

Extendiendo la Funcionalidad del Servidor Funambol DS Server de Código Abierto para la Sincronización de Datos con Dispositivos Móviles.

Joel Herrera Cabral

Universidad Tecnológica del Suroeste de Guanajuato, Valle de Santiago, Guanajuato, México,
mus2502@hotmail.com

José Luis Rico Moreno

Universidad Tecnológica del Suroeste de Guanajuato, Valle de Santiago, Guanajuato, México,
luis2173@yahoo.com

José Antonio Gordillo Sosa

Universidad Tecnológica del Suroeste de Guanajuato, Valle de Santiago, Guanajuato, México,
josant69@yahoo.com

ABSTRACT.

El cómputo móvil y la conectividad inalámbrica utilizando dispositivos móviles como teléfonos celulares y smartphones se han convertido en una herramienta tecnológica de gran importancia para las instituciones y las empresas. En la actualidad servicios como el acceso a datos personales, envío y recepción de correo electrónico, libretas de contactos, respaldo de información y consulta remota a bases de datos corporativas, están disponibles para el consumo de las masas.

Funambol DS Server es un servidor para la administración de dispositivos móviles que nos permite la sincronización de datos con una amplia gama de fabricantes y dispositivos, es de código abierto y libre de costo, su arquitectura nos permite consumir servicios que ya vienen incluidos en él, así como ampliar su funcionalidad incrustándole módulos desarrollados por terceros con algún propósito específico, también incluye algunos conectores para sincronizar datos para Card, iCalendar, FileSystem, Pop3 y conectores DB.

Este trabajo aborda la incrustación del módulo Funambol-utsoe-6.5.2.s4j para la sincronización de datos. En el presente artículo se describe la incrustación del módulo, su estructura de directorios fuente, archivos, sus clases, archivos de configuración, y scripts de inicialización SQL, así como el panel de configuración del módulo.

Palabras Clave: Móviles, dispositivos, sincronización, celulares, servidor.

ABSTRACT.

The mobile computing and the wireless connectivity using mobile devices like cellular telephones and smartphones they have turned into a technological tool of great importance for the institutions and the organizations. Nowadays services like the access to the personal information, sending and reception of email, addressbooks of contacts, information support and remote consulting to corporate databases are available for the consumption of the masses.

Funambol DS Server is a Server for the administration of mobile devices that allow us the synchronization of information with wide range of manufacturers and devices, is an open source one and without cost, its architecture allows us to consume services that are already included on it, as well as extend its functionality attaching it developed modules by third parties with some specific proposals, it also includes some connectors to synchronize data for Card, iCalendar, FileSystem, Pop3 and connectors DB.

This work approaches the attaching of the module Funambol-utsoe-6.5.2.s4j for the data synchronization. The present article describes the attaching of the module, its structure of source directories, files, classes, configuration files, and SQL initialization scripts, as well as the module's configuration panel.

Keywords: Mobiles, devices, synchronization, cellular, server.

1. INTRODUCCIÓN.

Realizar la instalación del servidor Funambol DS de dispositivos móviles de código abierto es relativamente sencillo, lo importante viene después al adquirir la experiencia necesaria par el desarrollo de módulos con propósitos específicos y poderlos incrustar en el servidor de una forma fácil de mantener.

Al principio la idea era configurar solamente el servidor Funambol DS Server para sincronizarlo con Outlook. Pero posteriormente el trabajo derivó en comenzar a experimentar con el servidor e instalarle el módulo Funambol-utsoe-6.5.2 para la sincronización de datos.

PANORAMA DE DESARROLLO DEL MÓDULO.

Para realizar lo anterior es necesario contar con:

- Instalación de Funambol DS Server y ejecutándose.
- Java 2 SDK.
- Instalación de Jakarta Ant.
- El sistema operativo puede ser Windows o Linux.

Una vez hecho esto, se comienza el desarrollo del módulo siguiendo los siguientes pasos:

- a. Crear lo siguiente, en secuencia:
 - Estructura del directorio fuente del módulo.
 - Módulo Utsoe Tipo SyncSource.
 - Panel de configuración de Utsoe SyncSource.
 - Scripts SQL para registrar el módulo.
 - Fichero del archivo del módulo.
- b. Instalar el módulo.
- c. Crear una instancia utsoe del tipo SyncSource.
- d. Probar el módulo con un cliente SyncML.

