

¿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 en Adictos a través de  

E-mail

Contraseña

Entrar [Deseo registrarme](#)
[Olvidé mi contraseña](#)

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

» Estás en: [Inicio](#) [Tutoriales](#) [Migrar repositorios de Mercurial a Git](#)



[Daniel Hernandez del Peso](#)

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

Ingeniero en Informática

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

Somos expertos en Java/J2EE



[Ver todos los tutoriales del autor](#)



Fecha de publicación del tutorial: 2012-03-07

Tutorial visitado 28 veces [Descargar en PDF](#)

Migrar repositorios de Mercurial a Git

Entorno del tutorial

El siguiente tutorial está desarrollado con el siguiente entorno:

- Portatil MacBook Pro
- Procesador Intel i7 2,8 GHz
- 8 GB memoria RAM
- Disco duro 256 GB SSD

Introducción

A lo largo de la vida de una organización, es normal que las herramientas que un día son las óptimas para nosotros se vean sustituidas por otras más modernas, eficientes o simplemente, que se adecúen mejor a nuestras nuevas necesidades

Una de estas herramientas puede ser el sistema de control de versiones... Es posible que en los inicios se usara CVS y luego se pasara a Subversion o, como se verá en este tutorial, se empleara Mercurial en un comienzo y se haya acabado usando Git

Cuando se realiza la migración de los datos de un repositorio de código a otro, tenemos que tener en cuenta dos cosas importantes: Por un lado, el número de recursos que necesitamos mover puede ser bastante grande; por otro, es más que interesante mantener los datos históricos tras la migración (saber por qué se ha hecho cada cambio en cada fichero de código, quién lo hizo, etc.).

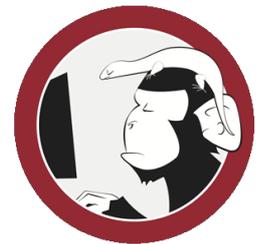
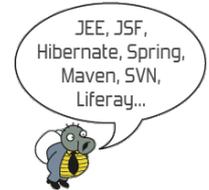
Por tanto, lo mejor es buscar herramientas que nos ayuden a resolver estos dos problemas. En este caso, vamos a ver como usar la herramienta "fast-export" para migrar los contenidos de un repositorio de Mercurial a un repositorio Git

Primer paso: obtener fast-export

Como puedes ver, estoy considerando que el primer paso va a ser directamente la descarga de "fast-export". Doy por sentado que, a estas alturas, dispones del cliente de Mercurial y del cliente de Git correctamente instalados y funcionando...

Hecha la aclaración, para obtener "fast-export", vamos a recogerlo de su propio repositorio de Git:
<http://repo.or.cz/w/fast-export.git>

Catálogo de servicios Autentia



Síguenos a través de:



Últimas Noticias

- » ¡Estrenamos Terrakas 1x03: Todo por la pasta!
 - » Material de la charla XVIII Charla Autentia - Mybatis: alternativa a Hibernate
 - » Autentia - Caso de éxito de las metodologías ágiles en la Administración Pública.
 - » XVIII Charla Autentia - Mybatis: alternativa a Hibernate
 - » Prohibidos Monos & Lagartos
- [Histórico de noticias](#)

Últimos Tutoriales

- » TDD con PHP gracias a PHPUnit
- » JBossESB y JBDS
- » Cuando llega el día de informar a la dirección de proyectos sobre el estado de un proyecto Ágil (Scrum).
- » Cómo forzar al navegador a que descargue recursos estáticos cuando los tiene cacheados.
- » Spring Integration: Ejemplo completo.

Public Git Hosting - fast-export.git/summary

fast-ex... Mercuri... Reposit... Convert... Publi... x > +

repo.or.cz/w/fast-export.g

Push URL: ssh://repo.or.cz/srv/git/fast-export.git

Content tags: fast-export git hg mercurial perforce python

Add:

shortlog

2012-01-27	Frej Drejhammar	Update documentation: The tool is no longer called... [master]	commit commitdiff tree snapshot (tar.gz zip)
2012-01-27	Frej Drejhammar	Mention the author map file in the readme	commit commitdiff tree snapshot (tar.gz zip)
2012-01-23	David Wragg	Add usage info for the --force option	commit commitdiff tree snapshot (tar.gz zip)
2011-12-09	Barry Wardell	Commits with no parent need a reset command.	commit commitdiff tree snapshot (tar.gz zip)
2011-11-26	Barry Wardell	Support the case where the author field has an empty...	commit commitdiff tree snapshot (tar.gz zip)
2011-10-18	Paul O'Shannessy	Use hg methods to extract name and email when doing...	commit commitdiff tree snapshot (tar.gz zip)

De este repositorio, descargamos una "snapshot", en nuestro caso la versión en "tar.gz", y la descomprimos donde más nos guste...

```
hg-fast-export.py
hg-fast-export.sh
hg-fast-export.txt
hg-reset.py
hg-reset.sh
hg2git.py
hg2git.pyc
Makefile
svn-archive.c
svn-fast-export.c
svn-fast-export.py
```

Como podéis ver, se trata de un script SH y poco más, no tiene componentes visuales...

