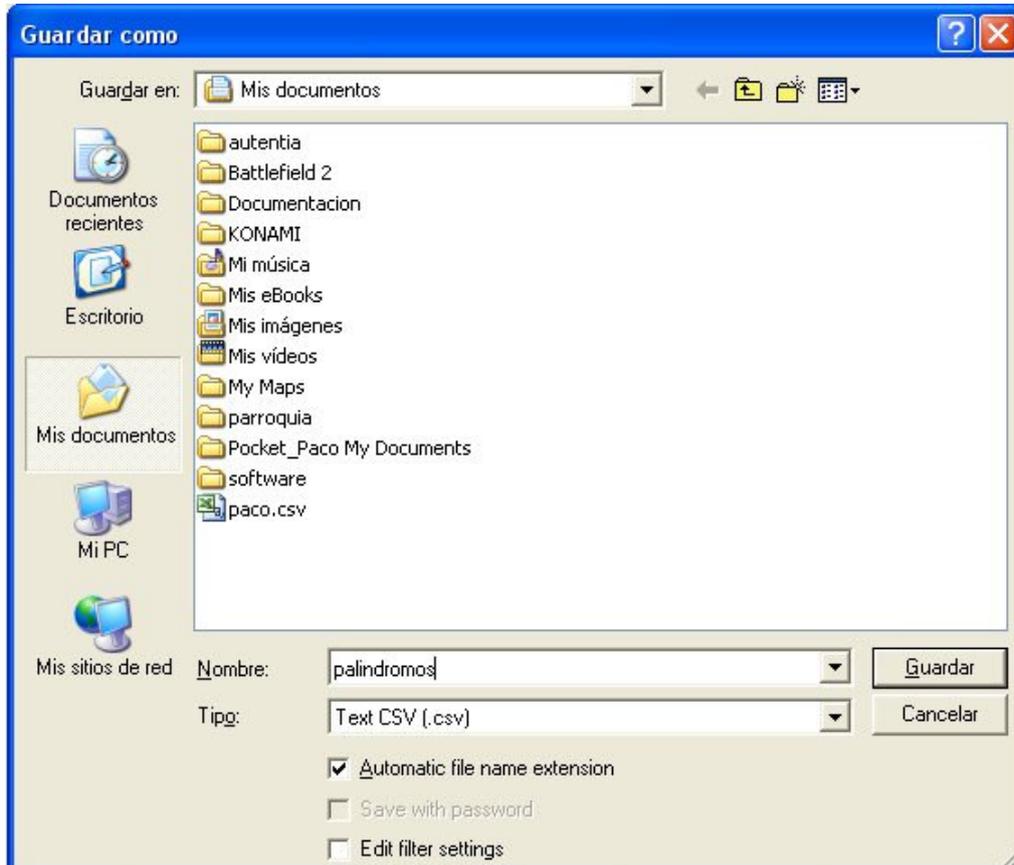
*"Los ficheros **CSV** (del inglés comma-separated values) son un tipo de documento sencillo para representar datos en forma de tabla, en las que las columnas se separan por comas y las filas por saltos de línea. Los campos que contengan una coma, un salto de línea o una comilla doble deben ser encerrados entre comillas dobles."*

Podemos usar cualquier editor de textos para generar este tipo de documentos, incluso también otros programas de hojas de cálculo (excel, openoffice calc...) Además, como norma se usa como delimitador el caracter coma, aunque se puede usar otro caracter como separador.

