

OBCOM ADAPI

Instalación y Configuración

Versión 2.1

OBCOM INGENIERÍA S.A. (Chile)

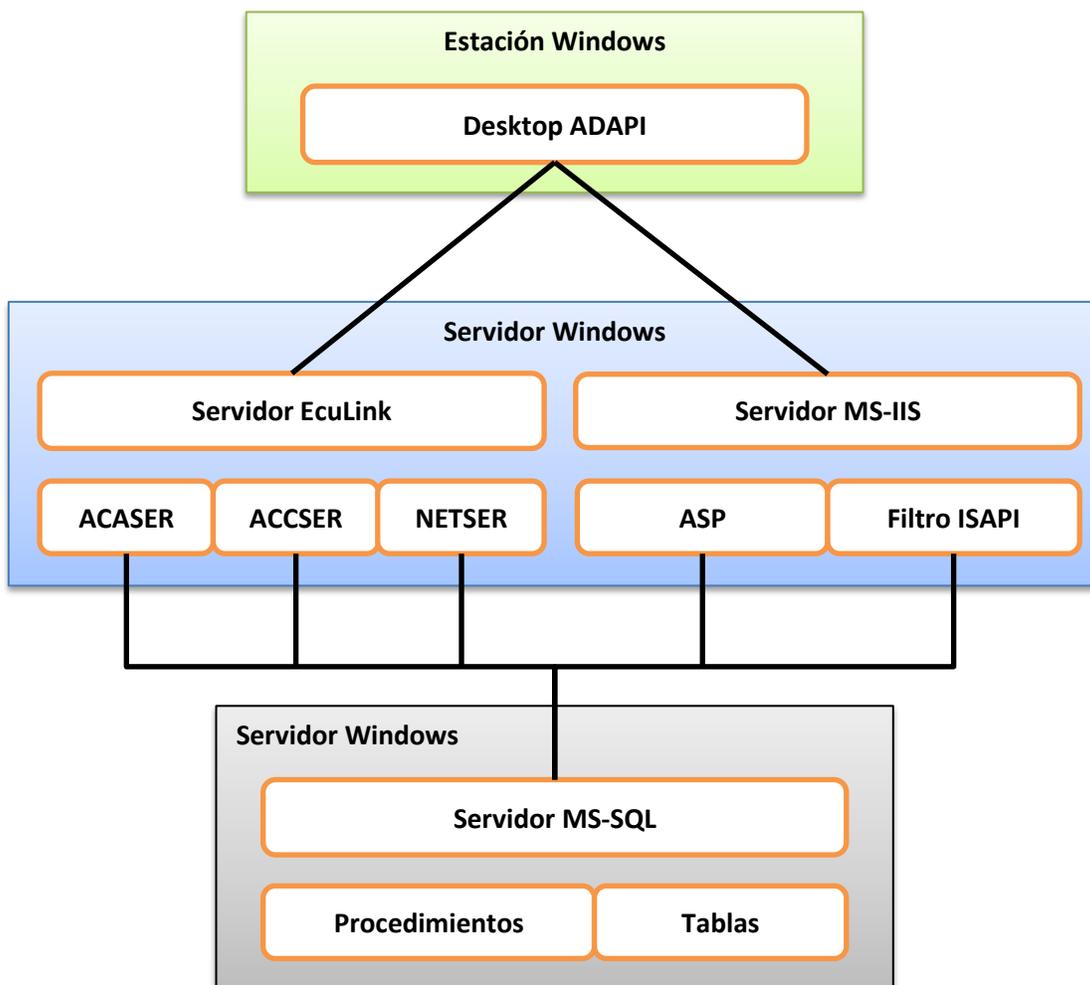
<http://www.obcom.cl>

Contenido

1	Introducción	2
2	Servidor SQL	2
2.1	Base de datos	2
2.2	Cuentas.....	3
2.3	Tablas	3
2.3.1	Tablas de seguridad.....	4
2.3.2	Tablas de preferencias	4
2.4	Procedimientos almacenados	5
3	Servidor EcuLink	5
3.1	Estructura de directorios.....	5
3.2	Servicio Windows	6
3.3	Archivo de configuración	10
4	Servidor IIS	11
4.1	Estructura del sitio web.....	11
4.1.1	Archivo Global.asa.....	12
4.1.2	Archivo EcuStartup.asp	13
4.1.3	Archivo EcuLogin.asp.....	13
4.2	Aplicación internet	13
4.2.1	Filtro de seguridad.....	16
5	Arquitectura HUB	20
5.1	Servidores EcuLink Front y HUB	22
5.2	Configuración servidores EcuLink Front.....	23
5.3	Configuración servidor EcuLink HUB.....	24

1 Introducción

Este documento explica cómo instalar y configurar el producto **OBCOM ADAPI**. Para entender este proceso de instalación y configuración, es importante familiarizarse con la arquitectura general de **OBCOM ADAPI**. La siguiente imagen muestra las principales componentes de la arquitectura:



2 Servidor SQL

El producto **OBCOM ADAPI** requiere un servidor SQL. Este servidor almacena la base de datos de usuarios y perfiles, así como las preferencias de los usuarios del sistema. El producto **OBCOM ADAPI** puede operar con **Microsoft SQL Server**, **Oracle** o **IBM DB2**. Este documento sólo describe la configuración de **Microsoft SQL Server**.

2.1 Base de datos

La base de datos del producto **OBCOM ADAPI** normalmente se llama **ECUBAS**. Esta base de datos tiene como propietario (dbo - database owner) la cuenta **EcuAdmin** (descrita más abajo). En esta

base de datos están todas las tablas y los procedimientos almacenados utilizados por el producto **OBCOM ADAPI**. Todos estos objetos se describen más adelante en este documento.

2.2 Cuentas

La configuración del producto **OBCOM ADAPI** utiliza tres cuentas (login). Estas cuentas tienen permisos (grants) muy específicos, y sólo pueden acceder y operar en la base de datos **ECUBAS**. No tienen acceso a ninguna otra base de datos del servidor SQL.

- 1) **EcuAdmin**: esta cuenta (login) es el propietario (dbo - database owner) de la base de datos **ECUBAS**. Como propietario, esta cuenta puede crear y modificar los objetos de la base de datos. No puede hacer absolutamente nada en otras bases de datos.
- 2) **AdapiSecUsers**: esta cuenta (login) la utilizan los servidores ACASER, ACCSER y NETSER para acceder directamente a las tablas de seguridad de usuarios y perfiles (descritas más adelante). Esta cuenta sólo tiene permisos (grants) para hacer SELECT, INSERT, UPDATE y DELETE en estas tablas. No puede hacer nada más.
- 3) **AdapiWebUsers**: esta cuenta (login) la utilizan las páginas ASP y el Filtro ISAPI para invocar a procedimientos almacenados. Esta cuenta sólo tiene permisos (grants) para hacer EXECUTE de los procedimientos almacenados del producto **OBCOM ADAPI** (descritos más adelante). No puede hacer nada más, ni siquiera leer tablas.

Una vez creada la base de datos **ECUBAS**, es necesario definir la cuenta propietaria. Para esto, el administrador del servidor SQL (sa – system administrator) debe ejecutar los siguientes comandos en una consola SQL:

```
use ECUBAS
go
exec sp_changedbowner EcuAdmin
go
```

Adicionalmente, el administrador del servidor SQL (sa – system administrator) debe agregar las cuentas **AdapiSecUsers** y **AdapiWebUsers** como usuarios (users) de la base de datos **ECUBAS**. Para esto, debe ejecutar los siguientes comandos en una consola SQL:

```
use ECUBAS
go
exec sp_adduser 'AdapiSecUsers'
go
exec sp_adduser 'AdapiWebUsers'
go
```

2.3 Tablas

El producto **OBCOM ADAPI** utiliza dos grupos de tablas: (1) tablas de seguridad, y (2) tablas de preferencias. Las próximas secciones explican cómo instalar y configurar estas tablas.

2.3.1 Tablas de seguridad

Las tablas de seguridad son leídas o modificadas directamente por los servidores **ACASER**, **ACCSER** y **NETSER** utilizando la cuenta **AdapiSecUsers**. Las tablas de seguridad son las siguientes:

- 1) **EcuACCCON**: estado de las conexiones activas.
- 2) **EcuACCCTX**: contexto de los servidores EcuLink.
- 3) **EcuACCFAM**: definición de las familias (grupos) de terminales del sistema.
- 4) **EcuACCLOG**: registro de las modificaciones realizadas a las tablas de seguridad.
- 5) **EcuACCNET**: definición de los terminales del sistema.
- 6) **EcuACCPER**: definición de los perfiles de usuarios.
- 7) **EcuACCPRV**: definición de los nombres de los privilegios de usuarios.
- 8) **EcuACCU2P**: relación muchos-a-muchos entre usuarios y perfiles.
- 9) **EcuACCUSU**: definición de los usuarios del sistema.
- 10) **EcuACCATR**: definición de los atributos del sistema.
- 11) **EcuACCATU**: relación muchos-a-muchos entre usuarios y atributos.

Para crear las tablas de seguridad en la base de datos **ECUBAS** es necesario ejecutar el archivo de comandos (script) llamado **DbTables.sql**. Este archivo de comandos se encuentra en el directorio **<DiskT>\Database\AccessControl\SQL**, donde **<DiskT>** es el directorio donde están instaladas las herramientas estándar del producto **OBCOM ADAPI**.

Una vez creadas las tablas de seguridad, es necesario asignar los permisos (grants) ejecutando el archivo de comandos (script) llamado **DbGrants.sql**. Este archivo de comandos se encuentra en el directorio **<DiskT>\Database\AccessControl\SQL**.

2.3.2 Tablas de preferencias

El otro grupo de tablas utilizado por el producto **OBCOM ADAPI** son las tablas de preferencias. Estas tablas son utilizadas por **Desktop ADAPI** para almacenar las preferencias y estado de las sesiones de los usuarios. Las tablas de preferencias son:

- 1) **EcuDeskProgNew**: preferencias a nivel de controles.
- 2) **EcuDeskPageNew**: preferencias a nivel de páginas.
- 3) **EcuDeskUserNew**: preferencias a nivel de usuario.