2. DESARROLLO.

El desarrollo del módulo requiere conocimiento de Java, Ant y SQL.

Los archivos usados para desarrollar el módulo están disponibles para descarga en el siguiente sitio:

http://forge.objectweb.org/project/download.php?group_id=96&file_id=5868

También es importante, revisar a detalle el tutorial de desarrollo de módulos Funambol que sirve de guía para la creación de conectores. Es importante aclarar que el módulo no es lo mismo que un conector. La diferencia es que un módulo es un componente adicional al servidor Funambol DS y puede contener uno o más conectores.

2.1 LA ESTRUCTURA DE DIRECTORIOS FUENTE DEL MÓDULO.

Es necesaria una estructura de directorios fuente, de configuración y el script sql para el módulo. Esta estructura es creada al descomprimir el archivo descargado del link anterior. La descompresión del archivo se realiza en algún directorio de nuestra elección. La estructura queda de la siguiente forma:

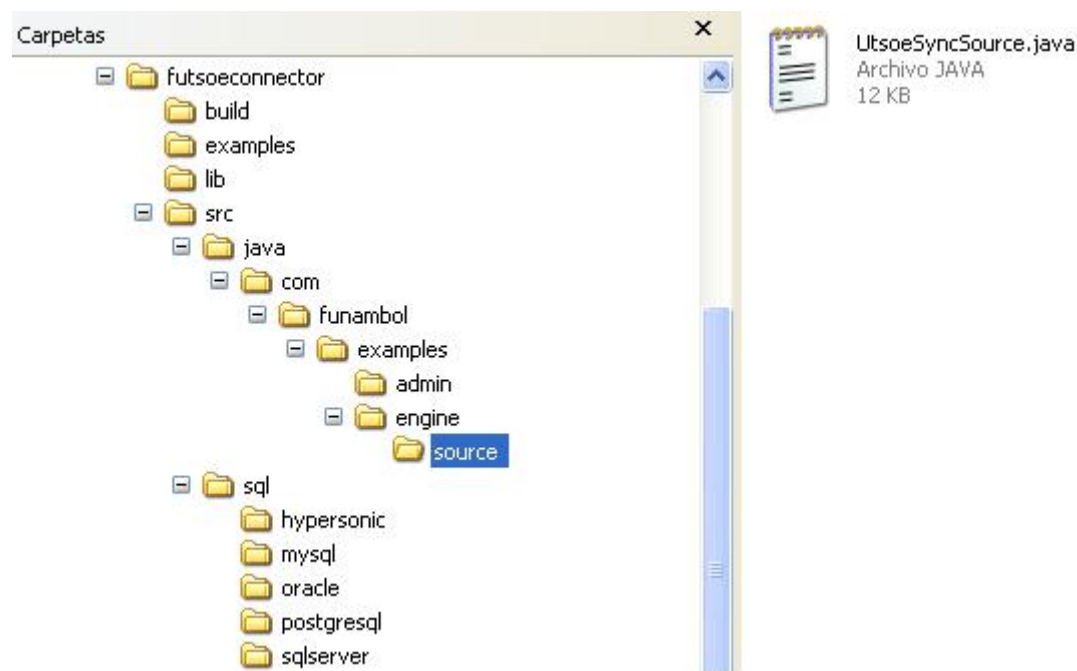


Figura 1: Estructura de directorios fuente del módulo

Como puede observarse de la estructura de directorios, el módulo está compuesto de las siguientes partes:

- Los archivos fuente Java del módulo.
- Los scripts SQL utilizados para instalar y desinstalar el módulo y el conector en el servidor Funambol DS.
- Un script ant para la construcción del módulo.
- Librerías necesarias para el módulo.

2.2 EL TIPO UTSOESYNCSOURCE.

Existen dos partes fundamentales al momento de desarrollar un conector. La clase de sincronización SyncSource que nos permite el acceso a los datos de forma sincronizada. Y un panel de configuración creado especialmente para el conector, que será desplegado en el panel de administración del servidor Funambol DS para configurar la sincronización con el nuevo conector.

