

# ¿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

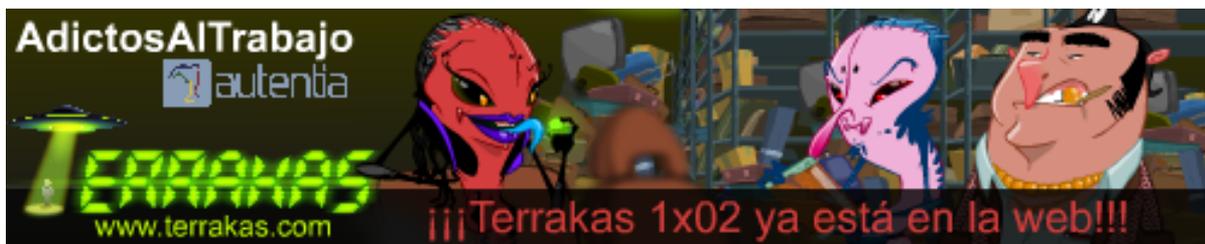
Tareas programadas (Quartz)  
 Gestor documental (Alfresco)  
 Inversión de control (Spring)

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)



[Inicio](#) [Quiénes somos](#) [Tutoriales](#) [Formación](#) [Comparador de salarios](#) [Nuest](#)

» Estás en: [Inicio](#) [Tutoriales](#) [Trabajando con múltiples cores con Apache Solr.](#)



[Miguel Arlandy Rodríguez](#)

Consultor tecnológico de desarrollo de proyectos informáticos.

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

Somos expertos en Java/JEE

[Ver todos los](#)

También puedes seguirme



**Fecha de publicación del tutorial: 2012-02-06**

Tutorial visitado

## Trabajando con múltiples cores con Apache Solr.

### 0. Índice de contenidos.

- [1. Introducción.](#)
- [2. Entorno.](#)
- [3. Instalando Solr.](#)
- [4. Configuración general.](#)
- [5. Configurando el core.](#)
- [6. Actualizando los índices.](#)
- [7. Consultando los datos.](#)

- 8. Referencias.
- 9. Conclusiones.

## 1. Introducción

Como ya nos explicó [Jose Manuel](#), Solr es un motor de búsqueda de documentos de código abierto que permite diferentes comportamientos en la búsqueda e indexación de dichos documentos.

Pueden existir situaciones en las que queramos definir diferentes índices y esquemas de datos en Solr. Definir diferentes cores.

En este tutorial vamos a ver cómo configurar Apache Solr para que, con una única instancia, podamos trabajar con múltiples cores con sus respectivos índices y esquemas de datos.

## 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
- Apache Solr 3.5.
- Mozilla Firefox 9.

## 3. Instalando Solr.

Lo primero que haremos será [descargarnos la última versión de Solr](#). En nuestro caso será la 3.5.0.

A continuación, descomprimos su contenido y observamos la estructura de directorios.

| Nombre              | Fecha            | Tamaño | Tipo            |
|---------------------|------------------|--------|-----------------|
| ▼ apache-solr-3.5.0 | 03/02/2012 12:48 | --     | Carpeta         |
| .DS_Store           | Hoy, 13:08       | 8 KB   | Documento       |
| CHANGES.txt         | 22/11/2011 14:03 | 160 KB | Texto normal    |
| ▶ client            | 03/02/2012 13:31 | --     | Carpeta         |
| ▶ contrib           | 22/11/2011 14:54 | --     | Carpeta         |
| ▶ dist              | 03/02/2012 12:48 | --     | Carpeta         |
| ▶ docs              | 03/02/2012 12:48 | --     | Carpeta         |
| ▼ example           | 03/02/2012 12:55 | --     | Carpeta         |
| .DS_Store           | Hoy, 13:08       | 8 KB   | Documento       |
| ▶ etc               | 22/11/2011 14:02 | --     | Carpeta         |
| ▶ example-DIH       | 22/11/2011 14:02 | --     | Carpeta         |
| ▶ exampledocs       | Hoy, 11:30       | --     | Carpeta         |
| ▶ lib               | 22/11/2011 14:02 | --     | Carpeta         |
| ▶ logs              | 22/11/2011 14:02 | --     | Carpeta         |
| ▼ multicore         | Hoy, 10:50       | --     | Carpeta         |
| .DS_Store           | Hoy, 13:09       | 8 KB   | Documento       |
| ▶ core0             | Hoy, 10:51       | --     | Carpeta         |
| ▶ core1             | Hoy, 10:51       | --     | Carpeta         |
| ▶ exampledocs       | 22/11/2011 14:02 | --     | Carpeta         |
| README.txt          | 22/11/2011 14:02 | 4 KB   | Texto normal    |
| solr.xml            | 22/11/2011 14:02 | 4 KB   | Docu...e texto  |
| README.txt          | 22/11/2011 14:02 | 4 KB   | Texto normal    |
| ▶ solr              | Hoy, 09:51       | --     | Carpeta         |
| start.jar           | 22/11/2011 14:02 | 20 KB  | Archiv...e Java |
| ▶ webapps           | 22/11/2011 14:54 | --     | Carpeta         |
| ▶ work              | 03/02/2012 12:50 | --     | Carpeta         |
| LICENSE.txt         | 22/11/2011 14:03 | 82 KB  | Texto normal    |
| NOTICE.txt          | 22/11/2011 14:03 | 20 KB  | Texto normal    |
| README.txt          | 22/11/2011 14:03 | 8 KB   | Texto normal    |