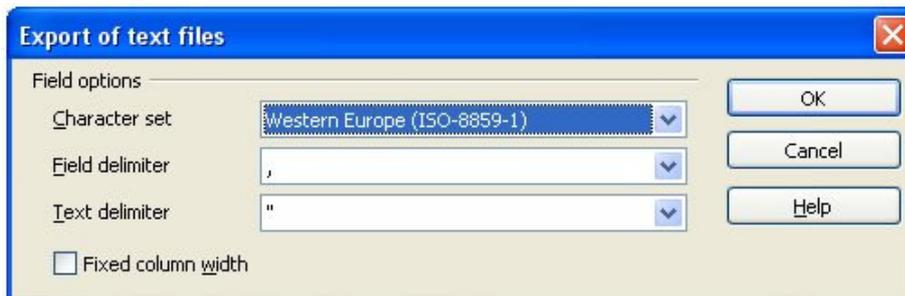
Vamos por lo tanto a generarnos nuestro primer documento CSV. Yo lo voy a hacer con OpenOffice:



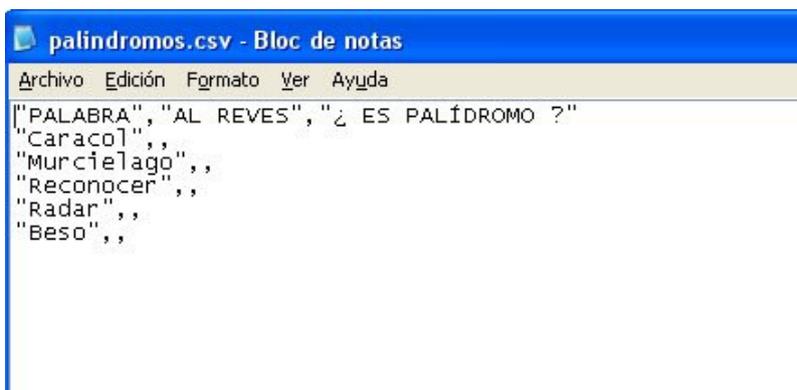
Vamos a guardar el documento como CSV:



Seleccionamos el caracter delimitador y el juego de caracteres:



Una vez guardado, comprobemos lo que es realmente un fichero CSV:



Se puede comprobar que un documento CSV no es más que un fichero de texto, donde se separan las columnas por un caracter delimitador y las filas por un salto de línea.

Sería sencillo generarnos nuestro propio código para tratar este tipo de ficheros, pero...¿Es necesario reinventar la rueda?.

Buscando en google, he encontrado este enlace:

<http://sourceforge.net/projects/javacsv/>

Aqui podemos encontrar una sencilla librería java para tratar este tipo de ficheros. Vamos a probarla:



Descomprimos en algun lugar del disco y encontramos la documentación de la librería y el fichero con los binarios: **csv.jar**

Os recomiendo echar un vistazo a la documentación (index.html), no perderéis mucho tiempo porque son dos clases:

All Classes
[CsvReader](#)
[CsvWriter](#)

Package Class Tree Deprecated Index Help
PREV PACKAGE NEXT PACKAGE

Package com.csvreader

Class Summary

CsvReader	A stream based parser for parsing delimited text data from a file or a stream.
CsvWriter	A stream based writer for writing delimited text data to a file or a stream.

