

# ¿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)



Estás en: Inicio » Tutoriales » Spring Security, Single Sign-On bajo plataformas Windows con Active Directo...



DESARROLLADO POR:  
Ángel García Jerez

Consultor tecnológico de desarrollo de proyectos informáticos. Co-autor del libro "Actualización y mantenimiento del PC (Edición de 2010) publicado por Anaya Multimedia  
Ingeniero Técnico en Informática de Sistemas e Ingeniero en Informática (premio al mejor expediente de su promoción)  
Puedes encontrarme en Autentia: Ofrecemos servicios de soporte a desarrollo, factoría y formación  
Somos expertos en Java/J2EE



Fecha de publicación del tutorial: 2011-07-26

Share |

Regístrate para votar



# SPRING SECURITY, SINGLE SIGN-ON BAJO PLATAFORMAS WINDOWS CON ACTIVE DIRECTORY Y KERBEROS V5

1. Introducción
2. Entorno
3. Spnego: Integrated Windows Authentication
4. Configuración del Active Directory
  - 4.1. Creación de un servidor de dominio
  - 4.2. Agregar el servidor aplicaciones al Active Directory
  - 4.3. Creación de un usuario para mapear con nuestro servidor
  - 4.4. Crear el fichero keytab para el servidor
5. Integrar Spnego en nuestra aplicación web
  - 5.1. Creación del fichero de configuración de Spring Security
  - 5.2. Añadir Spring Security al web.xml
  - 5.3. Añadir el fichero web.keytab al classpath de la aplicación
6. Configuración del Servidor Tomcat
7. Configurando el navegador
  - 7.1. Spnego en Firefox
  - 7.2. Spnego en IExplorer
8. Testeando la aplicación
9. Código fuente
10. Conclusiones

## 1. Introducción

Desde la aparición de Spring el desarrollo de aplicaciones ha evolucionado considerablemente. Su principal proyecto y por el que todos lo conocemos es por su contenedor de control de inversión. No sólo se queda aquí, Spring ha desarrollado otros proyectos no menos importantes que han seguido mejorando y simplificando el desarrollo de nuestras aplicaciones.

Uno de estos proyectos es Spring Security; se trata de un módulo no intrusivo que permite de forma más o menos sencilla añadir funcionalidades de control de acceso y autenticación a nuestras aplicaciones web. Por simplificarlo al máximo se trata de un conjunto de filtros que se ejecutan en cadena antes de acceder a los recursos de nuestra aplicación con el objetivo de securizarlos.

En este tutorial veremos una de las extensiones que ofrece Spring Security para poder integrar en nuestra aplicación autenticación Single Sign On a través de un Active Directory y Kerberos V5.

## 2. Entorno

Entorno utilizado para escribir este tutorial:

- **Hardware:** Mac Book Pro (Core 2 Duo 2,8 Ghz, 4 GB RAM, 500 GB)

Catálogo de servicios Autentia



Últimas Noticias

- VII Autentia Cycling Day
- Autentia patrocina la charla sobre Java SE 7 en Madrid
- Alfresco Day 2011
- XVII Charla Autentia - Grails - Vídeos y Material
- ¡¡¡ 15 millones de descargas de tutoriales !!!

Histórico de NOTICIAS

Últimos Tutoriales

- Spring MVC: acceder a las propiedades de un fichero desde una JSP con Expression Language (EL)
- Framework Scala liftweb
- Trabajando con JAXB y Eclipse
- Configurar Spring Security 3.1 para autenticarse contra un Active Directory
- Migración a ICEfaces 2.0

Últimos Tutoriales del Autor

- Configuración de aplicaciones multientorno con Maven
- Notificación de eventos en Nut
- Nut - Network UPS Tools
- Awstats - Herramienta de generación de estadísticas.
- JBoss autenticación basada en certificados cliente

Síguenos a través de:



Últimas ofertas de empleo



- **Sistema Operativo:** Lion 10.7.0
- **Active Directory:** Windows 2003 Server
- **JDK:** 1.6
- **Maven:** 3.0.3
- **Spring:** 3.0.1-RELEASE
- **Spnego:** 1.0.0-M2

### 3. Spnego: Integrated Windows Authentication

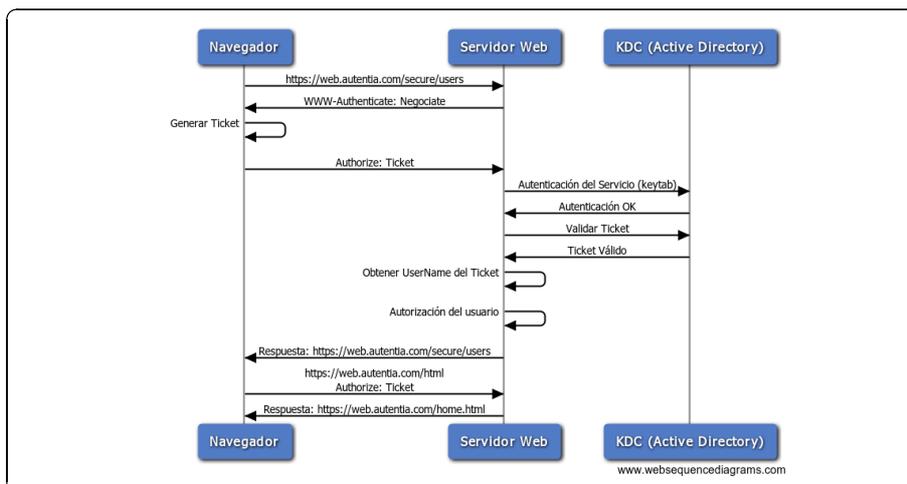
En la mayoría de las grandes empresas que utilizan una infraestructura cliente basada en Windows es relativamente habitual que sus aplicaciones de Intranet utilicen en el proceso de autenticación las credenciales de inicio de sesión en el dominio. De esta manera centralizan el control de acceso a las aplicaciones y mejoran la experiencia de usuario evitando la autenticación en todas aquellas aplicaciones que se accedan.

La plataforma Windows utiliza actualmente dos protocolos de autenticación: NTLM y Kerberos, ambos soportados por Active Directory. Aunque en la actualidad se siguen utilizando los dos, NTLM está deprecado por Microsoft que recomienda el uso de Kerberos como protocolo de autenticación de sus sistemas. Este tipo de autenticación utilizado por Windows en el inicio de sesión de los usuarios es posible usarlo en nuestras aplicaciones web proporcionando lo que habitualmente se llama Single Sign On.

Para llevarla a cabo se utiliza el mecanismo de autenticación tradicional por HTTP: Basic y Digest, junto a Spnego.

Spnego es un mecanismo de negociación que permite a dos partes acordar un protocolo/algoritmo para la comunicación; en nuestro caso poder recuperar las credenciales del usuario que está accediendo a la aplicación web. Aunque no vamos a entrar en detalle si que me gustaría que vierais el flujo básico de peticiones para que tengáis un conocimiento básico de este mecanismo. Para aquellos que quieran profundizar más en el tema os recomiendo la lectura de la RFC 4559 (<http://www.ietf.org/rfc/rfc4559.txt>).