Para crear las tablas de preferencias en la base de datos **ECUBAS** es necesario ejecutar el archivo de comandos (script) llamado **DbTablesNew.sql**. Este archivo de comandos se encuentra en el directorio **<DiskT>\Database\EcuDesktop\SQL**.

Las tablas de preferencias son siempre leídas y/o modificadas vía procedimientos almacenados. Por consiguiente, ninguna cuenta debe tener permisos (grants) para acceder directamente a ellas.

2.4 Procedimientos almacenados

Como se explicó en la sección anterior, las tablas de preferencias son siempre leídas o modificadas vía procedimientos almacenados, los cuales son invocados desde el Desktop ADAPI. Estos procedimientos almacenados son:

1. **EcuDeskNemoNew**: retorna la lista de símbolos.
2. **EcuPageDelNew**: borra una página del usuario.
3. **EcuPageDirNew**: retorna la lista de páginas del usuario.
4. **EcuPageGetNew**: retorna contenido de una página del usuario.
5. **EcuPageGidNew**: retorna la identificación de una página del usuario.
6. **EcuPagePutNew**: actualiza una página del usuario.
7. **EcuPageRenNew**: renombra una página del usuario.
8. **EcuProgDelNew**: borra un control del usuario.
9. **EcuProgGetNew**: retorna contenido de un control del usuario.
10. **EcuProgPutNew**: actualiza un control del usuario.
11. **EcuUserGetNew**: retorna las preferencias del usuario.
12. **EcuUserPutNew**: actualiza las preferencias del usuario.
13. **EcuWordGet**: utilizado por el filtro ISAPI de seguridad.

Para crear los procedimientos en la base de datos **ECUBAS** es necesario ejecutar los archivos de comandos (script) que se encuentran en el directorio **<DiskT>\Database\EcuDesktop\SQL**, donde **<DiskT>** es el directorio donde están instaladas las herramientas estándar del producto **OBCOM ADAPI**. Existe un archivo de comando (script) por cada procedimiento.

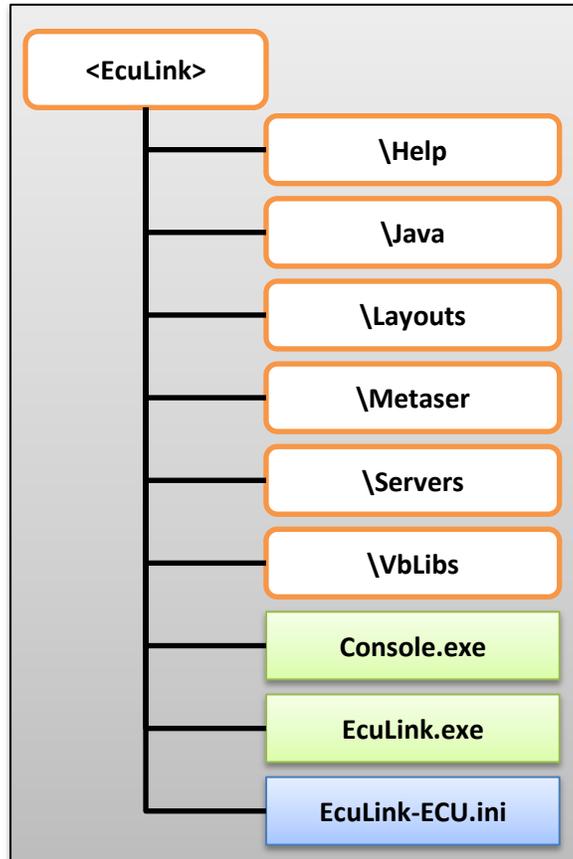
Una vez creado los procedimientos, es necesario asignar los permisos (grants) ejecutando el archivo de comandos (script) llamado **DbProcGrants.sql**. Este archivo de comandos se encuentra en el directorio **<DiskT>\Database\EcuDesktop\SQL**.

3 Servidor EcuLink

El producto **OBCOM ADAPI** requiere de uno o más servidores **EcuLink**. Estos servidores permiten al **Desktop ADAPI** comunicarse con las aplicaciones de seguridad **ACASER**, **ACCSE** y **NETSER**, las cuales autentican y autorizan a los usuarios del sistema. Además, los servidores **EcuLink** permiten la comunicación con otras aplicaciones como el **IMSER**, que implementa el servicio de mensajería instantánea entre los usuarios del sistema.

3.1 Estructura de directorios

Para poder operar, el servidor **EcuLink** requiere de varias componentes, todas las cuales están agrupadas en una estructura de directorios. La raíz de esta estructura de directorios es llamada **<EcuLink>**, y normalmente corresponde al directorio **C:\EcuLink**. La siguiente figura muestra los principales archivos y subdirectorios que componen esta estructura:



1. Directorio **Help**: contiene la documentación de los comandos del servidor **EcuLink**.
2. Directorio **Java**: contiene la JDK (Java Development Kit) para servidores Java.
3. Directorio **Layouts**: contiene las plantillas (records) de los mensajes **EcuLink**.
4. Directorio **Metaser**: contiene los servidores y bibliotecas escritas en Java.
5. Directorio **Servers**: contiene los servidores escritos en Visual Basic 6.
6. Directorio **VbLibs**: contiene las bibliotecas para servidores Visual Basic 6.
7. Programa **Console.exe**: consola de administración del servidor **EcuLink**.
8. Programa **EcuLink.exe**: ejecutable del servidor **EcuLink**.
9. Archivo **EcuLink-ECU.ini**: archivo de configuración del servidor **EcuLink**.

3.2 Servicio Windows

El servidor **EcuLink** normalmente opera como un **Servicio Windows**. Es decir, opera como un proceso no interactivo (sin consola), que presta servicios a través de enlaces de red (sockets), y deja sus mensajes en el **Visor de eventos** (Event Viewer) de Windows.

Para que el servidor **EcuLink** opere como un **Servicio Windows**, es necesario inscribirlo ejecutando el siguiente comando en una consola con privilegios de Administrador:

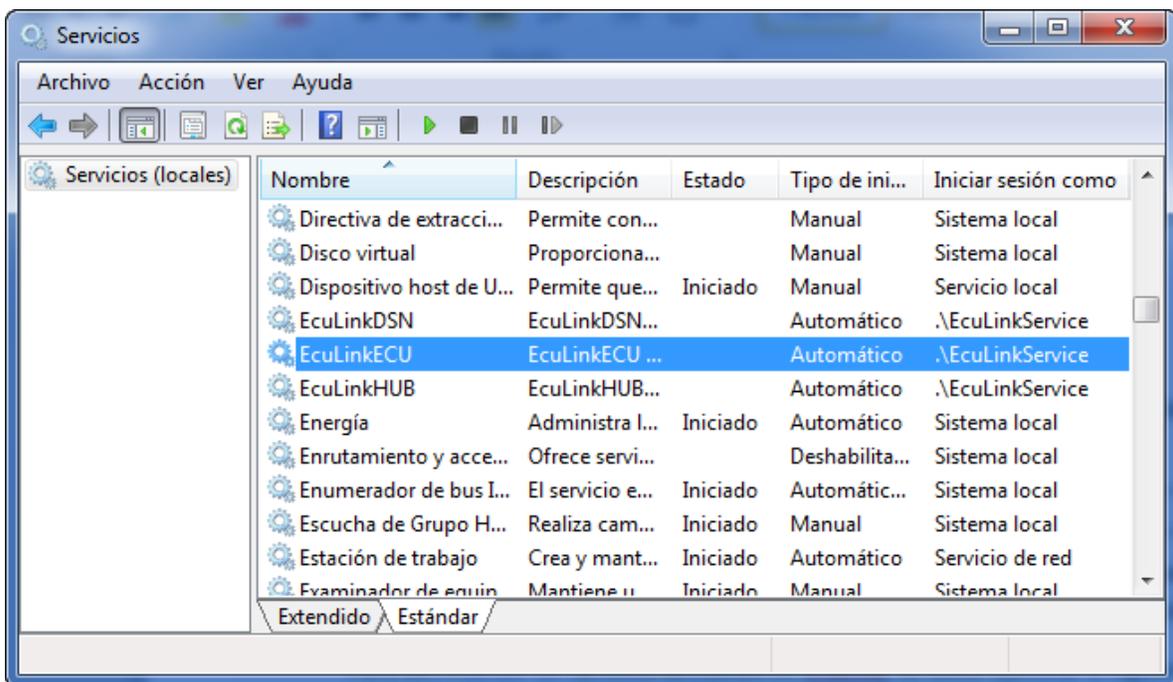
```
C:\EcuLink>EcuLink.exe Service ECU
```

El parámetro **ECU** corresponde al nombre que tendrá la instancia del **Servicio Windows** de **EcuLink**. Este nombre debe tener de tres a cuatro letras, y debe ser único en un servidor Windows.

Para eliminar la inscripción de un **Servicio Windows** de **EcuLink**, se debe ejecutar el siguiente comando en una consola con privilegios de Administrador:

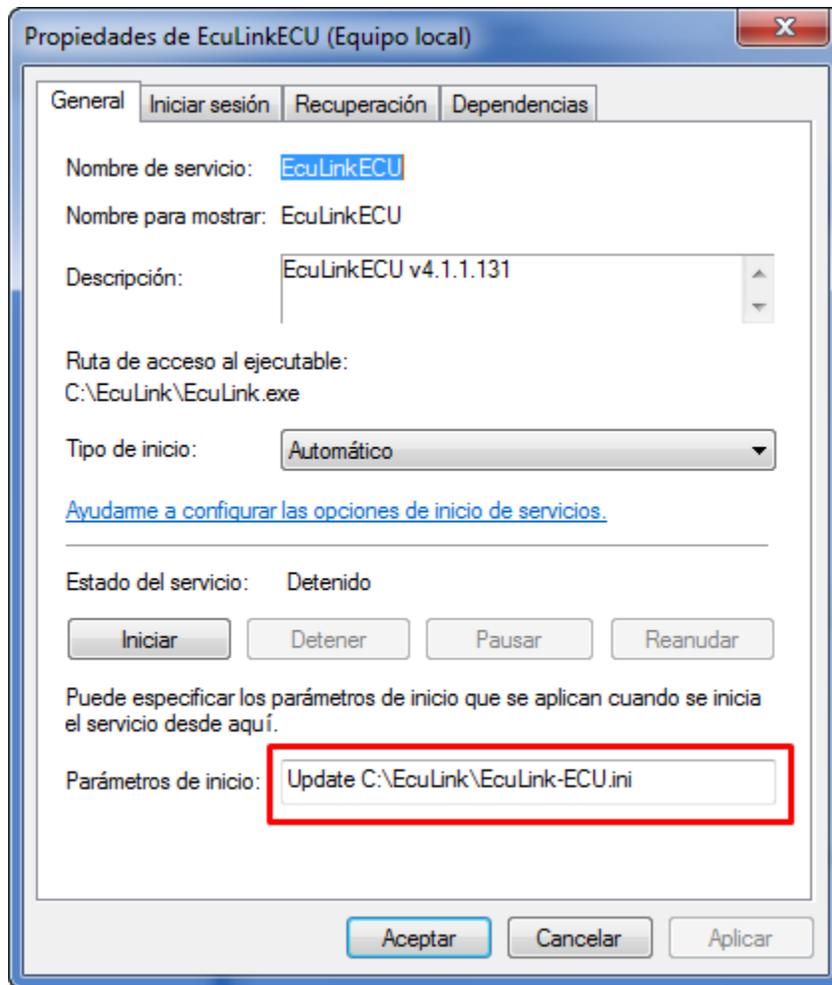
```
C:\EcuLink>EcuLink.exe UnService ECU
```