Como podemos observar existe un directorio multicore/ dentro del directorio example que viene (\$SOLR\_HOME/example/multicore/). Pues bien, vamos a destripar un poco este ejemplo para

## 4. Configuración general.

Abrimos el fichero \$SOLR\_HOME/example/multicore/solr.xml y vemos lo siguiente:

```

1 | <?xml version="1.0" encoding="UTF-8" ?>
2 |
3 | <solr persistent="false">
4 |   <cores adminPath="/admin/cores">
5 |     <core name="core0" instanceDir="core0" />
6 |     <core name="core1" instanceDir="core1" />
7 |   </cores>
8 | </solr>

```

Como podemos observar, se definen dos cores distintos: core0 y core1, que serán administrados /admin/cores relativo al despliegue de Solr. A cada uno de los cores le damos un nombre distinto donde se alojará la configuración propia de cada uno y el índice que maneje. Además, Solr permite propiedades a modo de nombre valor:

- dataDir: directorio sobre el que se almacenarán los datos del índice del core. Por defecto es /admin/cores/core0/data
- configName: nombre del fichero de configuración del core. Por defecto es solrconfig.xml
- schemaName: nombre del fichero donde se define el esquema a utilizar. Por defecto es schema.xml

## 5. Configurando el core.

Vamos ahora con la configuración propia de cada core. Por ejemplo la del core0. Si vamos al directorio `${SOLR_HOME}/example/multicore/core0/` vemos que tenemos dos directorios: conf y data.

|                |                  |       |                    |
|----------------|------------------|-------|--------------------|
| ▼ core0        | Hoy, 10:51       | --    | Carpeta            |
| .DS_Store      | Hoy, 15:55       | 8 KB  | Documento          |
| ▼ conf         | Hoy, 15:54       | --    | Carpeta            |
| .DS_Store      | Hoy, 15:54       | 8 KB  | Documento          |
| schema.xml     | Hoy, 11:36       | 37 KB | Documento de texto |
| solrconfig.xml | 22/11/2011 14:02 | 4 KB  | Documento de texto |
| ▼ data         | Hoy, 10:51       | --    | Carpeta            |
| .DS_Store      | Hoy, 13:27       | 8 KB  | Documento          |
| ▶ index        | Hoy, 11:40       | --    | Carpeta            |

En el directorio conf estarán los ficheros de configuración propios del core (del core0). Vamos a configurar el core0 para que se comporte de manera distinta a la que viene en la distribución. En el directorio `${SOLR_HOME}/example/multicore/core0/conf/` tenemos otro ejemplo de funcionamiento de Solr (este con un único core), así que vamos a hacer que el core0 se comporte como el de este ejemplo. Para ello nos vamos al directorio `${SOLR_HOME}/example/multicore/core0/conf/` y copiamos el fichero `schema.xml` y los `.txt` en el directorio de configuración del core0. Ahora el core0 indexará los documentos de la misma forma que en el ejemplo que tiene un único core.