El código fuente de la clase UtsoeSyncSource se encuentra en el directorio \futoconnector\src\java\com\funambol\examples\engine\source de la estructura de directorios creada al descomprimir el archivo.

2.3 REVISANDO EL CÓDIGO DE LA CLASE UTSOESYNCSOURCE.

La estructura de la clase UtsoeSyncSource hereda de la clase AbstractSyncSource, e implementa las interfaces SyncSource, Serializable, y LazyInitBean. Su constructor inicializa algunos elementos nota que se almacenan en las variables de instancia newItem, deletedItems y updatedItems.

Y son regresados mediante los métodos get[All/Updated/New/Deleted]Items(). El método crea los elementos recibiendo como parámetros el identificador del elemento (la clave del elemento), el contenido y su estado.

La estructura de la clase (métodos) refleja a la interfaz SyncSource. En adición, extiende a AbstractSyncSource ya que hereda métodos comunes.

El constructor crea algunos elementos nota que son almacenados en las variables de instancia newItem, deletedItems y updatedItems. Ellos son retornados cuando son solicitados por get[All/Updated/New/Deleted]Items().

Los elementos son creados en createItem(): dando el identificador del elemento (la clave del elemento), el contenido y el estado, se instancia un nuevo SyncItemImpl (una implementación simple de la interfaz SyncItem) y establece BINARY_PROPERTY a el contenido de la nota.

2.4 IMPLEMENTACIÓN DEL PANEL DE CONFIGURACIÓN.

Para crear un panel de configuración del conector e incrustarlo en la Herramienta de Administración del servidor Funambol DS, es necesario extender la clase Source ManagementPanel haciendo uso del mecanismo de herencia de la programación orientada a objetos. SourceManagementPanel es una clase JPanel, su método init() crea los controles necesarios que sea necesario desplegar en el Panel y observarlos en la Herramienta de Administración del servidor Funambol DS. Esos controles pueden servir para capturar información como:

- Nombre de la fuente.
- Tipos de datos soportados.
- Versiones de los tipos de datos soportados.
- source URI.

se puede colocar un JButton en el panel para la sincronización con el dispositivo móvil y almacenar los valores de configuración del conector en el servidor. El código para realizar esto es el siguiente:

```
confirmButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent event) {
        try {
            validateValues();
            getValues();
            if (getState() == STATE_INSERT) {
                UtsoeSyncSourceConfigPanel.this.actionPerformed(
                    new ActionEvent(UtsoeSyncSourceConfigPanel.this,
                        ACTION_EVENT_INSERT,
                        event.getActionCommand()));
            } else {
                UtsoeSyncSourceConfigPanel.this.actionPerformed(
                    new ActionEvent(UtsoeSyncSourceConfigPanel.this,
                        ACTION_EVENT_UPDATE,
                        event.getActionCommand()));
            }
        } catch (Exception e) {
            notifyError(new AdminException(e.getMessage()));
        }
    }
});
```

2.5 LA API SYNC SOURCE

También es necesario implementar algunos métodos de la interface SyncSource para el conector.

Tabla 1: API SyncSource

Propósito	Métodos	Descripción
Sincronización	public void beginSync(SyncContext ctx) throws SyncSourceException;	Inicia la sincronización con la fuente de los datos
	public void endSync() throws SyncSourceException;	Finaliza la sincronización con la fuente de los datos
	public void commitSync() throws SyncSourceException;	Concreta los cambios hechos durante la sincronización.
Getters	public abstract SyncItemKey[] getAllSyncItemKeys() throws SyncSourceException;	Regresa un arreglo de todas las claves únicas de cada elemento almacenado en la fuente y es usado en el modo de sincronización.
	public abstract SyncItemKey[] getDeletedSyncItemKeys(Timestamp sinceTs, Timestamp untilTs) throws SyncSourceException;	Estos tres métodos regresan los elementos borrados, nuevos y actualizados en la fuente
	public abstract SyncItemKey[] getNewSyncItemKeys(Timestamp sinceTs, Timestamp untilTs) throws SyncSourceException;	
	public abstract SyncItemKey[] getUpdatedSyncItemKeys(Timestamp since, Timestamp until) throws SyncSourceException;	
	public abstract SyncItem getSyncItemFromId(SyncItemKey syncItemKey) throws SyncSourceException;	Este método retorna el elemento especificado por la clave.
Setters	public abstract void removeSyncItem(SyncItemKey syncItemKey, Timestamp time,	Estos métodos remueven, actualizan y agregan elementos en

