

# SINCRONIZACIÓN DE DATOS DE USUARIO EN REDES DE PRÓXIMA GENERACIÓN



**EIVAR EDER ARMERO LUNA**

**DIEGO FERNANDO RODRIGUEZ CHAMORRO**

**ANEXO D**

**IMPLEMENTACIÓN DEL MECANISMO DE REPLICACIÓN DE INFORMACIÓN DE USUARIO EN  
IMS A TRAVÉS DE LA INTERFAZ Sh**

**Universidad del Cauca  
Facultad de Ingeniería Electrónica y Telecomunicaciones  
Departamento de Telemática  
Servicios Avanzados de Telecomunicaciones  
Popayán, Septiembre de 2008**

## **ANEXO D**

### **IMPLEMENTACIÓN DEL MECANISMO DE REPLICACIÓN DE INFORMACIÓN DE USUARIO EN IMS A TRAVES DE LA INTERFAZ Sh**

## TABLA DE CONTENIDO

1.	INTRODUCCIÓN.....	1
2.	DESCRIPCIÓN DEL MANEJO Y FUNCIONAMIENTO DE LA INTERFAZ Sh EN EL WLSS.....	1
3.	HABILITACIÓN DE LA INTERFAZ Sh EN EL WLSS .....	2
3.1	Configuración del canal DIAMETER .....	2
3.2	Extensión de la consola Web para DIAMETER y <i>Profile Service</i> .....	4
3.3	Configuración del Cliente DIAMETER.....	5
3.4	Configuración del <i>Profile Service</i> API.....	8
4.	CONFIGURACIÓN DE LOS NODOS PARA LA INTERFAZ Sh Y LA TABLA DE PERMISOS EN EL FHoSS.....	9
5.	APLICACIÓN DIAMETER DE PRUEBAS PARA LA INTERFAZ Sh .....	10
5.1	Introducción a la Aplicación DIAMETER de Pruebas .....	11
5.2	Descarga y despliegue de la Aplicación de Pruebas.....	11
5.3	Descripción detallada de la Aplicación de Pruebas.....	12
5.3.1	Conexión y Desconexión al FHoSS .....	12
5.3.2	Procedimiento Sh-Pull .....	13
5.3.3	Procedimiento Sh-Subs-Notif .....	13
5.3.4	Procedimiento Sh-Update.....	14
6.	HABILITACIÓN DEL REPOSITORY DATA EN LA BASE DE DATOS DEL FHoSS .....	14
6.1	Ubicación del <i>Repository Data</i> .....	14
6.2	Ingreso de los primeros datos al Repository Data usando un script PHP .....	15

## LISTA DE FIGURAS

Figura 1. . Implementación del <i>Profile Service</i> API y del Proveedor Sh .....	1
Figura 2. Ventana para crear un nuevo canal en el WLSS .....	2
Figura 3. Creando un nuevo canal DIAMETER.....	3
Figura 4. Habilitación del nuevo canal DIAMETER .....	3
Figura 5. Consola web para configuración del cliente DIAMETER .....	6
Figura 6. Configuración del canal DIAMETER .....	7
Figura 7. Configuración del prefijo de mapeo en el <i>Service Profile</i> .....	9
Figura 8. Ventana Inicial de la interfaz web de la Aplicación de Pruebas.....	11
Figura 9. Despliegue de la aplicación de pruebas.....	12
Figura 10. Ventana para el procedimiento Sh-Pull.....	13
Figura 11. Ventana para el procedimiento Sh-Subs-Notif.....	13
Figura 12. Ventana para el Sh-Update .....	14
Figura 13. Tabla del <i>Repository Data</i> en la base de datos del FHoSS.....	15

## 1. INTRODUCCIÓN

A continuación se describe la implementación del mecanismo de sincronización de información de usuario en el IMS a través de la interfaz Sh, usando el BEA Weblogic SIP Server (WLSS) y el HSS del Open IMS Core (FHoSS).

## 2. DESCRIPCIÓN DEL MANEJO Y FUNCIONAMIENTO DE LA INTERFAZ Sh EN EL WLSS

La interfaz Sh en el WLSS se implementa como un proveedor del protocolo base DIAMETER. Dicho proveedor genera y responde automáticamente a los códigos de comando definidos en la especificación de la aplicación Sh. Un *Profile Service* API habilita *SIP Servlets* para gestionar los datos del perfil de usuario como un documento XML usando *XML Document Object Model* (DOM). Las subscripciones y notificaciones para cambiar la información del perfil de usuario son manejadas con la implementación de un *Profile Listener* en un *SIP Servlet* **¡Error! No se encuentra el origen de la referencia.** En la Figura 1 se puede observar la implementación del *Profile Service* API y del Proveedor Sh.

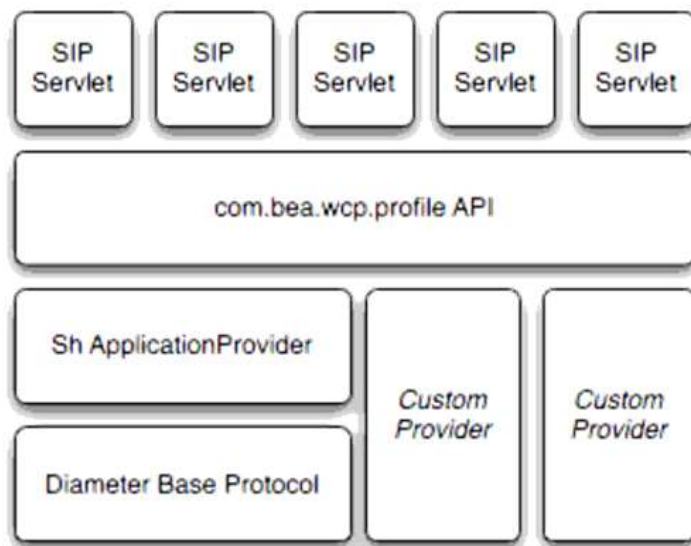


Figura 1. . Implementación del *Profile Service* API y del Proveedor Sh

Con la habilitación y configuración de un Cliente DIAMETER en el WLSS se garantiza la posibilidad de usar el protocolo base DIAMETER y con la configuración de la aplicación Cliente Sh se habilita el Proveedor Sh.

Posteriormente usando el *Profile Service* API se puede crear una aplicación para el manejo del perfil de usuario. Con los métodos de la clase *Profile Service*, se puede recuperar información del perfil de usuario obteniendo un documento XML al usar una llave como la siguiente: `protocol://uri/reference_type[/access_key]`, donde por ejemplo, para recuperar información del *Repository Data* la llave sería algo como `sh://sip:user@bea.com/RepositoryData/Call Screening/`.