Una vez que el servidor **EcuLink** ha sido inscrito como **Servicio Windows**, es necesario definir la ubicación y el nombre del archivo de configuración (archivo **EcuLink-ECU.ini** antes mencionado). Para esto, se utiliza la consola de administración de **Servicios de Windows**, y se selecciona el servicio **EcuLink** como se muestra en la siguiente figura:



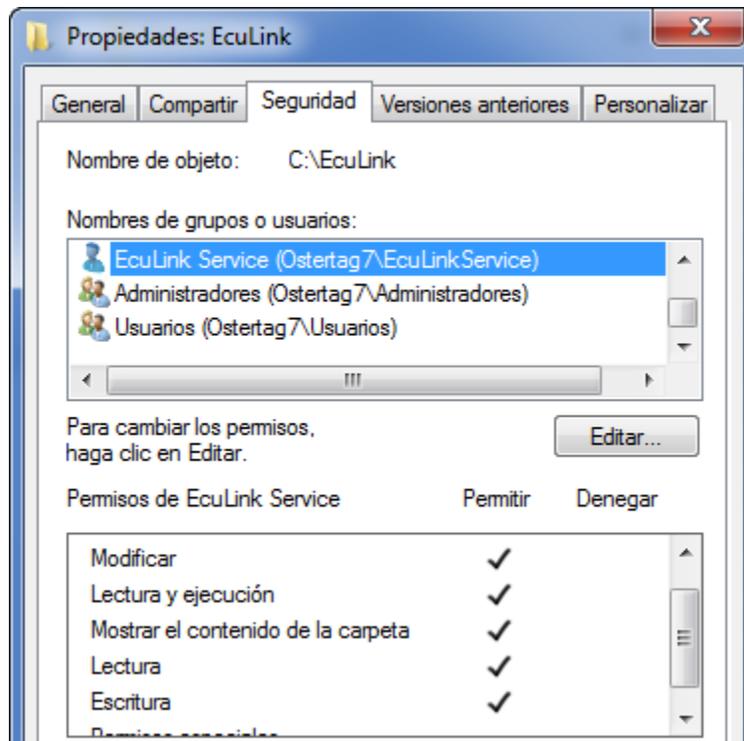
Luego se abre las propiedades del servicio seleccionado haciendo doble-clic, con lo cual aparece una ventana como la que se muestra en la siguiente figura. En el campo "**Parámetros de inicio:**" se escribe "**Update C:\EcuLink\EcuLink-ECU.ini**", ajustando el directorio y/o el nombre del archivo de configuración, según corresponda. Por último, se hace clic en el botón **Iniciar** para comenzar la ejecución del servicio.

Al iniciar el servicio con un comando **Update**, el servidor **EcuLink** almacena la información suministrada, y la utiliza en el futuro cada vez que es iniciado. Para cambiar la ubicación y/o el nombre del archivo de configuración del **EcuLink**, es necesario volver a iniciar el servicio con un comando **Update** actualizado.

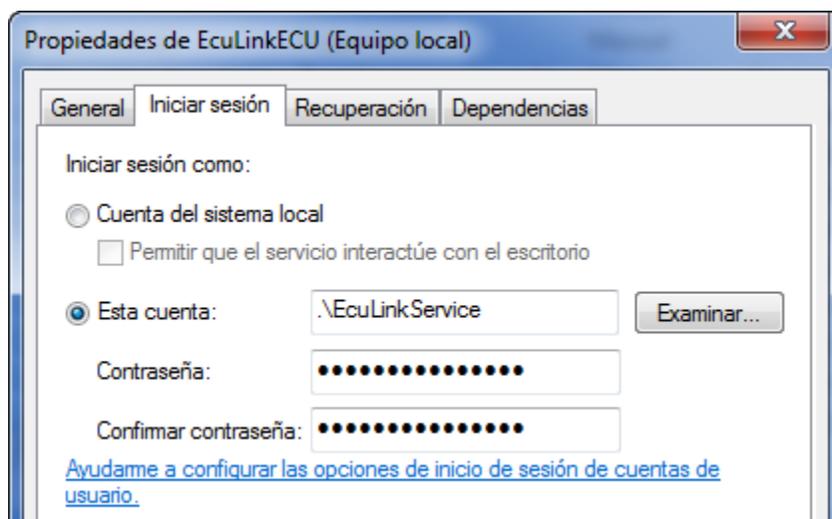


En forma predeterminada, los servicios de Windows son ejecutados con la cuenta **Servicio local** (SYSTEM). Esta cuenta es similar a la cuenta Administrador, y posee muchos privilegios. Para aumentar el nivel de seguridad, se recomienda (pero no es obligatorio) asignarle al **Servicio Windows** de **EcuLink** una cuenta sin privilegios.

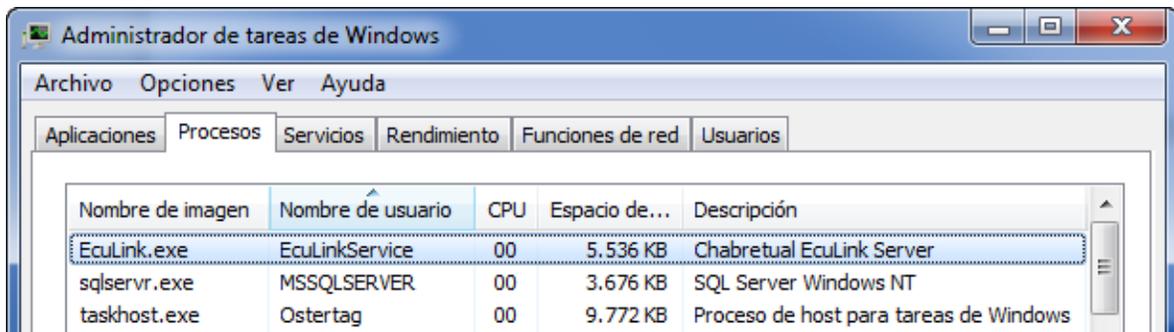
Para hacer esto, el Administrador debe crear una cuenta a nivel de dominio, la cual normalmente se llama **EcuLinkService**. Luego, debe dar a la cuenta **EcuLinkService** privilegios de lectura y modificación sobre toda la estructura de directorio **<EcuLink>**, como se muestra en la siguiente figura.



Luego, el Administrador debe modificar las propiedades del **Servicio Windows** de **EcuLink** para que utilice esta cuenta, como se muestra en la figura siguiente. Finalmente, hay que detener y luego iniciar el **Servicio Windows** de **EcuLink** para que tome esta nueva configuración.



Si este cambio de configuración es exitoso, se debe poder observar el proceso del **Servicio Windows** de **EcuLink** ejecutando con la cuenta **EcuLinkService**, como se observa en la siguiente ventana del **Administrador de tareas de Windows**:



3.3 Archivo de configuración

El archivo de configuración del servidor **EcuLink** permite definir un conjunto muy grande de parámetros. Estos parámetros permiten a **EcuLink** operar en distintos ambientes productivos. En esta sección sólo se explicarán aquellos parámetros relevantes al producto **OBCOM ADAPI**.

La convención de nombre del archivo de configuración es simple: comienza con "**EcuLink-**", luego viene el nombre del **Servicio Windows** de **EcuLink**, y se termina con **".ini"**. Como se explicó antes, el nombre del **Servicio Windows** de **EcuLink** tiene entre tres a cuatro letras, y debe ser único en un servidor Windows. Usando esta convención de nombres, se puede usar una sola estructura de directorios **<EcuLink>** para configurar varios **Servicios Windows** de **EcuLink**, cada uno de ellos usando un archivo de configuración distinto.

Por ejemplo, supongamos que se requiere configurar dos **Servicio Windows** de **EcuLink**, uno con el nombre **IMP**, y el otro con el nombre **HUB**. En este caso, existirán dos archivos de configuración, uno llamado **EcuLink-IMP.ini**, y el otro **EcuLink-HUB.ini**, ambos ubicados en la raíz de **<EcuLink>**. Para crear la versión inicial de estos dos archivos, se recomienda copiar el archivo **EcuLink-ECU.ini** provisto con el producto **OBCOM ADAPI**.