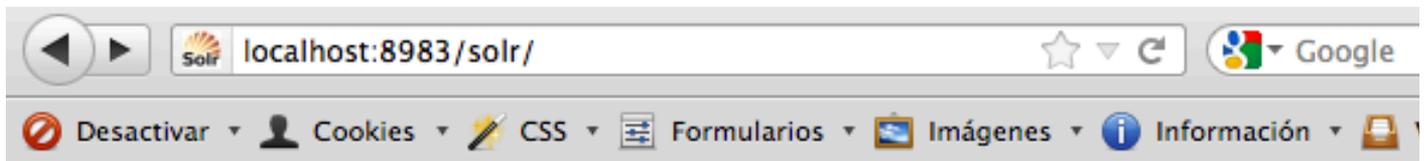
El core1 vamos a dejarlo tal y como está. Si abrimos los dos ficheros `schema.xml` (`${SOLR_HOME}/example/multicore/core0/conf/schema.xml` y `${SOLR_HOME}/example/multicore/core1/conf/schema.xml`) veremos que son distintos, por lo que la información con la que nutriremos a los índices será diferente.

## 6. Actualizando los índices.

Vale, pues lo siguiente que debemos hacer es actualizar los índices de nuestros dos cores. Para ello vamos a configurar Solr para que funcione acorde con el ejemplo de multicores. Nos vamos al directorio `${SOLR_HOME}/bin/` y ejecutamos lo siguiente:

```
1 | java -Dsolr.solr.home=multicore -jar start.jar
```

Si estamos en local y nos vamos a <http://localhost:8983/solr> veremos la consola de administración de los módulos.



# Welcome to Solr!

[Admin core0](#)

[Admin core1](#)

Y pulsando sobre cualquiera de los dos enlaces accederemos a la administración del core correspondiente.

Bien, lo siguiente que haremos será actualizar los datos de los índices. Para ello, en la propia consola de administración de Solr, vamos a introducir los datos de prueba. Recordemos que nuestros cores tienen esquemas de datos distintos, por lo que...

suministrar los datos correspondientes con cada core.

En el directorio `${SOLR_HOME}/example/exampledocs/` tenemos varios documentos de prueba de esquema de datos del core0. En concreto los que acaban en `.xml`.

La URL a la que enviaremos las peticiones con los datos del índice del core0 será (si estamos en `http://localhost:8983/solr/core0/update`). Para enviar los datos, la distribución de Solr nos trae un script `post.jar` con el propósito. Pues bien, en el directorio `exampledocs/` tecleamos lo siguiente:

```
1 | java -jar -Durl=http://localhost:8983/solr/core0/update post.jar
```

Lo mismo hacemos con el core1. Los datos que se corresponden con su esquema están en `${SOLR_HOME}/example/multicore/exampledocs/`. Por lo que tecleamos desde el directorio `${SOLR_HOME}/example/exampledocs/` lo siguiente:

```
1 | java -jar -Durl=http://localhost:8983/solr/core1/update post.jar
```

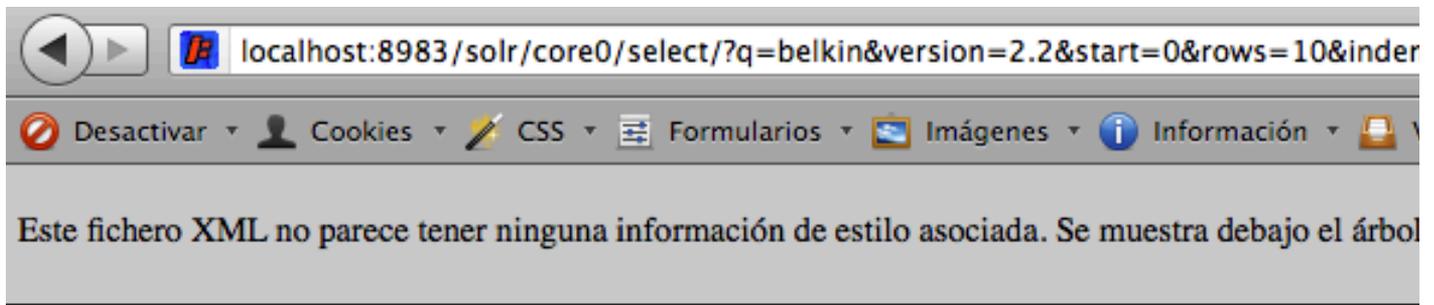
Y listo, ya tendríamos nuestros dos índices con unos datos de prueba.

## 7. Consultando los datos.

Bueno, pues vamos a ver si ahora somos capaces de consultar los datos que tenemos almacenados en los índices de nuestros cores.