	<pre>boolean softDelete) throws SyncSourceException; public abstract SyncItem updateSyncItem(SyncItem syncInstance) throws SyncSourceException; public abstract SyncItem addSyncItem(SyncItem syncInstance) throws SyncSourceException;</pre>	la fuente utilizando una clave para identificarlos.
Otros Métodos	<pre>public abstract SyncItemKey[] getSyncItemKeysFromTwin(SyncItem syncItem) throws SyncSourceException;</pre>	Este método regresa las claves de los elementos que son considerados como idénticos.
	<pre>public abstract void setOperationStatus(String operationName, int status, SyncItemKey[] keys);</pre>	Este método es utilizado para notificar el estado de una operación (“Agregar”/ “Reemplazar”/ “Borrar”).

2.6 CREANDO UN PANEL DE CONFIGURACIÓN.

Para configurar un SyncSource se utiliza la Herramienta de Administración.

USANDO LA HERRAMIENTA DE ADMINISTRACIÓN.

Iniciar la Herramienta de Administración, y ejecutar lo siguiente:

1. Iniciar el Servidor Funambol DS seleccionando **Inicio > Todos los programas > Funambol > Data Synchronization Server > Start**.
2. Iniciar la Herramienta de Administración seleccionando **Inicio > Todos los programas > Funambol > Administration Tool**. Se despliega la ventana Administration Tool.

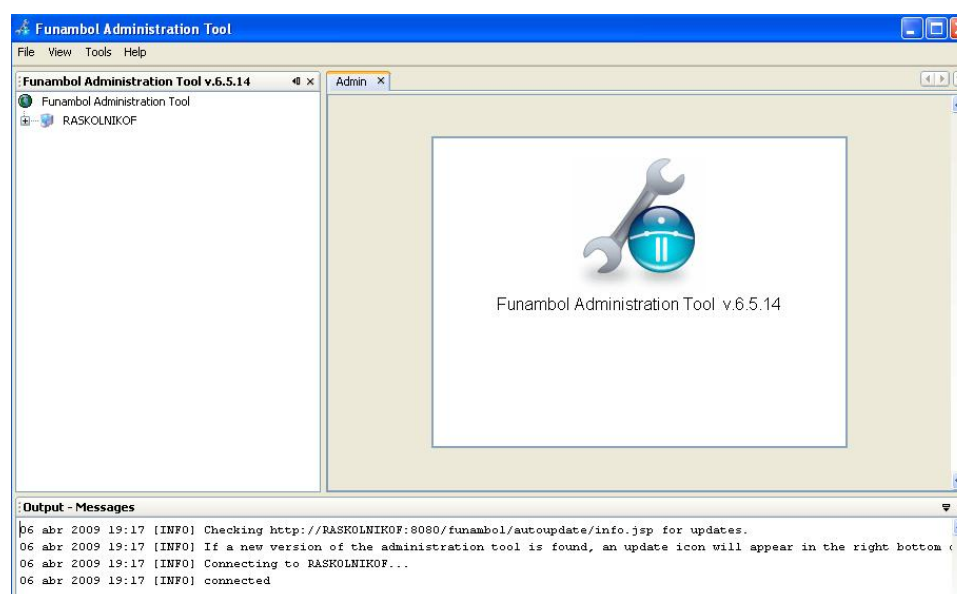


Figura 2: Herramienta de Administración

3. En la barra de menú principal seleccionar **File > Login**. Se despliega la ventana Login. Verificar que los campos se establecen como sigue, o especificar esos valores:

Hostname /IP: <localhost> (debe ser el nombre de tu máquina)
 Port: 8080
 User Name: admin.
 Password: sa.

Click **Login**: La ventana Output en el panel de abajo debe desplegar “conectado”.

4. Configurar un SyncSource, navegar en el panel izquierdo a el SyncSource deseado. Por ejemplo, para configurar un File System SyncSource, expandir el árbol **localhost** como sigue **localhost > Modules > pdi > FunambolPDICconnector**, entonces seleccionar **FileSystemSyncSource**. Se despliega la pantalla SyncSource Edit File System.