Con otros métodos de la misma clase, se puede crear una aplicación que copie el documento XML y posteriormente modificarlo y enviarlo donde se requiera. El uso del *Profile Service* API, BEA Weblogic se lo deja totalmente al usuario.

Así mismo, el WLSS provee un API para la implementación de un *Profile Listener* para recibir notificaciones automáticas de los cambios en el perfil de usuario.

### 3. HABILITACIÓN DE LA INTERFAZ Sh EN EL WLSS

Con la configuración descrita en este numeral se asegura el uso del protocolo base DIAMETER, del Proveedor Sh y la configuración del *Profile Service* API para su uso en una aplicación determinada.

#### 3.1 Configuración del canal DIAMETER<sup>1</sup>

En primer lugar, se ingresa un registro en el DNS para el WLSS, el cual se debe encontrar dentro del dominio de la red *home* IMS, puesto que la interfaz Sh se define para interconectar un AS y un HSS que se encuentren dentro de la misma red. Por ejemplo, para el caso específico de este proyecto, al WLSS se le asignó el registro findme.anubis.unicauca.edu.co, ya que todos los nodos de la red *home* IMS en están en el dominio anubis.unicauca.edu.co.

A continuación se habilita un canal DIAMETER en el WLSS. Para eso se dirige a la consola web del mismo, ingresando a la pestaña del servidor sobre el que se desea realizar la configuración. Posteriormente se dirige a la pestaña *Protocols* y dentro de esta a la sección *Channels*, en donde se procede a crear un nuevo canal haciendo clic en *New*, como se observa en la Figura 2.

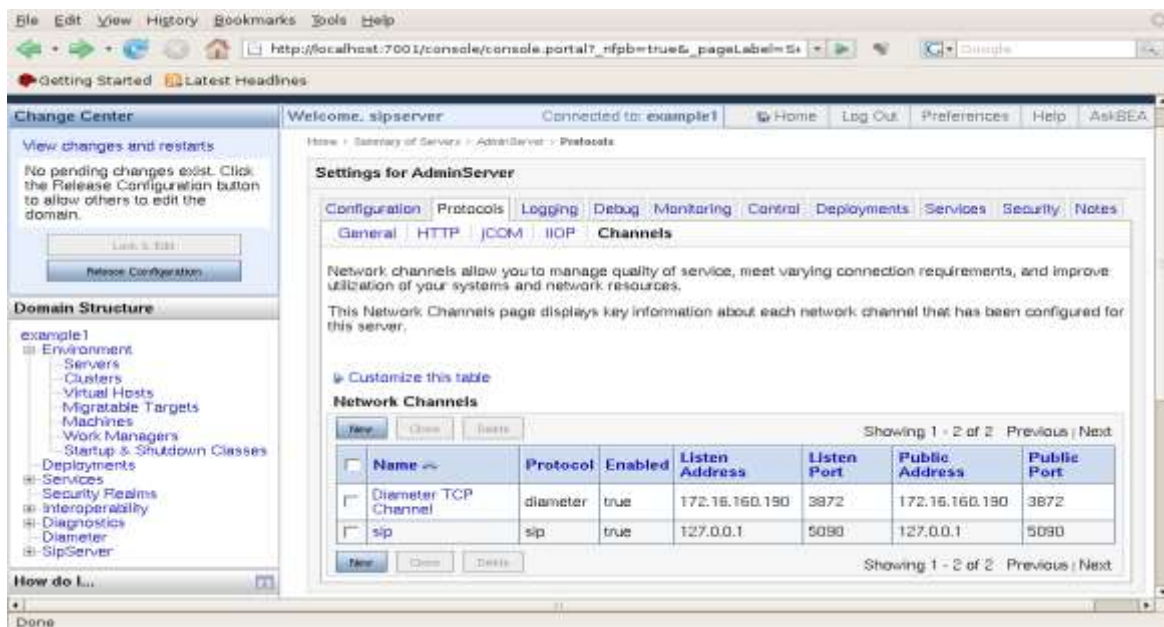


Figura 2. Ventana para crear un nuevo canal en el WLSS

1 Para mayor información sobre los diferentes tipos de configuración de un canal DIAMETER en el WLSS remítase al documento <http://e-docs.bea.com/wlcp/wlss31/confignetwork/diameterconfig.html#channels>

En la siguiente ventana se ingresa un nombre para el canal y se selecciona el protocolo que se va a usar en el mismo. Para usar DIAMETER sobre TCP se selecciona la opción *diameter* como en la Figura 3.

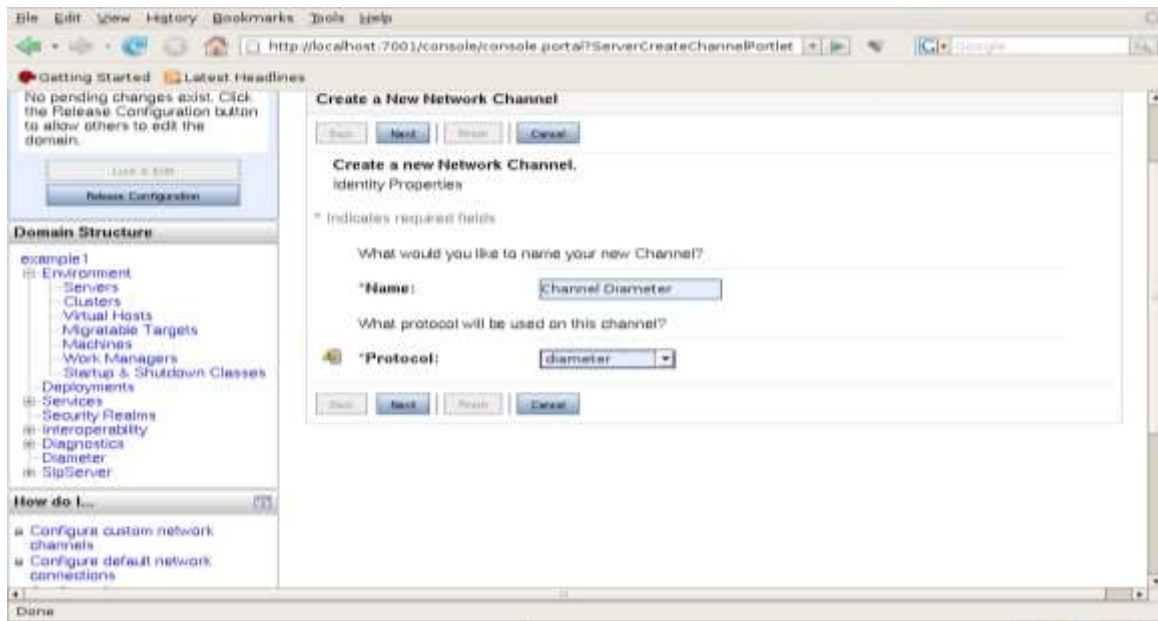


Figura 3. Creando un nuevo canal DIAMETER

A continuación se ingresa la dirección IP en la que el WLSS debe escuchar las peticiones DIAMETER, la cual debe corresponder con el registro agregado en el DNS, así como también el puerto de escucha, que para DIAMETER por defecto es el 3868, aunque se puede escoger cualquier otro que no esté siendo usado en la misma máquina. En este caso se usó el puerto 3872 ya que el 3868 se usa por el FHoSS que corre en la misma máquina.