En el proceso de autenticación se ven involucrados tres actores: navegador, Active Directory y servidor de aplicaciones. Para que quede más claro, en el siguiente gráfico vemos la comunicación entre los diferentes actores.



En el intercambio de peticiones entre las diferentes partes hay tres puntos a destacar:

- Cuando la autenticación es requerida para acceder a un recurso, el servidor envía una respuesta con la cabecera WWW-Authenticate con el valor Negotiate. Esto le dice al cliente que el servidor requiere autenticación mediante Spnego.
- El navegador recupera de alguna manera las credenciales del usuario y las envía al servidor mediante la cabecera Authorize. El ticket enviado es validado contra el KDC (Kerberos Distribution Center) y si todo es correcto se deja acceder al recurso.
- Una vez que el servidor requiera autenticación mediante Spnego el navegador enviará en todas las peticiones que se realicen a esa aplicación el ticket con las credenciales. Aunque no es importante este hecho, hay que tenerlo en cuenta para posteriormente entender el funcionamiento de la extensión de Spnego para Spring Security.

También hay que destacar que este tipo de autenticación es compatible con la mayoría de los navegadores existentes en el mercado desde Explorer pasando por Firefox, Chrome o Safari, evidentemente todos ellos corriendo bajo un sistema operativo Windows.

### 4. Configuración del Active Directory

Antes de añadir Spnego a nuestra aplicación tenemos que disponer de un entorno bien configurado para que todo funcione correctamente. En nuestro caso necesitaremos:

1. Un Active Directory con el protocolo de Kerberos activo (opción disponible por defecto).
2. Agregar nuestro servidor de aplicaciones como equipo en el Active Directory y habilitar la delegación de confianza.
3. Crear un usuario al que asignaremos el servicio HTTP de nuestro servidor.
4. Generar un fichero (denominado keytab) que permita autenticarse a nuestro servidor contra el Active Directory en el proceso de

2011-07-06  
 Otras Sin catalogar - LUGO.

2011-06-20  
 Comercial - Ventas - SEVILLA.

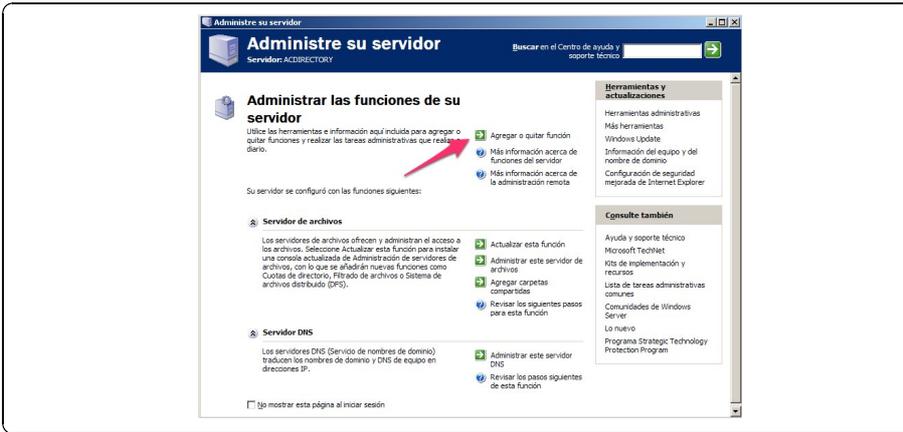
2011-05-24  
 Contabilidad - Especialista Contable - BARCELONA.

2011-05-14  
 Comercial - Ventas - TARRAGONA.

2011-04-13  
 Comercial - Ventas - VALENCIA.

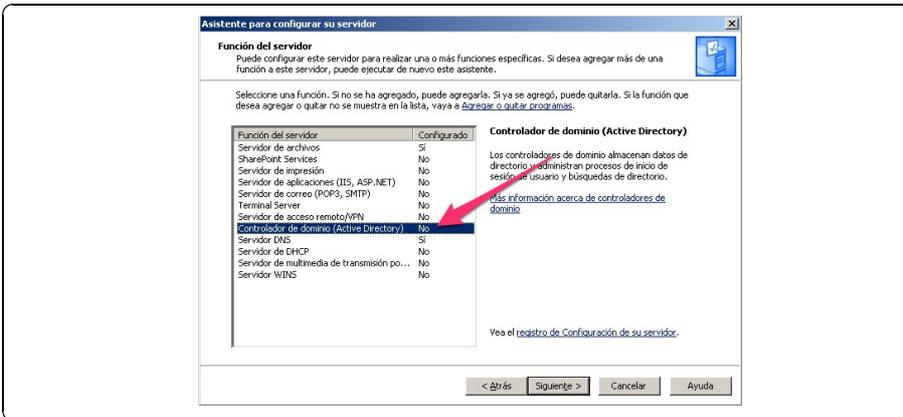
### 4.1 Creación de un servidor de dominio

Crear un servidor de dominio es bastante sencillo a través del asistente que proporciona Windows 2003 Server. Ejecutamos el programa situado en Inicio -> Todos los programas -> Herramientas administradas -> Administre su servidor.

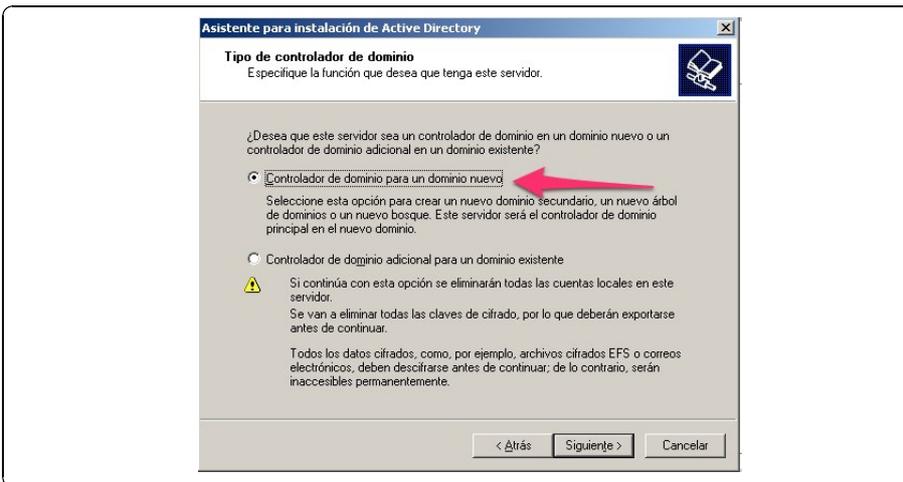


Desde allí lazamos el asistente mediante el enlace "Agregar o quitar función". Al ser relativamente sencillo el proceso de instalación del Active Directory sólo se van a comentar las pantallas más importantes en este proceso, evitando así información irrelevante que no aportan información alguna.

