

## Automatización de pruebas con SoapUI

**Fecha: 18-5-2011**

**Referencia:**

EJIE S.A.  
Mediterráneo, 14  
Tel. 945 01 73 00\*  
Fax. 945 01 73 01  
01010 Vitoria-Gasteiz  
Posta-kutxatila / Apartado: 809  
01080 Vitoria-Gasteiz  
[www.ejie.es](http://www.ejie.es)

Este documento es propiedad de EJIE, S.A. y su contenido es confidencial. Este documento no puede ser reproducido, en su totalidad o parcialmente, ni mostrado a otros, ni utilizado para otros propósitos que los que han originado su entrega, sin el previo permiso escrito de EJIE, S.A.. En el caso de ser entregado en virtud de un contrato, su utilización estará limitada a lo expresamente autorizado en dicho contrato. EJIE, S.A. no podrá ser considerada responsable de eventuales errores u omisiones en la edición del documento.

# Control de documentación

Título de documento: **Automatización de pruebas con SoapUI**

## Histórico de versiones

Código:

Versión: 1.1

Fecha:

Resumen de cambios:

## Cambios producidos desde la última versión

Se eliminan ejemplos y se referencia al manual de de SoapUI.

## Control de difusión

Responsable:

Aprobado por:

Firma:

Fecha:

Distribución:

## Referencias de archivo

Autor: CAC

Nombre archivo:

Localización:

	Capítulo/sección	Página
1	Introducción	2
2	A tener en cuenta en los proyectos de SoapUI	2
3	Pasos para la automatización	3
3.1	Estructura en SVN	3
3.2	Ejecutar las pruebas	3
3.3	Resultados	3

## 1 Introducción

A continuación se detallan los pasos a realizar con el objetivo de definir un proyecto en SoapUI que permita después ser ejecutado automáticamente en el entorno de desarrollo y pruebas.

Este documento debe leerse previo al desarrollo del caso de prueba de soapUI, dado que para conseguir la automatización de estos casos de prueba es necesario seguir ciertas normas que trataremos de detallar.

Conviene asimilar los ejemplos incluidos en este documento.

La automatización de soapUI actual está realizada con la versión **3.6.1** del producto.

Se recomienda la lectura del manual de SoapUI (

<https://ataria.ejie.eus/docs/DocBinBlob.aspx?Id={F55B7494-9273-4180-9D08-0FCB67489FB7}&Download=1>

) previo a la automatización de pruebas.

## 2 A tener en cuenta en los proyectos de SoapUI

### ES NECESARIO

1. Los proyectos de SoapUI se llamarán aaa<sup>1</sup> y los ficheros xml del proyecto aaa-soapui-project.xml.
2. Definir las propiedades que dependen del entorno a nivel del proyecto. VOLCAR A FICHERO ESTAS PROPIEDADES, QUE SERÁN DIFERENTES PARA CADA ENTORNO. Este fichero se deberá traspasar a cada entorno y deberá llamarse aaa-soapui.properties.  
**Es decir, si tengo un fichero aaa-soapui-project.xml dentro habrá un proyecto llamado “aaa”. Es importante que el proyecto se llame igual que el fichero, y que el fichero de properties se llame aaa-soapui.properties, para que aplique las propiedades correctamente.**
3. Todos los endpoints que no sean definidos como variables de proyecto deberán ser siempre de un mismo host:puerto. En este caso, no podemos definir un fichero de properties, y la tarea de automatización pasará como variable de entorno el host:puesto del endpoint. En este caso, entonces, el mismo proyecto de soapUI nos sirve para diferentes entornos, mientras que en el caso anterior, tendremos un fichero de properties por entorno. Tendremos que crear un tag para pruebas...
4. Definir las propiedades de transferencia de los casos de prueba en properties (o bien transferir directamente, de respuesta a petición con transformaciones xpath) del caso de prueba. NUNCA VOLCAR A FICHERO ESTAS PROPIEDADES.

### SE ACONSEJA

1. Crear un “Conjunto de pruebas” que agrupen en bloques lógicos casos funcionales es mejor que tener un solo “Conjunto de casos de pruebas”.
2. No crear casos de prueba con muchos pasos; es decir, intentar que un caso de prueba tenga los pasos necesarios y suficientes para su ejecución.

<sup>1</sup> aaa deberá ser [a..z][0..9] (1..n). Por ejemplo aaa=a, aaa=PruebaWSp21.

### 3 Pasos para la automatización

#### 3.1 Estructura en SVN

Subiremos los proyectos y sus ficheros de propiedad a las carpetas de SVN correspondientes.

Para cada módulo “WAR” que contenga la carpeta:

**test-system/soapui**

ó

**test-integration/soapui**

La descarga de SVN y mapeo en estructura de servidor la realizaran las tareas de compilación de Test. Las tareas de ejecución de test de soapui ejecutarán por defecto TODOS LOS CASOS DE PRUEBA que tengamos definidos en cada proyecto de soapui que dejemos en esas rutas.

Si necesitamos un fichero de propiedades, deberá llamarse aaa-soapui.properties y ubicarse en el mismo sitio que el proyecto de soapui. Caso de tener más de un proyecto de soapui en un mismo directorio de soapui se debe saber que el fichero de propiedades es único y común para todos.

Una vez probado el caso de prueba en el entorno de desarrollo de Ejie, deberemos dejar en traspaso los proyectos de soapUI y sus ficheros de propiedades, adecuándolos al entorno de pruebas.

**Si el proyecto de SoapUI usa un keystore “p12” porque es un servicio de tipo WS-Security (Ver apartado 3.1), será necesario cambiar las rutas; esto es, dejar en el repositorio de SVN el proyecto de soapui con referencias a los ficheros en /datos/aaa/p12-aaa.p12 por ejemplo.**

#### 3.2 Ejecutar las pruebas

Lanzaremos la tarea aprovisionada en Jenkins “xxx\_test”. Estas tareas lanzan todas las pruebas automatizadas del proyecto, tanto las de junit, como las de selenium como estas de soapui...

En desarrollo: [www.otc.ejiedes.net/jenkins/login](http://www.otc.ejiedes.net/jenkins/login)

En pruebas: [www.otc.ejiepru.net/jenkins/login](http://www.otc.ejiepru.net/jenkins/login)

#### 3.3 Resultados

Los resultados los podremos ver como casos de prueba ejecutados en TestLink, dentro del agrupamiento “automatizadas”. Tanto el entorno de desarrollo como el de pruebas alimentan la gestión de pruebas del entorno de desarrollo, pero se distinguen planes de prueba para cada entorno.

En desarrollo: [www.otc.ejiedes.net/testlink](http://www.otc.ejiedes.net/testlink) (Plan de pruebas “Automáticas Desarrollo”)

En pruebas: [www.otc.ejiedes.net/testlink](http://www.otc.ejiedes.net/testlink) (Plan de pruebas “Automáticas Pruebas”)

Caso de que un caso de prueba haya fallado, se debería haber creado una incidencia en Mantis.