Se debe ser modificar las líneas que definen el nombre del proceso **EcuLink**:

```
set ProcessName      IMP
set BootHostName     IMP
set DefaultHostName  IMP
set LocalHostName    IMP
set SecurityHostName IMP
```

Se debe modificar los puertos por los cuales opera **EcuLink**. Estos puertos deben ser únicos en un mismo servidor Windows:

```
alter daemon "daecon", address "*,10101"
alter daemon "daesrv", address "*,10102"
alter daemon "daelink", address "*,10103"
alter daemon "daecli", address "*,10104"
alter daemon "daedyn", address "*,10105"
```

Se debe ajustar la siguiente línea para que coincida con el puerto **daesrv** definido antes:

```
setenv ECUSERVER "127.0.0.1,10102"
```

Como se explicó antes, los servidores de seguridad **ACASER**, **ACCSER** y **NETSER** leen y modifican las tablas de seguridad en la base de datos **ECUBAS**. Es necesario ajustar las líneas que definen la conexión a la base de datos para estos tres servidores. En particular, hay que ajustar (1) el nombre del servidor SQL, (2) el nombre de la base de datos, (3) el nombre del usuario de la base de datos, y (4) la contraseña de este usuario:

```
server ACASER, parameters "...-CProvider...;Data+Source...;User+ID...;Password..."
server ACCSER, parameters "...-CProvider...;Data+Source...;User+ID...;Password..."
server NETSER, parameters "...-CProvider...;Data+Source...;User+ID...;Password..."
```

4 Servidor IIS

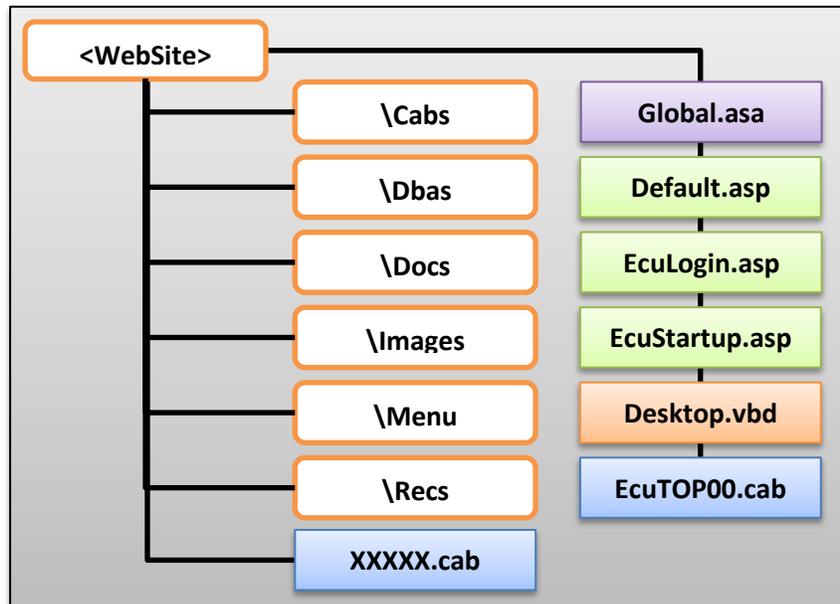
El producto **OBCOM ADAPI** requiere de uno o más servidores **IIS** (Internet Information Services). Estos servidores HTTP/HTTPS proveen al **Desktop ADAPI** varios servicios: (1) descargar nuevas versiones de controles, (2) invocar a procedimientos almacenados vía páginas ASP, (3) obtener plantillas (records) de mensajes **EcuLink**, (4) obtener parámetros de configuración, (5) obtener imágenes, (6) obtener los menús de los usuarios, etc.

4.1 Estructura del sitio web

El sitio web del producto **OBCOM ADAPI** está compuesto por los siguientes directorios y archivos:

- 1) Directorio **Cabs**: contiene los controles empaquetados en archivos CAB.
- 2) Directorio **Dbas**: contiene las páginas ASP para invocar procedimientos.
- 3) Directorio **Docs**: contiene la documentación estándar del producto.
- 4) Directorio **Images**: contiene las imágenes e íconos del producto.
- 5) Directorio **Menu**: contiene los menús de los usuarios.
- 6) Directorio **Recs**: contiene las plantillas de mensajes EcuLink.
- 7) Archivo **Global.asa**: define los parámetros de la aplicación IIS.
- 8) Archivo **Default.asp**: página de inicio predeterminada (default).
- 9) Archivo **EcuStartup.asp**: define los parámetros iniciales del **Desktop ADAPI**.
- 10) Archivo **EcuLogin.asp**: define los parámetros de una sesión de trabajo.
- 11) Archivo **Desktop.vbd**: la página ActiveX de la aplicación **Desktop ADAPI**.
- 12) Archivo **EcuTOP00.cab**: contiene la aplicación y las bibliotecas del **Desktop ADAPI**.
- 13) Archivos **XXXXX.cab**: contienen las bibliotecas de terceros utilizadas por el producto.

La estructura de este sitio web se muestra en la siguiente figura. De todos estos directorios y archivos, el Administrador del sistema debe ajustar algunos de los parámetros definidos en los archivos: **Global.asa**, **EcuLogin.asp** y **EcuStartup.asp**. El resto de los archivos se utilizan tal cual están provistos con el producto.



4.1.1 Archivo Global.asa

El archivo **Global.asa** puede contener declaraciones de objetos, variables y métodos que pueden ser usados por todas las páginas ASP de una aplicación ASP. El producto **OBCOM ADAPI** es una aplicación ASP que utiliza variables definidas en el archivo **Global.asa**. En primera instancia, se asignan las variables de la licencia del producto:

1. **Application("EcuTitle")**: título de la ventana principal del Desktop ADAPI.
2. **Application("EcuMessage")**: mensaje explicativo para los usuarios.
3. **Application("EcuCompany")**: nombre de la organización que licenció el producto.
4. **Application("EcuLicense")**: texto encriptado con la licencia de uso.

El valor de estas variables de licencia es provisto por la empresa **OBCOM INGENIERÍA S.A.** a todos los clientes que han adquirido una licencia de uso del producto **OBCOM ADAPI**.

En el archivo **Global.asa** también se asignan las variables que definen las conexiones a la base de datos. La siguiente línea define la variable estándar **Application("EcuConnString")** que utilizan las páginas ASP que invocan a los procedimientos almacenados de manejo de preferencias.

```
Application("EcuConnString") = "Provider=SQLOLEDB; Data Source=SQLSER;
Initial Catalog=ECUBAS; User ID=AdapiWebUsers; Password=<contraseña>"
```

El Administrador del sistema debe ajustar la línea anterior, cambiando el nombre del servidor (SQLSER), el nombre de la base de datos (ECUBAS), el nombre del usuario de la base de datos (AdapiWebUsers), o la contraseña del usuario, según corresponda.

4.1.2 Archivo EcuStartup.asp

Cuando el **Desktop ADAPI** comienza a ejecutar, invoca una sola vez al archivo **EcuStartup.asp**. Este archivo define varios parámetros, la mayoría de los cuales no es necesario ajustar. Los parámetros que el Administrador del sistema sí debe ajustar son los siguientes:

1. **IPAddress**: nombre o dirección IP del servidor donde está ejecutando **EcuLink**.
2. **PortNumber**: número del puerto **daegli** por el cual está escuchando el servidor **EcuLink**.
3. **HostName**: nombre asignado al servidor **EcuLink** en el archivo de configuración.

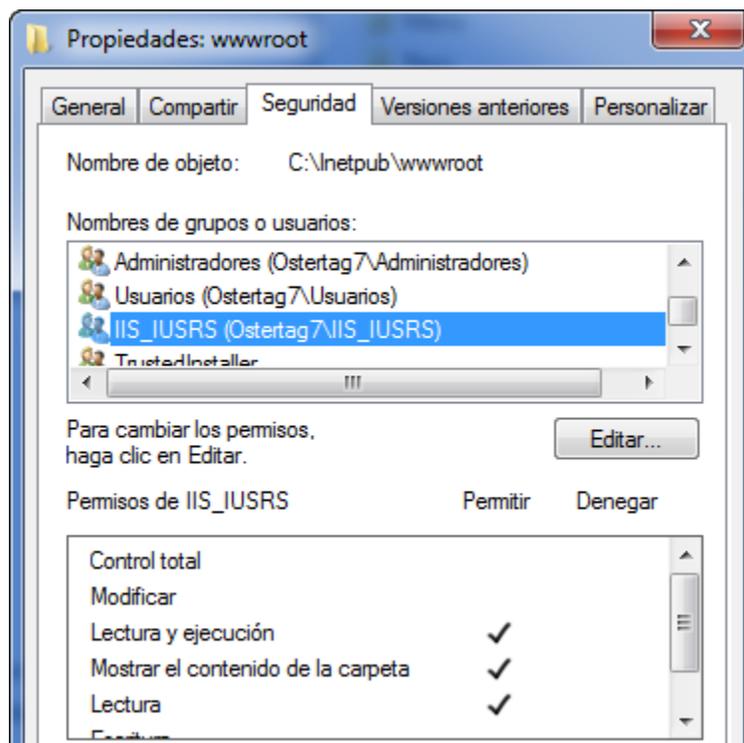
4.1.3 Archivo EcuLogin.asp

Cada vez que un usuario ingresa al sistema haciendo login, el **Desktop ADAPI** invoca al archivo **EcuLogin.asp**. Este archivo define varios parámetros específicos a la sesión de trabajo. En general, el Administrador del sistema no necesita ajustar estos parámetros.