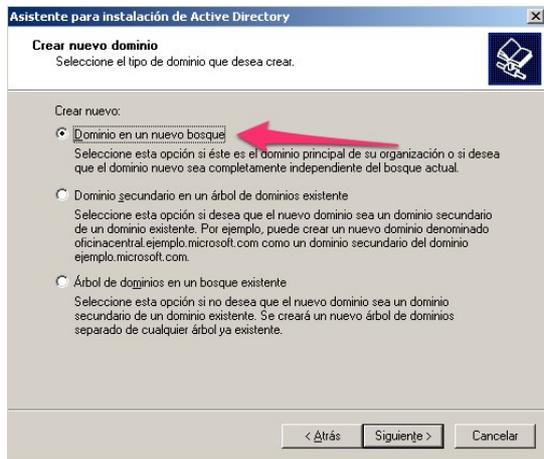
En la siguiente pantalla seleccionamos "Controlador de dominio (Active Directory)" y pulsamos sobre Siguiente.



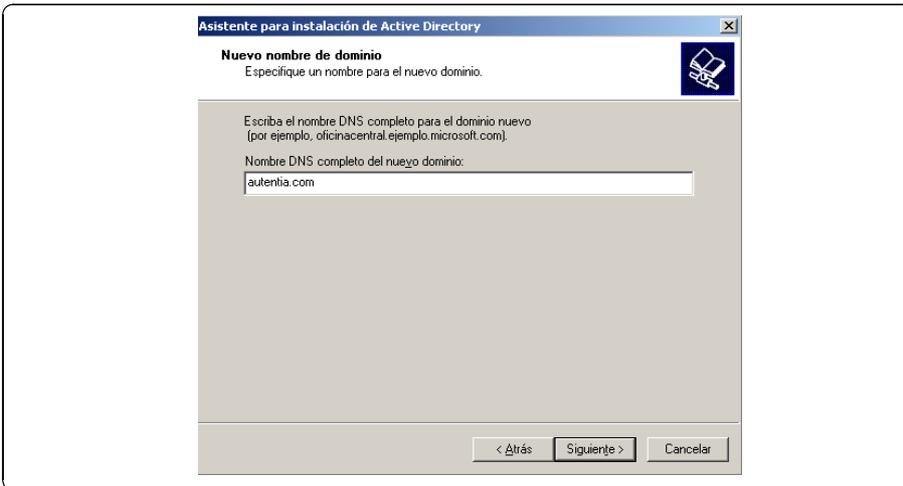
A continuación indicamos que el controlador de dominio que estamos creando es el principal.



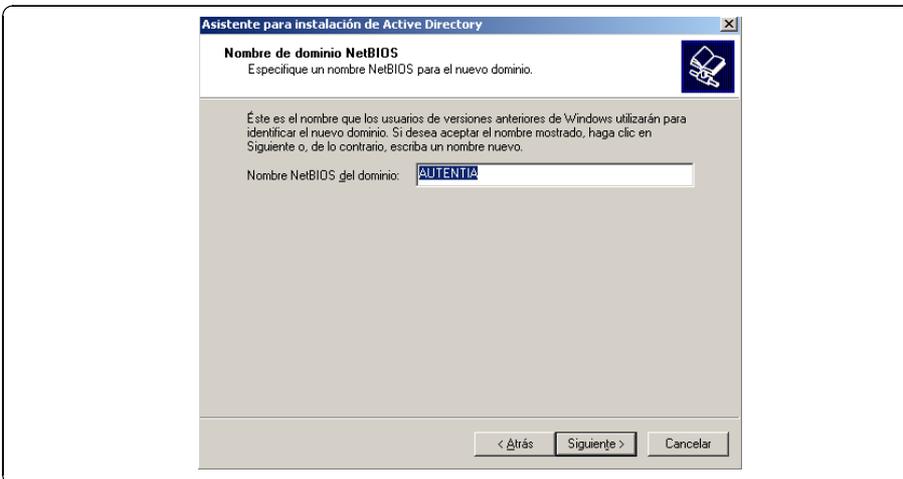
Al ser un nuevo controlador de dominio seleccionamos la opción "Dominio en un nuevo bosque".



A continuación introducimos el nombre de nuestro dominio, en nuestro caso "autentia.com".



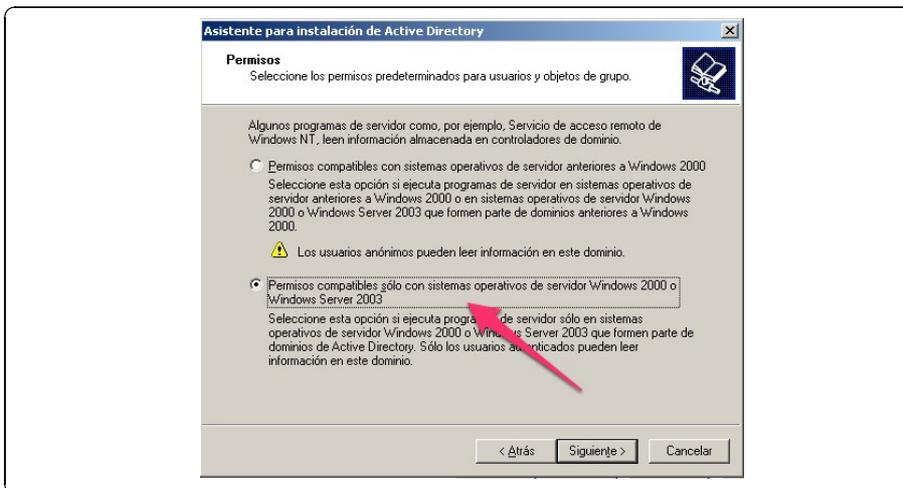
En la siguiente pantalla nos piden el nombre de NetBIOS de nuestro dominio en nuestro caso "AUTENTIA".



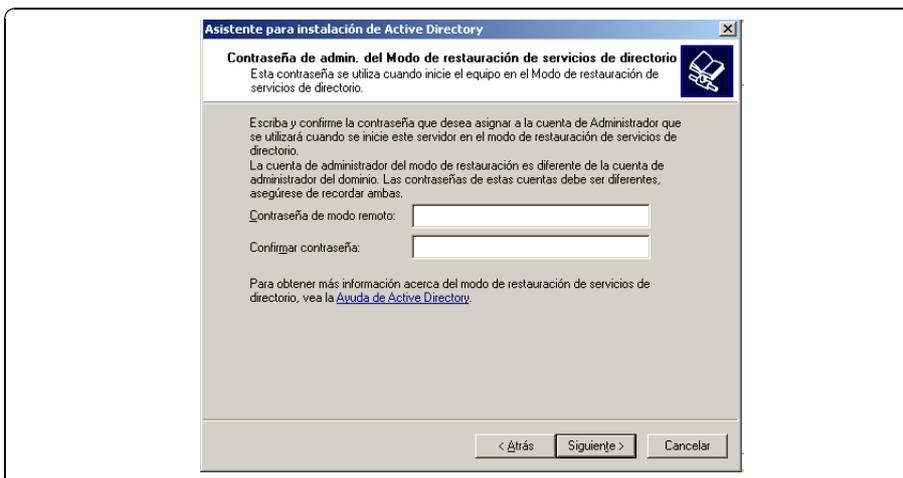
Más tarde nos pedirá los directorios donde Active Directory almacena la base de datos y el registro así como la copia para el servidor de los archivos públicos. En nuestro caso dejamos los valores por defecto.