Se continúa configurando las propiedades adicionales del canal, únicamente seleccionando la casilla *Enabled* y *Outbound Enabled* para el caso de un canal DIAMETER como en la Figura 4 y se hace click en *Finish* para completar el proceso.

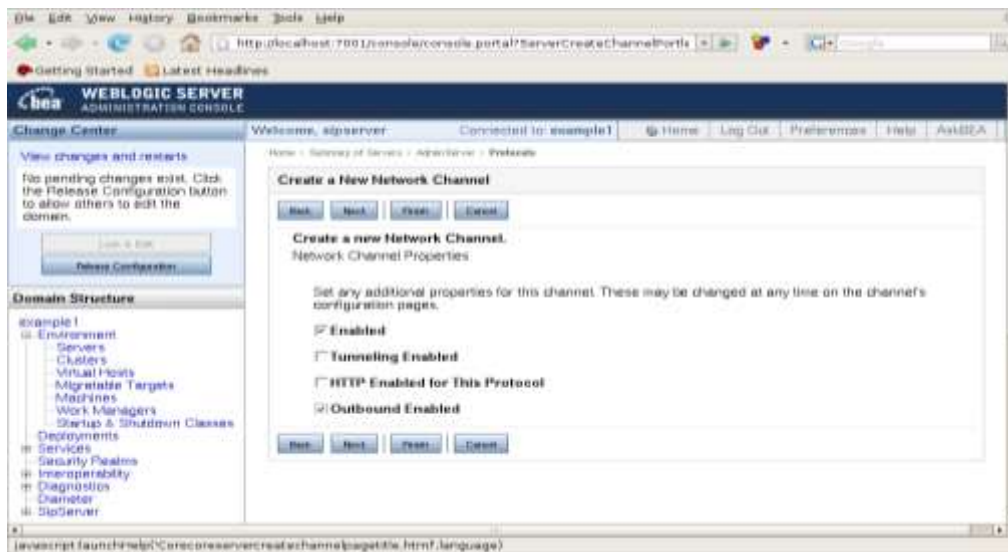


Figura 4. Habilitación del nuevo canal DIAMETER

### 3.2 Extensión de la consola Web para DIAMETER y *Profile Service*<sup>2</sup>

Para poder crear una extensión en la consola web tanto para configurar el Cliente DIAMETER como el *Profile Service*, se necesita agregar unas líneas al archivo config.xml ubicado en `~/bea/user_projects/domains/domainname/config/config.xml`.

Las líneas que se deben adicionar son las que se encuentran en negrilla:

```

<custom-resource>
  <name>sipserver</name>
  <target>AdminServer</target>
  <descriptor-file-name>custom/sipserver.xml</descriptor-file-name>
  <resource-
class>com.bea.wcp.sip.management.descriptor.resource.SipServerResource</resource-class>
  <descriptor-bean-
class>com.bea.wcp.sip.management.descriptor.beans.SipServerBean</descriptor-bean-class>
</custom-resource>
<custom-resource>
  <name>diameter</name>
  <target>AdminServer</target>
  <deployment-order>200</deployment-order>
  <descriptor-file-name>custom/diameter.xml</descriptor-file-name>
  <resource-class>com.bea.wcp.diameter.DiameterResource</resource-class>
  <descriptor-bean-
class>com.bea.wcp.diameter.management.descriptor.beans.DiameterBean</descriptor-bean-
class>

```

<sup>2</sup> Para mayor información sobre la extensión de la consola web remítase al documento <http://e-docs.bea.com/wcp/wlss31/confignetwork/diameterconfig.html#clients>



```
</custom-resource>
<custom-resource>
  <name>ProfileService</name>
  <target>AdminServer</target>
  <deployment-order>300</deployment-order>
  <descriptor-file-name>custom/profile.xml</descriptor-file-name>
  <resource-class>com.bea.wcp.profile.descriptor.resource.ProfileServiceResource</resource-
class>
  <descriptor-bean-class>com.bea.wcp.profile.descriptor.beans.ProfileServiceBean</descriptor-
bean-class>
</custom-resource>
<admin-server-name>AdminServer</admin-server-name>
</domain>
```

Ahora, para correr el WLSS otra vez es necesario que existan los archivos `diameter.xml` y `profile.xml`, que son los que describen la configuración del cliente DIAMETER y del *Profile Service*. Estos dos archivos se describen en las siguientes secciones.

### 3.3 Configuración del Cliente DIAMETER<sup>3</sup>

La configuración del Cliente DIAMETER se puede realizar a través del archivo de configuración `diameter.xml` o mediante la extensión en la consola web.

Si el archivo `diameter.xml` no existe, no se puede correr el servidor y por ende no se puede usar la consola web. Por tal motivo, primero se crea el archivo basado en una configuración pre-existente y posteriormente se explica dicha configuración y la forma de modificarla usando la consola web.