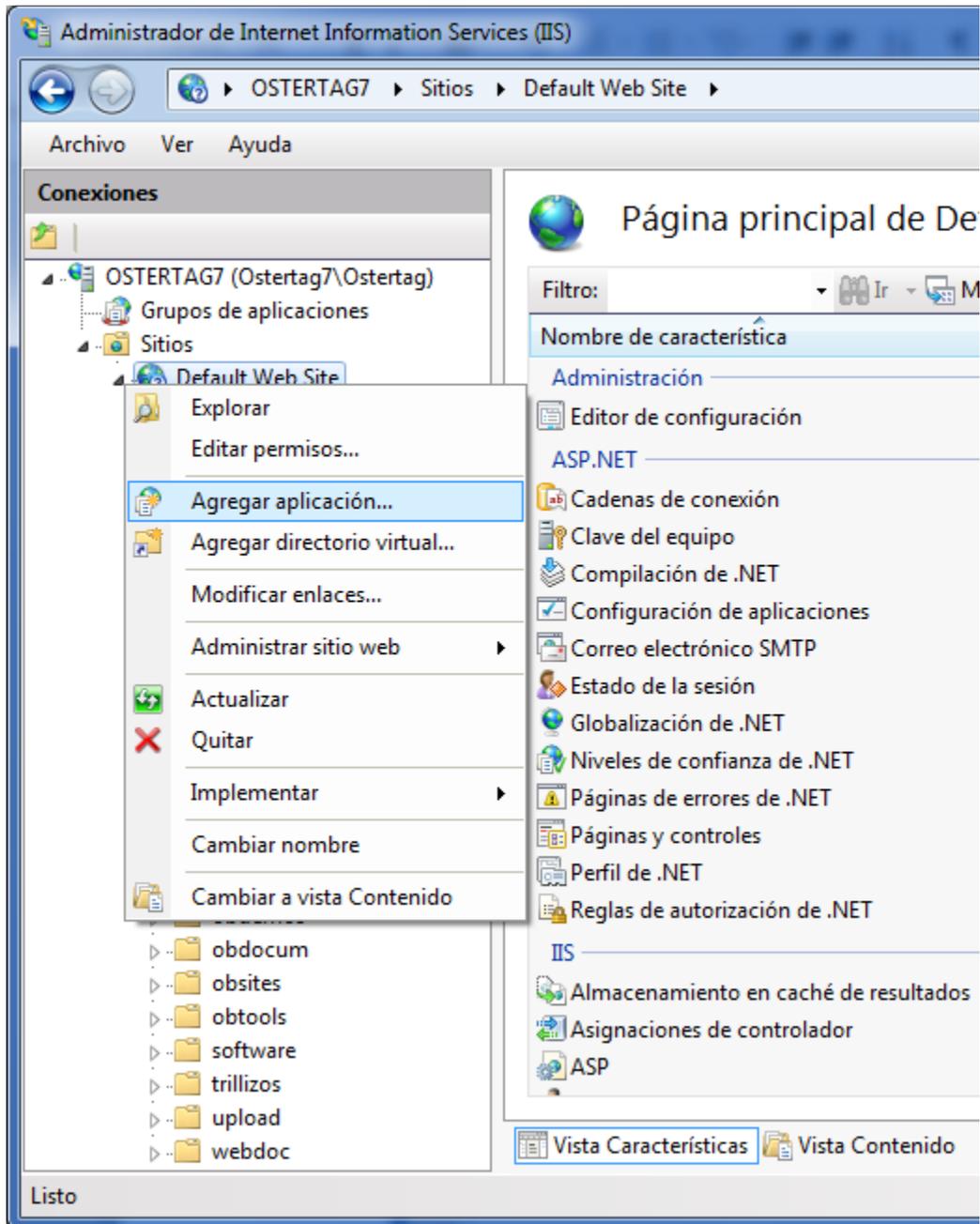
4.2 Aplicación internet

Para que pueda operar el producto **OBCOM ADAPI**, es necesario configurar la estructura de su sitio web **<WebSite>** como una aplicación internet IIS. El primer paso consiste en copiar esta estructura completa al directorio de sitios web de la organización. El directorio predeterminado por IIS es **C:\inetpub\wwwroot**, pero se puede utilizar cualquier otra ubicación.

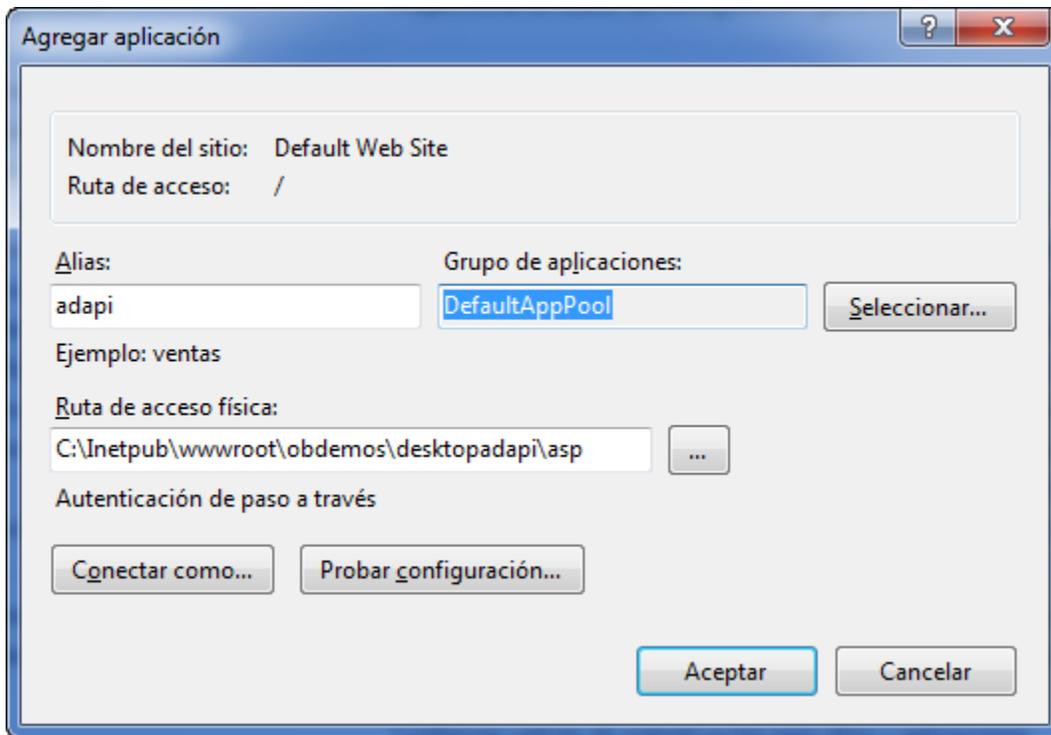
Una vez copiada la estructura completa, es necesario ajustar los permisos de todos sus archivos y subdirectorios de modo que la cuenta del usuario anónimo de IIS puede leerlos, pero no pueda modificarlos. La cuenta del usuario anónimo normalmente se llama **IIS_IUSRS**.



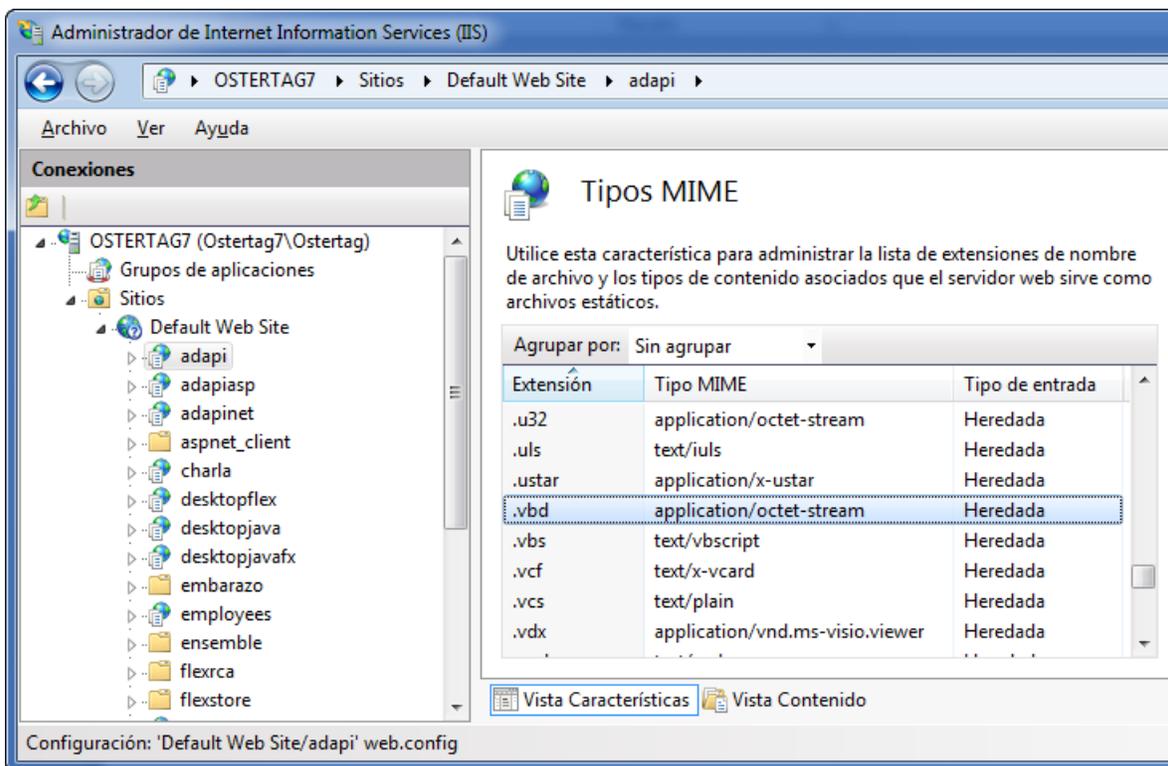
Una vez realizados los pasos anteriores, se utiliza la consola de administración de IIS para configurar la aplicación internet del producto **OBCOM ADAPI**. Para esto, se selecciona la opción “**Agregar aplicación...**” como se muestra en la siguiente figura:



Al seleccionar esta opción, aparece una ventana como la que se muestra en la próxima figura. En esta ventana se suministra el nombre del “**Alias:**”, y la “**Ruta de acceso física:**”. También es posible especificar un “**Grupo de aplicaciones:**” distinto del predeterminado. Finalmente, se hace clic en el botón **Aceptar**, y IIS creará la aplicación internet.



Luego de creada la aplicación internet, es necesario modificar su configuración de modo que IIS procese correctamente el archivo **Desktop.vbd** del producto **OBCOM ADAPI**. Para esto, se agrega la extensión **.vbd** con tipo **application/octet-stream** a los **Tipos MIME** de la aplicación IIS.



Una vez creada la aplicación internet, y configurado el tipo MIME, el producto **OBCOM ADAPI** está listo para ser utilizado. Para esto, basta utilizar el navegador **Microsoft Internet Explorer** para navegar al servidor IIS y la aplicación creada en esta sección.

