**EEUtil-misc**

**Manual de Instalación**

*Documento de Integración*

**Sistemas Desarrollo**

*Versión* del documento

**0001**

*Fecha de revisión*

**31/10/2018**

*Realizado por*

**Sistemas Desarrollo**

**ÍNDICE**

[1 Control de modificaciones 3](#_Toc528766401)

[2 Objetivo de este documento 4](#_Toc528766402)

[3 Requisitos previos 5](#_Toc528766403)

[4 Instalación 6](#_Toc528766404)

[4.1 Base de datos 6](#_Toc528766405)

[Instalación de EEUtils (MySQL) 6](#_Toc528766406)

[4.2 Ficheros de configuración 7](#_Toc528766407)

[log4j.properties 7](#_Toc528766408)

[database.properties (MySQL) 7](#_Toc528766409)

[afirma.properties 7](#_Toc528766410)

[eeutil.properties 8](#_Toc528766411)

[clave.properties 8](#_Toc528766412)

[SignModule\_SP.xml 11](#_Toc528766413)

[StorkSamlEngine\_SP.xml 12](#_Toc528766414)

[SamlEngine.xml 12](#_Toc528766415)

[credential.properties 12](#_Toc528766416)

[eni.properties 12](#_Toc528766417)

[facturae.properties 12](#_Toc528766418)

[loadTables-config.properties 12](#_Toc528766419)

[pdfA.properties 13](#_Toc528766420)

[siaService.properties 13](#_Toc528766421)

[wsCrypto.properties 13](#_Toc528766422)

[messages.properties 13](#_Toc528766423)

[4.3 Instalación de EEUtil-misc a partir del código fuente 15](#_Toc528766424)

[Instalación de librerías externas: 15](#_Toc528766425)

[Descarga de dependencias: 16](#_Toc528766426)

[Especificar versión del conector de base de datos MySQL: 16](#_Toc528766427)

[Ejecutar EEUtil-misc en entorno local 16](#_Toc528766428)

[Construcción del WAR a partir del código fuente 16](#_Toc528766429)

[4.4 Instalación de EEUtil-misc a partir del war. 17](#_Toc528766430)

[5 Alta de nuevas aplicaciones 18](#_Toc528766431)

[Tabla inside\_aplicaciones 18](#_Toc528766432)

[Tabla inside\_aplicaciones\_propiedad 18](#_Toc528766433)

[6 Autenticación por Cl@ve 19](#_Toc528766434)

[7 Web Services externos a EEUtil-misc 21](#_Toc528766435)

[8 Anexo Tomcat 21](#_Toc528766436)

[Instalación en Tomcat 21](#_Toc528766437)

# Control de modificaciones

Revisión Actual: 001

Fecha: 31/10/2018

Autor: DSIC - Sistemas Desarrollo

Descripción:

*-* Primera versión del manual

# Objetivo de este documento

Este documento pretende ser una guía de instalación del sistema EEUtil-misc.

Para conseguir el WAR (fichero de despliegue de aplicaciones Web) existen dos alternativas:

* + Descargarlo directamente.
  + Realizar la compilación del código fuente y posterior empaquetamiento.

Si se ha descargado el WAR directamente puede ignorarse el punto 4.3

# Requisitos previos

Para poder realizar una instalación correcta y completa de EEUtil-misc, es necesario que el sistema cumpla los siguientes requisitos:

* + Base de datos: MySQL. Versión 5.5.9. **La aplicación no es compatible con la versión 5.6 de MySQL.**
  + **JDK 1.7** o superior.
  + Servidor de aplicaciones Apache Tomcat (versión 6 o superior), o similar.
  + Apache Maven 2 (si se desea trabajar con el código fuente).

# Instalación

En este apartado se detallan los pasos para realizar la instalación y configuración correcta de EEUtil-misc, tanto a través del war como del código fuente.

## Base de datos

En este apartado se describen los pasos para realizar la instalación de EEUtil-misc.

