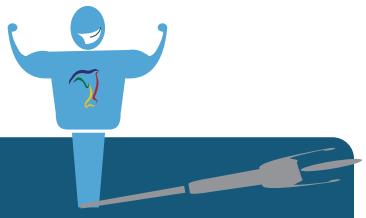
autentia

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...

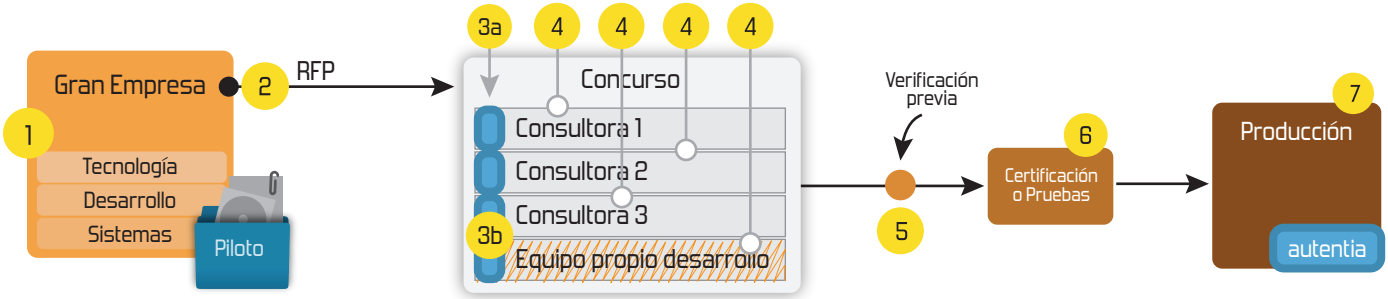
1. Desarrollo de componentes y proyectos a medida



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

timas Noticias

» Vota AdictosAltrabajo.com en DZone
» Liberada TNTConcept 0.16.1
» iAdictos Renovado!
» Resumen de la cuarta charla gratuita de Autentia: SCRUM
» Si se pregunta ¿Qué ofrece este Web?
» Autentia cumple 6 años
» Alimarket.es: Primera aplicación pública del framework wuija by Autentia
» Entrevista a Roberto Canales

Noticias Destacadas

» Resumen de la cuarta charla gratuita de Autentia: SCRUM
» Autentia cumple 6 años
» Alimarket.es: Primera aplicación pública del framework wuija by Autentia
» Liberada TNTConcept 0.16.1

Comentarios Cómic

Enlaces

Tutorial desarrollado por

Borja Lázaro de Rafael

Consultor tecnológico de desarrollo de proyectos informáticos.

Ingeniero en Informática

Puedes encontrarme en Autentia

Somos expertos en Java/J2EE

Descargar este documento en formato PDF: Certificados_Servidor_Keytool_OpenSSL_para_IE7.pdf

Fecha de creación del tutorial: 2008-07-04

Certificados de Servidor con Keytool y OpenSSL para IE7

Índice de contenido

1. Introducción

En varios tutoriales ya hemos visto como generar certificados de servidor para dar soporte SSL en distintos servidores. El motivo de este tutorial es crear un certificado de servidor con la herramienta "keytool" de java y "OpenSSL" que sea válido en un navegador IE7, ya que hemos detectado que certificados generados por otros métodos pueden no ser aceptados por IE7, impidiendo el acceso a nuestras páginas.

2. Entorno

El tutorial se ha escrito bajo el siguiente entorno:

Últimos tutoriales

2009-05-11
Introducción a TortoiseSVN

2009-05-07
Hacer 'scp' de varios ficheros sin solicitud de clave

2009-05-02
Plugin Hibernate3 para Maven

2009-04-26
AgileDraw: una técnica rápida de modelado

2009-04-24
Spring AOP: Cacheando aplicaciones usando anotaciones y aspectos con Aspectj

2009-04-20
Modelos de conocimiento con CmapTools

2009-04-16
Informes Crosstab con iReport

2009-04-16
Registro de un fichero de datos personales con el formulario NOTA

2009-04-15
Estadísticas de www.adictosaltrabajo.com Abril 2009

- Hardware: Portatil Samsung R70 (Intel(R) Core(TM)2 Duo 2,5Ghz, 2046 MB RAM, 232 Gb HD)
- Sistema Operativo: Windows Vista Home Premium
- Máquina Virtual Java: JDK 1.5.0_14 de Sun Microsystems
- OpensSSL 0.9.8g

3. Crear la CA

Realmente no es necesario que tengamos nuestra propia CA para emitir los certificados, ya que podríamos utilizar los servicios que nos ofrecen varias CA's como VeriSign, FNMT, etc. Casi todas ellas previo pago.

