

¿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: Juan Alonso Ramos (Autentia) es consultor tecnológico de desarrollo de proyectos informáticos. Contacta en jalonso@autentia.com</p>	<p>www.adictosaltrabajo.com es el Web de difusión de conocimiento de www.autentia.com</p>  <p>autentia real business solutions</p> <p>Catálogo de cursos</p>
--	---

Descargar este documento en formato PDF [tomcatUTF8.pdf](#)

[Firma en nuestro libro de Visitas <----->](#) [Asociarme al grupo AdictosAlTrabajo en eConozco](#)

[Master Java Certificado](#)

Temario Actualizado-UML-JSF-AJAX Trabajo Garantizado-Bolsa de Empleo
www.exes.es

[Noaris Innovación](#)

Desarrollo software y web 2.0
 Servicios informáticos a medida
www.noaris.com

[Java Struts o J2EE](#)

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

[More Design, Less Debug](#)

High productivity real-time Java: Predictable, reliable, elegant.
www.javelocity.com

[Anuncios Goooooogle](#)

[Anunciarse en este sitio](#)

Configuración de una Aplicación Web en Tomcat con codificación UTF-8

15/02/2007

Índice de Contenidos

- 1. Internacionalización**
- 2. Problemas en la codificación de caracteres**
- 3. Soporte para codificación de caracteres en UTF-8**
 - 3.1 Configuración del cliente**
 - 3.2 Configuración del Tomcat**
- 4. Conclusión**

1. Internacionalización

Muy a menudo nos encontramos con el problema de tener que traducir una aplicación web a un idioma que contiene caracteres diferentes al que se tenía actualmente definido. Si inicialmente no se contempló esta posibilidad ya que se pensó que nunca se debería traducir a otro idioma nos encontramos con un grave problema. Llegados a este punto y teniendo en cuenta que no podemos dar marcha atrás para volver a definir los requisitos iniciales, nos acarreará mucho tiempo y esfuerzo la modificación de todas las páginas para dotarlas de soporte internacional. Por todo esto es importante tener en cuenta desde el principio la codificación de caracteres que tendrá nuestra aplicación web.

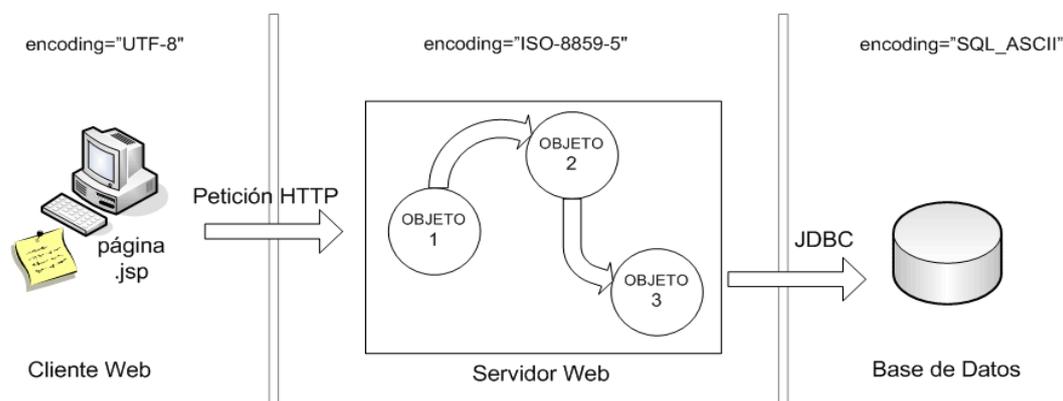
Puedes leer más sobre internacionalización [aquí](#)

2. Problemas en la codificación de caracteres

La codificación de caracteres en una aplicación web es un tema que suele dar bastantes problemas a los desarrolladores ya que no está del todo claro qué formato hay que seguir y cuál es el que suelen utilizar los servidores web, bases de datos o navegadores web.