En el directorio `~/bea/user_projects/domains/example1/config/custom` se crea el archivo `diameter.xml` con el siguiente contenido.

```
<?xml version='1.0' encoding='utf-8'?>
<diameter xmlns="http://www.bea.com/ns/wlcp/diameter/300"
xmlns:sec="http://www.bea.com/ns/weblogic/90/security"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:wls="http://www.bea.com/ns/weblogic/90/security/wls">
  <configuration>
    <name>DiameterConfig</name>
    <target>AdminServer</target>
    <host>findme.anubis.unicauca.edu.co</host>
    <realm>anubis.unicauca.edu.co</realm>
    <message-debug-enabled>true</message-debug-enabled>
    <application>
      <name>WlssShApplication</name>
      <class-name>com.bea.wcp.diameter.sh.WlssShApplication</class-name>
      <param>
        <name>destination.realm</name>
        <value>anubis.unicauca.edu.co</value>
      </param>
    </application>
    <peer-retry-delay>10</peer-retry-delay>
    <allow-dynamic-peers>false</allow-dynamic-peers>
    <peer>
      <host>hss.anubis.unicauca.edu.co</host>
      <address>172.16.160.190</address>
      <port>3868</port>
      <protocol>tcp</protocol>
      <watchdog-enabled>true</watchdog-enabled>
    </peer>
    <route>
      <name>FoHSS</name>
      <realm>anubis.unicauca.edu.co</realm>
      <application-id>16777217</application-id>
      <action>redirect</action>
    </route>
  </configuration>
</diameter>
```

---

<sup>3</sup> Para mayor información sobre la configuración de un cliente DIAMETER refiérase al documento <http://e-docs.bea.com/wlcp/wlss31/confignetwork/diameterconfig.html#clients>

```

    <server>hss.anubis.unicauca.edu.co</server>
  </route>
</default-route>
  <action>redirect</action>
  <server>hss.anubis.unicauca.edu.co</server>
</default-route>
</configuration>
</diameter>

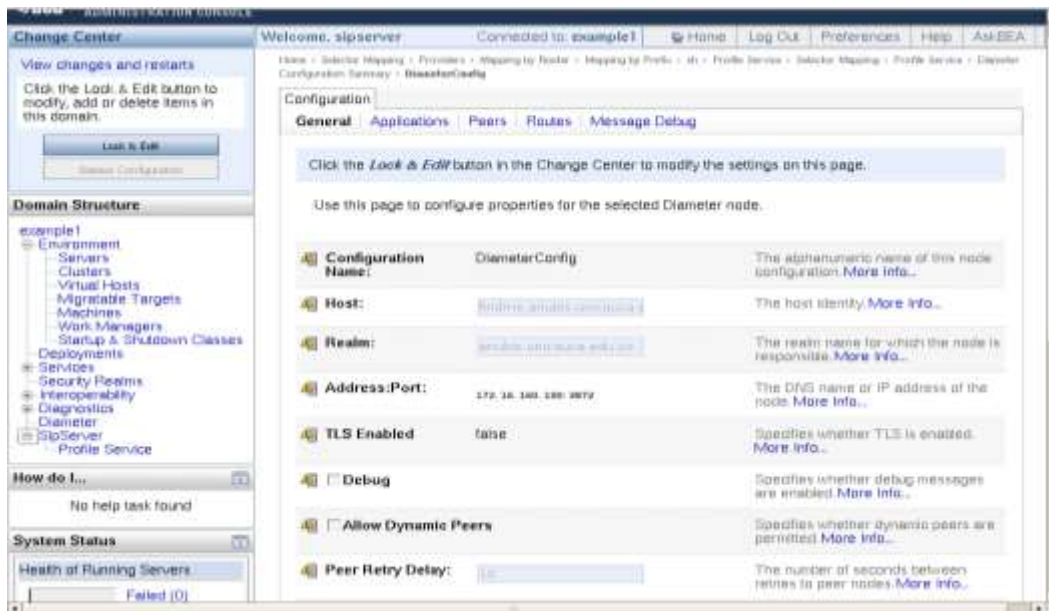
```

En la consola web ahora se puede observar la extensión para el Cliente DIAMETER en el panel izquierdo (Figura 5). A continuación se describe la configuración que se obtuvo con los datos que se ingresaron en el archivo diameter.xml.

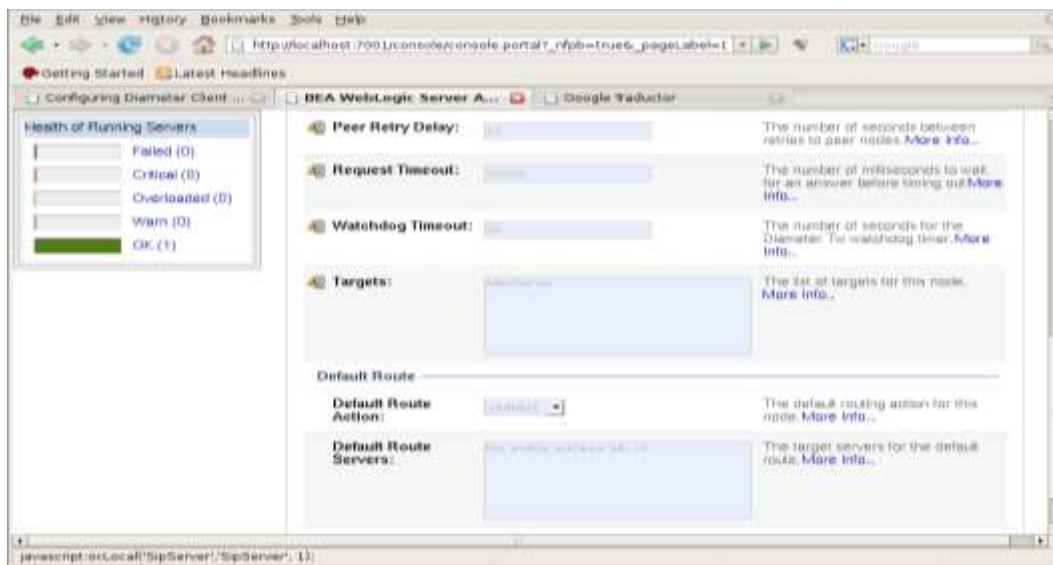


Figura 5. Consola web para configuración del cliente DIAMETER

En *Diameter Configurations* (Figura 6) se creó una configuración llamada DiameterConfig, la cual tiene como host destino findme.anubis.unicauca.edu.co y como dirección y puerto las del canal DIAMETER que se creó anteriormente, es decir 172.16.160.190:3872.



(a)



(b)

Figura 6. Configuración del canal DIAMETER

Ingresando a la configuración general también se puede verificar o modificar otros valores que se ingresaron directamente en el archivo diameter.xml. Por ejemplo, el *Peer Retry Delay* hace referencia al tiempo en segundos entre cada intento de conexión con otro nodo DIAMETER.

Así mismo, la casilla *Allow Dynamic Peers* no se encuentra seleccionada, indicando que no se ha habilitado la opción de que otros nodos se conecten al WLSS sin que hayan sido previamente configurados.

También se especifica una ruta por defecto para enviar las peticiones que no coincidan con una ruta configurada con anterioridad. Entonces, se usa la acción *redirect* y se ingresa el nombre del nodo al cual que debe enviar dichas peticiones.