4.2.1 Filtro de seguridad

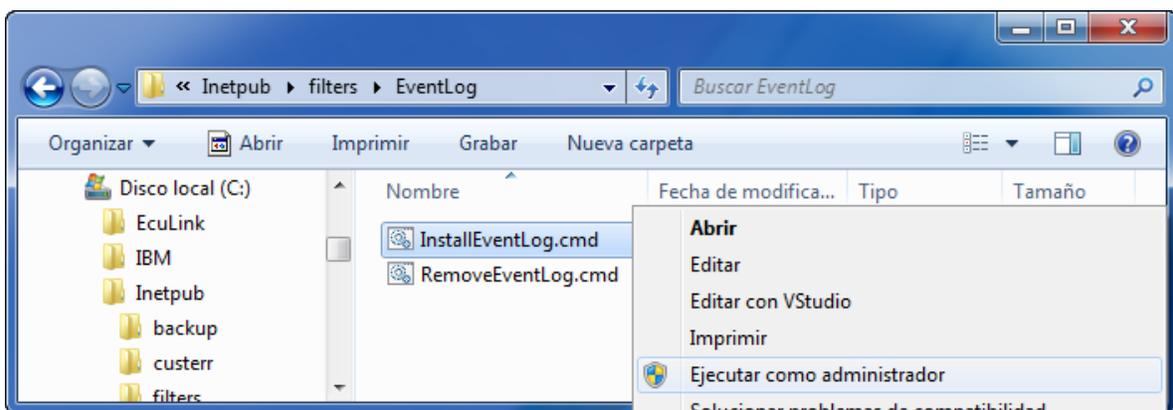
El producto **OBCOM ADAPI** puede (opcionalmente) hacer uso de un **Filtro ISAPI** de seguridad. Este filtro revisa las invocaciones a todas páginas ASP de una aplicación internet, y sólo permitir que se ejecute una página ASP si las credenciales suministradas en la invocación son autenticadas. Si la invocación no incluye credenciales, o si estas credenciales son inválidas, el **Filtro ISAPI** rechaza la invocación, y retorna el código HTTP 401 (Unauthorized). De ser necesario, el **Filtro ISAPI** genera mensajes de advertencia o error que el Administrador puede leer con la herramienta **Visor de eventos** (Event Viewer) de Windows.

Existe una versión del filtro para sistemas de **32 bits**, y otra versión para sistemas de **64 bits**. En ambos casos, los archivos asociados al filtro son los siguientes:

1. **adapi_filter.dll**: corresponde al **Filtro ISAPI** de seguridad.
2. **adapi_filter.ini**: parámetros de configuración del **Filtro ISAPI**.
3. **EcuWordGet.sql**: procedimiento almacenado usado por el **Filtro ISAPI**.
4. **InstallEventLog.cmd**: instalan los mensajes del **Filtro ISAPI** en el **Visor de eventos**.
5. **RemoveEventLog.cmd**: remueve los mensajes del **Filtro ISAPI** del **Visor de eventos**.

El primer paso para configurar el **Filtro ISAPI** consiste en copiar los archivos anteriores a un directorio adecuado. Se recomienda usar el directorio **C:\inetpub\filters\adapi**, el cual debe tener permiso de lectura a la cuenta del usuario anónimo de IIS (**IIS_IUSRS**).

El segundo paso consiste en ejecutar el archivo **InstallEventLog.cmd** para instalar los mensajes del **Filtro ISAPI** en el **Visor de eventos** (Event Viewer). Este archivo de comandos debe ser ejecutado con privilegios de Administrador, como se muestra en la siguiente figura:



El tercer paso consiste en cargar a la base de datos **ECUBAS** el procedimiento almacenado definido en el archivo **EcuWordGet.sql**. Para esto, se recomienda utilizar la consola de administración del servidor SQL Server.

El cuarto paso consiste en ajustar los parámetros de configuración del **Filtro ISAPI**. Para esto hay que editar el archivo **adapi_filter.ini**, el cual tiene un contenido similar al que se muestra en la siguiente figura:

```
[Settings]
InitialPasswordCapacity = 4096
PasswordTableLoadFactor = 0.75
InitialNonceCapacity = 4096
NonceTableLoadFactor = 0.75
LogNonceTableSwap = 0
NonceSwapTimeSecs = 300
MaxDeltaTimeSecs = 300
ConnectionString = Driver={SQL Server};Server=...;Database=...;Uid=...;Pwd=...;
VirtualDir1 = /adapi/
VirtualDir2 = ...
VirtualDir3 = ...
```

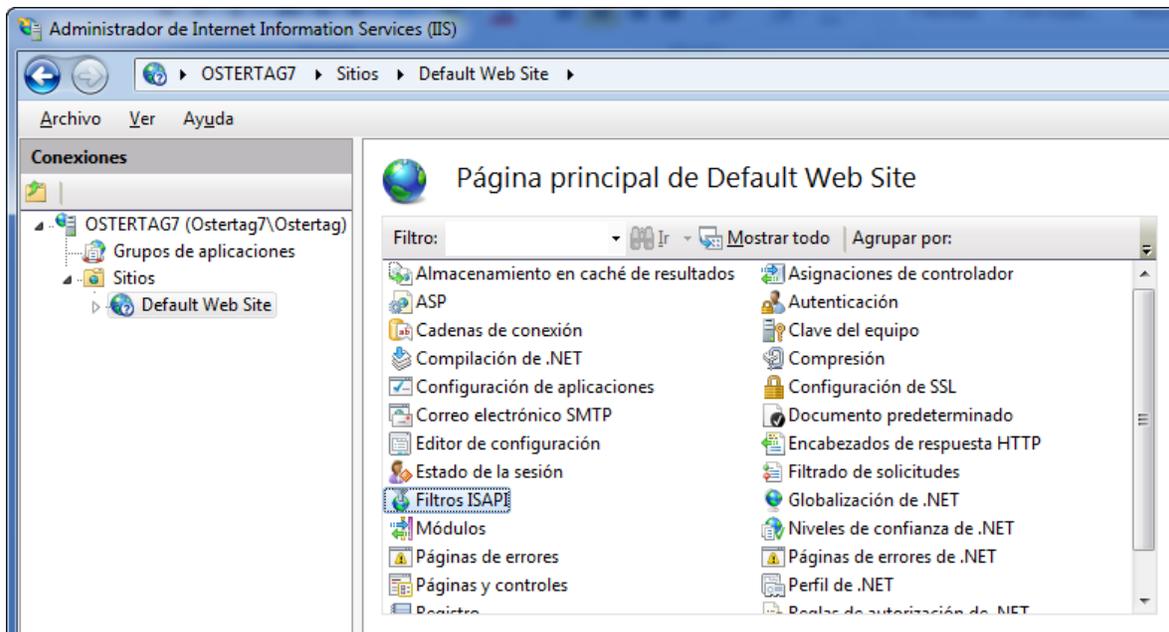
De todos estos parámetros, normalmente se modifican sólo **ConnectionString** y **VirtualDirN**. En **ConnectionString** se debe indicar (1) el nombre o número IP del servidor SQL (**Server**), (2) el nombre de la base de datos (**Database**), (3) el nombre del usuario de la base de datos (**Uid**), y (4) la contraseña del usuario (**Pwd**).

Importante: la base de datos indicada en **ConnectionString** debe contener el procedimiento almacenado **EcuWordGet.sql**. Además, el usuario especificado debe tener permiso (grant) para ejecutar este procedimiento. Se recomienda usar la base de datos **ECUBAS**, dado en esta base de datos se encuentran las tablas de seguridad que lee este procedimiento almacenado.

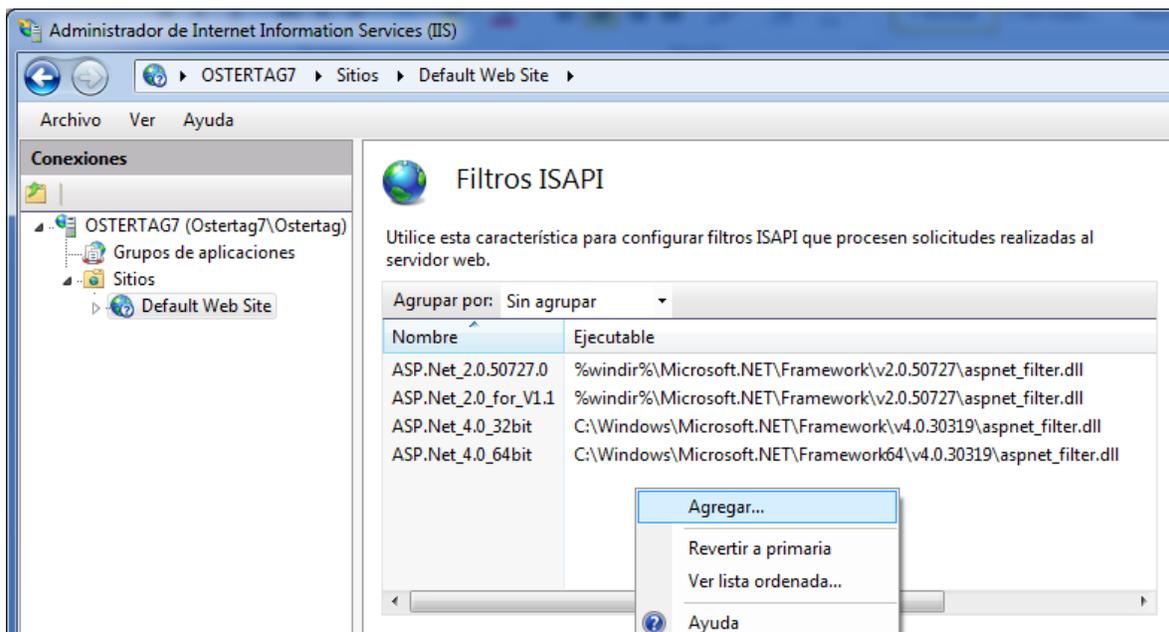
Por su parte, los parámetros **VirtualDirN** se usan para definir los directorios virtuales que el **Filtro ISAPI** debe proteger. Si no se especifica ningún parámetro **VirtualDirN**, el **Filtro ISAPI** proteger todos los directorios virtuales del sitio web IIS donde fue configurado.

Finalmente, para que el **Filtro ISAPI** comience a operar, es necesario indicarle a IIS que utilice este filtro. Para esto, se utiliza la consola de administración de IIS como se muestra en la siguiente figura. Primero se selecciona el sitio web que se desea configurar (en este caso, **Default Web Site**), y luego se hace clic en la opción **Filtros ISAPI**.

Observación: sólo se pueden configurar filtros ISAPI a nivel de sitios web. No es posible configurar filtros a nivel de directorios virtuales. Por este motivo, cuando uno configura un filtro ISAPI, se puede afectar a todos los directorios virtuales (aplicaciones) del sitio web.