### Instalación de EEUtils (MySQL)

En primer lugar (tanto para el war como para el código fuente) es necesario crear las tablas y objetos necesarios para el funcionamiento de EEUtils. Los scripts se encuentran tanto en la carpeta de código fuente como en la de los binarios. Podemos encontrarlos en la ruta resources/scripts\_bbdd.

Existen varios scripts que incluye creación de tablas y los datos mínimos necesarios que estas deben llevar para poder arrancar la aplicación:

* + 01-create-tables.sql
  + 02-inserts.sql
  + 03-create\_procedures.sql
  + 04-insert-prueba-test-user.sql
  + 05-insert-admin-eeutil-misc.sql

Estos scripts son comunes a todos los proyectos de EEUtils, por lo que deben ejecutarse una única vez.

Se incluye el usuario “prueba” con la contraseña “test” para realizar pruebas.

Para crear nuevos usuarios, modifique el script 04-insert-prueba-test-user.sql con los datos del nuevo usuario.

El script 05-insert-admin-eeutil-misc.sql debe ser modificado para incluir los identificadores de los certificados con el que se vayan a logar en caso de utilizar eeutil-misc.

## Ficheros de configuración

Los ficheros de configuración se proporcionan en la ruta resources/config y pueden colocarse en la ruta que se desee. Además, este directorio debe indicarse en el arranque de la aplicación, asignando la ruta a la variable de entorno eeutil-misc.config.path:

java ….. -Deeutil-mic.config.path=<ruta\_ficheros\_configuracion>

También debe especificarse esta misma ruta para la variable local\_home\_app:

java ….. -Dlocal\_home\_app=<ruta\_ficheros\_configuracion>

Además configurar almacen de confianza para incluir los certificados de los servicios a los que llame . Ejemplo sería:

java ….. - Djavax.net.ssl.trustStore=<ruta\_almacen\_configurado\_como\_truststore>

java ….. - Djavax.net.ssl.trustStorePassword =<password\_para\_abrir\_almacen>

### log4j.properties

En este fichero se configurarán las propiedades de log de la aplicación. El que se proporciona en la distribución contiene necesita ser configurado en la siguiente propiedad, para establecer la ruta en la que dejar el fichero de log:

log4j.appender.GENERAL.File=/XXX/eeutil-misc.log

### database.properties (MySQL)

En este fichero se configuran las propiedades de la base de datos.

database.driverClassName=com.mysql.jdbc.Driver

database.url=jdbc:mysql://host:port/database

database.username=XXX

database.password=XXX

database.showSql=0

database.hibernateDialect=org.hibernate.dialect.MySQLDialect

No se proporciona ningún driver por lo que deberán incluir el driver especificado según la versión de su MySQL en las librerías del servidor de aplicaciones.

### afirma.properties

En este fichero se configuran las propiedades para consumir los servicios web de @firma:

# PARÁMETROS PARA LA CONEXIÓN CON @FIRMA

afirma.validarfirma.url=http://afirma.redsara.es/afirmaws/services/ValidarFirma

afirma.validarcertificado.url=http://afirma.redsara.es/afirmaws/services/ValidarCertificado

afirma.verifydss.url=http://afirma.redsara.es/afirmaws/services/DSSAfirmaVerify

afirma.idaplicacion=XXX

afirma.user=XXX

afirma.password=XXX

afirma.truststore=/XXX/truststore.jks

afirma.truststore.password=XXX

afirma.idaplicacionServidorDedicado=XXX

afirma.connectionTimeOut=120000

afirma.receiveTimeOut=120000

Se deben especificar las direcciones donde se encuentran los web services de @firma a utilizar en las variables afirma.validarfirma.url, afirma.validarcertificado.url y afirma.verifydss.url.

Se deben especificar los datos de autenticación de @firma en las variables afirma.idaplicacion, afirma.user y afirma.password.

