

¿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)



Powered by **autentia** Hosting patrocinado por **enredados**

- [Inicio](#)
- [Quienes somos](#)
- [Tutoriales](#)
- [Formación](#)
- [Empleo](#)
- [Colabora](#)
- [Comunidad](#)
- [Libro de Visitas](#)
- [Comic](#)

NUEVO ¿Quieres saber cuánto ganas en relación al mercado? pincha aquí...

[Ver cursos que ofrece Autentia](#)

[Descargar comics en PDF y alta resolución](#)



[NUEVO!] 2008-04-20

2008-04-14

2008-04-07

2008-04-01

Estamos escribiendo un libro sobre la profesión informática y estas viñetas formarán parte de él. Puedes opinar en la seccion [comic](#).

Tutorial desarrollado por



Jose Manuel Sánchez Suárez

Consultor tecnológico de desarrollo de proyectos informáticos. Diseñador de Adictos Al Trabajo 2.0

Puedes encontrarme en [Autentia](#)

Somos expertos en Java/J2EE

Catálogo de servicios de Autentia

[Descargar \(6,2 MB\)](#)

[Descargar en versión comic \(17 MB\)](#)

[AdictosAlTrabajo.com](#) es el Web de difusión de conocimiento de [Autentia](#).



[Catálogo de cursos](#)

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

Fecha de creación del tutorial: 2008-05-05

Prototype.js: la sombra que se esconde detrás de todo.

0. Índice de contenidos.

- 1. Introducción
- 2. La sombra.
- 3. Funciones principales.
- 4. Ejemplo: recarga controlada de un selector con prototype.js.
- 5. Conclusiones

1. Introducción

[Prototype.js](#) se define así mismo como un framework, escrito en javascript, para desarrollar aplicaciones web dinámicas. Si bien, más que un framework, yo diría que es una librería que implementa toda esa funcionalidad que siempre quisimos que formase parte del estándar ECMAScript, proporciona:

- una extensión del lenguaje,
- un acceso sencillo al árbol DOM, con independencia del navegador del cliente,
- un soporte claro y sencillo de Ajax,
- soporte para JSON,
- y otras muchas utilidades que siempre hubieses querido tener trabajando con javascript.

Más de uno puede decir "a buenas horas mangas verdes", pues no tiene años... y es verdad, y otros dirán "yo no quiero saber nada de javascript!!!!", no con menos razón si tenemos en cuenta que nosotros mismos estamos proponiendo el uso de frameworks y, dentro de estos, librerías de componentes que nos aislen de la problemática de trabajar con un lenguaje del lado del cliente.

Voy a intentar justificar el por qué de este tutorial, aún hoy día. De todos los proyectos que conocéis, ¿cuántos son nuevos desarrollos y cuántos están en fase de mantenimiento?

- de los nuevos desarrollos, ¿todos se basan en JSF y librerías de componentes del tipo ICEfaces o RichFaces?
- de los que están en fase de mantenimiento, ¿cuántos están contruidos sobre un framework conocido y con soporte de ajax?
- de los que no responden a ningún framework o no tienen soporte de ajax, ¿cuántos tienen restricciones (software en el que corren, hardware, presupuesto,...) de modo que no pueden ser migrados a un sistema "más moderno", o no pueden incluir nuevas librerías?

Si no conocéis ningún caso que pueda encuadrarse dentro del último grupo, sentiros afortunados, y pedir por no caer en un mantenimiento o incluso evolutivo de una aplicación que tenga más de 4 años, que no responda a patrones y tenga su propia arquitectura.

Este tutorial va dirigido:

- a quien necesite imprimir algo de dinamismo a la aplicación que está manteniendo o sobre la que tiene que realizar un evolutivo y no puede evitar en el manejo "a mano" de javascript. Vamos a intentar que ese manejo sea más sencillo y a

Catálogo de servicios Autentia (PDF 6,2MB)



En formato comic...



Web
 www.adictosaltrabajo.com

Últimos tutoriales

2008-05-05
[Prototype.js: la sombra que se esconde detrás de todo](#)

2008-05-05
[Creación de una aplicación web con SpringMVC desde 0](#)

2008-05-05
[Cómo integrar Eastwood en nuestras aplicaciones web](#)

2008-04-28
[Cómo lanzar aplicaciones web desde Maven con Jetty](#)

2008-04-28
[Solución al problema de la exportación a HTML de informes JasperReports](#)

2008-04-21
[Proyecto Sakai: Una plataforma de e-learning libre \(II\)](#)

2008-04-21
[Proyecto Sakai: Una plataforma de e-learning libre \(I\)](#)

2008-04-19
[Ampliación de la comparativa de antivirus freeware y opensource](#)

2008-04-19
[Log4J, SMTPAppender: Envío de trazas de log por email](#)

2008-04-17
[Indentación del código fuente](#)

Últimas ofertas de empleo

2008-04-29

incluir soporte para Ajax de una forma muy simple.