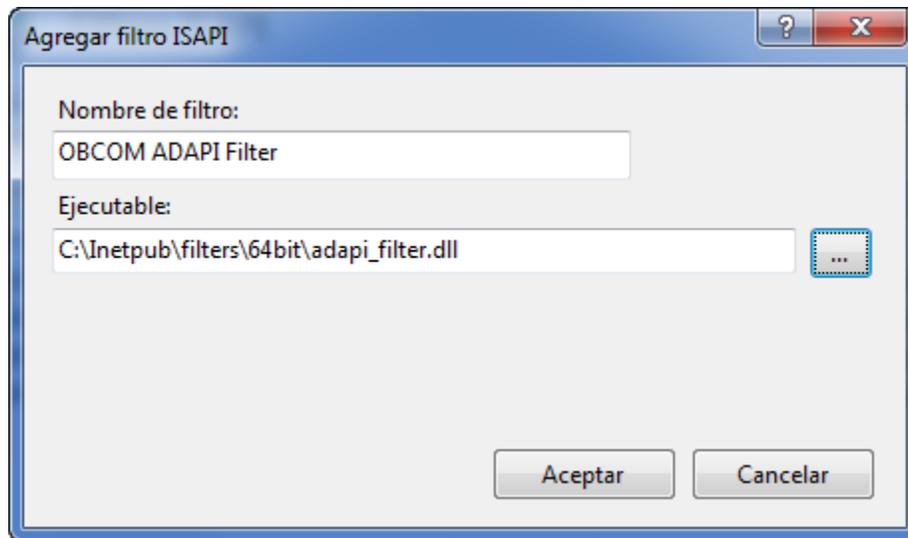


Después de hacer clic en la opción **Filtros ISAPI**, aparece una ventana similar a la que se muestra en la siguiente figura.

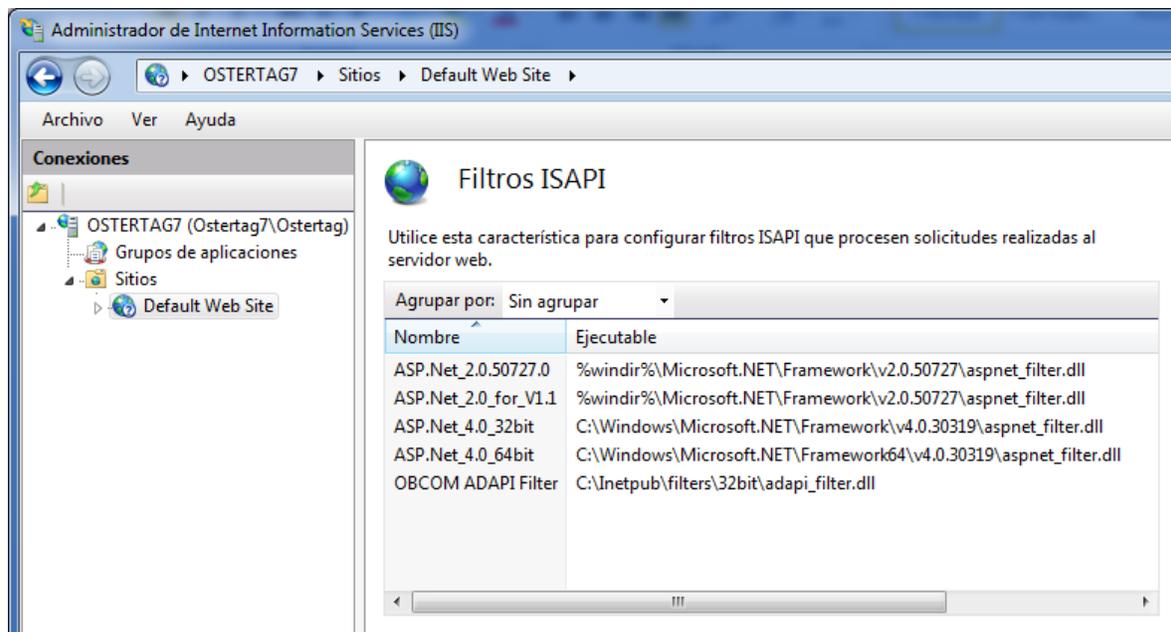


Luego se hace clic con el botón derecho, y se selección la opción **“Agregar...”**. Aparece una ventana como se muestra en la siguiente figura. Aquí es necesario escribir el **“Nombre de filtro:”** y seleccionar el **“Ejecutable:”** del filtro.

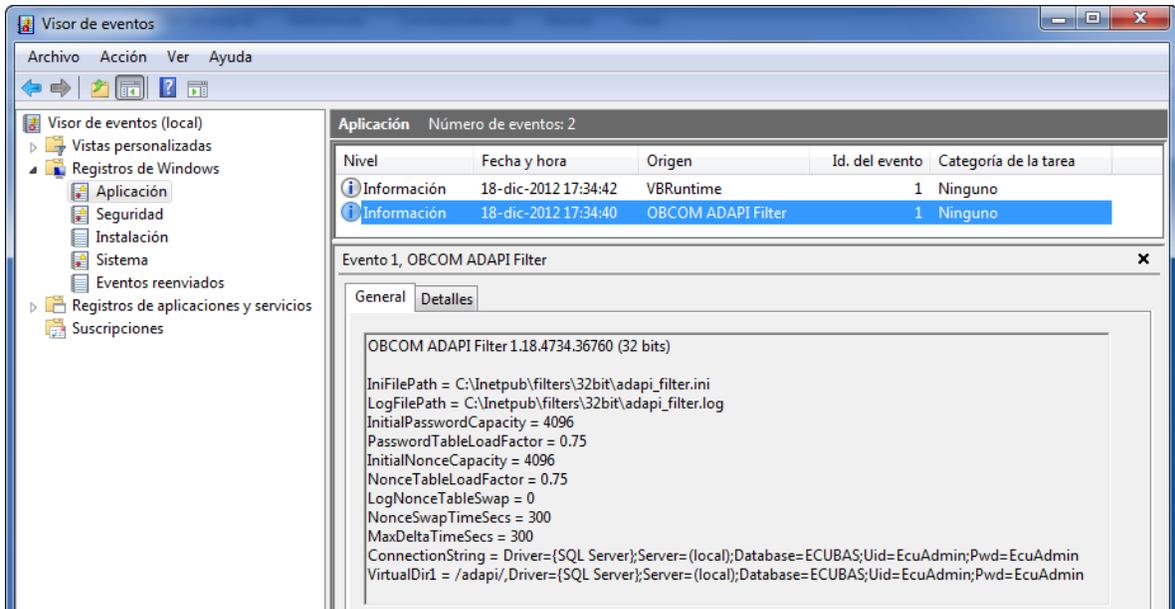
Importante: seleccione la versión de **32 bits** o **64 bits** que corresponda al sistema operativo.



Al hacer clic en el botón **Aceptar**, debe aparecer la siguiente ventana mostrando el nombre y la ubicación del **Filtro ISAPI** en la lista de filtros del sitio web.

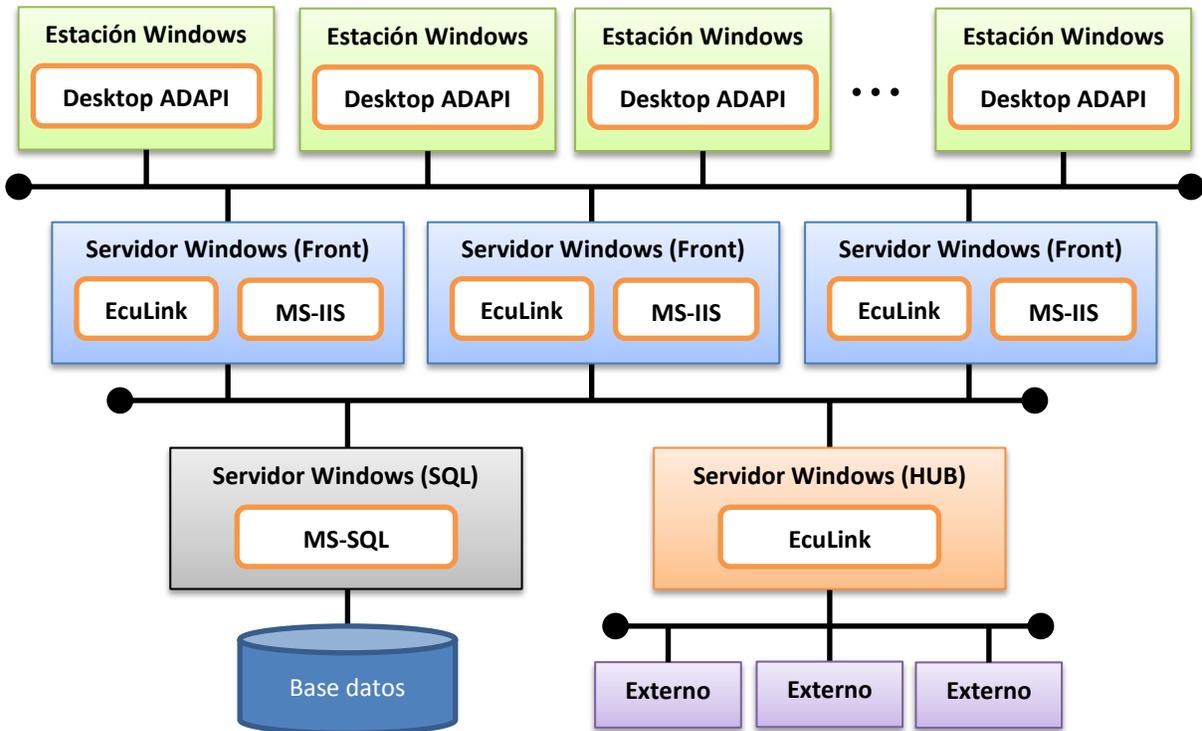


Para verificar que el **Filtro ISAPI** de seguridad está funcionando correctamente, se debe navegar con **Microsoft Internet Explorer** a la aplicación internet del producto **OBCOM ADAPI**. Luego hay que revisar los mensajes del **Visor de eventos** (Event Viewer), y buscar el mensaje que genera el filtro cuando comienza a operar correctamente (ver siguiente figura). Todos los mensajes que genera el filtro tienen un **Origen** llamado “**OBCOM ADAPI Filter**”.