Segundo paso: Configurar los repositorios de origen y destino

Una vez que tenemos fast-export listo para usar, descargaremos en nuestro equipo el repositorio de Mercurial que queremos migrar. Para ello, iremos a la carpeta en que deseamos que se descargue y ejecutamos el comando

```
hg clone https://mihost/mirepositorio/proyectoTutorial
```

y con eso tenemos en local nuestro proyecto

A continuación, tenemos que acceder al repositorio Git. Yo ya tengo un repositorio vacío creado y voy a descargarlo para hacer la importación sobre él:

```
git clone https://host.de.mi.repo/tutorialEjemplo
```

Tercer paso: empieza la "magia"

Llegamos al momento crucial... Una vez que tenemos todo lo necesario (los dos repositorios y el script) sólo nos queda juntarlo todo

Vamos al directorio donde se ha creado el directorio de trabajo de Git y ejecutamos el comando:

```
/ruta/a/fast-export/hg-fast-export.sh -r /ruta/a/MercurialRepo/repoTutorial/
```

lo que nos genera la siguiente salida, signo de que todo ha ido bien

```
master: Exporting full revision 1/3 with 83/0/0 added/changed/removed files
master: Exporting simple delta revision 2/3 with 15/10/0 added/changed/removed files
master: Exporting simple delta revision 3/3 with 0/0/3 added/changed/removed files
Issued 3 commands
git-fast-import statistics:
-----
Alloc'd objects:      5000
Total objects:       273 (      43 duplicates      )
blobs :              83 (      25 duplicates      39 deltas of      82 attempts)
trees :             187 (      18 duplicates      25 deltas of     187 attempts)
commits:              3 (         0 duplicates      0 deltas of      0 attempts)
tags :                0 (         0 duplicates      0 deltas of      0 attempts)
Total branches:      1 (         1 loads      )
marks:             1024 (         3 unique      )
atoms:                66
Memory total:        2344 KiB
pools:               2110 KiB
objects:             234 KiB
-----
pack_report: getpagesize() = 4096
pack_report: core.packedGitWindowSize = 1073741824
pack_report: core.packedGitLimit = 8589934592
pack_report: pack_used_ctr = 1
pack_report: pack_mmap_calls = 1
pack_report: pack_open_windows = 1 / 1
pack_report: pack_mapped = 61978 / 61978
-----
```

A continuación, hacemos un "git checkout" para traer al directorio de trabajo los cambios del repositorio local

Últimos Tutoriales del Autor

- » Creación de cursos con Moodle
- » Administración de sitios Moodle
- » Primeros pasos con Moodle
- » Creación de contenidos SCORM: eXe
- » Ficheros de mapeo de Hibernate desde las clases

Categorías del Tutorial

- Gestión Contenidos
- Patrones de Integración
- Herramientas
- www.adictosaltrabajo.com

Últimas ofertas de empleo

- 2011-09-08 Comercial - Ventas - MADRID.
- 2011-09-03 Comercial - Ventas - VALENCIA.
- 2011-08-19 Comercial - Compras - ALICANTE.
- 2011-07-12 Otras Sin catalogar - MADRID.
- 2011-07-06 Otras Sin catalogar - LUGO.

```

340 Mar 6 08:00 .
272 Mar 6 08:00 ..
544 Mar 6 08:00 .git
180 Mar 6 08:00 .hgignore
136 Mar 6 08:00 icefaces1-showcase
136 Mar 6 08:00 icefaces2-showcase
136 Mar 6 08:00 jsf2-showcase
136 Mar 6 08:00 primefaces2-showcase
136 Mar 6 08:00 richfaces3-showcase
136 Mar 6 08:00 showcase-core

```

Ahora ya tenemos en nuestro directorio local y en la copia local del repositorio Git remoto los commits. Para subirlo al repositorio remoto, ejecutamos el comando

```
/ruta/a/git/repo/git push origin master
```

```

Counting objects: 273, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (98/98), done.
Writing objects: 100% (273/273), 60.49 KiB, done.
Total 273 (delta 64), reused 273 (delta 64)
To https://... proyectoTutorial
* [new branch]      master -> master

```

Y ya lo tenemos :)

name	age	message	history
 icefaces1-showcase	a year ago	He añadido una página en el show case de jsf2 para la realización ...	
 icefaces2-showcase	a year ago	He añadido una página en el show case de jsf2 para la realización ...	
 jsf2-showcase	a year ago	He añadido una página en el show case de jsf2 para la realización ...	
 primefaces2-showcase	a year ago	He añadido una página en el show case de jsf2 para la realización ...	
 richfaces3-showcase	a year ago	He añadido una página en el show case de jsf2 para la realización ...	
 showcase-core	a year ago	Gestión del evento de cambio de valor en JSF2: valueChangeListener.	
 .hgignore	a year ago	Primera subida del jsf-showcase	

Conclusiones

Ya sabemos todos que la tecnología avanza cada vez más rápido, y las soluciones de hoy serán distintas a las que podamos aplicar mañana. Muchas veces nos quedamos anclados en tecnologías antiguas por miedo a que el cambio se complique...

Sin embargo, ya hemos visto que casi siempre se pueden encontrar herramientas para que estos cambios sean "triviales". No podemos olvidar que, gracias a internet, vivimos en un mundo global, y que es raro que mis problemas o dudas no los haya tenido alguien antes. Por tanto, buceando un poco vais a poder encontrar soluciones a muchas de las situaciones con las que os vais a encontrar

Así que ya sabéis, perded el miedo y si no lo conseguís, podéis contratar a [Autentia](#) para que os ayude

A continuación puedes evaluarlo:

[Regístrate para evaluarlo](#)

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

Share |

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?](#)

sin clicks

0 personas han traído clicks a esta página

+ + + + + + + +

powered by [karmacracv](#)

Copyright 2003-2012 © All Rights Reserved | [Texto legal y condiciones de uso](#) | [Banners](#) | [Powered by Autentia](#) | [Contacto](#)