En el Cliente DIAMETER interesa sobre todo configurar la Aplicación Cliente Sh, lo que se lleva a cabo en la pestaña *Applications*, donde se encuentra la configuración ingresada en el archivo con el nombre *WlssShApplication* y la clase del API que se encarga de manejar los procesos de la interfaz Sh, la cual es *com.bea.wcp.diameter.sh.WlssShApplication*. Así mismo, también se puede cambiar la configuración de esta aplicación, por ejemplo en los parámetros que recibe la clase, como la ruta para las peticiones DIAMETER que en este caso se reenvían al dominio *anubis.unicauca.edu.co*.

En la pestaña *Peers*, se configura el o los nodos a los que se va a conectar el WLSS, que para este caso es el FHSS. Se especifica el nombre, la dirección, puerto y protocolo de transporte.

Y para finalizar se configura una ruta para re direccionar las peticiones de una aplicación específica. Esto se hace ingresando el *Application ID* - que para la interfaz Sh es *16777217* -, una acción para reenviar los mensajes, el dominio y el nodo DIAMETER con el que se va a tener contacto.

### 3.4 Configuración del *Profile Service API*

Aquí se puede seleccionar, con qué prefijo mapear las peticiones de documentos del perfil y a cuál *Profile Service Provider* hacerla. A su vez, el *Profile Service Provider* se contacta con la Aplicación Cliente para la Interfaz Sh.

Como para el caso del Cliente DIAMETER, para el *Service Profile* también existe un archivo en el que se guarda la configuración, dicho archivo es el *profile.xml* que se debe ubicar en *~/bea/user\_projects/domains/example1/config/custom* con el siguiente contenido.

```
<profile-service xmlns="http://www.bea.com/ns/wlcp/wlss/profile/300">
  <mapping>
    <map-by>prefix</map-by>
    <map-by-prefix>
      <provider-prefix-set>
        <name>sh</name>
        <prefix>sh</prefix>
      </provider-prefix-set>
    </map-by-prefix>
  </mapping>
  <provider>
    <name>sh</name>
    <provider-class>com.bea.wcp.profile.ShProviderCached</provider-class>
  </provider>
</profile-service>
```

Las peticiones para la Aplicación Cliente Sh se hacen mediante una llave que inicia con el prefijo *sh://*, por lo tanto como método de mapeo se escoge *by prefix*.

En la extensión de la consola web se puede modificar la configuración que se ingresó en el archivo profile.xml. En la pestaña *Mapping by Prefix* se puede modificar el proveedor y su respectivo prefijo, tal como se muestra en la Figura 7.

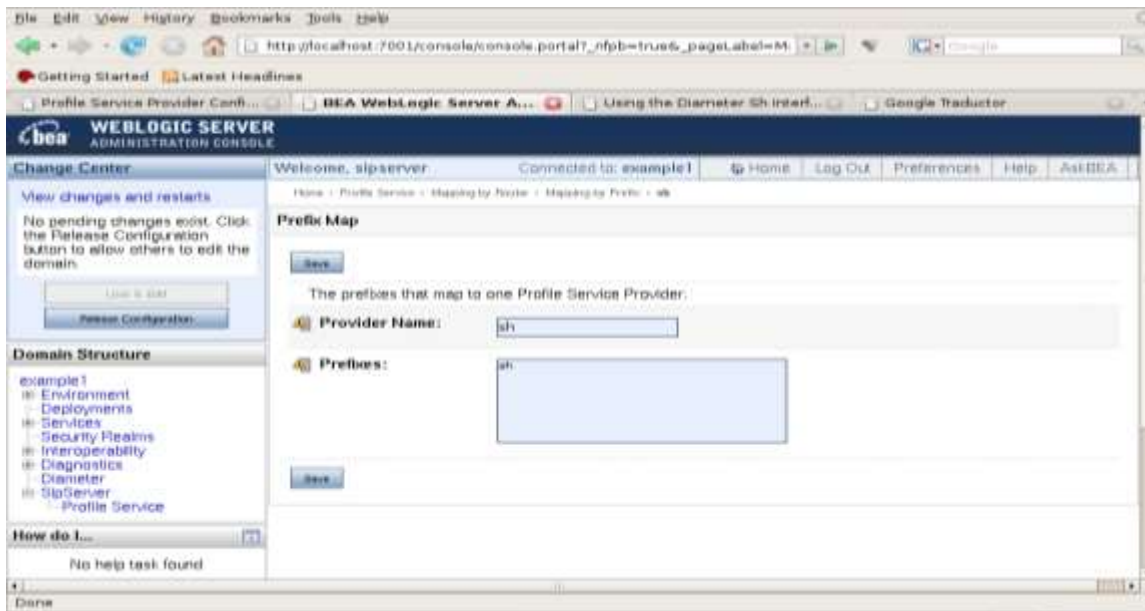


Figura 7. Configuración del prefijo de mapeo en el *Service Profile*

En la pestaña *Providers* se escoge la clase que implementa el *Profile Service Provider*, que para este caso es `com.bea.wcp.profile.ShProviderCached`.

#### 4. CONFIGURACIÓN DE LOS NODOS PARA LA INTERFAZ Sh Y LA TABLA DE PERMISOS EN EL FHoSS

Además de configurar la Aplicación Sh en el WLSS con el canal DIAMETER y los nodos al los cuales se va a conectar a través de dicha interfaz, también es necesario configurar el FHoSS para que este acepte la comunicación con el WLSS. Este procedimiento es muy sencillo y simplemente se debe agregar a la lista de nodos permitidos por el FHoSS la dirección del WLSS.

El archivo XML que se debe modificar es el `/opt/OpenIMSCore/FHoSS/deploy/DiameterPeerHSS.xml` ingresando la FQDN (*Fully Qualified Domain Name*) del WLSS, el dominio al que pertenece y el puerto por el cual este nodo escucha los mensajes DIAMETER.

