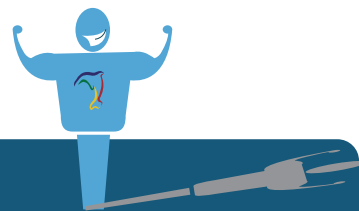


¿Qué ofrece Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
Ese apoyo que siempre quiso tener...

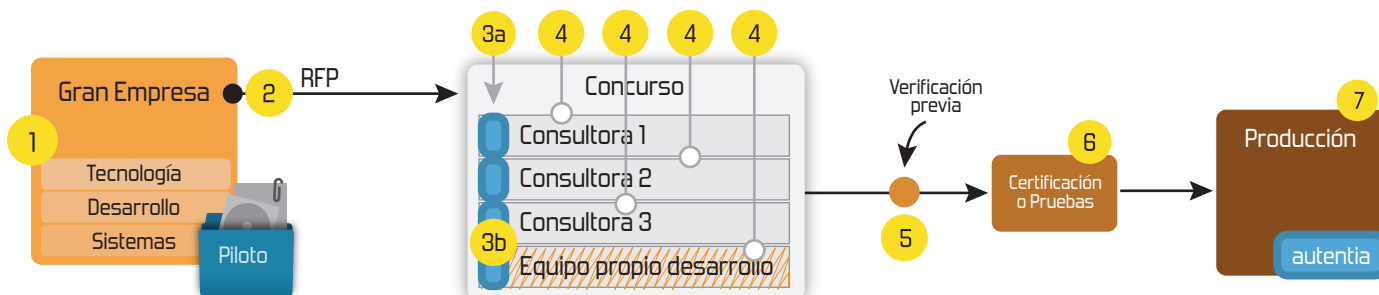
1. Desarrollo de componentes y proyectos a medida



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

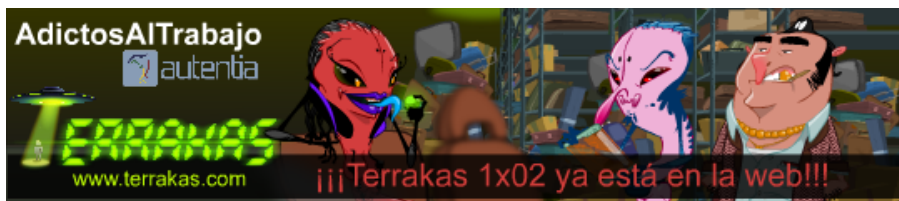
Tareas programadas (Quartz)
Gestor documental (Alfresco)
Inversión de control (Spring)

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)


[Entra er](#)

[Inicio](#)
[Quiénes somos](#)
[Tutoriales](#)
[Formación](#)
[Comparador de salarios](#)
[Nuestro libro](#)
[Charlas](#)

» Estás en: [Inicio](#) [Tutoriales](#) [Cómo forzar al navegador a que descargue recursos estáticos cuando los tien...](#)



Miguel Arlandy Rodríguez

Consultor tecnológico de desarrollo de proyectos informáticos.

Puedes encontrarme en [Autentia](#): Ofrecemos servicios de soporte a desarrollo, factoría y formación

Somos expertos en Java/JEE

[Ver todos los tutoriales del autor](#)

**Cat
Aut**

Fecha de publicación del tutorial: 2012-02-27

Tutorial visitado 2 veces [Descargar en PDF](#)

Cómo forzar al navegador a que descargue recursos estáticos cuando los tiene cacheados.

0. Índice de contenidos.

- [1. Introducción.](#)
- [2. Entorno.](#)
- [3. El problema.](#)
- [4. La solución.](#)
- [5. Conclusiones.](#)

1. Introducción

Los recursos estáticos, tales como hojas de estilo, librerías javascript o imágenes son componentes habituales en las aplicaciones web. El navegador cuando accede a una página, se descarga estos componentes para poder presentar el contenido de la misma. Los navegadores cuentan con una caché para este tipo de recursos estáticos que le permite no tener que descargárselos cada vez que se accede a una página que requiere el uso de este componente. De esta manera se consigue reducir el tráfico entre cliente y servidor reduciendo asimismo el tiempo de carga de la página.

No obstante, esta magnífica solución puede convertirse a veces en un problema. Supongamos que nuestra aplicación ya está en producción y, en la siguiente versión que desplegamos cambiamos una hoja de estilos (cambiamos su contenido no su nombre). ¿Qué sucederá cuando un usuario que ya ha accedido a una de nuestras páginas vuelva a acceder tras los cambios en hoja de estilos? Pues muy probablemente, que el navegador no descargará el contenido de la .css ya que la tendrá cacheada de otras veces y, por tanto, la página no se verá correctamente.

En este tutorial vamos a intentar explicar cómo forzar al navegador a que se vuelva a descargar un recurso estático de nuestra aplicación web cuando hemos cambiado su contenido.