Por último activamos la compatibilidad para sistemas operativos a partir de Windows 2000.



Finalmente establecemos una contraseña para el acceso remoto del usuario administrador al dominio.

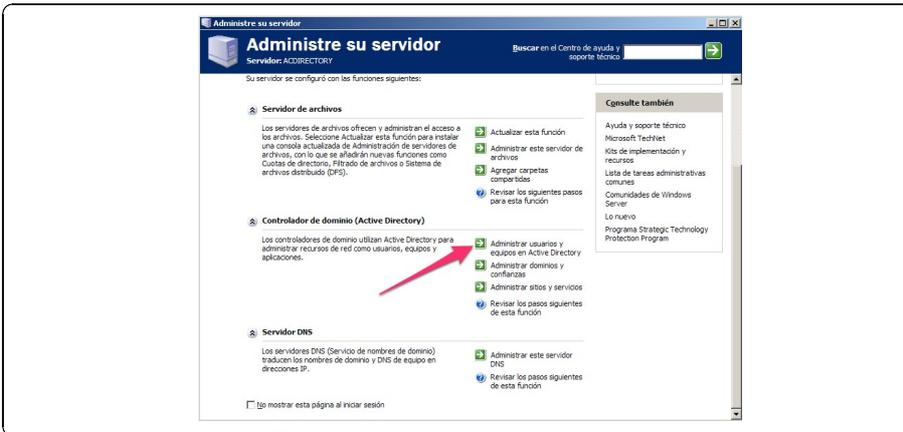


El resto del proceso es automático, finalmente nos pedirá reiniciar y a partir de eso momento tenemos creado nuestro Active Directory con Kerberos V5.

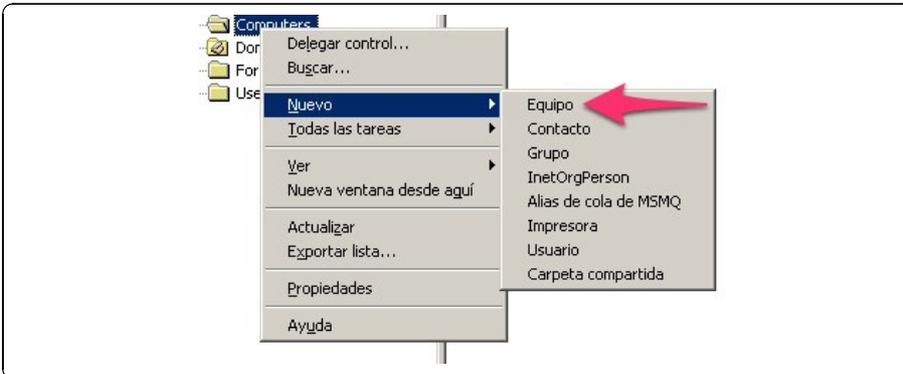


#### 4.2 Agregar el servidor aplicaciones al Active Directory

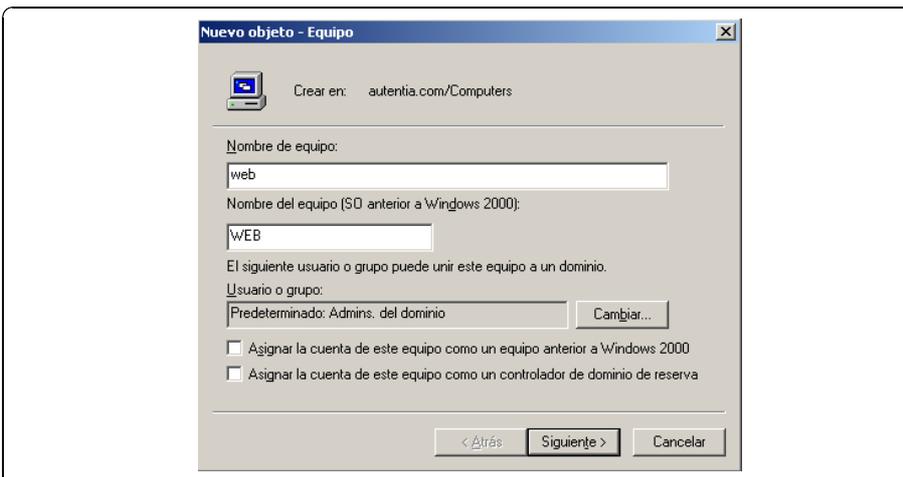
Creado el Servidor de Dominio, el siguiente paso es añadir el servidor de aplicaciones al Active Directory. Accedemos a la aplicación "Administre su servidor" y ejecutamos "Administrar usuarios y equipos en Active Directory".



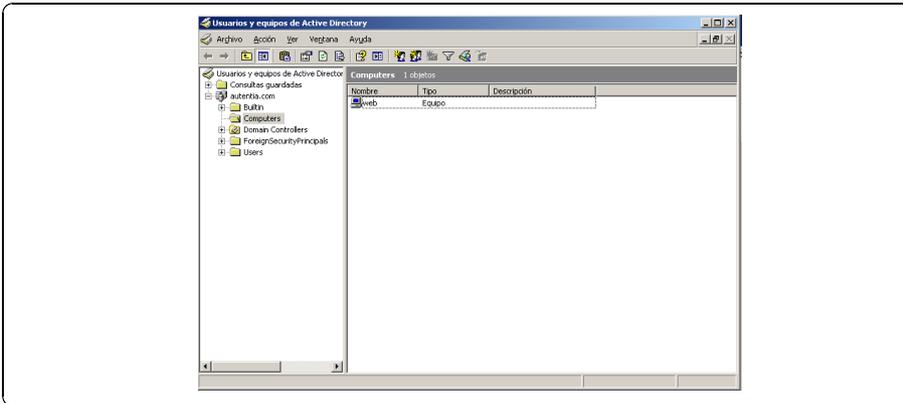
Ahora desplegamos el dominio "autentia.com", seleccionamos la opción "Computers" y pulsando el botón derecho hacemos click en "Nuevo" -> "Equipo".



Introducimos los datos de nuestro servidor, en nuestro caso "web" y pulsamos Finalizar.