La ruta del truststore se especifica en afirma.truststore y la contraseña del mismo en afirma.truststore.password.

### eeutil.properties

En este fichero se configura la ubicación del servicio web de eeutil-misc:

eeutil.misc.ws.url=http://XXX:XXX/eeutil-misc/ws/EeUtilService?wsdl

### clave.properties

Contiene información de configuración como es el nombre del proveedor, la url del módulo, la url a la que se retornará la respuesta del proxy (después de realizar la petición SAML) y el nivel de QAA. Además contiene todos los atributos que se le mostrarán al usuario para que elija los que quiere solicitar. Éstos están divididos en tres categorías: atributos personales, de negocio y legales.

#Provider Name

provider.name=XXX

#Sp Sector

sp.sector=XXX

#OPCIONAL (RELLENAR CON EL CODIGO SIA)

sp.aplication=XXX

#Sp Country

sp.country=LOCAL

#Sp Url

sp.url=http://HOST:PORT/eeutil-misc/login

#Sp return url

sp.return=http://HOST:PORT/eeutil-misc/accesoRedirectClave

#Sp QAA Level

sp.qaalevel=3

#SSO

# dónde enviar la confirmación de la petición

sp.logout.response.url=https://HOST:PORT/SPProxy/logoutvalidateResponse

# Dónde enviar la solicitud de Logout

cpeps.logout.url=Proxy/LogoutAction

#Avaliable Attributes for this SP

attribute.number=24

attribute1.name=eIdentifier

attribute2.name=givenName

attribute3.name=surname

attribute4.name=isdnie

attribute5.name=inheritedFamilyName

attribute6.name=adoptedFamilyName

attribute7.name=dateOfBirth

attribute8.name=countryCodeOfBirth

attribute9.name=placeOfBirth

attribute10.name=nationalityCode

attribute11.name=gender

attribute12.name=maritalStatus

attribute13.name=textResidenceAddress

attribute14.name=canonicalResidenceAddress

attribute15.name=eMail

attribute16.name=title

attribute17.name=residencePermit

attribute18.name=pseudonym

attribute19.name=age

attribute20.name=isAgeOver

attribute20.value=18

attribute21.name=signedDoc

attribute21.value=<dss\:SignRequest xmlns\:dss\='urn\:oasis\:names\:tc\:dss\:1.0\:core\:schema' RequestID\='\_%s' DocUI\='' Profile\='urn\:oasis\:names\:tc\:dss\:1.0\:profiles\:CAdES\:forms\:BES' NumberOfSigners\='1'><dss\:OptionalInputs/><dss\:InputDocuments><dss\:Document><dss\:Base64Data MimeType\='text/plain'>VGVzdCB0ZXh0</dss\:Base64Data></dss\:Document></dss\:InputDocuments></dss\:SignRequest>

attribute22.name=citizenQAALevel

#attribute22.name=fiscalNumber

attribute23.name=unknown

attribute24.name=registerType

businessAttribute.number=41

businessAttribute1.name=surnameOfHolder

businessAttribute2.name=givenNameOfHolder

businessAttribute3.name=dateOfBirthOfHolder

businessAttribute4.name=studentIdentificationCode

businessAttribute5.name=nameOfQualification

businessAttribute6.name=nameOfTitle

businessAttribute7.name=mainFieldsOfStudy

businessAttribute8.name=nameOfAwardingInstitution

businessAttribute9.name=statusOfAwardingInstitution

businessAttribute10.name=languageOfInstruction

businessAttribute11.name=languageOfAssesment

businessAttribute12.name=levelOfQualification

businessAttribute13.name=officialLengthOfProgramme

businessAttribute14.name=accessRequirement

businessAttribute15.name=modeOfStudy

businessAttribute16.name=programmeRequirements

businessAttribute17.name=programmeDetails

businessAttribute18.name=gradingScheme

businessAttribute19.name=gradingDistributionGuidance

businessAttribute20.name=overallClassification

businessAttribute21.name=accessToFurtherStudy

businessAttribute22.name=professionalStatus

businessAttribute23.name=additionalInformation

businessAttribute24.name=additionalInformationSources

businessAttribute25.name=certificationDate

businessAttribute26.name=certificationCapacity

businessAttribute27.name=higherEducationSystemInformation

businessAttribute28.name=yearOfStudy

businessAttribute29.name=averageGradeOfStudy

businessAttribute30.name=studyRecommendation

businessAttribute31.name=isEligibleForInternship

businessAttribute32.name=isStudent

businessAttribute33.name=isAcademicStaff

businessAttribute34.name=isTeacherOf

businessAttribute35.name=isCourseCoordinator

businessAttribute36.name=isAdminStaff

businessAttribute37.name=habilitation

businessAttribute38.name=acTitle

businessAttribute39.name=hasDegree

businessAttribute40.name=hasAccountInBank

businessAttribute41.name=isHealthCareProfessional

legalAttribute.number=14

legalAttribute1.name=eLPIdentifier

legalAttribute2.name=legalName

legalAttribute3.name=alternativeName

legalAttribute4.name=type

legalAttribute5.name=translatableType

legalAttribute6.name=status

legalAttribute7.name=activity

legalAttribute8.name=registeredAddress

legalAttribute9.name=registeredCanonicalAddress

legalAttribute10.name=contactInformation

legalAttribute11.name=fiscalNumber

legalAttribute12.name=representative

legalAttribute13.name=represented

legalAttribute14.name=mandateContent

clave.service.url=https://HOST:PORT/Proxy/ServiceProvider

# Si se quiere excluir alguno de los servicios de identificación

# podemos indicar el servicios separador por punto y coma

# aFirma;Stork;SS;AEAT;

clave.service.excludedIdPList=none

# Si se quiere forzar alguno de los servicios de identificación

# podemos indicar el servicio, en caso negativo none

# aFirma o Stork o SS o AEAT;

clave.service.forcedIdP=none

A continuación se explica cada de uno de los parámetros que hay que establecer:

* + **provider.name:** Indica el nombre del proveedor de acceso al servicio. Este valor debe de ser suministrado por Cl@ve.
  + **sp.sector:** Utilizar el mismo valor que la propiedad 'provider.name'.
  + **sp.aplication:** Este valor es utilizado por clave para diferenciar entre aplicaciones del mismo proveedor. Indicar el código SIA del proveedor en caso de disponer de él o un texto que identifique la aplicación.
  + **sp.url:** URL del proveedor del servicio.
  + **sp.return:** URL de retorno al proveedor del servicio.
  + **clave.service.url:** Indica la URL a la que el servicio invoca para realizar la autenticación.
  + **clave.service.excludedIdPList:** Si se quiere excluir alguno de los servicios de identificación podemos indicar el servicios separador por punto y coma aFirma;Stork;SS;AEAT;. Por defecto indicar el valor 'none'.
  + **clave.service.forcedIdP:** Si se quiere forzar alguno de los servicios de identificación podemos indicar el servicio, en caso negativo none aFirma o Stork o SS o AEAT; . Por defecto indicar el valor 'none'

El resto de propiedades que aparecen en el fichero no modifican su valor y deben de mantenerse en los diferentes entornos.

### SignModule\_SP.xml

Configuración de los parámetros de acceso al almacén donde debe encontrarse la clave privada para la generación del token de acceso a Clave.

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">

<properties>

<comment>SWModule sign with JKS.</comment>

<entry key=*"keystorePath"*>almacen.jks</entry>

<entry key=*"keyStorePassword"*>password</entry>

<entry key=*"keyPassword"*>XXX</entry>

<entry key=*"issuer"*>XXX</entry>

<entry key=*"serialNumber"*>XXX</entry>

<entry key=*"keystoreType"*>JKS</entry>

</properties>

A continuación se explica cada de uno de los parámetros que hay que establecer:

* + **keystorePath:** Indicar el nombre del almacén donde se encuentra el certificado para el acceso a Clave.
  + **keyStorePassword:** Indicar la contraseña de acceso al almacén.
  + **keyPassword**: Indicar la contraseña de acceso a la clave privada.
  + **issuer:** Indicar la información del emisor del certificado.
  + **serialNumber:** Indicar el número de serie del certificado.
  + **keystoreType:** Indicar el tipo de almacén.

### StorkSamlEngine\_SP.xml

Este fichero es utilizado por Clave y no precisa ningún cambio.

### SamlEngine.xml

Este fichero es utilizado por Clave y no precisa ningún cambio.

### credential.properties

En este fichero se configura la ruta del web service de Inside.

inside.admin.url=http://XXX:XX/inside/ws/AdminService?wsdl

### eni.properties

En este fichero se configuran las rutas para validar los esquemas ENI.

### facturae.properties

Este fichero se utiliza para configurar la ruta donde se encuentra el archivo para validar si un documento es una factura-e.

### loadTables-config.properties

Este fichero se utiliza para cargar las unidades orgánicas y las oficinas. Es necesario rellenar los siguientes datos:

ws.unidadesOrganicas.login=XXX

ws.unidadesOrganicas.password=XXX

ws.oficinas.login=XXX

ws.oficinas.password=XXX

### pdfA.properties

Este fichero se utiliza para establecer la configuración de la conversión de los documentos a pdf. Los datos para rellenar son:

# PDFA converter

pdfa.converter.key=XXX

# PDFA validator

pdfa.validator.key=XXX

### siaService.properties

Este fichero se utiliza para configurar el servicio web de consulta de códigos SIA.

ws.sia.login=XXX

ws.sia.password=XXX

### wsCrypto.properties

Este fichero se utiliza para configurar llamadas vía certificado en peticiones SOAP. Es necesario rellenar los siguientes datos:

org.apache.ws.security.crypto.merlin.keystore.password= XXX

org.apache.ws.security.crypto.merlin.keystore.private.password= XXX

passwordCertificado= XXX

org.apache.ws.security.crypto.merlin.truststore.password= XXX

### messages.properties

En este fichero se configuran los textos de la aplicación. No es necesario modificar nada.

## Instalación de EEUtil-misc a partir del código fuente

El código fuente de EEUtil-misc se distribuye en distintos módulos:

* + eeutil-mvn-base: Proyecto “padre”, no contiene código fuente, tan solo las propiedades y dependencias comunes.
  + eeutil-afirmaws-stub: Módulo que consume los servicios de @firma.
  + eeutil-comunes: Módulo que da soporte a otros módulos y que contiene interceptores SOAP, capa de seguridad, definición de excepciones de servicios web...
  + eeutil-inside-stub: Módulo que consume los servicios web de inside.
  + eeutil-model: Módulo que da soporte a otros módulos y que contiene clases POJO comunes.
  + eeutil-pdf-conversion: Módulo que da soporte sobre la conversión de documentos a formato pdf utilizando la librería de iText.
  + eeutil-pdf-conversion-igae: Módulo que da soporte sobre la conversión de documentos a formato pdf compatible con las necesidades de IGAE.
  + eeutil-services: Módulo que da soporte sobre clases DAO, servicios de aplicaciones y usuarios, rutas de configuración...
  + eeutil-util: Módulo que da soporte sobre certificados, archivos, mimetypes, xml...
  + load-tables:Módulo que da soporte a las transacciones con la base de datos.
  + eeutil-misc: Proyecto principal

### Instalación de librerías externas:

Para el desarrollo de EEUtil-misc a partir del código fuente es necesaria la inclusión de librerías externas que no se encuentran publicadas en los repositorios públicos de Maven. Estas librerías se encuentran en la carpeta resources/external-libs:

* + afirma
  + clave
  + igae
  + jodconverter
  + pdftools

Dentro de cada una de estas carpetas se incluyen las librerías necesarias y un documento install\_jars.txt con los comandos de Maven necesarios para la instalación de cada una de ellas en nuestro repositorio local.

### Descarga de dependencias:

La instalación de dependencias se puede realizar instalando los proyectos en el siguiente orden:

* + eeutil-afirmaws-stub
  + eeutil-comunes
  + eeutil-inside-stub
  + eeutil-model
  + eeutil-pdf-conversion
  + eeutil-pdf-conversion-igae
  + eeutil-services
  + eeutil-util
  + load-tables
  + eeutil-mvn-base
  + eeutil-misc

### Especificar versión del conector de base de datos MySQL:

En la raíz del proyecto eeutil-misc se encuentra el fichero pom.xml. Se debe buscar mysql-connector-java en el fichero y, si es necesario, cambiar la versión del conector de base de datos por el que corresponda a la versión de MySQL a la que se va a atacar.

### Ejecutar EEUtil-misc en entorno local

Una vez se han rellenado correctamente los ficheros de configuración, descargado las dependencias y compilado el código, la aplicación puede ser ejecutada en un entorno de servidor local, ejecutando la siguiente orden en el directorio eeutil-misc del proyecto:

mvn -Deeutil-misc.config.path=<ruta\_ficheros\_configuracion> -Dlocal\_home\_app=<ruta\_ficheros\_configuracion> clean jetty:run

Se puede deducir que la aplicación ha arrancado correctamente si en la consola no aparecen errores y si se puede acceder desde un navegador a la dirección:

Página de login de la aplicación web:

[http://](http://localhost:8090/inside)localhost:8091/eeutil-misc

Muestra la lista con los servicios web disponibles:

[http://](http://localhost:8090/inside)localhost:8091/eeutil-misc/ws

### Construcción del WAR a partir del código fuente

Para crear el WAR a partir del código fuente se ejecuta la siguiente orden en cada uno de los proyectos y en el orden indicado:

mvn clean package install

* + eeutil-afirmaws-stub
  + eeutil-comunes
  + eeutil-inside-stub
  + eeutil-model
  + eeutil-pdf-conversion
  + eeutil-pdf-conversion-igae
  + eeutil-services
  + eeutil-util
  + load-tables
  + eeutil-mvn-base
  + eeutil-misc

Finalmente, para el proyecto eeutil-misc se ejecutará:

mvn -P homovivus package war:war

En el directorio target del proyecto eeutil-misc se habrá creado el fichero de despliegue eeutil-misc.war, que contendrá dentro todas las librerías necesarias para la ejecución de la aplicación.

## Instalación de EEUtil-misc a partir del war.

El war sumistrado por el distribuible no contiene la librería de conexión de la base de datos, por tanto, se debe incluir en las librerías del servidor de aplicaciones el conector adecuado según la versión de la Base de Datos de MySQL a la que se va a atacar.

# Alta de nuevas aplicaciones

Las aplicaciones consumidoras de los Web Services de EEUtil-misc deberán estar dadas de alta en las tablas inside\_aplicaciones e inside\_aplicaciones\_propiedad.

La gestión de estas tablas puede realizarse directamente en la base de datos o bien desde la aplicación web de eeutil-misc.

El script de base de datos 05-insert-admin-eeutil-misc.sql debe ser modificado para incluir los identificadores de los certificados de los usuarios administradores que se vayan a logar en la aplicación web de eeutil-misc. Una vez dentro de la aplicación se pueden crear nuevas aplicaciones mediante un formulario.

En los scripts de generación de la base de datos se incluye en el documento 04-insert-prueba-test-user.sql un ejemplo con las sentencias SQL INSERT a ejecutar para crear un usuario. A continuación se describen los campos de las tablas.

### Tabla inside\_aplicaciones

* + **idaplicacion**: identificador de la aplicación.
  + **password**: hash del password de la aplicación, calculado con el algoritmo md5.
  + **activo**: Indica si la aplicación está activa, si su valor es false no le dejará realizar ninguna operación
  + **descripcion**: nombre o descripción de la aplicación.
  + **activa**: Indica si la aplicación puede está activa.
  + **tramitar**: Indica si la aplicación puede realizar tareas de trámites.
  + **sello**: Indica si la aplicación puede realizar tareas de sello electrónico.
  + **firma**: Indica si la aplicación puede realizar tareas de firma.
  + **email**: Email del responsable de la aplicación.
  + **telefono**: Teléfono de contacto del responsable de la aplicación.
  + **responsable**: Nombre del responsable de la aplicación.
  + **unidad**: DIR3 de la unidad orgánica a la que pertenece la aplicación.

### Tabla inside\_aplicaciones\_propiedad

* + **idaplicacion:** identificador de la aplicación.
  + **propiedad:** nombre de la propiedad.
  + **valor:** valor de la propiedad.

El valor que puede tomar la columna **propiedad** puede ser los siguientes:

* + **algoritmoFirmaDefecto:** algoritmo de firma que se utilizará por defecto
  + **formatoFirmaDefecto:** formato de la firma que se utilizará por defecto
  + **modoFirmaDefecto:** modo de la firma que se utilizará por defecto
  + **rutaKS:** ruta donde se encuentra el keystore a utilizar. Se recomienda hacer uso de la variable ${local\_home\_app} para que los keystores se encuentren en la misma ruta que los ficheros de configuración.
  + **tipoKS:** tipo del keystore especificado en la propiedad rutaKS.
  + **passwordKS:** password para poder acceder al keystore especificado en rutaKS.
  + **aliasCertificado:** alias del certificado a utilizar que se encuentra dentro del keystore especificado en rutaKS.
  + **passwordCertificado:** password del certificado especificado en aliasCertificado.
  + **ip.openoffice:** dirección IP donde se encuentra el servicio de OpenOffice.
  + **port.openoffice:** puerto donde se encuentra el servicio de OpenOffice.
  + **rutaLogo:** ruta donde se encuentra el logotipo a utilizar en la estampación de los documentos. Se recomienda hacer uso de la variable ${local\_home\_app} para que el logotipo se encuentre en la misma ruta que los ficheros de configuración.

# Autenticación por Cl@ve

Para la autenticación de un usuario es obligatorio el uso de [Cl@ve](http://clave.gob.es/clave_Home/clave.html). Para el acceso a Cl@ve es necesario:

1. Autenticarse contra Cl@ve a través de alguno de sus métodos. (DNI Electrónico -> Tener un certificado válido instalado en el navegador; Clave Pin -> Estar registrado; etc...)
2. Dar de alta al usuario en la aplicación EEUtil-misc (NIF Obligatorio) para trabajar en modo administrador en la aplicación web.

A continuación se explica el proceso general de atención en Cl@ve y posteriormente cómo configurarlo en EEUtil-misc.

El proceso de autenticación en Cl@ve consta de los siguientes pasos:

1. El usuario solicita la autenticación con Cl@ve
2. EEUtil-misc genera un token basado en un certificado proporcionado por Cl@ve y envía una solicitud de autenticación.
3. Cl@ve proporciona los distintos mecanismos de autenticación de usuario (DNI electrónico, Pin, etc)
4. El usuario se autentica con el mecanismo elegido
5. Una vez cl@ve lo autentica, redirige la petición de nuevo a EEUtil-misc suministrando la información del usuario autenticado (DNI, Nombre, etc)
6. EEUtil-misc valida los datos proporcionados por Cl@ve contra la Base de Datos (chequea el DNI)
7. Si se encuentran coincidencias se da acceso a la aplicación

El proceso de autenticación en EEUtil-misc consta de dos partes:

1. Configurar el uso de Cl@ve como herramienta de autentificación

Se necesita solicitar a Cl@ve un certificado para hacer uso del servicio de Cl@ve. El certificado proporcionado por Cl@ve se instalará en un almacén JKS (en forja existe un almacén vacío para tal efecto "almacen.jks" con contraseña "password"). Además se configurará en el fichero "SignModule\_SP.xml" con los datos de acceso al almacén y al certificado.

Cl@ve proporcionará información del "provider" que se deberá de configurar en el fichero **"**clave.properties" las siguientes propiedades:

* provider.name=Información suministrada por Cl@ve
* sp.sector=Información suministrada por Cl@ve
* sp.aplication=Opcionalmente rellenar con el código SIA, si no se dispone, usar un texto descriptivo para identificar accesos.
* sp.url=https://nombre\_del\_servidor/eeutil-misc/acceso
* sp.return=http://nombre\_del\_servidor/eeutil-misc/accesoClave
* clave.service.url=https://pasarela.clave.gob.es/Proxy/ServiceProvider

1. Dar de alta a los usuarios en la tabla usuario\_credencial con su NIF. Editar el script de base de datos 05-insert-admin-eeutil-misc.sql para insertar los NIF de los administradores.

# Web Services externos a EEUtil-misc

Algunas funcionalidades de EEUtil-misc dependen de la implementación de Web Services externos. Estos Web Services tienen que cumplir una especificación determinada.

La especificación de estos servicios web se encuentra en la carpeta /Web-Services-Externos.

- **Inside**: Especificación del WS de inside para creación de usuarios desde interfaz de eeutil-misc.

Se implementa un Cliente Web Service de la especificación establecida por el fichero WSDL que se encuentran en la ruta /Web-Services-Externos /inside.

- **EEUtil**: Especificación del WS de eeutil-oper-firma para la obtención de información y validación de firmas y certificados.

Se implementa un Cliente Web Service de la especificación establecida por el fichero WSDL que se encuentran en la ruta /Web-Services-Externos /eeutil.

- **DIR3**: Especificación de los WS de DIR3 para la obtención de unidades orgánicas y oficinas.

Se implementa un Cliente Web Service por cada una de las especificaciones establecidas por los ficheros WSDL que se encuentran en la ruta /Web-Services-Externos /dir3.

- **SIA**: Especificación del WS de SIA para la consulta de códigos SIA.

Se implementa un Cliente Web Service de la especificación establecida por el fichero WSDL que se encuentran en la ruta /Web-Services-Externos /sia.

# Anexo Tomcat

Los ficheros de configuración se proporcionan en la ruta resources/config y pueden ubicarse en la ruta que se desee**, teniendo en cuenta se accederá a ellos mediante la variable ${local\_home\_app}**.

### Instalación en Tomcat

Esta es la instalación más sencilla de la aplicación. Es necesario un contenedor de servlets tipo Tomcat. Se recomienda en su versión 6 o mayor. Para llevar a cabo la instalación del war descargue la última versión del área de descargas.

Para el caso de Tomcat, el war puede ser instalado a través del manager de despliegue o simplemente ubicarlo en la carpeta de aplicaciones webapp. Busque la forma si no se trata de Tomcat.

En la carpeta /lib debe ubicar la librería con el conector de base de datos de la versión adecuada a la versión de MySQL que ataque en caso de utilizar directamente el war proporcionado por el distribuible.

Por último, queda indicar las variables de configuración de EEUtil-misc. Recomendamos que para dichos parámetros se utilice la variable de entorno JAVA\_OPTS, de forma que esté disponible al ejecutar Tomcat:

|  |
| --- |
| JAVA\_OPTS="$JAVA\_OPTS -Dlocal\_home\_app=<ruta\_ficheros\_configuracion> -Deeutil-misc.config.path<ruta\_ficheros\_configuracion>" |