Cuando se recibe o envía texto a una página web, lo que en un principio parece algo muy simple en realidad no lo es tanto ya que para que ese texto sea interpretado y escrito en pantalla por el navegador web y manejado por el servidor web hasta llegar a la base de datos, en ocasiones "pasa por muchas manos".

A menudo ocurre que se tiene configurado un formato de codificación de caracteres diferente en el navegador del cliente, otro para el servidor web y otro en la base de datos. Hay que tener especial cuidado con esto ya que si tenemos una aplicación web como la del ejemplo de abajo, si el cliente envía los caracteres en UTF-8, el servidor web los espera en ISO-8859-5 y en la base de datos deben guardarse como SQL_ASCII no hay forma de que la aplicación funcione correctamente.



3. Soporte para codificación de caracteres en UTF-8

UTF-8 es un estándar de codificación de longitud variable que utiliza caracteres Unicode y que soporta la mayoría de los lenguajes del mundo por lo que es muy recomendable. Con UTF-8 no tendremos que preocuparnos de escribir el código HTML de cada carácter especial (ñ, vocales acentuadas ...) como se hace algunas veces ya que se establece internamente una referencia entre el carácter introducido y su código de representación UTF-8.

Para dotar a nuestras páginas con soporte para codificación de caracteres en UTF-8 habrá que seguir los siguientes pasos:

3.1 Configuración del cliente

1. En la primera línea de cada JSP se debe indicar el encoding de la página a utilizar de manera que cuando se envía la petición por HTTP entre el cliente y el servidor, el navegador sepa en qué formato debe enviar los caracteres para que el servidor los procese correctamente.

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
```

3.2 Configuración del Tomcat

2. En la parte del servidor también tenemos que especificar el encoding a utilizar. En Tomcat por defecto se especifica el formato de codificación ISO-8859-1. Para cambiar la codificación tenemos que modificar el archivo *server.xml* que se encuentra en *DIRECTORIO_INSTALACION_TOMCAT\conf\server.xml*.

Añadimos el atributo **URIEncoding="UTF-8"** en la etiqueta `<Connector port="8080" ... />` ya que es aquí donde se configura el comportamiento que sigue Tomcat cuando recibe peticiones a través del puerto 8080.

```
<Connector port="8080" maxHttpHeaderSize="8192" maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
enableLookups="false" redirectPort="8443" acceptCount="100" connectionTimeout="20000" disableUploadTimeout="true"
URIEncoding="UTF-8"/>
```

3. Podría ser necesario especificar al contenedor de servlets Catalina la codificación a utilizar indicándole en el arranque que los caracteres siempre van en UTF-8. Para ello editamos el fichero *catalina.bat* (Windows) o el *catalina.sh* (Linux) y añadimos lo siguiente:

```
(Windows) set CATALINA_OPTS=-Dfile.encoding="UTF-8"
```

```
(Linux) export CATALINA_OPTS=-Dfile.encoding="UTF-8"
```

A la vez indicamos las opciones del arranque de la JVM en UTF-8 por lo que añadimos al mismo fichero.

```
(Windows) set JAVA_OPTS=%JAVA_OPTS% -Dfile.encoding="UTF-8"
```

```
(Linux) export JAVA_OPTS=%JAVA_OPTS% -Dfile.encoding="UTF-8"
```

4. Por último debemos forzar a que la request soporte los caracteres en UTF-8 ya que cuando se envía la petición por post, el servidor Tomcat no es capaz de saber cuál es el formato de codificación del cliente. Como posible solución está crear un filtro para que todas las peticiones pasen por él y donde se especifica que la request vaya en codificación UTF-8.