Dicho archivo debe quedar con la siguiente información, agregando la línea en negrita y modificándola según sea el caso.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- HSS Server config -->
<DiameterPeer
    FQDN="hss.anubis.unicauca.edu.co"
    Realm="anubis.unicauca.edu.co"
```

```

Vendor_Id="10415"
Product_Name="JavaDiameterPeer"
AcceptUnknownPeers="1"
DropUnknownOnDisconnect="1"
Tc="30"
Workers="4"
QueueLength="32"
>
<Peer FQDN="icscf.anubis.unicauca.edu.co" Realm="anubis.unicauca.edu.co" port="3869" />
<Peer FQDN="scscf.anubis.unicauca.edu.co" Realm="anubis.unicauca.edu.co" port="3870" />
<Peer FQDN="findme.anubis.unicauca.edu.co" Realm="anubis.unicauca.edu.co"
port="3872" />

<Acceptor port="3868" bind="127.0.0.1" />

<Auth id="16777216" vendor="10415"/><!-- 3GPP Cx -->
<Auth id="16777216" vendor="4491"/><!-- CableLabs Cx -->
<Auth id="16777216" vendor="13019"/><!-- ETSI/TISPAN Cx -->
<Auth id="16777217" vendor="10415"/><!-- 3GPP Sh -->
<Auth id="16777221" vendor="10415"/>

</DiameterPeer>

```

Por otra parte, para permitir la lectura, escritura y subscripción a notificaciones para la información almacenada en el FHoSS, se deben asignar los permisos al WLSS en la tabla de permisos del FHoSS (Ver Anexo B numeral 11).

## 5. APLICACIÓN DIAMETER DE PRUEBAS PARA LA INTERFAZ Sh

Después de habilitar correctamente la interfaz Sh en el WLSS con todos sus componentes, el siguiente paso es desarrollar una aplicación usando el *Profile Service* API, la cual pueda responder a los comandos especificados para la interfaz Sh y manejar la información de usuario en archivos XML.

Como se había expresado, BEA Weblogic le deja la implementación de esta aplicación al usuario, sin embargo han desarrollado una aplicación de pruebas para los procedimientos del manejo de la información de usuario.

Así, con el propósito de ahorrar tiempo en las pruebas y reutilizar las herramientas proporcionadas, en la implementación del mecanismo de replicación de datos de usuario de IMS, se hizo uso de esta Aplicación DIAMETER de Pruebas, la cual se describe a continuación.

## 5.1 Introducción a la Aplicación DIAMETER de Pruebas<sup>4</sup>

La Aplicación DIAMETER de Pruebas permite hacer peticiones y actualizar la información guardada en cualquier HSS al manejar de los procedimientos Sh-Pull, Sh-Update, Sh-Subs-Notif y Sh-Notif consignados en el TS 29.328 del 3GPP.

Estos procedimientos se pueden ejecutar a través de una interfaz web en la que se puede escoger el nombre de usuario, el tipo de petición y el nombre del servicio para el caso del Sh-Pull y los datos del servicio y el valor en el caso del Sh-Update. Así mismo, en el caso del procedimiento Sh-Subs-Notif permite seleccionar el nombre de usuario, el tipo de datos y el nombre del servicio.

En la Figura 8 se puede apreciar la ventana inicial de la interfaz web de la aplicación de pruebas.

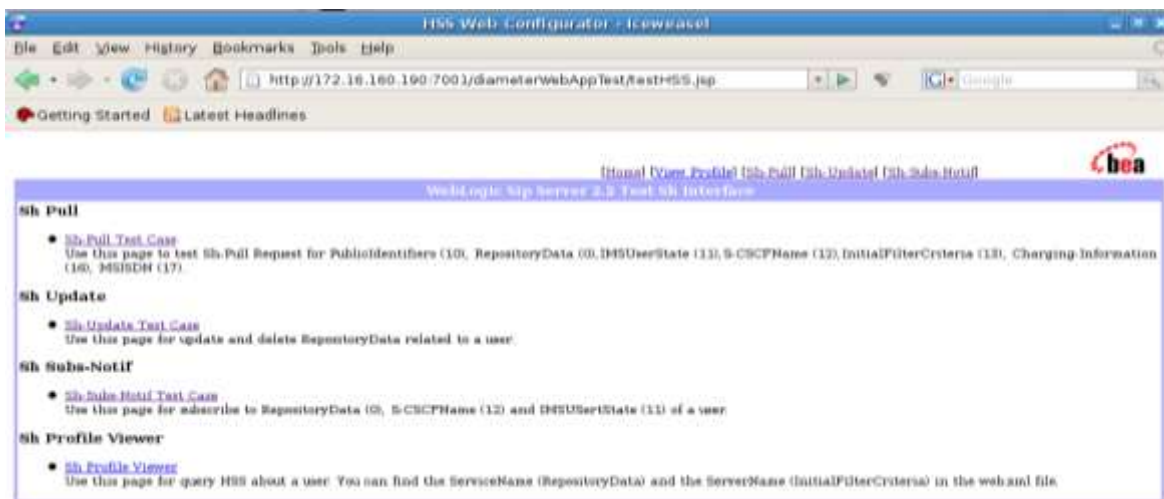


Figura 8. Ventana Inicial de la interfaz web de la Aplicación de Pruebas

## 5.2 Descarga y despliegue de la Aplicación de Pruebas

La aplicación se la puede encontrar en el enlace <http://dev2dev.bea.com/2006/10/DiameterTest.zip> y se debe desplegar en el WLSS de la siguiente forma.

Al extraer el contenido del archivo .zip se encuentra un directorio /bin donde está el archivo diameterWebAppTest.war, el cual se debe desplegar haciendo uso de la consola web del WLSS. Se entra a la consola y se dirige al enlace *Deployments* localizado en el panel izquierdo, se localiza el archivo .war, se instala y se inicia como en la Figura 9.

<sup>4</sup> Para mayor información sobre la Aplicación DIAMETER de Pruebas refiérase al documento <http://www.oracle.com/technology/pub/articles/dev2arch/2006/10/home-subscriber-server.html>





Figura 9. Despliegue de la aplicación de pruebas

A la interfaz web de la aplicación se ingresa por el enlace <http://direccióndelWLS:7001/diameterWebAppTest/testHSS.jsp>.

### 5.3 Descripción detallada de la Aplicación de Pruebas

Una vez desplegada la aplicación se describe su funcionamiento.

#### 5.3.1 Conexión y Desconexión al FHoSS

Dado que hasta el momento ya se tiene habilitada la aplicación de pruebas y se han agregado como nodos DIAMETER permitidos el FHoSS y el WLSS mutuamente, una vez los dos estén corriendo se intentan conectar automáticamente. El WLSS envía un Capability-Exchange-Request (CER) al FHoSS y este retorna un Capability-Exchange-Answer (CEA) indicándole al WLSS que sí soporta la aplicación Sh.

En la consola del FHoSS ya no se despliega más el mensaje:

```
ERROR de.fhg.fokus.diameter.DiameterPeer.peer.StateMachine - I_Snd_Conn_Req StateMachine:
Peer findme.anubis.unicauca.edu.co not responding to connection attempt.
```

Y en la consola del WLSS se observa el mensaje de conexión abierta:

```
<Notice> <Diameter.Core> <BEA-340012> <Peer "hss.anubis.unicauca.edu.co" transitioned from
state "Closed" to "R-Open">.
```

Con esto se asegura que los dos nodos se encuentran en comunicación por la interfaz Sh y se puede proseguir con las demás pruebas.

### 5.3.2 Procedimiento Sh-Pull

En la Figura 10 se observa la interfaz web para el procedimiento Sh-Pull. Con este procedimiento se puede recuperar información de usuario almacenada en el FHoSS en un formato XML.

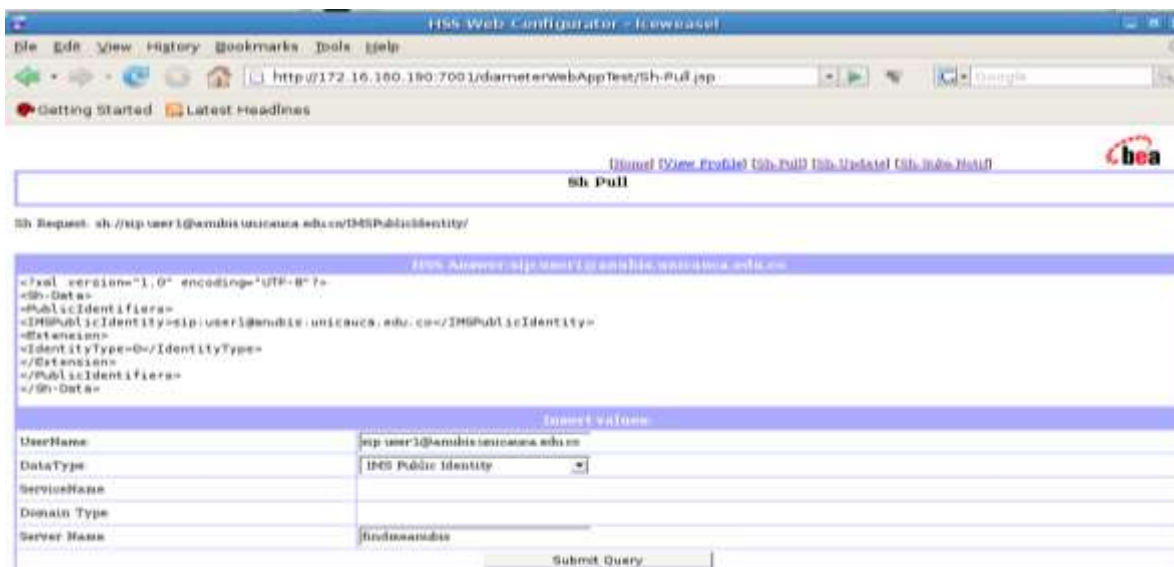


Figura 10. Ventana para el procedimiento Sh-Pull

### 5.3.3 Procedimiento Sh-Subs-Notif

Haciendo uso de este procedimiento se puede suscribir a notificaciones de cambios en la información del *Repository Data*, *S-CSCF Name* y *IMS User State*.

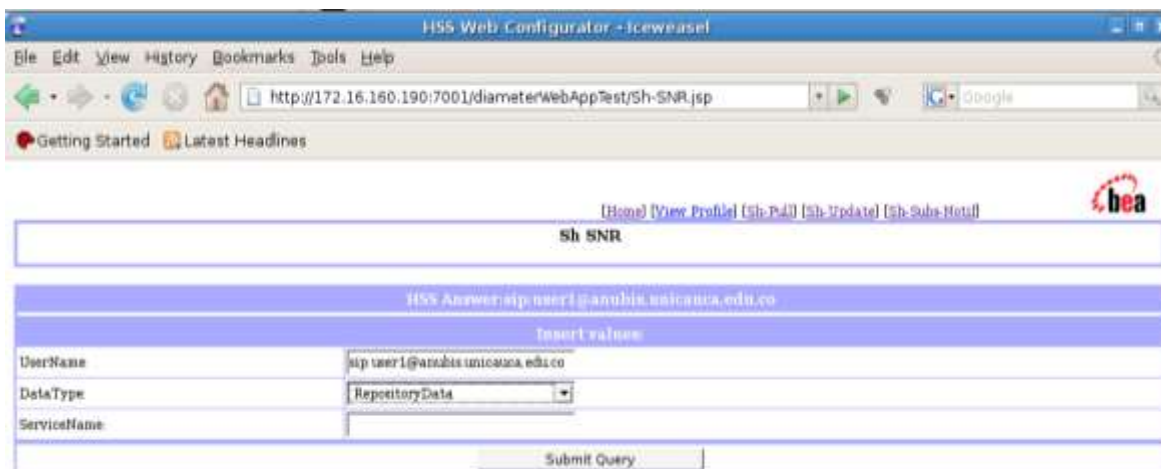


Figura 11. Ventana para el procedimiento Sh-Subs-Notif

### 5.3.4 Procedimiento Sh-Update

Dependiendo de los permisos que tenga el WLSS en la lista del FHoSS se pueden hacer actualizaciones en la información contenida en este último usando el procedimiento Sh-Update. En la Figura 12 se observa la ventana para este procedimiento.

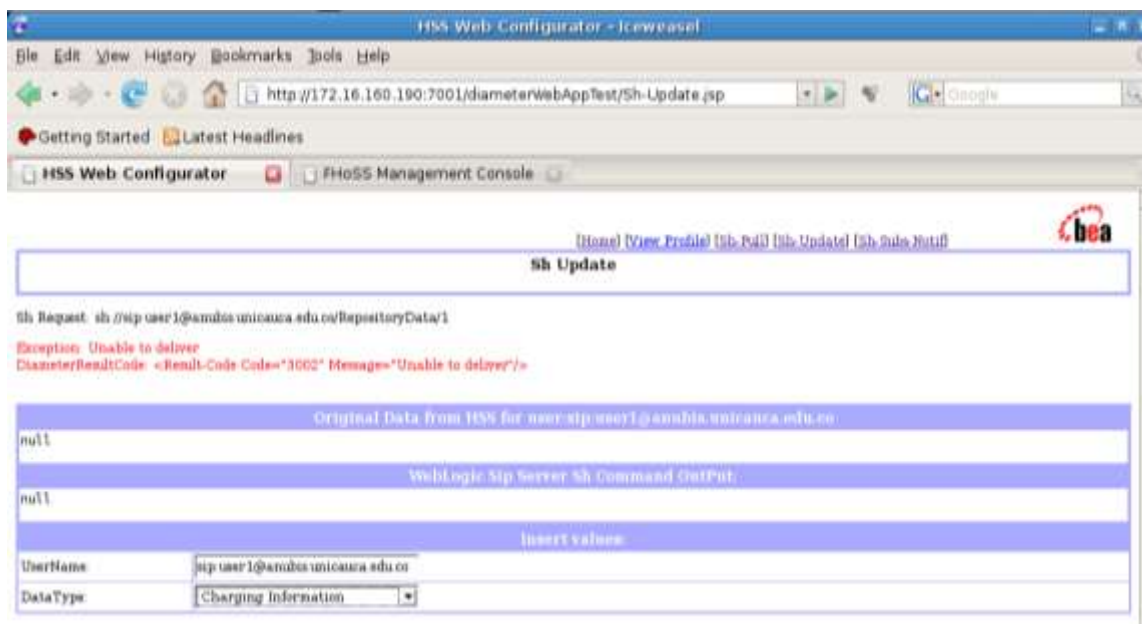


Figura 12. Ventana para el Sh-Update

Para poder actualizar información en el *Repository Data* se debe haber introducido antes alguna información manualmente en la base de datos del FHoSS, procedimiento que se explica en la sección 6.

## 6. HABILITACIÓN DEL REPOSITORY DATA EN LA BASE DE DATOS DEL FHoSS

### 6.1 Ubicación del *Repository Data*

El FHoSS tiene una base de datos llamada hss\_db. Haciendo uso de la plataforma para gestión de bases de datos phpMyAdmin, se ubica dicha base de datos y se observa sus diferentes tablas correspondientes a los tipos de datos que este guarda.

Así, en la Figura 13 se aprecia que la tabla repository\_data es la que guarda la información de usuario que puede actualizarse desde el WLSS. Esta tabla tiene los siguientes campos:

- id: corresponde al identificador para uno de los campos de información del *Repository Data*.