2. Entorno.

El tutorial está escrito usando el siguiente entorno:

- Hardware: Portátil MacBook Pro 15' (2.2 Ghz Intel Core I7, 8GB DDR3).
- Sistema Operativo: Mac OS Snow Leopard 10.6.7
- Mozilla Firefox 10.0.2.

Síg



Últi

» Al
las r
Adrr

» X
Myb
Hibe

» Pr
Lag

» Cu
prep
apai

» jii
trafc

Histi

Últi

3. El problema.

Pues un poco lo que hemos explicado en la introducción. Imaginemos que tenemos una web con una única página como la que vemos a continuación:

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1?
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4   <title>Ejemplo.</title>
5   <meta content="text/html; charset=ISO-8859-1" http-equiv="content-type" />
6   <link rel="stylesheet" type="text/css" href="http://www.misitio.com/css/estilos.css">
7 </head>
8 <body>
9   <h3 class="titulo">Esto es un título</h3>
10  <p>
11    Y esto es un párrafo
12  </p>
13 </body>
14 </html>
```

Como vemos, nuestra página hace uso de un recurso estático. Una hoja de estilos llamada estilos.css cuyo contenido es el siguiente:

```
1 body {
2     font-family: sans-serif;
3 }
4
5 h3.titulo {
6     color: red;
7 }
```

Y nuestra página se vería así:

Esto es un título

Y esto es un párrafo

Pues bien, imaginemos que tenemos nuestra página en producción durante unos meses y que, un día, decidimos cambiar el color del título. Queremos que no sea de color rojo sino azul. Para ello, lo que tendríamos que hacer es cambiar la propiedad color:red a color:blue. Luego reemplazamos la css en producción y listo.

Sin embargo, es más que probable que, un usuario que haya accedido a la página poco antes de los cambios en producción, no los vea reflejados la siguiente vez que acceda a la página debido a que su navegador no se ha descargado estilos.css con los nuevos cambios puesto que ya la tiene cacheada. Deberá esperar a sucesivos accesos a dicha página, dependiendo de la política de reemplazo de la caché de su navegador, para que pueda ver el título en color azul.

4. La solución.

Pues la solución es MUY MUY sencilla, consiste en "engañar" al navegador haciéndole creer que la css es otra, o lo que es lo mismo, cambiando la URL de la que se debe descargar la hoja de estilos en el .html que hace uso de la misma. No se trata de cambiar el recurso de ubicación o de cambiarlo de nombre, simplemente añadimos un parámetro en la URL y listo.

```
1 Ejemplo 1:
2 <link rel="stylesheet" type="text/css" href="http://www.misitio.com/css/estilos.css?versi
3
4 Ejemplo 2:
5 <link rel="stylesheet" type="text/css" href="http://www.misitio.com/css/estilos.css?20120
```

Como se puede apreciar en los ejemplos anteriores hemos añadido un parámetro a la URL donde está el recurso, pero el recurso sigue siendo el mismo, estilos.css. De esta forma, el navegador no tendrá constancia de esta nueva URL por lo que descargará el recurso.

Los parámetros de los ejemplos anteriores son pura convención. El primero es version=1.1 lo que nos ayudará a mantener un control sobre la versión de la css que estamos exponiendo. El segundo es 201202241835 que no es más que la fecha en formato yyyyMMddHHmm en la que se hicieron los últimos cambios en el recurso. Por supuesto vale cualquier otro parámetro aunque es recomendable seguir siempre la misma convención.

En el momento en que volvamos a hacer cambios en el recurso bastará con cambiar este parámetro al importar el recurso desde la página (ej: version=1.2 ó 201205160937) y nos aseguraremos al 100% que el navegador del cliente descargará el recurso la primera vez que acceda a la página una vez subidos los cambios a producción.

5. Conclusiones.

En este tutorial (mejor dicho mini-tutorial) hemos visto cómo evitar que el navegador cachee los recursos estáticos de nuestras webs evitando que se vean los cambios futuros en dichos recursos.

En el ejemplo hemos visto cómo hacerlo con una css pero esto es válido para cualquier otro tipo de recurso como una librería javascript (.js) o una imagen. Especialmente delicado es el tema de las .js ya que es posible que añadamos funcionalidad nueva y, si el cliente no tiene la última versión de este recurso es probable que la página le deje de funcionar como se espera (hasta que expire la caché, lógicamente...).

Espero que este tutorial os haya sido de ayuda. Un saludo.

Miguel Arlandy

marlandy@autentia.com

Twitter: [@m_arlandy](#)

A continuación puedes evaluarlo:

[Regístrate para evaluarlo](#)

Por favor, vota +1 o compártelo si te pareció interesante

[Share](#) |

0

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.

Enviar comentario

(Sólo para usuarios registrados)

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



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

IMPULSA

Impulsores

Comunidad

[¿Ayuda?](#)

0 personas han traído clicks a esta página

sin clicks

+ + + + + + + +

powered by [karmacracy](#)