```
package com.autentia.tutoriales;

import java.io.IOException;

import javax.servlet.Filter;

import javax.servlet.FilterChain;

import javax.servlet.FilterConfig;

import javax.servlet.ServletException;

import javax.servlet.ServletRequest;
```

```

import javax.servlet.ServletResponse;

/**
 * Filtro para que la aplicación acepte codificación en formato UTF-8
 */

public class UTF8Filter implements Filter {

    private String encoding;

    /**
     * Recogemos el tipo de codificación definido en el web.xml
     * Si no se hubiera especificado ninguno se toma "UTF-8" por defecto
     */

    public void init( FilterConfig filterConfig ) throws ServletException {

        encoding = filterConfig.getInitParameter( "requestEncoding" );

        if( encoding == null ) {

            encoding = "UTF-8";

        }

    }

    /**
     * Metemos en la request el formato de codificación UTF-8
     */

    public void doFilter( ServletRequest request, ServletResponse response, FilterChain fc )

        throws IOException, ServletException {

        request.setCharacterEncoding( encoding );

        fc.doFilter( request, response );

    }

    public void destroy() {}

}

```

Añadimos el filtro al *web.xml* antes de la declaración de cualquier servlet.

```

<!-- Filtro para codificación de caracteres en formato UTF-8 -->

<filter>

    <filter-name>UTF8Filter</filter-name>

    <filter-class>com.autentia.tutoriales.UTF8Filter</filter-class>

    <init-param>

        <param-name>requestEncoding</param-name>

        <param-value>UTF-8</param-value>

    </init-param>

</filter>

<filter-mapping>

    <filter-name>UTF8Filter</filter-name>

    <url-pattern>/*</url-pattern>

</filter-mapping>

```

4. Conclusión

Como podéis ver la codificación de caracteres no es nada trivial y afecta en muchas partes de nuestra aplicación web por lo que es algo a tener muy en cuenta a la hora de planificar un proyecto. Para evitarnos muchos problemas sobre todo cuando tenemos mucho código escrito hay que tener muy presente que nuestras páginas web serán accedidas desde diferentes sitios con distintos idiomas. Todo esto queda resuelto eligiendo un estándar en la codificación de los caracteres, en este caso UTF-8.



This work is licensed under a [Creative Commons Attribution-NonCommercial-No Derivative Works 2.5 License](https://creativecommons.org/licenses/by-nc-nd/2.5/).
[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

Formación en nuevas tecnologías

[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	<input type="text"/>
	<input type="button" value="Enviar"/>

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

Nombre Corto	Descripción
Activar soporte SSL en Tomcat	Os mostramos como activar el acceso SSL en Tomcat, utilizando certificados generados por Keygen (java)
Monitorización de Tomcat con JMX	Este tutorial nos cuenta, paso a paso, como monitorizar Tomcat usando su implementación JMX
Como configurar Tomcat + IIS	Os mostramos como activar el filtro ISAPI de IIS para conectarlo con Tomcat. De este modo el servidor Web IIS 5.5 de Microsoft servirá las páginas estáticas y Tomcat los JSPs y Servlets dinámicos
Generador automático de Webservices	Os mostramos como crear un servicio Web a partir de una clases, gracias a generadores automáticos de código y NetBeans
Creación de Un Web básico	Como como construir nuestras primeras páginas HTML
Seguridad en Tomcat	Os mostramos como proteger de un modo básico el acceso a recursos dentro de nuestro servidor de componentes Tomcat
Múltiples hosts en Tomcat	En este tutorial se va a presentar un ejemplo práctico para ver como Tomcat soporta el alojamiento de múltiples hosts
Filtros de Servlets en Tomcat	En este tutorial os enseñamos la técnica (poco conocida) del encadenamiento de filtros en la activación de servlets, dentro del entorno Tomcat
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.
Plantear una aplicación Web y Struts	Os mostramos un posible modo de plantear una aplicación Web (análisis) y darla forma. El FrameWork utilizado es struts y tratamos de identificar qué depende de este FrameWork y qué no.

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](#)



**¿Buscas un hospedaje de calidad
con soporte JAVA?**

www.AdictosAlTrabajo.com Optimizado 800X600