Lo que vamos a ver en este punto es como crearnos nuestra propia CA, de forma que podamos emitir el certificado del servidor nosotros mismos.

Para crearnos nuestra "pequeña" CA necesitamos una herramienta como OpenSSL que nos ofrece un conjunto de herramientas criptográficas open source que podemos descargar de http://www.openssl.org/related/binaries.html.

3.1 Instalación de OpenSSL

El proceso de instalación de OpenSSL para Windows es muy sencillo, tendremos el clásico asistente donde tendremos que aceptar la licencia y seleccionar el destino donde queremos instalar.

Nota: Para facilitar la ejecución de los comandos de OpenSSL añadimos el directorio "bin" del OpenSSL a la variable de entorno "PATH". En nuestro caso añadimos "C:\OpenSSL\bin".

3.2 Crear estructura de la CA

Aunque no es necesario crear una estructura determinada para gestionar los certificados y peticiones de la CA, siempre es conveniente tener una estructura adecuada para organizar la información que gestionamos.

Para esto nos vamos a crear un directorio para nuestra CA en "C:\autentiaCA", y dentro creamos los siguientes directorios:

- keys: Directorio donde guardaremos las claves generadas.
- request: Directorio donde guardaremos las peticiones (CSR's) de nuevos certificados.
- certs: Directorio donde se guardarán los nuevos certificados.
- crl: Directorio donde guardaremos las listas de certificados revocados.

También crearemos dos ficheros más en el directorio de nuestra CA:

- database.txt: Sirve como base de datos para los certificados emitidos. El fichero lo crearemos vacío.
- serial.txt: Sirve para llevar un control del número de serie para los certificados emitidos. El fichero lo crearemos con la cadena "00" en la primera línea.
- crlNumber.txt: Sirve para llevar un control del número de serie para las listas de certificados revocados. El fichero lo crearemos con la cadena "00" en la primera línea.

3.3 Generar el certificado de la CA

Ahora generaremos el certificado de nuestra CA. Abrimos una consola de comandos y nos situamos en el directorio de nuestra CA.

2009-04-15
Iniciación a OSWorkflow con Spring

2009-04-14
Tests de Selenium con librerías de componentes JSF: Apache Tomahawk.

2009-04-13
JTAPI. El API de Telefonía para Java

2009-04-13
Registro de Web Services con Apache jUDDI. Configuración y ejemplo

2009-04-13
Cómo hacer UML con Eclipse y el plugin UML2

2009-04-09
Spring WS: Servicios Web a través del correo electrónico

2009-04-02
Creación de cursos con Moodle

2009-03-31
Integrar Liferay Portal 5.2.1 con Pentaho BI 2.0.0 sobre MySQL 5.1

2009-03-31
Spring WS: Construcción de Clientes de Servicios Web con Spring

2009-03-30
Administración de sitios Moodle

2009-03-29
Empaquetamiento de aplicaciones de escritorio (standalone) con Maven

2009-03-27
Primeros pasos con Moodle

2009-03-26
Introducción a JSF Java

2009-03-25
A1 Website Analyzer

2009-03-24
Cómo ver el correo de Gmail sin conexión a Internet

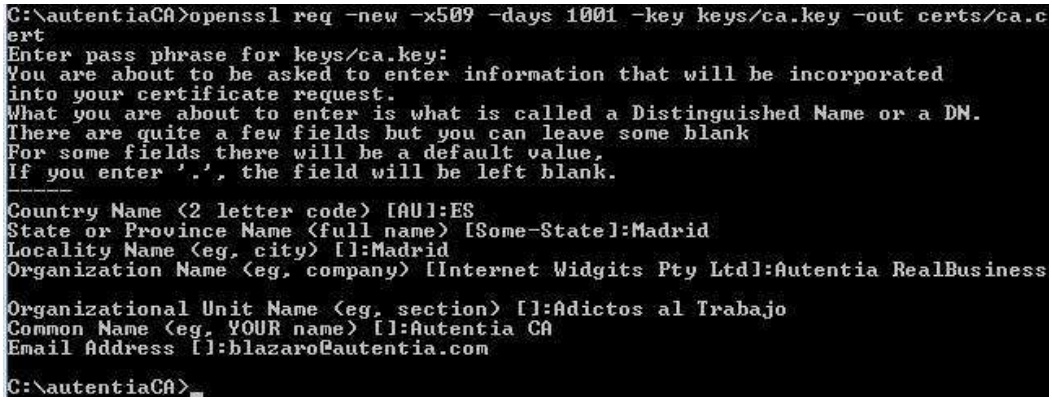
2009-03-20
JasperReports Maven Plugin

2009-03-16
Creación de contenidos SCORM: eXe

Primero crearemos el par de claves con el comando:

- -out: Es la ubicación donde generar el fichero para el par de claves.
- 1024: Tamaño de las claves.

Una vez tenemos las claves de nuestra CA, tenemos que crear un certificado público X509 que distribuiremos para que los diversos clientes reconozca a nuestra CA y. Para generar el certificado lo hacemos con:



- req: Petición para un CSR.
- -new: Nueva petición.
- -X509: Tipo de certificado.
- -days: Duración de la validez del certificado.
- -key: Fichero con el par de claves asociadas al certificado. (Nota:En el certificado X509 solo se incluye la clave pública)
- -out: Ubicación del certificado generado.

Durante el proceso de generación nos preguntará los datos públicos que va a tener el certificado de la CA (país, ciudad, organización, etc.)

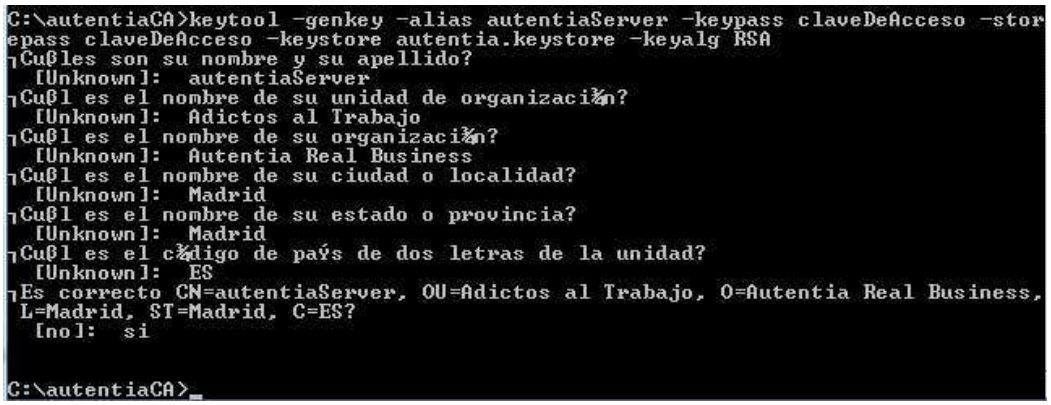
4. Establecer SSL en un servidor

Para establecer una comunicación segura en nuestro servidor necesitaremos tener un certificado de servidor que sirva tanto para identificar al propio servidor como para cifrar la comunicación entre el servidor y los clientes que se conecten al mismo. En los siguientes puntos vamos a ver como generar nuestras claves de servidor, y como conseguir un certificado para dichas claves.

4.1 Generar las claves del servidor

En nuestro caso vamos a generar las claves utilizando la herramienta "keytool" de Java y fichero de claves del tipo Java Key Store (JKS).

Para la generación de las claves de servidor ejecutamos:



- -genkey: Petición para generar el par de claves.
- -alias: Nombre con el que haremos referencia al par de claves creado.
- -keypass: Contraseña de acceso a la clave privada.
- -storepass: Contraseña de acceso al fichero JKS.
- -keystore: Fichero JKS donde almacenar el par de claves.

2009-03-15
Spring WS: Creación de Servicios Web con Spring

2009-03-13
Instalación Alfresco (Labs)

2009-02-26
Maven JXR Plugin: publica el código fuente en el site

2009-03-15
Generación de XML Schema (XSD) y DTD a partir de documentos XML

2009-03-04
Persistencia con Spring

2009-02-26
Vistas materializadas

2009-02-03
Instalación de MySQL 5.1 en Windows

2009-03-03
Instalación de Java Virtual Machine

2009-03-03
Primeros Pasos con Liferay 5.2.1

2009-02-27
Edicion de video MPEG2

2009-02-26
Introducción teórica a XPath

2009-02-26
Integración Selenium / Maven 2 / Surefire / Cargo / Tomcat 6

2009-02-24
Selenium Remote Control

2009-02-22
Integración de Groovy, JRuby y BeanShell con Spring 2

2009-02-18
Instalación de Pentaho BI Suite Community Edition 1.7.0

2009-02-18
Replicar Web PHP en máquina local

2009-02-16
Selenium Core : El motor de Selenium.