Esta pantalla se utiliza para especificar valores de configuración y agregar una instancia del tipo SyncSource. Más adelante se crea una pantalla similar a esta para el SyncSource Utsoe.

2.7 CREANDO EL PANEL DE CONFIGURACIÓN.

Para crear un panel de configuración SyncSource Utsoe, se debe crear una extensión de la clase `sync4j.syncadmin.ui.SourceManagmentPanel` a través del mecanismo de herencia y se llamará `UtsoeSyncSourceConfigPanel.java`.

El código para `UtsoeSyncSourceConfigPanel` está en el directorio `\futsoeconnector\src\java\com\funambol\examples\admin`.

2.8 CREANDO LOS SCRIPTS SQL PARA REGISTRAR EL MÓDULO.

El servidor Funambol DS es conectado a la base de datos describiendo los conectores y su información. Así que es necesario almacenar información del nuevo conector como el nombre de la clase del conector, y el nombre del panel. Los scripts se mencionan a continuación:

- `drop_schema.sql` – es utilizado para desinstalar el módulo y remover los datos.
- `create_schema.sql` – crea nuevas tablas y bases de datos.
- `init_schema.sql` – inicializa los datos para nuestro conector en la base de datos.

Para este módulo se utiliza Hypersonic. El único script requerido es `init_squema`, el cual incluye las siguientes sentencias SQL:

```
--
-- Initialization data for the Utsoe module
-- @version $Id: init_schema.sql,v 1.1 2006/01/18 16:04:56 luigiafassina Exp $
--

--
-- Module structure registration
--

delete from fnbl_sync_source_type where id='utsoe';
insert into fnbl_sync_source_type(id, description, class, admin_class)
values('utsoe','Utsoe
SyncSource','com.funambol.examples.engine.source.UtsoeSyncSource','com.funambol.exam
ples.admin.UtsoeSyncSourceConfigPanel');

delete from fnbl_module where id='utsoe';
insert into fnbl_module (id, name, description)
values('utsoe','utsoe','Utsoe');

delete from fnbl_connector where id='utsoe';
insert into fnbl_connector(id, name, description, admin_class)
values('utsoe','FunambolUtsoeConnector','Funambol Utsoe Connector','');

delete from fnbl_connector_source_type where connector='utsoe' and
sourcetype='utsoe';
insert into fnbl_connector_source_type(connector, sourcetype)
values('utsoe','utsoe');

delete from fnbl_module_connector where module='utsoe' and connector='utsoe';
insert into fnbl_module_connector(module, connector)
values('utsoe','utsoe');
```

Los comando SQL informan al Servidor Funambol DS que hay un nuevo módulo llamado **Utsoe**, el cual contiene un Conector llamado **Utsoe**, el cual contiene un tipo SyncSource llamado **Utsoe**. El tipo SyncSource es especificado por la clase `SyncSource` `com.funambol.examples.engine.source.UtsoeSyncSource` y el panel de configuración por `com.funambol.examples.admin.UtsoeSyncSourceConfigPanel`.