- a mi cuñado Chema, que está realizando su primer formulario en php dentro de una página web estática y sencilla, en la que no tiene sentido el uso de ningún framework, y cada vez que el cliente selecciona una provincia, se recarga toda la página para dar valor a las poblaciones de la misma, perdiéndose el valor de los campos de tipo `<input type="file" />`. Vamos a ver un ejemplo de recarga controlada de selectores vía Ajax con prototypejs.
- a quien utilice librerías de componentes Ajax, que ya encapsulen todo el código javascript que se ejecuta en cliente, porque prototypejs está detrás de todo...

2. La sombra que se esconde detrás de todo.

A finales de 2006, una encuesta de popularidad de ajaxian.com afirmaba que prototypejs era el framework más usado con un 43%, y script.aculo.us el segundo con un 33%, sin tener en cuenta que por debajo de script.aculo.us está prototypejs (43 + 33 = 76).

Ahora seguramente esto no sea así y el ranking de popularidad haya cambiado. Tenemos otras librerías que, sobre la idea base de prototype, implementan mucha más funcionalidad:

- [dojo](#)
- [qooxdoo](#)
- [rico](#)
- [jQuery.js](#) más sencilla, aporta compatibilidad con el propio prototypejs.

Si tuviésemos que escribir hoy día una librería de componentes JSF basándonos en un framework de javascript, seguro que examinaríamos primero estas librerías, pero la realidad es que las librerías de componentes que usamos hoy día se escribieron hace un par de años y tomaron como base la tecnología más popular en ese momento.

Afirmar que "prototypejs es la sombra que se esconde detrás de todo" es un poco ambicioso, pero podéis comprobar que:

- ICEfaces se basa en script.aculo.us para incluir "efectos especiales" y ésta a su vez prototypejs, solo tenéis que abrir el fichero `icefaces-d2d.js` y ver las primeras líneas de código,
- `Ajax4j` que ahora forma parte de `RichFaces` importa el fichero `framework.pack.js`, al igual que en el caso anterior, revisad las primeras líneas del código del fichero,
- cuando en Ruby On Rails necesitamos realizar alguna recarga controlada de la página se importan las librerías vía `<%= javascript_include_tag 'prototype' %>` que importa directamente la librería prototypejs que incorpora el RoR.
- el framework de desarrollo de aplicaciones php de [Zend](#) importa, del mismo modo, directamente la librería prototypejs, ...

Podríamos continuar con algún ejemplo más, pero con los dos primeros, para los que desarrollamos habitualmente aplicaciones JEE, nos debería bastar, puesto que son los más usados.

3. Funciones principales.

Para obtener una información detallada sobre las múltiples posibilidades que nos proporciona prototypejs, echad un vistazo a la [documentación del site](#). Ahora ya tenemos una documentación detallada en el site de la librería, y no solo existe la documentación de antaño, la de [Sergio Pereira](#).

Aquí solo voy a comentar brevemente las funciones que me parecen más relevantes.

3.1. Acceso al árbol DOM.

Prototypejs nos da una serie de funciones que nos facilitan el acceso a los nodos del árbol DOM y sus atributos:

- `$("#")`: la función dolar obtiene un nodo por id como si de un `getElementById()` se tratase, pero abstrayéndonos del browser en el que se ejecute,
- `F()`: la función `$F` obtiene todos los valores de un formulario, para enviarlos a través de una petición Ajax por ejemplo,
- `$A()`: la función `$A` convierte un objeto a un array de objetos
- `$H()`: la función `$H` convierte un objeto en un hash
- `document.getElementsByTagName()`: devuelve una colección de nodos que responden al mismo nombre,
- mediante el objeto `Form`: podemos obtener una serialización del formulario: `Form.serialize()`

3.2. Uso de Ajax.

Prototypejs proporciona soporte de ajax a través del objeto `Ajax`, de modo que no tenemos que preocuparnos si el navegador da soporte al objeto `XMLHttpRequest` de forma nativa o a través de un control `ActiveX`. Podemos realizar:

- peticiones síncronas al servidor:

```
view plain print ?
01. var response = new Ajax.Request('/some_url', {asynchronous: false, method: 'post', parameters: par
02. var items = resp.responseXML.getElementsByTagName( "item" );
```

- peticiones asíncronas al servidor:

```
view plain print ?
01. new Ajax.Request('/some_url', {method: 'get', onSuccess: function(transport){
02. var response = transport.responseText || "no response text";
03. alert("Success! \n\n" + response);
04. },
05. onFailure: function(){ alert( 'Something went wrong...' ) }}
06. );
```

- recarga controlada de componentes:

```
view plain print ?
01. new Ajax.Updater( 'products', '/some_url', {method: 'get' }
```

4. Ejemplo: recarga controlada de un selector con prototypejs.

Vamos a realizar un ejemplo sencillo, la recarga controlada de un selector en una página html básica, en función de la selección realizada en otro.

El código de la aplicación se podría ejecutar en cualquier servidor web, de aplicaciones,... salvo por el hecho de que la página que da servicio al listado de poblaciones es un jsp, pero podría ser un php, un cgi, un servlet, cualquier "cosa" del lado del servidor que devolviese un xml en función de un parámetro.

Vamos a por el html:

Otras - Mecánica - ASTURIAS.

2008-04-28
T. Información -
Administrador Sistemas UNIX
/ NT - CIUDAD REAL.

2008-04-23
T. Información - Analista /
Programador - BARCELONA.

2008-04-23
T. Información - Analista /
Programador - BARCELONA.

2008-04-23
T. Información - Analista /
Programador - BARCELONA.

Anuncios Google

```

view plain print ?
01. <?xml version="1.0" encoding="ISO-8859-1" ?>
02. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-ti
03. <html xmlns="http://www.w3.org/1999/xhtml" >
04. <head>
05. <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
06. <title>Recarga de selectores</title>
07. <script type="text/javascript" src="script/prototype-1.6.0.2.js"></script>
08. <script type="text/javascript" src="script/reloaders.js"></script>
09. <link rel="stylesheet" href="style/default.css" type="text/css"></link>
10. </head>
11. <body>
12. <div id="layout">
13. <fieldset>
14. <legend>Ajax select loader</legend>
15. <form name="ftest01" action="#">
16. <label for="provincias">Provincias</label>
17. <select name="provincias" id="provincias"
18. <option value="0">(Seleccione una provincia)</option>
19. <option value="1">Madrid</option>
20. <option value="2">Málaga</option>
21. </select>
22. <label for="poblaciones">Población</label>
23. <select name="poblaciones" id="poblaciones">
24. <option value="">(Seleccione una provincia)</option>
25. </select>
26. </form>
27. </fieldset>
28. </div>
29. </body>
30. </html>
31.

```

- **07:** Nos descargamos [la última librería de prototypejs](#) y la importamos en la cabecera.
- **08:** Importamos la librería que controlará la recarga del selector, cuyo código incluimos más abajo.
- **17:** Incluimos el selector de provincias con un par de ellas para la prueba y asignamos al evento onchange la llamada a una función javascript que implementaremos en la librería que importamos en la línea 08, pasándole como parámetro todo el objeto selector de provincias.
- **24:** Incluimos el selector de provincias que se recargará de forma dinámica. Le asignamos un name, en el ámbito del formulario (para poder obtener los parámetros de la request en su caso) y un id, para poder acceder al mismo a través de la lectura del árbol DOM.

La librería de javascript **script/reloaders.js**, que controla las peticiones y la recarga del selector, sería la siguiente (incluye un comentario para cada línea):

```

view plain print ?
01. function reloadPoblaciones(selector)
02. {
03. // La url del servicio que nos devuelve el listado de poblaciones para una provincia
04. var surl = "services/poblaciones.jsp";
05.
06. // parámetro con el identificador de la provincia
07. var params = "id="+selector.value;
08. // debugger;
09.
10. // prototype Ajax.Request: petición de servicio
11. var response = new Ajax.Request(surl, {asynchronous: false, method: 'post', parameters: params}).tr;
12.
13. // obtenemos el listado de poblaciones del xml de respuesta del servicio
14. var items = response.responseXML.getElementsByTagName( "poblacion");
15.
16. // prototype getElementById: obtenemos el selector de poblaciones del árbol DOM
17. var select = $("poblaciones");
18. select.options.length = 0;
19.
20. // si tenemos alguna población para la provincia
21. if( items.length != 0 )
22. {
23. // asignamos un valor por defecto en el primer option del selector de poblaciones
24. select.options.add( new Option("Seleccione una población", ""));
25. // por cada una de las poblaciones obtenidas
26. for(i = 0 ; i < items.length; i++)
27. {
28. // añadimos un option al selector de poblaciones
29. if (items[i].firstChild != null)
30. {
31. select.options.add( new Option(items[i].firstChild.data,items[i].attributes[ 0 ].value));
32. }
33. }
34. // seleccionamos el option por defecto
35. select.options[0].selected = "selected";
36. }
37. }