2009-02-16

- -keyalg: Algoritmo para el par de claves.

Nota: En algunos servidores es necesario que la contraseña de acceso al par de claves tiene que ser la misma que la contraseña de acceso al almacén de claves JKS.

4.2 Generar CSR

Ahora tenemos que generar un fichero CSR (Certificate Signing Request) para que la CA emita un certificado asociado al par de claves que hemos generado. Para conseguir el CSR ejecutamos:

```
C:\autentiaCA>keytool -certreq -alias autentiaServer -keypass claveDeAcceso -sto
repass claveDeAcceso -keystore autentia.keystore -file request/autentiaServer.cs
r

C:\autentiaCA>
```

- -certreq: Opción para generar una nueva petición de certificado (CSR).
- -alias: Nombre con el que haremos referencia al par de claves creado.
- -keypass: Contraseña de acceso a la clave privada.
- -storepass: Contraseña de acceso al fichero JKS.
- -keystore: Fichero JKS donde almacenar el par de claves.
- -file: Ubicación del fichero CSR a generar.

4.3 Enviar petición a la CA y firmar con la CA

Una vez hemos generado nuestra petición para un nuevo certificado hay que enviarlo a una CA para que nos emita un nuevo certificado.

En nuestro caso utilizaremos la CA que nos hemos creado, pero antes de generar el certificado debemos retocar el fichero de configuración "openssl.cnf". Este fichero se encuentra en el directorio "bin" donde hayamos instalado el OpenSSL (en nuestro caso "C:\OpenSSL\bin\openssl.cnf"); haremos una copia de este fichero al directorio de nuestra CA (en nuestro caso "C:\autentiaCA\openssl.cnf"). Ahora tenemos que modificar algunos parámetros de la sección "[CA_default]" quedando los parámetros modificados:

- dir = . # Where everything is kept
- certs = $dir/certs # Where the issued certs are kept
- crl_dir = $dir/crl # Where the issued crl are kept
- database = $dir/database.txt # database index file.
- new_certs_dir = $dir/certs # default place for new certs.
- certificate = $dir/certs/ca.cert # The CA certificate
- serial = $dir/serial.txt # The current serial number
- crlnumber = $dir/crlNumber.txt # the current crl number
- crl = $dir/crl/crl.pem # The current CRL
- private_key = $dir/keys/ca.key# The private key

Ahora generaremos nuestro certificado X509 para el servidor con el comando:

```
C:\autentiaCA>openssl ca -in request/autentiaServer.csr -out certs/autentiaServe
r.cert -config openssl.cfg -policy policy_anything
Using configuration from openssl.cfg
Loading 'screen' into random state - done
Enter pass phrase for ./keys/ca.key:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 0 (0x0)
        Validity
            Not Before: Jul  3 14:49:11 2008 GMT
            Not After : Jul  3 14:49:11 2010 GMT
        Subject:
            countryName               = ES
            stateOrProvinceName       = Madrid
            localityName              = Madrid
            organizationName          = Autentia Real Business
            organizationalUnitName    = Adictos al Trabajo
            commonName                = autentiaServer
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                50:03:29:14:DB:A2:FE:79:AD:2B:05:D6:B7:D9:0A:99:2F:9E:FA:DB
            X509v3 Authority Key Identifier:
                keyid:1F:A6:BD:60:5A:5A:AC:E5:5C:93:E4:3D:FF:48:04:7E:F4:0C:95:C
0

Certificate is to be certified until Jul  3 14:49:11 2010 GMT (730 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated

C:\autentiaCA>
```

```
C:\autentiaCA>
```

- ca: Indica que actúe como una CA.
- -in: Ubicación del fichero CSR de petición.
- -out: Ubicación del certificado generado.
- -config: Ubicación del fichero de configuración de OpenSSL.
- -policy: Política que se utiliza para la firma de certificados.

Esto nos generará el certificado de nuestro servidor y queda llevarlo de vuelta a nuestro almacén de claves JKS que utiliza el servidor.