Exception Summary

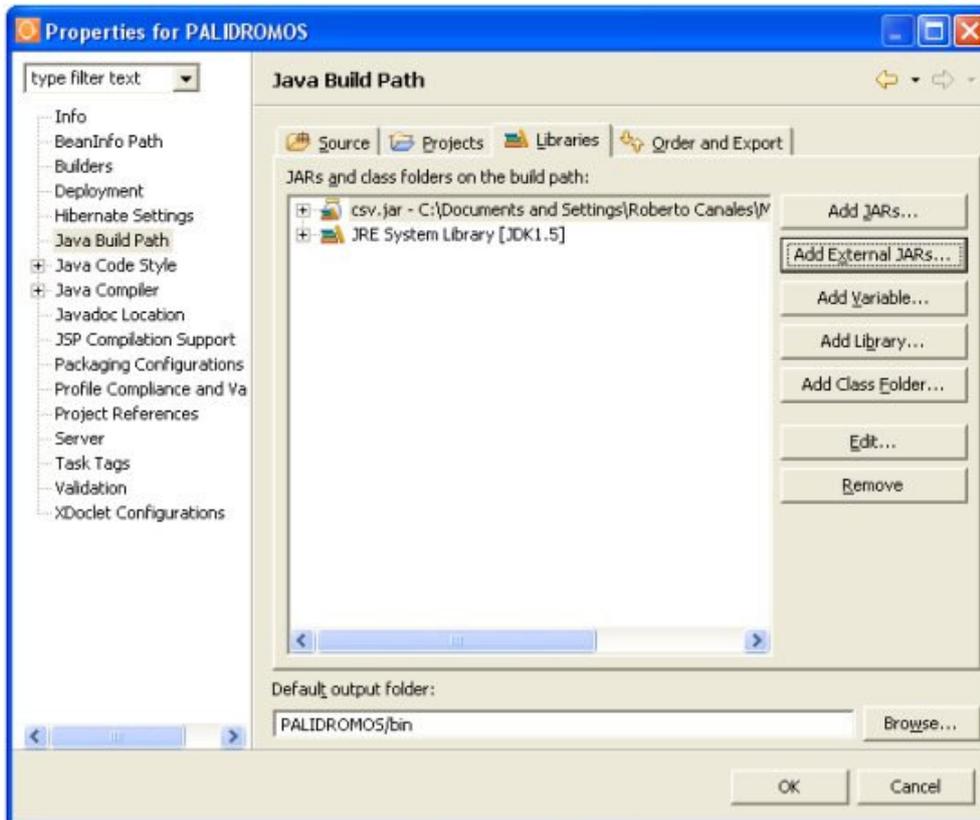
EMPEZANDO A TRABAJAR.

Vamos a empezar a hacer algo interesante. Os propongo el siguiente programa:

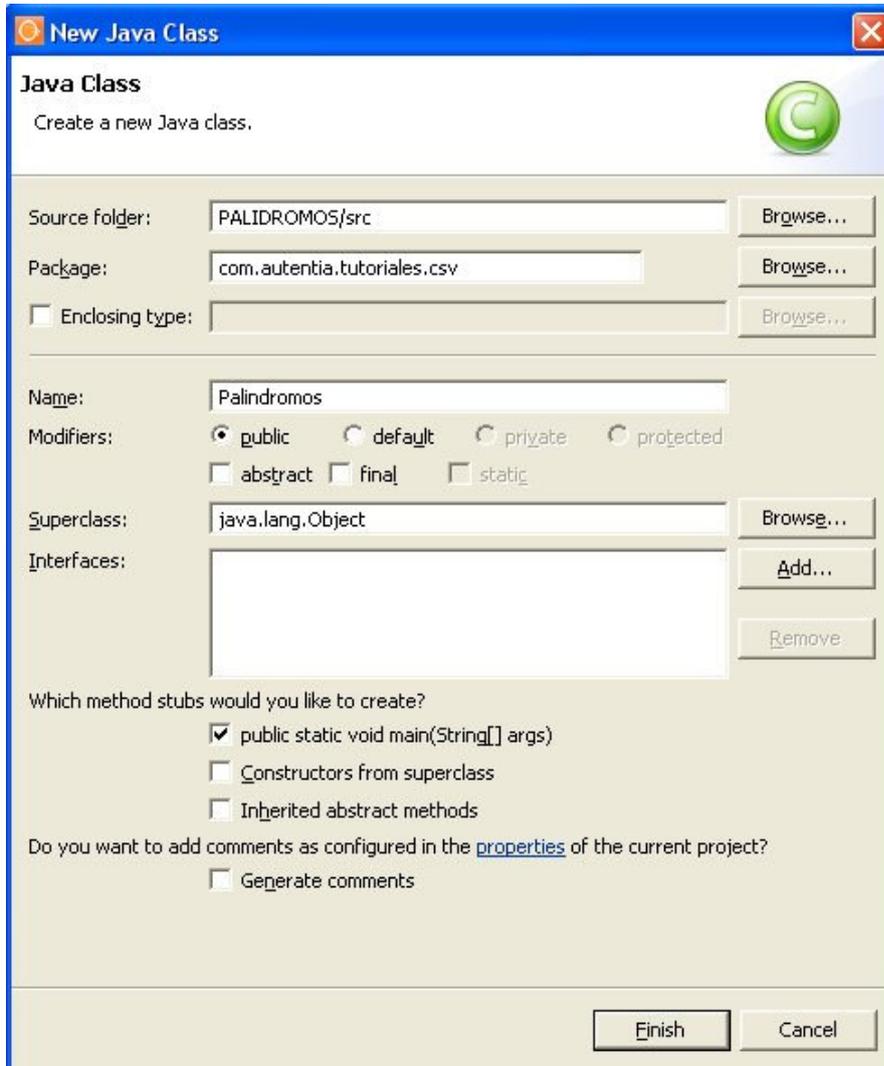
1. Leer el documento CSV creado arriba,
2. Para cada palabra, ponerla al revés y rellenar la segunda columna de cada fila.
3. Comprobar si cada palabra es palíndroma (se lee igual de derecha a izquierda que de izquierda a derecha) y rellenar la tercera columna con el resultado.

Para empezar, abriremos eclipse y crearemos un proyecto java nuevo (yo le he llamado PALINDROMOS).

Añadiremos al CLASSPATH de nuestro proyecto la librería javacvs (cvs.jar):



Nos crearemos una clase que nos sirva de base para nuestro programa:
com.autentia.tutoriales.csv.Palindromos:



Nos vamos a crear otra clase que representará un fila en el fichero CSV. La vamos a denominar VOPalindromo, a continuación os muestro el código:

```
public class VOPalindromo implements Serializable {
    private String palabra;
    private String reves;
    private boolean palindromo;

    public String getPalabra() {
        return palabra;
    }
    public void setPalabra(String palabra) {
        this.palabra = palabra;
    }
    public boolean isPalindromo() {
        return palindromo;
    }
    public void setPalindromo(boolean palindromo) {
        this.palindromo = palindromo;
    }
    public String getReves() {
        return reves;
    }
    public void setReves(String reves) {
        this.reves = reves;
    }
}
}
```

Vamos a escribir ahora un método en la clase Palindromos que lea todas las palabras del fichero CSV, calcule si cada palabra es palíndromo (capicúa) y las almacene en memoria:

```
/**
 * Lee el fichero CSV llamado pathFichero, usando como delimitador de
 * columnas: "delimitador" y crea una lista en memoria de
 * Objetos de tipo VOPalindromo
 * @param pathFichero
 * @param delimitador
 */
```

```

    * @return
    */

    public List leePalindromos(String pathFichero, String delimitador)
    throws Exception {

        CsvReader cvsReader = null;

        try {
            File fichero = new File(pathFichero);
            FileReader freader = new FileReader(fichero);
            cvsReader = new CsvReader(freader,delimitador.charAt(0));
            String[] headers = null;
            List listaRegistros = new ArrayList();
            // Leemos las cabeceras del fichero (primera fila).
            if(cvsReader.readHeaders()) {
                headers = cvsReader.getHeaders();
                System.out.println("----- CABECERAS DEL FICHERO -----");
                for(int i=0;i<headers.length;i++) {
                    System.out.println(headers[i]);
                }
                System.out.println("-----");
            }
            // Leemos los registros
            System.out.println("----- PALABRAS DEL FICHERO -----");
            while(cvsReader.readRecord()) {
                // Podemos usar get con el nombre de la cabecera o por posición
                String palabra = cvsReader.get(headers[0]);

                // Usamos StringBuffer para darle la vuelta a la palabra
                StringBuffer aux = new StringBuffer(palabra);
                String revés = aux.reverse().toString();

                // Almacenamos el objeto VOPalindromo
                VOPalindromo palindromo = new VOPalindromo();
                palindromo.setPalabra(palabra);
                palindromo.setReves(revés);

                // Calculamos si es palindromo y lo almacenamos.
                palindromo.setPalindromo(palabra.equalsIgnoreCase(revés));
                listaRegistros.add(palindromo);
                // Mostramos por pantalla:
                System.out.println("-" + palabra + " == " + revés + (palindromo.isPalindromo()?" : SI":" : NO"));
            }
            System.out.println("-----");

            return listaRegistros;
        } catch(Exception e) {
            throw e;
        } finally {
            if(cvsReader!=null) {
                cvsReader.close();
            }
        }
    }
}

```

Invoquemos al método desde main:

```

public static void main(String[] args) {

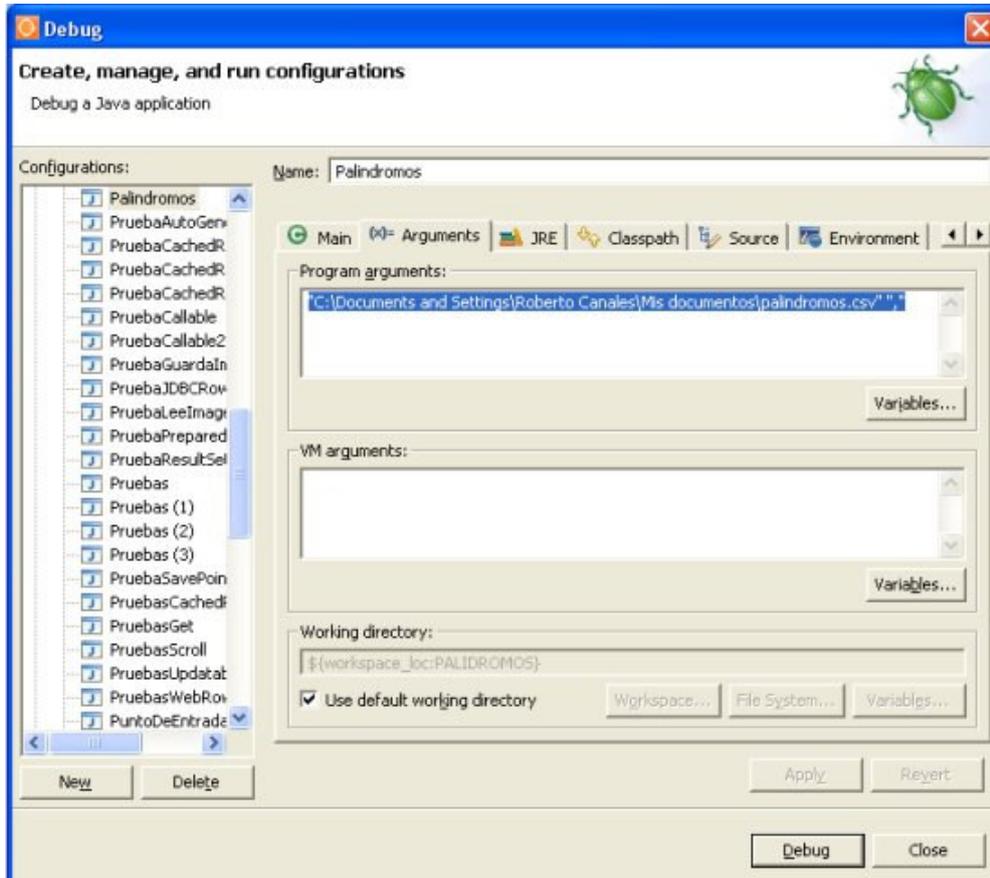
    // Obtenemos el fichero y el delimitador
    String fichero = args[0];
    String delim = args[1];

    Palindromos palindromo = new Palindromos();

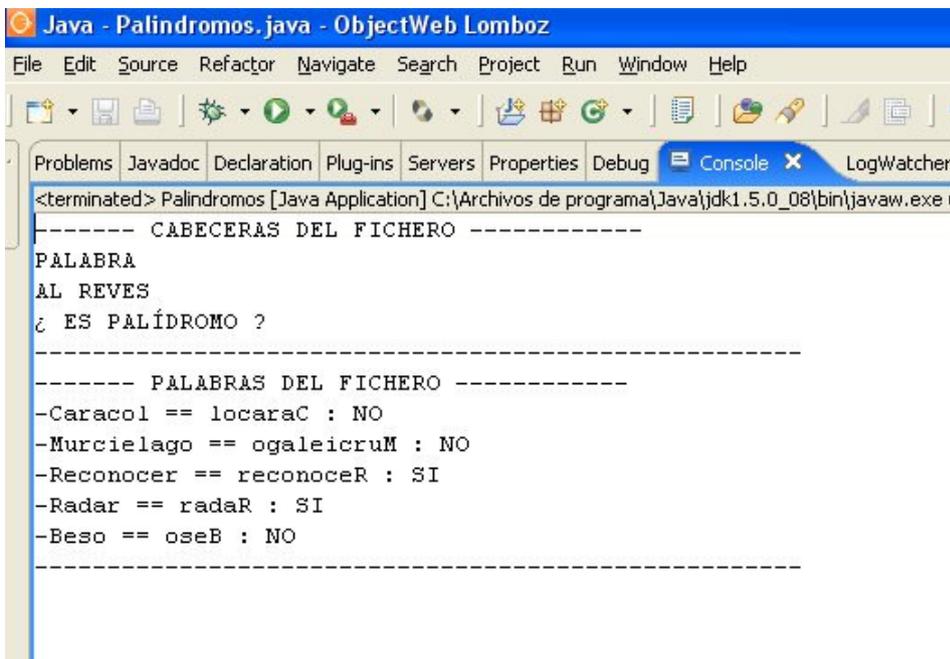
    List listaPalindromos = null;
    try {
        listaPalindromos=palindromo.leePalindromos(fichero,delim);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

Ejecutemos el código (no olvidar pasárle los parametros a main):



Invocamos y mostramos la consola:



Bueno, tan sólo nos falta la parte de almacenar los resultados en el fichero CSV. Nos crearemos otro método en la clase Palindromos al que llamaremos: escribePalindromos