2.9 CREANDO EL FICHERO DE LOS ARCHIVOS DEL MÓDULO.

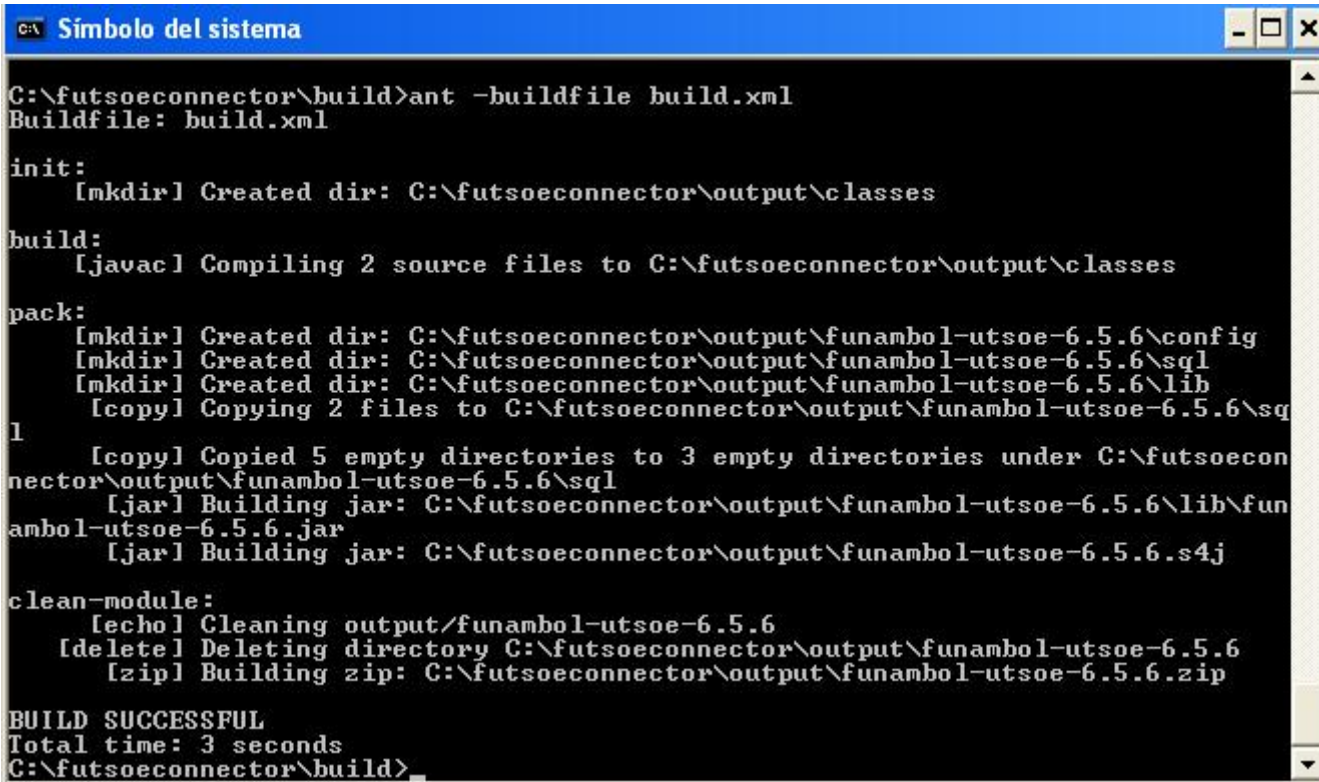
En este paso se automatiza el proceso de compilar las clases y empacarlas dentro de los archivos del módulo. La Figura 1 muestra la estructura de directorios de los archivos del módulo.

Se utiliza Jakarta Ant para construir el módulo, pero es posible usar otra o IDE para crear el módulo si se mantiene la estructura mostrada arriba. El archivo `build.xml` para generar el archivo `funambol-utsoe-6.5.2.s4j`, está disponible en el directorio `\futsoeconnector\build`.

Para ejecutar la construcción del módulo, ir al directorio `\futsoeconnector\build` y ejecutar en la línea de comandos el comando siguiente (con Jakarta Ant en la variable de entorno path):

```
C:\futsoeconnector\build>ant -buildfile build.xml
```

La salida debe parecer de forma similar a la Figura 3.



```
C:\futsoeconnector\build>ant -buildfile build.xml
Buildfile: build.xml

init:
[mkdir] Created dir: C:\futsoeconnector\output\classes

build:
[javac] Compiling 2 source files to C:\futsoeconnector\output\classes

pack:
[mkdir] Created dir: C:\futsoeconnector\output\funambol-utsoe-6.5.6\config
[mkdir] Created dir: C:\futsoeconnector\output\funambol-utsoe-6.5.6\sql
[mkdir] Created dir: C:\futsoeconnector\output\funambol-utsoe-6.5.6\lib
[copy] Copying 2 files to C:\futsoeconnector\output\funambol-utsoe-6.5.6\sql
1
[copy] Copied 5 empty directories to 3 empty directories under C:\futsoecon
nector\output\funambol-utsoe-6.5.6\sql
[jar] Building jar: C:\futsoeconnector\output\funambol-utsoe-6.5.6\lib\fun
ambol-utsoe-6.5.6.jar
[jar] Building jar: C:\futsoeconnector\output\funambol-utsoe-6.5.6.s4j

clean-module:
[echo] Cleaning output/funambol-utsoe-6.5.6
[delete] Deleting directory C:\futsoeconnector\output\funambol-utsoe-6.5.6
[zip] Building zip: C:\futsoeconnector\output\funambol-utsoe-6.5.6.zip

BUILD SUCCESSFUL
Total time: 3 seconds
C:\futsoeconnector\build>
```