- sqn: es el número de la secuencia de actualización del campo.
- id\_impu: es el identificador de la Identidad Pública correspondiente al usuario al que pertenece la información del campo en cuestión.
- service\_indication: corresponde al nombre del servicio para el que se guarda la información.
- rep\_data: es el valor de la información que se guarda en el *Repository Data*.

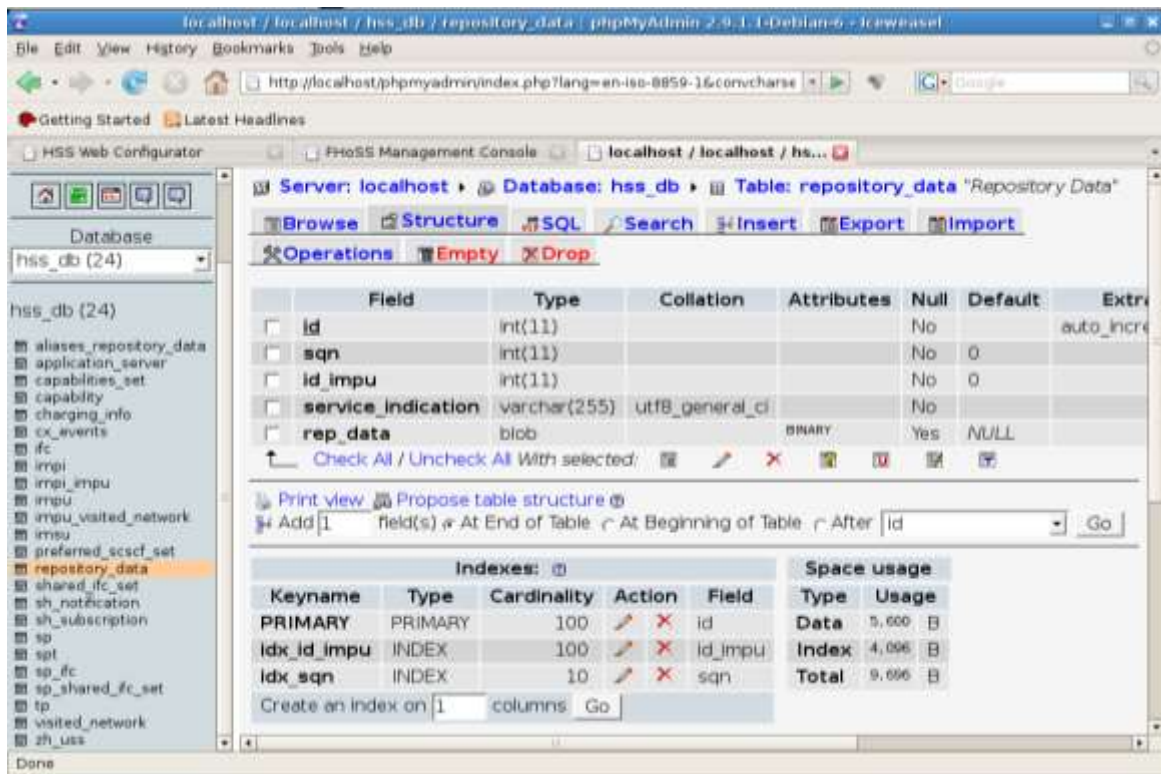


Figura 13. Tabla del *Repository Data* en la base de datos del FHoSS

## 6.2 Ingreso de los primeros datos al Repository Data usando un script PHP

Para ingresar por primera vez datos a la tabla `repository_data` de la base de datos del hss, se utilizó un script PHP cuya única función es generar un número determinado de usuarios insertando en el campo `rep_data` - que es donde se almacena la información de usuario -, una cadena de texto que se lee desde la misma carpeta donde se encuentra ubicado el script. Con esto, la lista de usuarios ya se habilita para ser actualizada.

El código del script PHP utilizado se muestra a continuación:

```
<?php
$dbhost="localhost";
$dbusuario="root";
$dbpassword="";
$db="hss_db";
$conexion = mysql_connect($dbhost, $dbusuario, $dbpassword);
mysql_select_db($db, $conexion);

$archivo = "/var/www/test/inicial.txt";
$fileHandle = fopen($archivo, "r");
$fileContent = fread($fileHandle,11);
$fileContent = addslashes($fileContent);
```

```
for ($i = 1; $i < 100; $i++){
    $query = 'INSERT INTO `repository_data` (`id`, `sqn`, `id_impu`, `service_indication`,
    `rep_data`) VALUES (NULL, \'0\',\'$i\', \'IOTService\', '$\'$fileContent\');';
    mysql_query($query);
}
echo "datos ",$i," insertados";
?>
```

Como se puede observar únicamente se realiza la carga del contenido del archivo inicial.txt y posteriormente se hace un ciclo de inserciones desde 1 hasta 100 para generar 100 usuarios en la base de datos, insertando para cada uno de ellos el texto por defecto del contenido que se cargo desde el archivo de texto.