```

El jsp **services/poblaciones.jsp**, que da servicio a las peticiones desde el cliente devolviendo un xml en función de un parámetro, sería el siguiente:

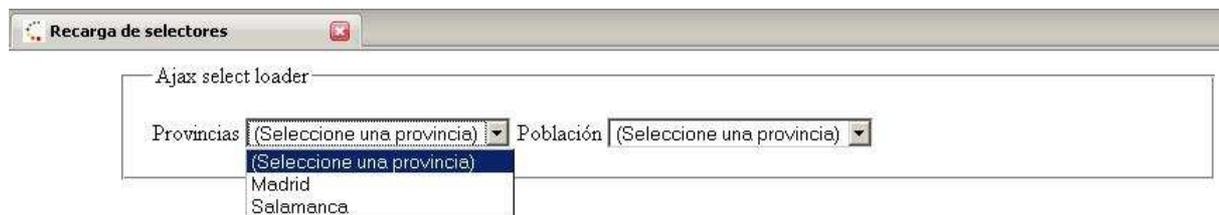
```

view plain print ?
01. <?xml version="1.0" encoding="ISO-8859-1"?>
02. <%
03.     response.setContentType("text/xml");
04.     String[] poblaciones = null;
05.     String id = request.getParameter("id");
06.     if (id == null) id = "1";
07. %>
08. <provincia id="<%=id%>">
09. <%
10.     if (id.equals("1"))
11.     {
12.         poblaciones = new String[]{"Alcobendas","Miraflores de la Sierra","San Fernando de Henares"}
13.     }
14.     else if (id.equals("2"))
15.     {
16.         poblaciones = new String[]{"El Palo","La Cala del Moral","Rincon de la Victoria"};
17.     }
18.     else
19.     {
20.         poblaciones = new String[]{" "};
21.     }
22.     for(int i = 0; i < poblaciones.length; i++)
23.     {
24. %>
25. <poblacion id="<%=i%>"><%=poblaciones[i]%></poblacion>
26. <%
27.     }
28. %>
29. </provincia>

```

Es un jsp básico y realizado solo para este ejemplo!, me abstengo de comentar como devuelve el resultado en función del parámetro.

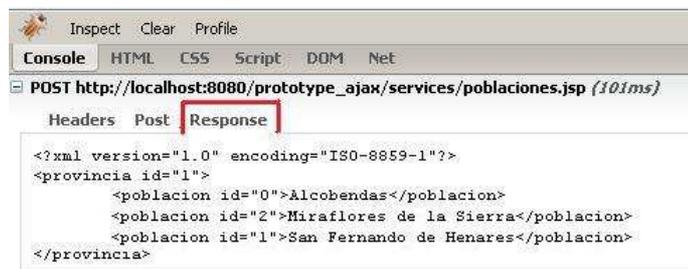
El resultado del ejemplo es la siguiente página con la funcionalidad prometida:



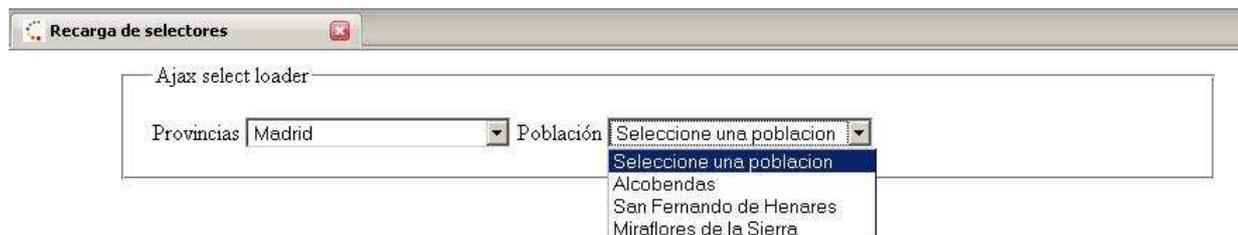
Comprobamos la conexión vía Ajax usando **Firebug**, el plugin imprescindible para desarrolladores de Mozilla Firefox. Bajo la opción **console** podemos ver los detalles de la conexión, en este caso síncrona, con el servidor:



Bajo la pestaña post, podemos ver el valor del parámetro que recibe la jsp que nos da servicio.



Bajo la pestaña response, el valor del xml que devuelve la jsp.



El selector de poblaciones se recarga en función de la provincia seleccionada.

Para que os quede con el mismo aspecto, podéis incluir la siguiente css (**style/default.css**):

```

view plain print ?
01. #layout {
02.     width:780px;
03.     margin:auto;
04. }
05. fieldset{
06.     padding:20px;
07. }

```

5. Conclusiones.

El desarrollo con el tipo de frameworks que proponemos desde **Autentia** en **nuestros tutoriales** nos abstrae del problema de bajar al lenguaje del cliente para realizar el tipo de operaciones que se plantean en este tutorial.

Si bien, no es bueno olvidarse de cómo realizar este tipo de tareas a mano, puesto que, en función del proyecto en el que estemos y el tipo de arquitectura en el que se base, quizás necesitemos "tirar de prototypejs". Máxime si tenemos en cuenta que es la librería de javascript en la que se basan las librerías de componentes JSF con las que estamos trabajando.

Un saludo.

Jose

- Puedes opinar sobre este tutorial [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. 



Esta obra está licenciada bajo [licencia Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5](#)

Recuerda

Autentia te regala la mayoría del conocimiento aquí compartido ([Ver todos los tutoriales](#)). Somos expertos en: J2EE, Struts, JSF, C++, OOP, UML, UP, Patrones de diseño ... y muchas otras cosas.

¿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?

Somos pocos, somos buenos, estamos motivados y nos gusta lo que hacemos ...

Autentia = Soporte a Desarrollo & Formación.

info@autentia.com



Servicio de notificaciones:

Si deseas que te enviemos un correo electrónico cuando introduzcamos nuevos tutoriales.

Formulario de suscripción a novedades:

E-mail

Tutoriales recomendados

Nombre	Resumen	Fecha	Visitas	pdf
Ajax con Java Fácil	En este tutorial realizado por Javier Antoniucci vamos a ver una implementación de Ajax llamada Direct Web Remoting (DWR).	2007-05-03	6372	pdf
Lucene: Analyzers, stemming y búsqueda de documentos similares.	En este tutorial vamos a ver cómo implementar un analizador semántico en nuestro idioma, potenciando la indexación y búsqueda, para terminar analizando la viabilidad de realizar búsquedas de documentos similares.	2008-02-22	1168	pdf
Ejemplo de web con ICEfaces	Creación de una web paso a paso con ICEFaces, Tomcat 5.5 y Eclipse	2008-01-16	2029	pdf
Una aplicación AJAX hecha a mano	En este tutorial se muestra cómo hacer una página web que, usando AJAX, accede a un servicio web SOAP. Para ello sólo se usa JavaScript, sin nada de código en el servidor.	2006-11-02	7874	pdf
Icefaces, JBoss, Maven2 y EJB3: Parte 1	Primer tutorial para construir proyecto con ICEfaces, JBoss, Maven2 y EJB3	2008-01-17	2401	pdf
Introducción a la tecnología AJAX	En este tutorial os mostramos los fundamentos de la tecnología AJAX, junto a un ejemplo y orientación sobre su utilización en el desarrollo de aplicaciones web en general	2006-02-17	24311	pdf
Integración de JSF 1.2, Facelets e ICEFaces en Tomcat 6	Integración de JSF 1.2, Facelets e ICEFaces en Tomcat 6	2007-12-10	2558	pdf
JAAS (Java Authentication and Authorization Service)	Acceso a servicios de control de autenticación y acceso mediante JAAS	2007-12-10	3212	pdf
Icefaces, JBoss, Maven2 y EJB3: Parte 2	Segunda parte del tutorial. Aquí haremos la configuración de la parte del módulo Web y del módulo Modelo	2008-01-17	1750	pdf
Framework para indexación de documentos en Lucene: LIUS	En este tutorial vamos a ver como usar el framework LIUS (Lucene Index Update and Search) para indexar documentos en el motor de búsqueda textual Lucene	2007-10-23	1907	pdf

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.