Figura 3: Construcción del módulo .s4j

El proceso de construcción crea el directorio `\output` conteniendo el fichero de archivos del módulo `funambol-utsoe-6.5.6`.

2.10 INSTALANDO EL MÓDULO.

En este procedimiento `<DS_SERVER_HOME>` representa el directorio que contiene al Servidor Funambol DS (por ejemplo `c:\Archivos de Programa\Funambol\ds-server`).

1. Copiar el fichero de archivos del módulo `funambol-utsoe-6.5.2.s4j` al directorio `<DS_SERVER_HOME>\modules`.
2. Usando un editor de textos, abrir el fichero `<DS_SERVER_HOME>\install.properties`.
3. Encontrar la línea que comienza `modules-to-install` en la sección de definiciones de módulo. Esta línea específica, en una lista separada por comas, los módulos a instalar durante la instalación.
4. Agregar `funambol-utsoe-6.5.2` a la lista separada por comas (sin la extensión del nombre del fichero `.s4j`).
5. Guardar y cerrar `install.properties`.
6. En **Windows**, abrir una ventana de comandos seleccionando **Start > All Programs > Accessories > Command Prompt** y ejecutar el script de instalación del servidor escribiendo lo siguiente en el prompt:
`bin\install -modules funambol`

Solo contestar si a la serie de preguntas que aparecen para el momento en que se instalará el nuevo módulo funambol-utsoe-6.5.2.s4j

2.11 CREANDO UNA INSTANCIA DUMMY SYNC SOURCE.

La Herramienta de Administración se utiliza para crear la instancia de Utsoe SyncSource, como sigue:

1. Iniciar el Servidor Funambol DS seleccionando **Start > All Programs > Funambol > Data Synchronization Server > Start**.
2. Iniciar la herramienta Funambol Administration seleccionando **Start > All Programs > Funambol > Administration Tool**. La ventana Administration Tool se despliega.
3. Sobre la barra de Menú Principal, seleccionar **File > Login**. La ventana Login despliega. Verificar que los campos estén establecidos como sigue, o especificar esos valores:

Hostname /IP: <localhost> (debe ser el nombre de tu máquina)
Port: 8080
User Name: admin.
Password: sa.

Click **Login**: La ventana Output en el panel de abajo debe desplegar “conectado”.

4. En el panel de la izquierda, expandir el árbol **localhost** como sigue: **localhost > Modulos > utsoe > FunambolUtsoeConnector**, entonces seleccionar **Utsoe SyncSource**. La pantalla Edición Utsoe SyncSource despliega en el panel derecho.