5 Arquitectura HUB

El producto **OBCOM ADAPI** se puede configurar en una arquitectura de varias capas llamada **HUB**. Esta arquitectura permite (1) distribuir la carga de las estaciones, (2) aumentar la disponibilidad del sistema, y (3) centralizar servicios comunes a toda la organización.



Como se observa en la figura, la arquitectura **HUB** posee las siguientes capas y componentes:

- 1) **Estaciones de trabajo:** computadores utilizados por los usuarios del sistema, donde se ejecuta la aplicación **OBCOM Desktop ADAPI**. Esta aplicación provee sus servicios conectándose a los servidores MS-IIS y EcuLink, que se encuentran en los Servidores Front. La conexión se realiza usando un mecanismo de balanceo que distribuye las estaciones de trabajo uniformemente entre los distintos Servidores Front.
- 2) **Servidores Front:** computadores que tienen instalados y configurados los servidores MS-IIS y EcuLink, como se explicó anteriormente en este documento. Los Servidores Front son todos iguales, y pueden haber tantos como sea necesario para una organización. Los servidores MS-IIS se conectan al Servidor MS-SQL para invocar procedimientos almacenados. Por su parte, los servidores EcuLink proveen sus servicios ya sea conectándose al servidor MS-SQL, o conectándose al Servidor HUB, el cual se explica más abajo.
- 3) **Servidor MS-SQL:** computador que provee los servicios de base de datos, el cual está configurado como se explicó anteriormente en este documento. Dada la importancia única de este servidor, se recomienda configurarlo en un clúster de conmutación por error (failover cluster).
- 4) **Servidor HUB:** computador que tiene instalado y configurado un servidor EcuLink. Este servidor provee servicios comunes a toda la organización, los cuales sólo pueden operar en una, y solo una, instancia. Debido a esto, estos servicios únicos no se pueden instalar en los Servidores Front, porque, en ese caso, habría más de una instancia. En particular, el Servidor HUB permite obtener servicios de entidades externas a la organización, tales como: DICOM o el Servicio de Registro Civil e Identificación de Chile (SRCel).
- 5) **Servicios Externos:** servicios de datos provistos por entidades externas a la organización. Para obtener estos servicios externos, es necesario establecer enlaces, e implementar protocolos específicos a cada entidad externa, lo cual se puede hacer con bastante facilidad usando servidores EcuLink. Además, por razones de seguridad, es común que las entidades externas permitan conexiones desde un solo punto, lo cual obliga a establecer estos enlaces desde el Servidor HUB, el cual es único, pero puede ser usado por todos en la organización.

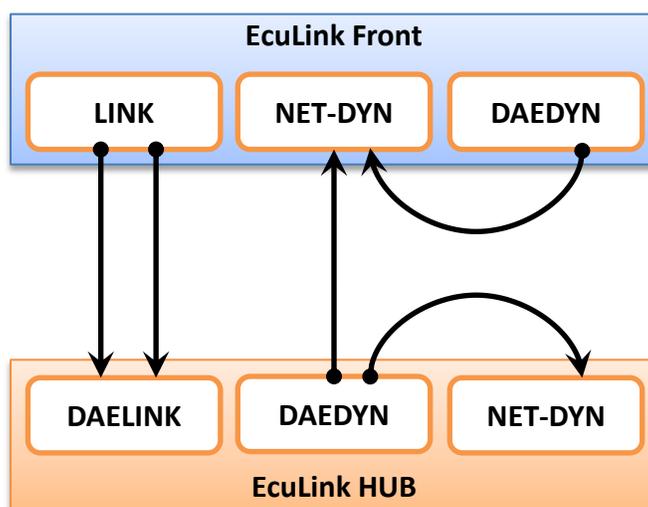
La arquitectura **HUB** provee altos niveles de disponibilidad debido a que la mayoría de los servicios están entregados por dos o más servidores. Los servicios de base de datos están configurados en un cluster de conmutación por error (failover cluster). Por su parte, los servicios de MS-IIS, así como los servicios de EcuLink, están disponibles en cada Servidor Windows Front. Si cualquiera de estos servidores falla, los restantes continúan atendiendo a las estaciones clientes. La única excepción es el Servidor Windows HUB, el cual no puede ser configurado en modalidad de alta disponibilidad, debido a que provee servicios que no se pueden duplicar (singletons).

5.1 Servidores EcuLink Front y HUB

Los servidores EcuLink Front son utilizados por las estaciones clientes de la organización. La principal labor de los servidores EcuLink Front es proveer servicios de autenticación a través de los servidores estándar ACASER, ACCSER y NETSER. Los servidores EcuLink Front también pueden ser configuradores para que provean otros servicios que puedan estar duplicados.

Como se explicó anteriormente, aquellos servicios que no pueden ser duplicados (singleton), deben ser provistos por el servidor EcuLink HUB. Como las estaciones clientes sólo pueden acceder a los servidores EcuLink Front, es necesario que éstos canalicen los requerimientos de los clientes a el servidor EcuLink HUB.

La tecnología EcuLink permite configurar un conjunto de servidores de modo que algunos de ellos puedan solicitar servicios a los otros. También es posible configurar estos servidores de modo que los mensajes dinámicos (broadcast, unsolicited) generados en uno de ellos puedan fluir a los otros.



Para que un servidor EcuLink **A** pueda enviar una solicitud a otro servidor EcuLink **B**, es necesario configurar uno (o más) conectores **LINK** en **A** para que se conecten al conector **DAELINK** en **B**. Los conectores **LINK** sólo pueden procesar un requerimiento (solicitud y respuesta) a la vez. En ambientes de mucha carga, se hace necesario configurar varios conectores **LINK** para poder tener procesamiento en paralelo.

Para que un servidor EcuLink **A** pueda recibir mensajes dinámicos desde un servidor EcuLink **B**, es necesario configurar el conector **NET-DYN** en **A** para que se conecte al conector **DAEDYN** en **B**. Además, todos los conectores **NET-DYN** de un servidor EcuLink deben conectarse a los conectores **DAEDYN** del mismo servidor EcuLink. Esto último es necesario para poder distribuir los mensajes dinámicos dentro del mismo servidor EcuLink.

5.2 Configuración servidores EcuLink Front

La sección 3 de este documento describe los pasos para configurar un servidor EcuLink estándar. Esta sección describe los cambios que es necesario realizar al archivo de configuración de los servidores **EcuLink Front** para que éstos puedan conectarse con un servidor **EcuLink HUB**.

Primero, hay que configurar el conector **DAEDYN** para que pueda recibir conexiones desde cualquier computador "*" en un puerto "<Puerto-DAEDYN-FRT>" (por ejemplo, el puerto 10105). Luego hay que iniciar (start) el conector **DAEDYN**. Opcionalmente, se puede borrar (delete) el conector **DAELINK**, debido a que los servidores **EcuLink Front** no lo usan.

```
alter daemon "daedyn", address "*",<Puerto-DAEDYN-FRT>"
start daemon "daedyn"
delete daemon "daelink"
```

Segundo, hay que definir un servidor especial (tipo "requester") con nombre "HUB". Este servidor estará encargado de encolar los requerimientos que van hacia el servidor **EcuLink HUB**. Este servidor especial podría tener cualquier nombre, y no es necesario que se llame "HUB", pero normalmente se usa el mismo nombre del servidor EcuLink que atenderá.

```
server HUB, new
server HUB, requester 1
```

Tercero, hay que definir uno (o más) conectores **LINK** para que se conecten al conector **DAELINK** del servidor **EcuLink HUB**. Cada conector **LINK** tiene asociado el servidor especial (requester) definido en el paso anterior. Finalmente, se debe iniciar (start) los conectores **LINK**.

```
define link "LNK-HUB1" like "link", address "<IP-HUB>,<Puerto-DAELINK-HUB>"
alter link "LNK-HUB1", server "HUB"
define link "LNK-HUB2" like "link", address "<IP-HUB>,<Puerto-DAELINK-HUB>"
alter link "LNK-HUB2", server "HUB"
start link "LNK-HUB1"
start link "LNK-HUB2"
```

Cuarto, se debe configurar dos conectores **NET-DYN**. El primero, llamado "**FRT-DYN**", se conecta al puerto del conector **DAEDYN** del mismo servidor **EcuLink Front**. El segundo, llamado "**HUB-DYN**", se conecta al puerto del conector **DAEDYN** del servidor **EcuLink HUB**. Finalmente, se debe iniciar (start) los dos conectores **NET-DYN** definidos.

```
define netdyn "FRT-DYN" like "net-dyn", address "localhost,<Puerto-DAEDYN-FRT>"
define netdyn "HUB-DYN" like "net-dyn", address "<IP-HUB>,<Puerto-DAEDYN-HUB>"
start netdyn "FRT-DYN"
start netdyn "HUB-DYN"
```

5.3 Configuración servidor EcuLink HUB

La sección 3 de este documento describe los pasos para configurar un servidor EcuLink estándar. Esta sección describe los cambios que es necesario realizar al archivo de configuración del servidor **EcuLink HUB** para que pueda operar con los servidores **EcuLink Front**.

Primero, hay que configurar el conector **DAELINK** para que pueda recibir conexiones desde cualquier computador "*" en el puerto "<Puerto-DAELINK-HUB>" (por ejemplo, el puerto 10103). También hay que configurar el conector **DAEDYN** para que pueda recibir conexiones desde cualquier computador "*" en el puerto "<Puerto-DAEDYN-HUB>" (por ejemplo, el puerto 10105). Por último, se debe iniciar (start) los conectores **DAELINK** y **DAEDYN**.

```
alter daemon "daelink", address "*", <Puerto-DAELINK-HUB>"
alter daemon "daedyn", address "*", <Puerto-DAEDYN-HUB>"
start daemon "daelink"
start daemon "daedyn"
```

Segundo, se debe configurar un conector **NET-DYN** llamado "**HUB-DYN**", el cual se conecta al puerto del conector **DAEDYN** del mismo servidor **EcuLink HUB**. Finalmente, se debe iniciar (start) el conector **NET-DYN** definido.

```
define netdyn "HUB-DYN" like "net-dyn", address "localhost,<Puerto-DAEDYN-HUB>"
start netdyn "HUB-DYN"
```