La url donde irán las consultas al core0 será `http://localhost:8983/solr/core0/select` mientras que al core1 será `http://localhost:8983/solr/core1/select`

Consultando en el índice del core0 todos los equipos electrónicos que contengan la palabra "beats" lo siguiente:

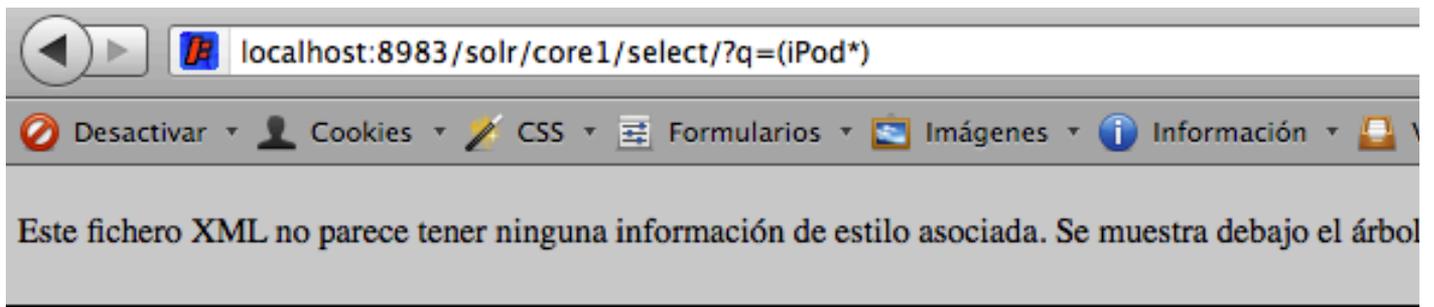


```

- <response>
  - <lst name="responseHeader">
    <int name="status">0</int>
    <int name="QTime">1</int>
  </lst>
  - <result name="response" numFound="2" start="0">
    - <doc>
      - <arr name="cat">
        <str>electronics</str>
        <str>connector</str>
      </arr>
      - <arr name="features">
        <str>car power adapter, white</str>
      </arr>
      <str name="id">F8V7067-APL-KIT</str>
      <bool name="inStock">>false</bool>
      <str name="manu">Belkin</str>
      <date name="manufacturedate_dt">2005-08-01T16:30:25Z</date>
      <str name="name">Belkin Mobile Power Cord for iPod w/ Dock</str>
      <int name="popularity">1</int>
      <float name="price">19.95</float>
      <str name="store">45.17614,-93.87341</str>
      <float name="weight">4.0</float>
    </doc>
    - <doc>
      - <arr name="cat">
        <str>electronics</str>
        <str>connector</str>
      </arr>
      - <arr name="features">
        <str>car power adapter for iPod, white</str>
      </arr>
      <str name="id">IW-02</str>
      <bool name="inStock">>false</bool>
      <str name="manu">Belkin</str>
      <date name="manufacturedate_dt">2006-02-14T23:55:59Z</date>
      <str name="name">iPod & iPod Mini USB 2.0 Cable</str>
      <int name="popularity">1</int>
      <float name="price">11.5</float>
      <str name="store">37.7752,-122.4232</str>
      <float name="weight">2.0</float>
    </doc>
  </result>
</response>

```

Consultando en el índice del core1 todos los equipos cuyo nombre comience por "iPod" obtend



```

- <response>
  - <lst name="responseHeader">
    <int name="status">0</int>
    <int name="QTime">0</int>
  </lst>
  - <result name="response" numFound="1" start="0">
    - <doc>
      <str name="id">IW-02</str>
      <str name="name">iPod & iPod Mini USB 2.0 Cable</str>
    </doc>
  </result>
</response>

```

Observemos que los tipos de datos devueltos son muy diferentes, como diferentes son los esquemas de los cores que tenemos configurados.

## 8. Referencias.

- [Solr: Core Admin.](#)

## 9. Conclusiones.

En este tutorial hemos visto cómo configurar varios cores en una misma instancia de Solr. Esto es interesante cuando tenemos varios esquemas para indexar diferentes tipos de documentos, pero centralizado en un único sitio.

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

**¿Te gusta adictosaltrabajo.com? Síguenos a través de:**



**Anímate y coméntanos lo que pienses sobre este TUTORIAL:**

Puedes opinar o comentar cualquier sugerencia que quieras comunicarnos sobre este tutorial 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](#)

IMPULSA

Impulsores

Comunidad

[¿Ayuda?](#)

0 personas han traído clicks a esta página

sin clicks



powered by [karma](#)

---

Copyright 2003-2012 © All Rights Reserved | [Texto legal y condiciones c](#)