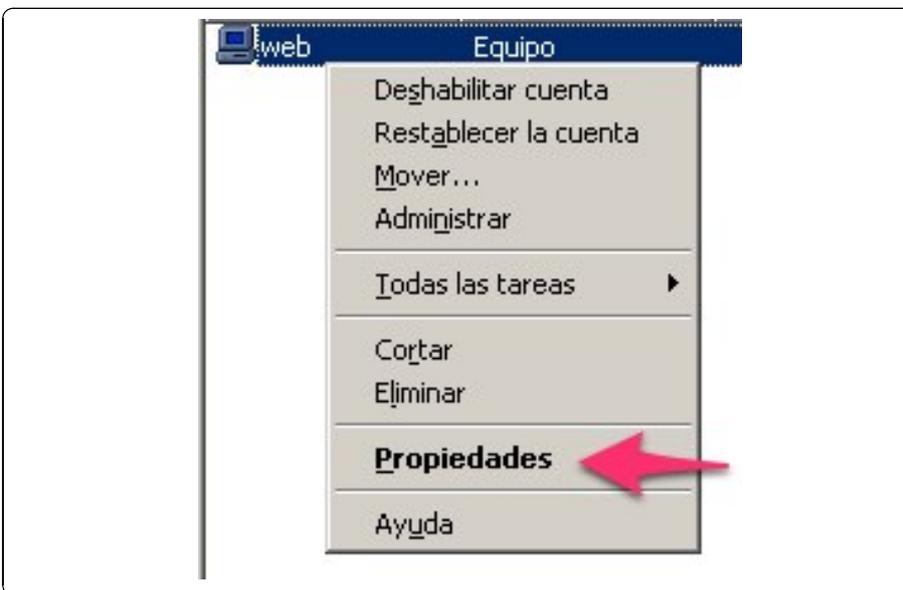


Una vez creado el equipo debemos tener algo como:

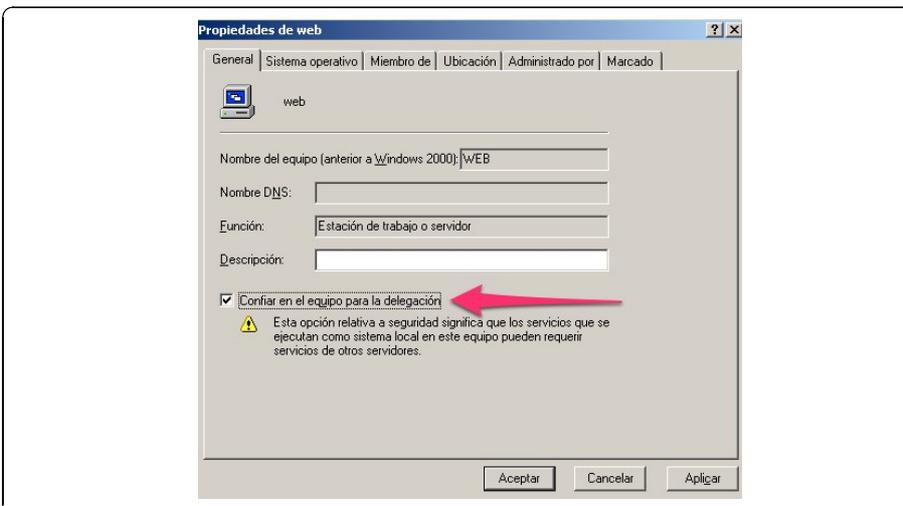


Ahora tenemos que habilitar el equipo creado para que nuestro dominio confíe en él. Este paso es muy importante pues de lo contrario el proceso de autenticación mediante protocolo Kerberos V5 no funcionará. ¿Por qué? Porque cuando el cliente envía el ticket al servidor con las credenciales no contendrá entre los algoritmos el protocolo Kerberos, produciéndose un error "GSSEException: Defective token detected".

Seleccionamos el equipo, pulsamos el botón derecho del ratón y hacemos click en "Propiedades".



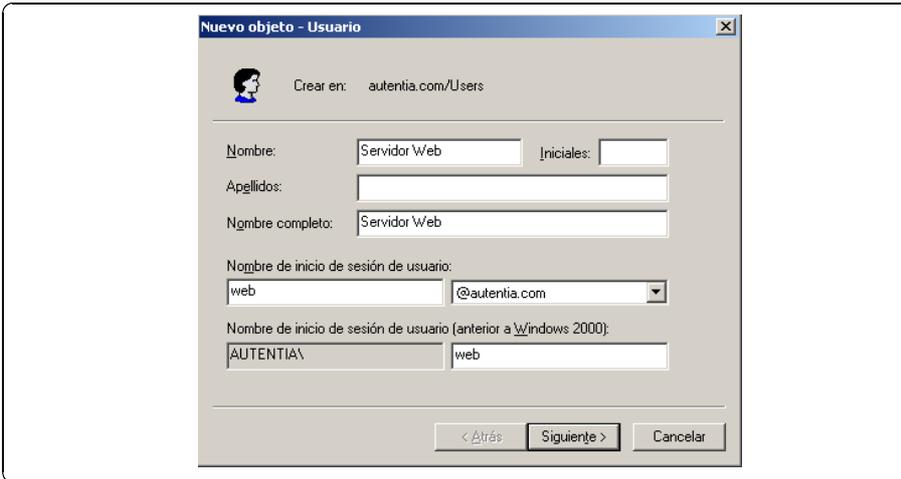
Activamos la opción "Confiar en el equipo para la delegación" y pulsamos "Aceptar".



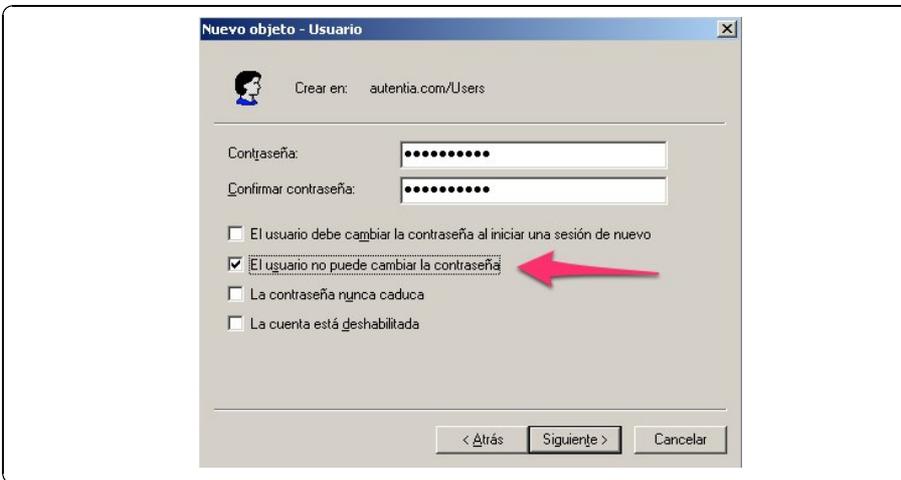
#### 4.3 Creación de un usuario para mapear con nuestro servidor.

Como hemos comentado antes, el servidor debe autenticarse contra el Active Directory antes de validar el token enviado por el cliente. Esta autenticación se realiza mediante un fichero (keytab) que mapea el servicio (servidor de aplicaciones) con un usuario del dominio. Éste debe ser utilizado única y exclusivamente para este propósito ya que en el caso de utilizar un usuario del dominio no podrá volverse a logar en una estación cliente.