```
public void escribePalindromos(List listaPalindromos, String pathFichero, String delimitador) throws Exception {
    CsvWriter writerscsv = null;

    try {
        File fichero = new File(pathFichero);
        FileWriter fwriter = new FileWriter(fichero);

        // Creamos la clase que nos permite escribir en el fichero CSV.
        writerscsv = new CsvWriter(fwriter, delimitador.charAt(0));

        // Escribimos las cabeceras.
        writerscsv.write("PALABRA");
    }
}
```

```

writercsv.write("AL REVES");
writercsv.write("¿ ES PALÍNDROMO ?");
writercsv.endRecord();

// Escribimos los resultados.
for(int i=0;i<listaPalindromos.size();i++) {
    VOPalindromo palin = (VOPalindromo)listaPalindromos.get(i);
    writercsv.write(palin.getPalabra());
writercsv.write(palin.getReves());
writercsv.write((palin.isPalindromo())?"SI":"NO");
writercsv.endRecord();
}

} catch(Exception e) {
    throw e;
} finally {
    if(writercsv!=null) {
writercsv.close();
    }
}
}
}

```

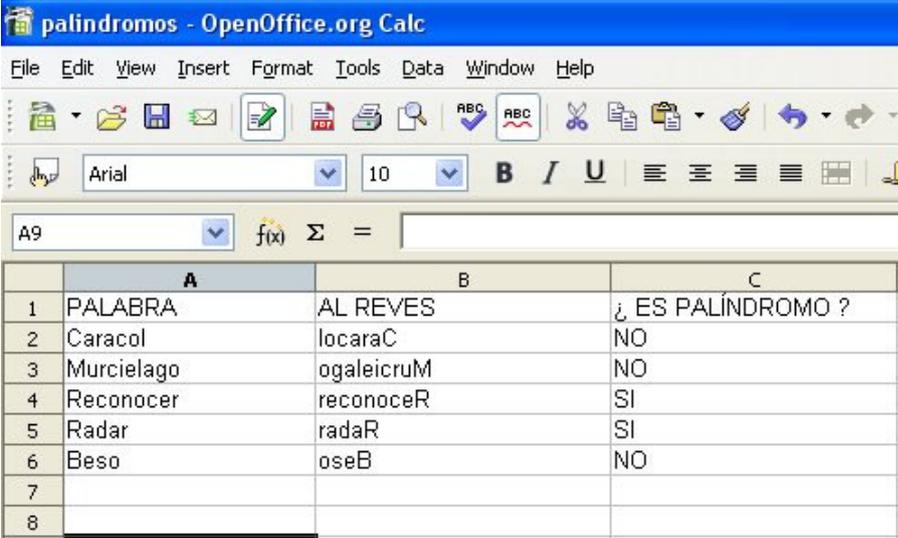
Lo invocamos desde main:

```

....
....
try {
    palindromo.escribePalindromos(listaPalindromos,fichero,delim);
} catch (Exception e) {
    e.printStackTrace();
}
}

```

Vamos a comprobar el resultado en el fichero:



	A	B	C
1	PALABRA	AL REVES	¿ ES PALÍNDROMO ?
2	Caracol	locaraC	NO
3	Murcielago	ogaleicruM	NO
4	Reconocer	reconoceR	SI
5	Radar	radaR	SI
6	Beso	oseB	NO
7			
8			

Ha sido sencillo ¿ verdad ?.

Os comento que hay gran cantidad de librerías para tratar ficheros CSV, esta es una más y funciona bien. También podéis generaros vuestra propia librería, ya que los ficheros CSV como habéis visto son muy sencillos de tratar.

Si necesitáis ayuda, ya sabéis donde encontrarlos: <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	
<i>e-mail</i>	<input type="text"/>
	<input type="button" value="Enviar"/>

Otros Tutoriales Recomendados ([También ver todos](#))

Nombre Corto	Descripción
Optimización de Serialización Java	Os mostramos una sencilla técnica para mejorar el rendimiento de la serialización de objetos en Java, a través de Streams asociados a buffers en memoria.
Introducción a JavaHelp	En este tutorial os mostramos los primeros pasos para utilizar el API JavaHelp de cara a la generación de ayudas con Java
Integración de Struts y eclipse	Alejandro Perez nos enseña como construir un entorno de alta eficiencia de desarrollo on Struts a través de plugins de eclipse
Analizar ejecución de programa Java	Os mostramos como investigar el comportamiento de vuestros programas Java, en ejecución, a través del profiling.
Novedades en Java 1.5	Ya está disponible la versión Beta del J2SDK 1.5. Os mostramos algunas de las nuevas características introducidas en el lenguaje Java: Clases genéricas, enumeraciones, bucles simplificados, etc.
Optimización Java con Eclipse Profiler Plugin	Alejandro Pérez nos enseña como analizar el rendimiento de nuestras aplicaciones con Eclipse Profiler Plugin.
Añadir Scripting a programas Java	Os mostramos como añadir a vuestras aplicaciones soporte de Scripting, gracias a BeansShell. Esto nos permite ejecutar Java generado dinámicamente.
Generación .exe con ficheros Java	Os mostramos como, utilizando la herramienta gratuita JSmooth, podemos generar un fichero .exe a partir de nuestros programas Java

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 Optimizado 800X600