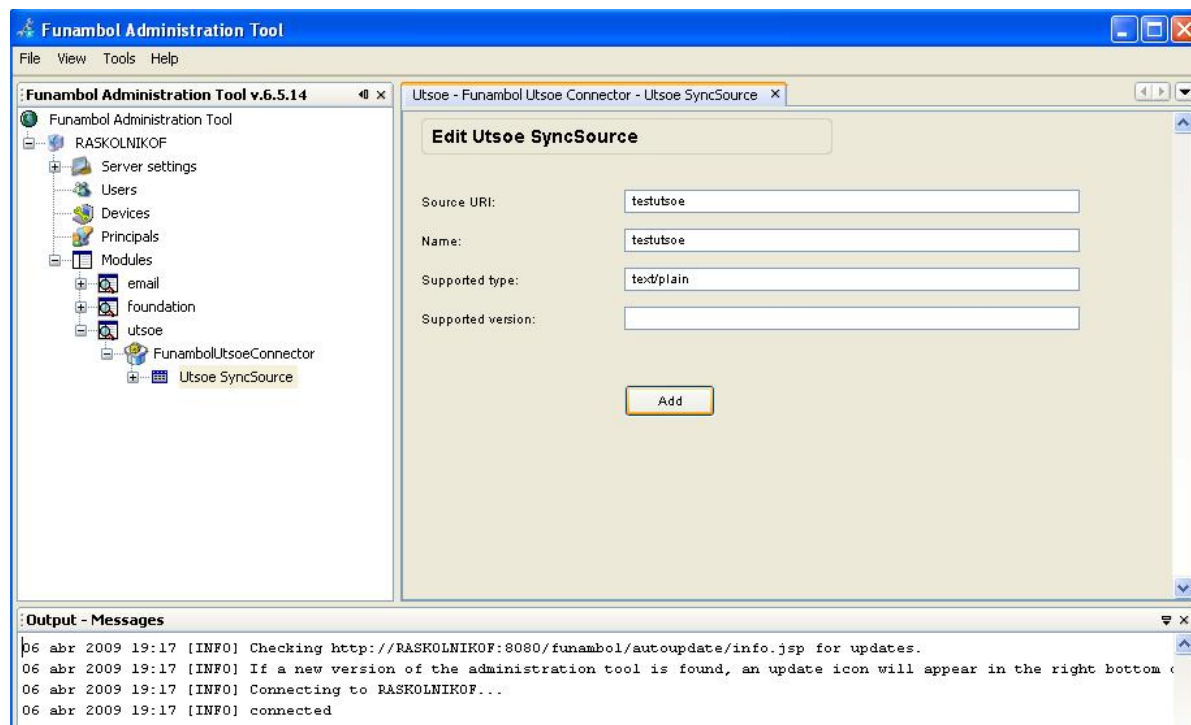


Figura 4: Visualización del módulo incrustado en el servidor Funambol.

5. Especificar los siguientes valores de campo:

Source URI: testutsoe
Name: testutsoe
Type: text/plain

6. Click en **Add**.

2.12 PROBANDO EL MÓDULO CON UN CLIENTE SYNCML.

Para probar el módulo con un cliente SyncML, ejecutar lo siguiente:

1. Descargar el ejemplo Funambol Java Command Line y desempacar el archivo.

San Cristóbal, Venezuela

2. Copiar el fichero `\futsoeconnector\examples\utsoe.properties\` al directorio `\cl-client\config\spds\sources`. No debe haber otros ficheros de propiedades en este directorio.
3. Crear el directorio `\cl-client\db\utsoe`.
4. Verificar que la variable de entorno `JAVA_HOME` este establecida correctamente.
5. Ejecutar `run.cmd` (o `run.sh` en Linux).

Si se tiene éxito, el directorio `\db\utsoe` contiene tres nuevos archivos llamados **10**, **30** y **40**; esos son los elementos generados por `UtsoeSyncSource`. Se puede también inspeccionar el contenido para verificar que corresponde al texto establecido en el código `SyncSource`.

También se puede ver el resultado de la sincronización en el log del servidor.

3. CONCLUSIONES.

Con Funambol DS Server es posible tener una plataforma para administrar dispositivos móviles de código abierto que permite concentrar el esfuerzo desarrollo de componentes utilizando los servicios de autenticación y seguridad ya incluidos en la plataforma. La extensión de la funcionalidad es muy práctica y relativamente sencilla, a través de este artículo hemos tratado de indicar la forma de incrustar el conector `funambol-utsoe-6.5.2.s4j` en su arquitectura beneficiándonos de una plataforma segura y confiable, con posibilidades de sincronizarse con una amplia gama de dispositivos móviles de diversos fabricantes así como a fuentes de datos muy populares.

4. REFERENCIAS.

- [1] Funambol <https://www.forge.funambol.org/download/>
- [2] Funambol Connector Development Tutorial
http://download.forge.objectweb.org/sync4j/funambol_ds_server_module_development_tutorial.pdf
- [3] *SyncSource API* http://download.forge.objectweb.org/sync4j/funambol_ds_server_syncsource_api.pdf.
- [4] Apache ant <http://ant.apache.org/>.

Authorization and Disclaimer .

Authors authorize LACCEI to publish the paper in the conference proceedings. Neither LACCEI nor the editors are responsible either for the content or for the implications of what is expressed in the paper.