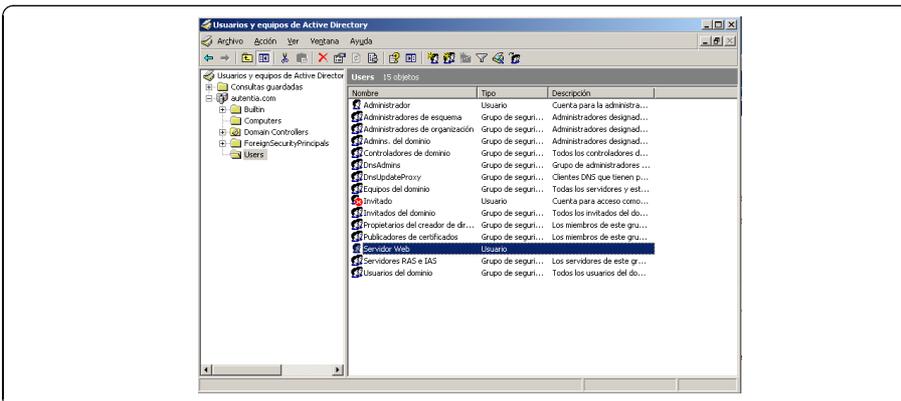
El proceso de creación es muy similar al anterior; accedemos a la opción "Administrar usuarios y equipos en Active Directory", damos al botón derecho sobre la opción "Users" y hacemos click en "Nuevo" -> "Usuario". Rellenamos los datos para el nuevo usuario y pulsamos sobre "Siguiente".



A continuación introducimos la contraseña, desactivamos la opción "El usuario debe cambiar la contraseña al iniciar una sesión de nuevo" y activamos "El usuario no puede cambiar la contraseña" y finalizamos el proceso de creación.



En el listado de usuarios deberá aparecer el usuario creado.



#### 4.4 Crear el fichero keytab para el servidor.

Una vez que hemos dado de alta el equipo donde se encuentra nuestro servidor y el usuario con el que se autenticará para validar los ticket, nos queda generar el fichero keytab. Este fichero es una copia cifrada de la clave privada de un usuario en kerberos y es usada por la aplicación web como proxy de autenticación de las credenciales enviadas por el usuario.

Para poder generar este fichero es necesario el uso del comando ktpass distribuido en el paquete "Microsoft Server 2003 Resources Kit Tools" que debéis instalaros en el caso de no tenerlo.

En nuestro ejemplo abrimos una consola y ejecutamos el siguiente comando:

```
ktpass -princ HTTP/web.autentia.com@AUTENTIA.COM -mapuser web@AUTENTIA.COM -pass * -
ptye KBR5_NT_PRINCIPAL -out web.keytab.
```

```
C:\Documents and Settings\Administrador.ACDIRECTORY.001>ktutil -princ HTTP/web.a
autentia.com@AUTENTIA.COM -maguser web@autentia.com -pass * -ptype KRBS_NT_PRINCIPAL
-rl -out web.keytab
Targeting domain controller: acdirectory.autentia.com
Using legacy password setting method
Successfully mapped HTTP/web.autentia.com to web.
Type the password for HTTP/web.autentia.com:
Type the password again to confirm:
Key created.
Output keytab to web.keytab:
Keytab version: 0x502
Keysize 59 HTTP/web.autentia.com@AUTENTIA.COM ptype 1 KRBS_NT_PRINCIPAL vno 3
etype 0x17 KR4-HMAC keylength 16 0x67973d390c2a426c9fafa28bfcaa9e43
C:\Documents and Settings\Administrador.ACDIRECTORY.001>
```

Lo que estamos haciendo es generar el fichero web.keytab para el servicio HTTP que se ejecuta en la máquina del dominio web.autentia.com y que va estar mapeado contra el usuario web.

El fichero generado lo utilizaremos más adelante cuando configuremos la extensión de Spnego para Spring Security en nuestro proyecto web.

## 5 Integrar Spnego en nuestra aplicación web.

El último paso es integrar la extensión de Spnego en nuestra aplicación web. Supongamos que los recursos situados bajo "/secure" sólo van a poder ser accedidos por aquellos usuarios validos en el dominio. Para conseguirlo debemos añadir varios elementos a la configuración básica de Spring Security. Sin entrar mucho en detalle serían:

- Un entry point  
(org.springframework.security.extensions.kerberos.web.SpnegoEntryPoint): responsable de enviar la cabecera "WWW-Authenticate: Negotiate" cuando se produce un acceso denegado sobre un recurso de nuestra aplicación.
- un filtro  
(org.springframework.security.extensions.kerberos.web.SpnegoAuthenticationProcessingFilter) que se ejecuta siempre que en la cabecera de la petición del usuario venga "Authorize". Utiliza el proveedor de autorización de Spnego para validar las credenciales del usuario.
- y un proveedor de autenticación  
(org.springframework.security.extensions.kerberos.KerberosServiceAuthenticationProvider) encargado de validar el token enviado en la cabecera Authorize.

### 5.1 Creación del fichero de configuración de Spring Security.

Bajo el directorio WEB-INF creamos un xml llamado spnego-spring-security.xml con:

```
01. <beans xmlns="http://www.springframework.org/schema/beans" xmlns:xsi="http://www.w3.org,
02. instance" xmlns:sec="http://www.springframework.org/schema/security" xsi:schemaLocation=
03. beans-
04. 2.0.xsd http://www.springframework.org/schema/security http://www.springframework.org/s
05. security-3.0.xsd">
06.
07. <sec:http entry-point-ref="spnegoEntryPoint">
08.   <sec:intercept-url pattern="/secure/**" access="IS_AUTHENTICATED_FULLY" />
09.   <sec:custom-
10.     filter ref="spnegoAuthenticationProcessingFilter" position="BASIC_AUTH_FILTER" />
11. </sec:http>
12.
13. <bean id="spnegoEntryPoint" class="org.springframework.security.extensions.kerberos.web
14.
15. <bean id="spnegoAuthenticationProcessingFilter" class="org.springframework.security.ext
16.   <property name="authenticationManager" ref="authenticationManager" />
17. </bean>
18.
19. <sec:authentication-manager alias="authenticationManager">
20.   <sec:authentication-provider ref="kerberosServiceAuthenticationProvider" />
21. </sec:authentication-manager>
22.
23. <bean id="kerberosServiceAuthenticationProvider" class="org.springframework.security.ext
24.   <property name="ticketValidator">
25.     <bean class="org.springframework.security.extensions.kerberos.SunJaasKerberosTi
26.       <property name="servicePrincipal" value="HTTP/web.autentia.com@AUTENTIA.COM"
27.       <property name="keyTabLocation" value="classpath:web.keytab" />
28.     </bean>
29.   </property>
30.   <property name="userDetailsService" ref="dummyUserDetailsService" />
31. </bean>
32. <!-- Solo retorna el usuario autenticado por Kerberos con el rol ROLE_USER -->
33. <bean id="dummyUserDetailsService" class="com.autentia.security.extensions.kerberos.sam
34. </beans>
```

Destacar varias líneas del fichero:

- Línea 3: Cambiamos el punto de entrada que viene configurado por defecto en Spring Security por el de Spnego.
- Línea 5: Indicamos a Spring Security que todos los recursos en /secure necesitan autenticación.
- Línea 5: Añadimos en la posición de BASIC\_AUTH\_FILTER de la cadena de filtros de Spring el filtro de Spnego.
- Línea 15: Añadimos el proveedor de autenticación de Spnego al gestor de autenticación de Spring Security.

- Línea 18: Se añade el bean encargado de validar el ticket enviado desde el cliente al proveedor de autenticación de Spnego. Este bean necesita ser configurado con el nombre del servicio dado de alta en el Active Directory y la localización del fichero keytab que más tarde añadiremos.
- Línea 28: Se añade al proveedor el servicio encargado de recuperar la información del usuario. En este ejemplo hemos creado un Dummy que retorna el usuario con el rol `ROLE_USER`.

## 5.2 Añadir Spring Security al web.xml

Ahora añadimos la configuración necesaria para Spring Security al web.xml :

```

01. <?xml version="1.0" encoding="UTF-8"?>
02. <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns="http://java.sun.com/xml/ns/javaee" xmlns:web="http://java.sun.com/xml/ns/
app_2_5.xsd" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/;
app_2_5.xsd" id="WebApp_ID" version="2.5">
03.
04.     <display-name>spring-security-kerberos</display-name>
05.     <filter>
06.         <filter-name>springSecurityFilterChain</filter-name>
07.         <filter-class>org.springframework.web.filter.DelegatingFilterProxy</filter-
class>
08.     </filter>
09.
10.     <filter-mapping>
11.         <filter-name>springSecurityFilterChain</filter-name>
12.         <url-pattern>/*</url-pattern>
13.     </filter-mapping>
14.
15.     <context-param>
16.         <param-name>contextConfigLocation</param-name>
17.         <param-value>/WEB-INF/spnego-spring-security.xml</param-value>
18.     </context-param>
19.
20.     <!-- Bootstraps the root web application context before servlet initialization --
>
21.     <listener>
22.         <listener-
class>org.springframework.web.context.ContextLoaderListener</listener-class>
23.     </listener>
24. </web-app>

```

De las líneas 5 a la 13 se configura el filtro de Spring Security que se encarga del control de acceso a los recursos de nuestra aplicación y de la 15 a la 23 la configuración para cargar el contexto de Spring.

## 5.3 Añadir el fichero web.keytab al classpath de la aplicación.

Para finalizar la configuración de la aplicación tendremos que añadir el fichero web.keytab generado a nuestro classpath. Este tipo de localización no es el recomendable en un entorno de producción, lo correcto sería que este fichero estuviera en una localización del servidor y bien securizado. De lo contrario podríamos comprometer la seguridad de nuestro dominio si el fichero cae en malas manos.

## 6 Configuración del Servidor Tomcat

Antes de desplegar la aplicación debemos añadir a la máquina virtual la configuración de kerberos necesaria para poder validar correctamente el ticket. Se debe añadir la propiedad "java.security.krb5.conf" con el path al fichero de configuración a kerberos que habitualmente en sistemas basados en Unix se encuentran en "/etc/krb5.conf".

En este fichero se debe indicar el dominio, el servidor KDC y los algoritmos soportados. Los más importantes son arcfour-hmac-md5 y rc4-hmac ya que son los algoritmos soportado por el Active Directory de Microsoft.

```

[domain_realm]
.autentia.com = autentia.com
autentiaautentia.com = autentia.com
[libdefaults]
default_realm = autentia.com
permitted_enctypes = aes128-cts aes256-cts arcfour-hmac-md5 rc4-hmac
default_tgs_enctypes = aes128-cts aes256-cts arcfour-hmac-md5 rc4-hmac
default_tkt_enctypes = aes128-cts aes256-cts arcfour-hmac-md5 rc4-hmac

[realms]
AUTENTIA.COM = {
    kdc = acdirectory.autentia.com
    admin_server = autentia.com
    default_domain = autentia.com
}

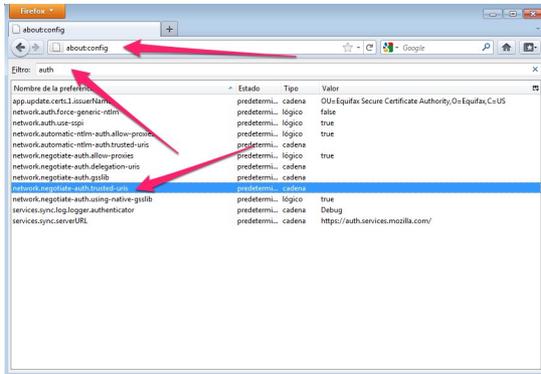
```

## 7 Configurando el navegador

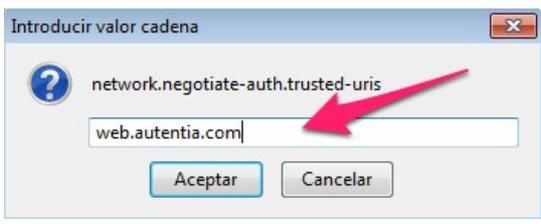
Por último nos queda configurar el navegador de los clientes. Dependiendo de cual se utilice la configuración será diferente. En este tutorial veremos la de Firefox y la de Explorer.

### 7.1 Spnego en Firefox

Para activar la autenticación debemos introducir en la barra de dirección "about:config" y filtrar por "auth".



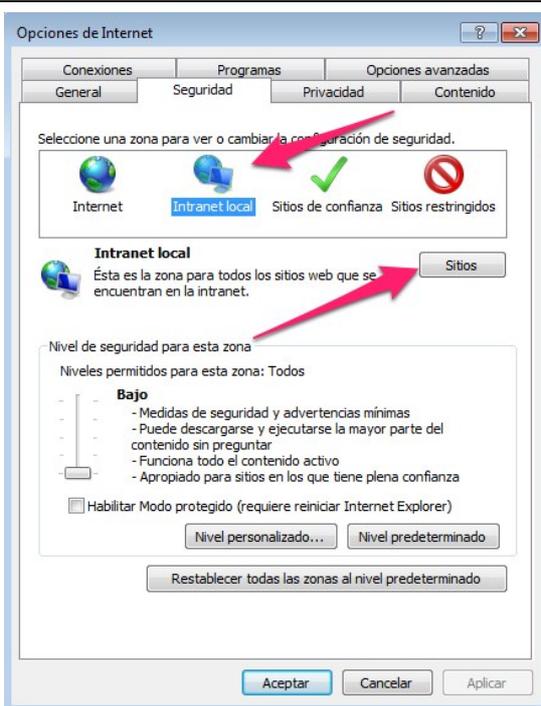
Ahora pulsamos sobre la propiedad "network.negotiate-auth.trusted-uris" e introducimos el dominio donde se hospeda nuestra aplicación "web.autentia.com".



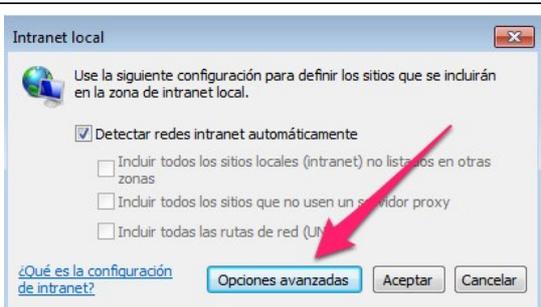
Ahora Firefox esta listo para enviar los ticket al dominio web.autentia.com

### 7.2 Spnego en IExplorer.

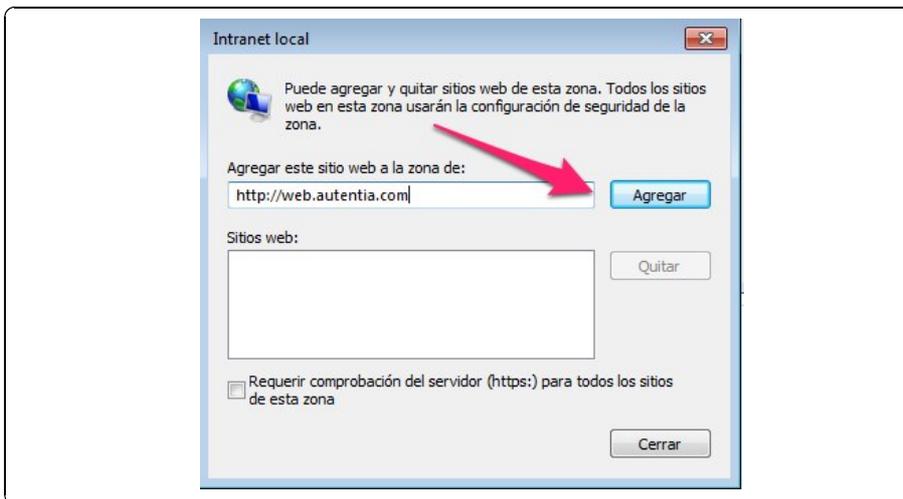
Para habilitar la autenticación integrada en IExplorer debemos añadir el sitio web como sitio en Intranet Local. Accedemos a las opciones de internet del navegador, seleccionamos la pestaña seguridad, hacemos click sobre Intranet Local y luego sobre el botón Sitios.



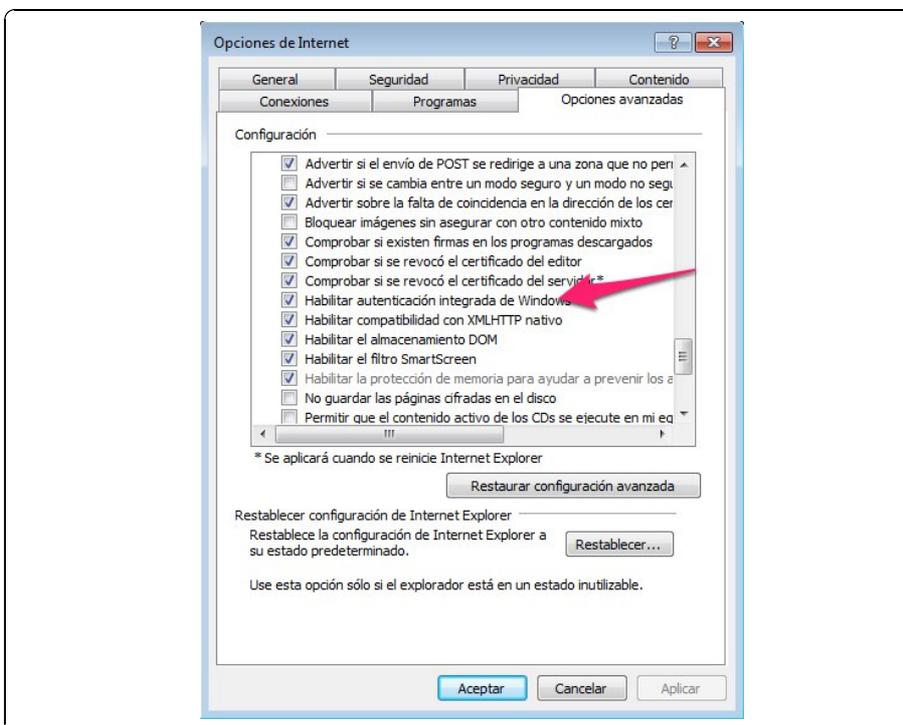
A continuación pulsamos sobre Opciones avanzadas.



Ahora añadimos la dirección como sitios de intranet pulsando Agregar.



Y por último nos vamos a la pestaña "Opciones avanzadas" y nos aseguramos tener habilitada la opción "Habilitar autenticación integrada de Windows".



## 8 Testeando la aplicación

Ya tenemos todo configurado y listo para testear. Arrancamos el Tomcat e iniciamos sesión en un PC dentro del dominio, abrimos un navegador y tecleamos `http://web.autentia.com:8080/spnego-web/secure/index.jsp`. Si todo ha ido bien deberá aparecer la siguiente pantalla.



## 9 Código fuente

Os dejo el ejemplo utilizado en este tutorial para que podáis usarlo (código).

## 10 Conclusión

El uso de credenciales de dominio en aplicaciones de Intranet es algo habitual y como habéis podido observar interactúan muchos actores. Hay tener muy claro lo que se esta haciendo en cada momento y configurar todo de forma correcta sino el sistema no funcionará.

Este tutorial ha abordado un ejemplo donde se utiliza un único método de autenticación, pero puede ocurrir que queramos integrarlo con otros como el acceso mediante login, es decir, si la autenticación basada en Kerberos fallase redireccionar al usuario a un formulario para acceder a la aplicación.

En este caso debemos implementar una extensión sobre Spnego Spring Security. Tendríamos que crear nuestro propio Entry Point que redireccionara a una página encargada de detectar si el usuario se encuentra en el dominio, un filtro que recoja esa información y en función de ella redirigir al formulario de login o enviar la cabecera "WWW-Authenticate". Este filtro debería ejecutarse delante del filtro de Spnego.

Animáte y coméntanos lo que pienses sobre este **TUTORIAL**:

Puedes opinar o comentar cualquier sugerencia que quieras comunicarnos sobre este tutorial; con tu ayuda, podemos ofrecerte un mejor servicio.

Enviar comentario

(Sólo para usuarios registrados)

» **Regístrate** y accede a esta y otras ventajas «

## COMENTARIOS



SOME RIGHTS RESERVED

Esta obra está licenciada bajo licencia Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5

Copyright 2003-2011 © All Rights Reserved | [Texto legal y condiciones de uso](#) | [Banners](#) | [Powered by](#)

[W3C XHTML 1.0](#)

[W3C CSS](#)

[XML RSS](#)

[XML ATOM](#)