4.4 Instalar la respuesta de la CA

El formato del certificado de servidor que hemos generado con OpenSSL no lo podemos importar directamente, tenemos que "limpiarlo" para quedarnos sólo con la parte comprendida entre "-----BEGIN CERTIFICATE-----" y "-----END CERTIFICATE-----" con ambas líneas incluidas. Esto lo podemos hacer bien copiando y pegando en un nuevo fichero o ejecutando el siguiente comando:

```
C:\autentiaCA>openssl x509 -in certs/autentiaServer.cert -outform PEM -out auten
tiaServerPEM.cert

C:\autentiaCA>_
```

- -in: Ubicación del certificado de servidor que hemos generado.
- -outform: Formato de la salida.
- -out: Ubicación del fichero de salida.

Antes de importar el certificado de servidor a nuestro JKS, deberemos importar el certificado público de la CA y marcarlo como de confianza. Esto lo hacemos ejecutando:

```
C:\autentiaCA>keytool -import -alias autentiaCa -keypass claveDeAcceso -file cer
ts/ca.cert -storepass claveDeAcceso -keystore autentia.keystore
Propietario: EMAILADDRESS=blazaro@autentia.com, CN=Autentia CA, OU=Adictos al Tr
abajo, O=Autentia RealBusiness, L=Madrid, ST=Madrid, C=ES
Emisor: EMAILADDRESS=blazaro@autentia.com, CN=Autentia CA, OU=Adictos al Trabajo
, O=Autentia RealBusiness, L=Madrid, ST=Madrid, C=ES
N·mero de serie: ffa6e155511248ea
Vßlido desde: Thu Jul 03 16:27:09 CEST 2008 hasta: Thu Mar 31 16:27:09 CEST 2011

Huellas digitales del certificado:
         MD5:  20:75:27:06:0E:5E:0E:78:A0:66:8E:6D:FC:66:5F:D6
         SHA1: D4:44:CC:EB:0D:3F:DE:C3:05:E0:B6:55:51:C8:F3:A4:99:44:F7:FE
¬Confiar en este certificado? [no]:  si
Se ha a±adido el certificado al almacún de claves

C:\autentiaCA>_
```

- -import: Acción de importación.
- -alias: Alias que damos al certificado de la CA.
- -keypass: Clave de acceso que damos para acceder al certificado de la CA.
- -file: Ubicación del certificado de la CA.
- -storepass: Contraseña de acceso al fichero JKS.
- -keystore: Ubicación del fichero JKS.

Por último sólo nos queda importar el certificado de nuestro servidor con el comando:

```
C:\autentiaCA>keytool -import -alias autentiaServer -keypass claveDeAcceso -file
 autentiaServerPEM.cert -storepass claveDeAcceso -keystore autentia.keystore
Se ha instalado la respuesta del certificado en el almacún de claves

C:\autentiaCA>
```

Para comprobar que en nuestro almacén de claves JKS tenemos todas las claves necesarias hacemos un listado del mismo:

```
C:\autentiaCA>keytool -list -v -storepass claveDeAcceso -keystore autentia.keyst
ore

Tipo de almacún de claves: jks
Proveedor de almacún de claves: SUN

Su almacún de claves contiene 2 entradas

Nombre de alias: autentiaserver
Fecha de creaci¾n: 03-jul-2008
Tipo de entrada: keyEntry
Longitud de la cadena de certificado: 2
Certificado[1]:
Propietario: CN=autentiaServer, OU=Adictos al Trabajo, O=Autentia Real Business,
 L=Madrid, ST=Madrid, C=ES
Emisor: EMAILADDRESS=blazaro@autentia.com, CN=Autentia CA, OU=Adictos al Trabajo
, O=Autentia RealBusiness, L=Madrid, ST=Madrid, C=ES
N·mero de serie: 2
Vßlido desde: Thu Jul 03 18:14:26 CEST 2008 hasta: Sat Jul 03 18:14:26 CEST 2010
```

- -list: Opción de listado.
- -v: Opción "verbose".
- -storepass: Contraseña de acceso al almacén de claves JKS.
- -keystore: Ubicación del almacén de claves JKS.

5 Conclusiones

En este tutorial hemos visto un método para crear un certificado de servidor con "keytool" y "OpenSSL", aunque existen muchos otros métodos para crear certificados igualmente válidos. Sea cual sea el método en el que hemos generado nuestro certificado hay que tener en cuanta los siguientes puntos para evitar las alertas de seguridad que nos muestran los clientes al conectarse a un servidor seguro:

- El nombre del certificado (CN, Common Name), a quién está emitido, debe coincidir con el nombre del dominio del servidor (p.e. www.adictosaltrabajo.com).
- Los certificados tienen un periodo de validez, por lo que siempre debe haber un responsable que se encargue de gestionar los certificados y asegurarse que no han caducado.
- Los clientes deben confiar explícitamente en la CA que ha emitido el certificado del servidor. En el caso de CA's conocidas suelen venir instalados por defecto sus certificados públicos para que nuestros clientes confíen en los certificados que emiten. Si hemos emitido el certificado del servidor con nuestra propia CA, tendremos que instalar el certificado público de nuestra CA en todos los clientes que se conecten a nuestro servidor si queremos que no les aparezca la alerta de seguridad.

Anímate 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.

Nombre: E-Mail:

Comentario:

Enviar comentario **Texto Legal y condiciones de uso**

Autor	Mensaje
Javier Conde	Fecha de envío: 2008-09-26 - 12:32:45 PM Excelente tutorial. Muchas gracias. Me habéis ayudado mucho.

Página 1 de 1 Anterior **Saltar a la página 1** Siguiente

- Puedes inscribirte en nuestro servicio de notificaciones haciendo clic aquí.
- Puedes firmar en nuestro libro de visitas haciendo clic aquí.
- Puedes asociarte al grupo AdictosAlTrabajo en XING haciendo clic aquí.
- Añadir a favoritos Technorati. ADD THIS BLOG TO MY Technorati FAVORITES

utoriales recomendados

Nombre	Resumen	Fecha	Visitas	Valoración	Votos	Pdf
Registro de un fichero de datos personales con el formulario NOTA	En este tutorial se presenta a modo de iniciación cómo registrar los ficheros de datos en la AEPD y haciendo algunas aclaraciones prácticas a tener en cuenta para rellenar el formulario de registro	2009-04-16	480	Muy bueno	1	
Tests de Selenium con librerías de componentes JSF: Apache Tomahawk.	En este tutorial vamos a hablar de cómo escribir tests funcionales con Selenium IDE sobre interfaces de usuario construidas con librerías de componentes visuales JSF y, en concreto, con Apache Tomahawk y uno de sus componentes.	2009-04-14	526	Muy bueno	1	
Cómo ver el correo de Gmail sin conexión a Internet	En este tutorial vamos a ver como podemos configurar el navegador Firefox 3 para poder acceder a todo nuestro histórico de correos sin necesitar de un conexión a Internet.	2009-03-24	1304	Bueno	7	
Integración Selenium / Maven 2 / Surefire / Cargo / Tomcat 6	Con este tutorial se pretende integrar en nuestro proyecto : Maven, Selenium, Surefire, Cargo y Tomcat 6 con el objetivo de incluir y ejecutar las pruebas de integración dentro del ciclo de vida de Maven.	2009-02-26	698	Muy bueno	3	
Selenium Remote Control	Selenium Remote Control es una herramienta que permite automatizar las pruebas sobre aplicaciones web	2009-02-24	1143	Muy bueno	6	
Selenium Core : El motor de Selenium.	Selenium Core es un aplicación perteneciente al juego de herramientas SeleniumHQ que permite realizar juegos de pruebas sobre aplicaciones web.	2009-02-16	1169	Muy bueno	4	
Manual Avanzado de Firebug	Firebug es una extensión gratuita para Mozilla Firefox especialmente diseñada para todos aquellos programadores que se dedican al desarrollo web. En este tutorial os enseñaremos en profundidad su funcionamiento.	2009-02-09	2017	Muy bueno	12	
Web Service Security	En este tutorial vamos a ver como crear un servicio web seguro con autenticación mediante usuario y contraseña. WS-Security define cómo utilizar los tokens de seguridad, XML Signature y Xml Encryption en los mensajes SOAP para proporcionar autenticación,	2009-02-09	1614	Muy bueno	3	
Executor : Un programa para ejecutarlos a todos.	Nuestro amigo Victor nos enseña éste utilísimo programa para programar la ejecución de aplicaciones de manera sencilla y rápida	2009-01-19	1527	Muy bueno	6	
Restaurar una Base de Datos en SQL Server o como cambiar el propietario de los objetos de la base de datos	En este tutorial vamos a ver como podemos replicar una base de datos de SQL Server en un servidor distinto al original sin que exista conexión entre ellos.	2009-01-16	1930	Bueno	6	

Nota: