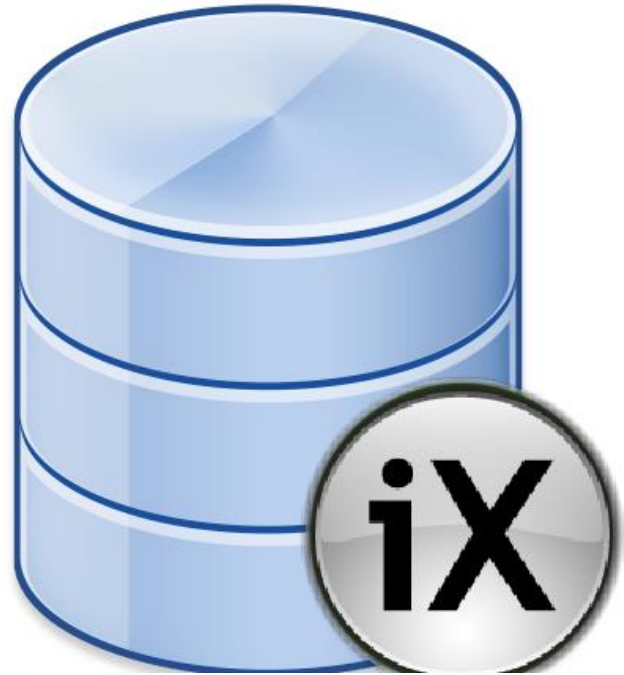


Guía Rápida Sql Proxy HMI – SQL Server



www.elion.es

Comunicación panel HMI con SQL Server

Versión: 1.0

01/11/2017

Servicio Asistencia Técnica
Farell, 5
08014 Barcelona
Tel. 932 982 040
soporte.tecnico@elion.es



Contenido

1. Alcance.....	3
2. Configuración PC.....	4
a. Información.....	4
b. Creación Base de Datos	5
c. Creación login.....	7
d. Configuración acceso (SqlProxy).....	10
3. Configuración Ix Developer	11



1. Alcance

En esta guía se pretende mostrar cómo se configura y se da acceso a una base de datos SQL Server mediante el Software Sql Proxy.

Las bases de datos SQL Server son, generalmente, las bases de datos que se hospedan en un ordenador para almacenar datos. El Software SQL Proxy, nos hace “de puente” con un panel HMI de Beijer el cual no contempla lenguaje de conectividad SQL Server.

Al tener este acceso a esta información, o datos, podemos visualizar en cualquier momento los valores para tratarlos, podremos editarlos, guardar nuevos o eliminar los propios valores si tenemos conocimientos de lenguaje SQL o .Net.

Siendo más específicos en esta guía, podremos hacer uso tanto de la base de datos SQL Lite que implementan las pantallas Beijer, como las bases de datos SQL Server de un ordenador.

Podremos mover registros como Alarmas, DataLogger, Recetas, Tendencias, campos de texto, etc. A esta base de datos o, por el contrario, leer valores, previamente inscritos en la base de datos, en el HMI.

Dando un paso más allá, podremos llegar a desarrollar un software para ordenador (SCADA) y tratar los datos escritos por el HMI desde el ordenador, liberando espacio en el panel cuando estos datos, pueden ser devueltos al HMI ya tratados.



2. Configuración PC

a. Información

En primer lugar, deberemos descargarnos el software que deberá correr en el ordenador donde esté instalado el SQL Server, el cual hará de puente entre HMI y BBDD SQL Server.

El programa, lo podemos descargar de la web de Beijer con el siguiente enlace:

[SQL Proxy - Beijer](#)

O con una descarga directa en Mediafire con los tres archivos a utilizar durante la configuración e implementación.

[SQL Proxy - Mediafire](#)

Para esta guía, hemos utilizado la versión más reciente a fecha de 01 noviembre de 2017 la cual es: **“SqlProxy_170929”**.

Una vez descargado y descomprimido, nos dirigiremos la ruta **“C:\Users\Public”** y crearemos una carpeta con nombre: **“SqlProxy”** donde descomprimiremos los tres archivos que trae el archivo .rar descargado.

- SqlProxy.ini
- tcpServer.dll
- SqlProxy.exe

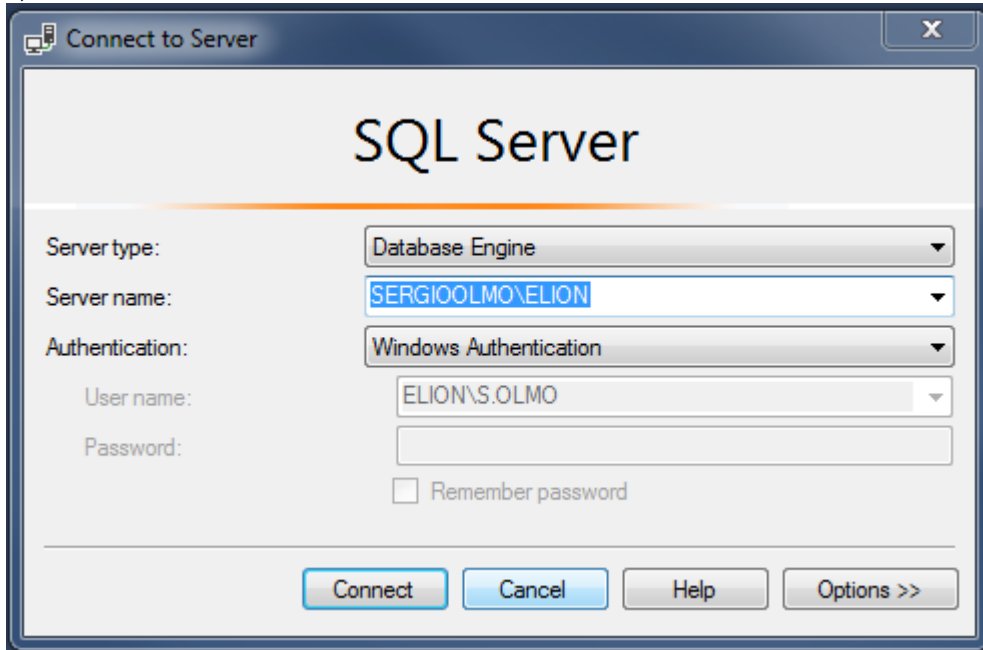
Antes de ejecutar el .exe o pasar a editar el archivo .ini como haremos, deberemos abrir nuestra base de datos SQL Server con el programa: **SQL Server Management Studio**, indiferentemente de la versión, que para el caso, utilizamos la 17.3 y accediendo al servidor, crearemos una nuevo login de acceso.



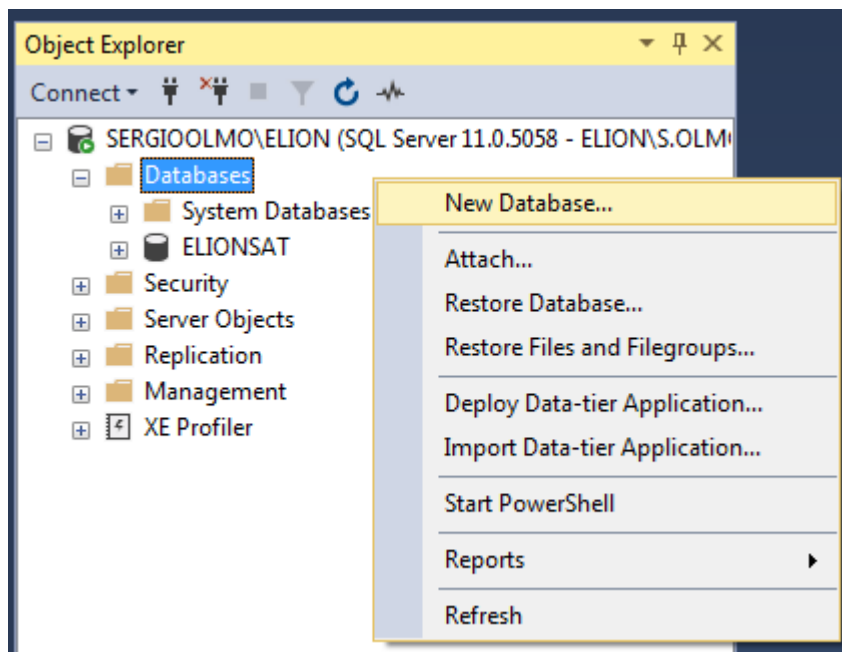
Información versión SQL Server Management Studio de guía

b. Creación Base de Datos

En primer lugar, deberemos acceder a nuestro motor de base de datos.

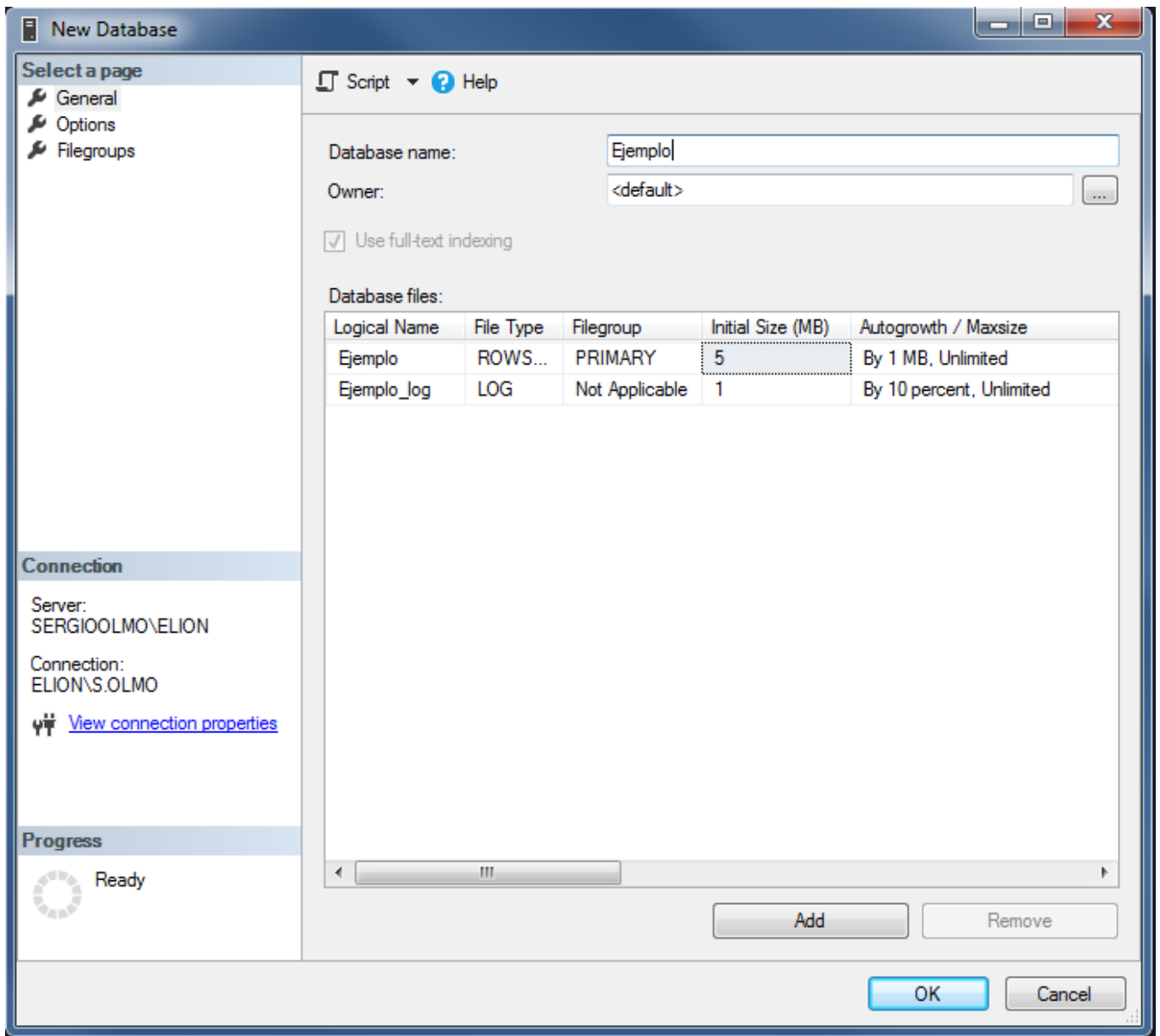


Una vez hayamos accedido, en el árbol de carpetas de la parte izquierda, encabezado por nuestra instancia de servidor. Desplegaremos la carpeta Databases y con el botón derecho, crearemos una nueva base de datos.



Sólo deberemos asignar un nombre a la nueva Base de Datos, ya que más adelante le daremos permisos de acceso según el usuario que intente acceder.



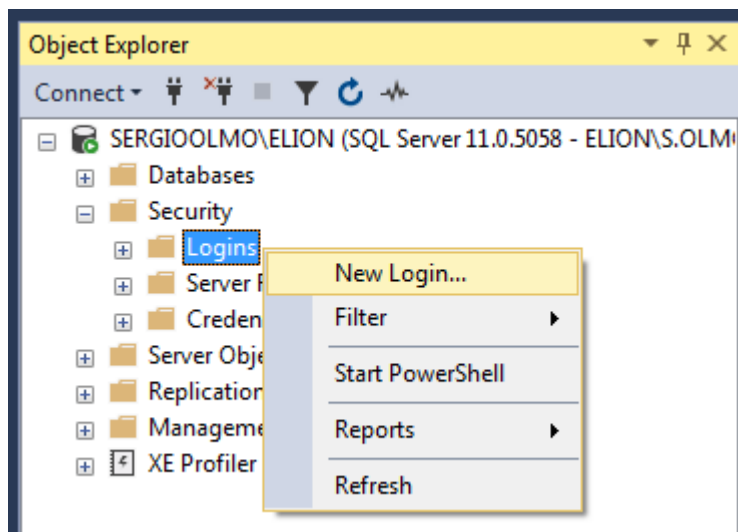


Dentro de esta BBDD podremos crear tablas y procedimientos que podrán ser llamados en el IX o trabajar desde ellos en otros softwares.

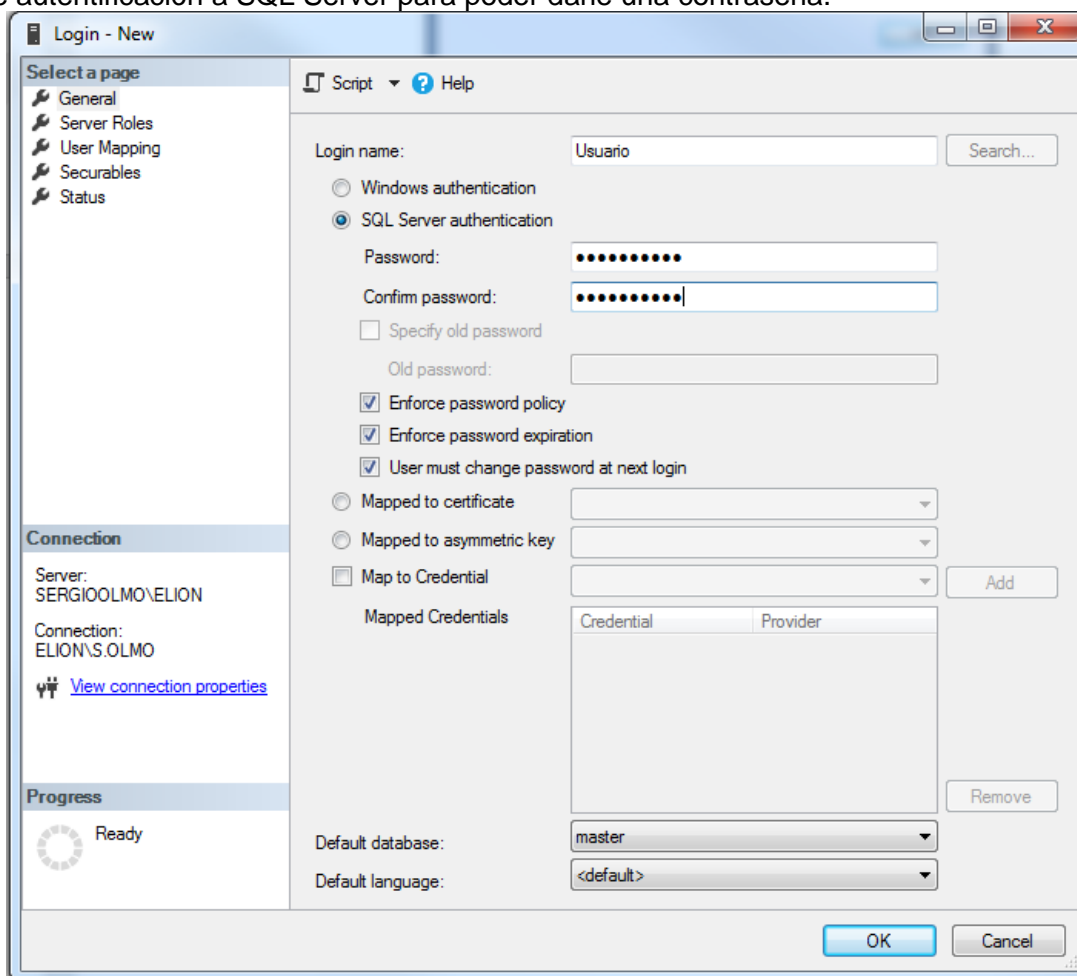


c. Creación login

Para crear un login de acceso, nos situaremos con el cursor sobre *Security* y desplegando el contenido, en *Login* con el botón derecho haremos click en “Nuevo”

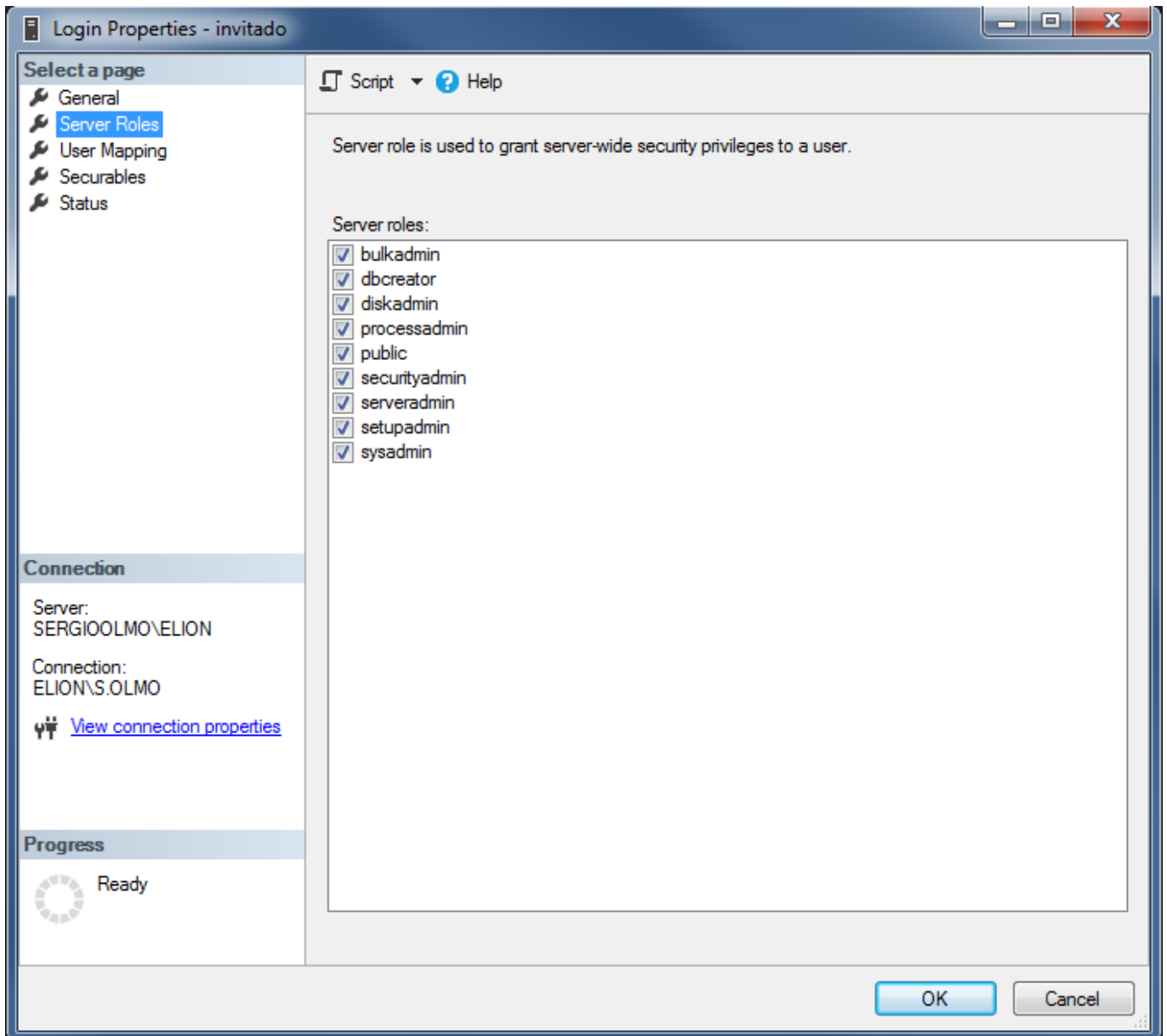


Con la nueva ventana que se nos aparece, deberemos asignarle un nombre de usuario y cambiar el método de autenticación a SQL Server para poder darle una contraseña.

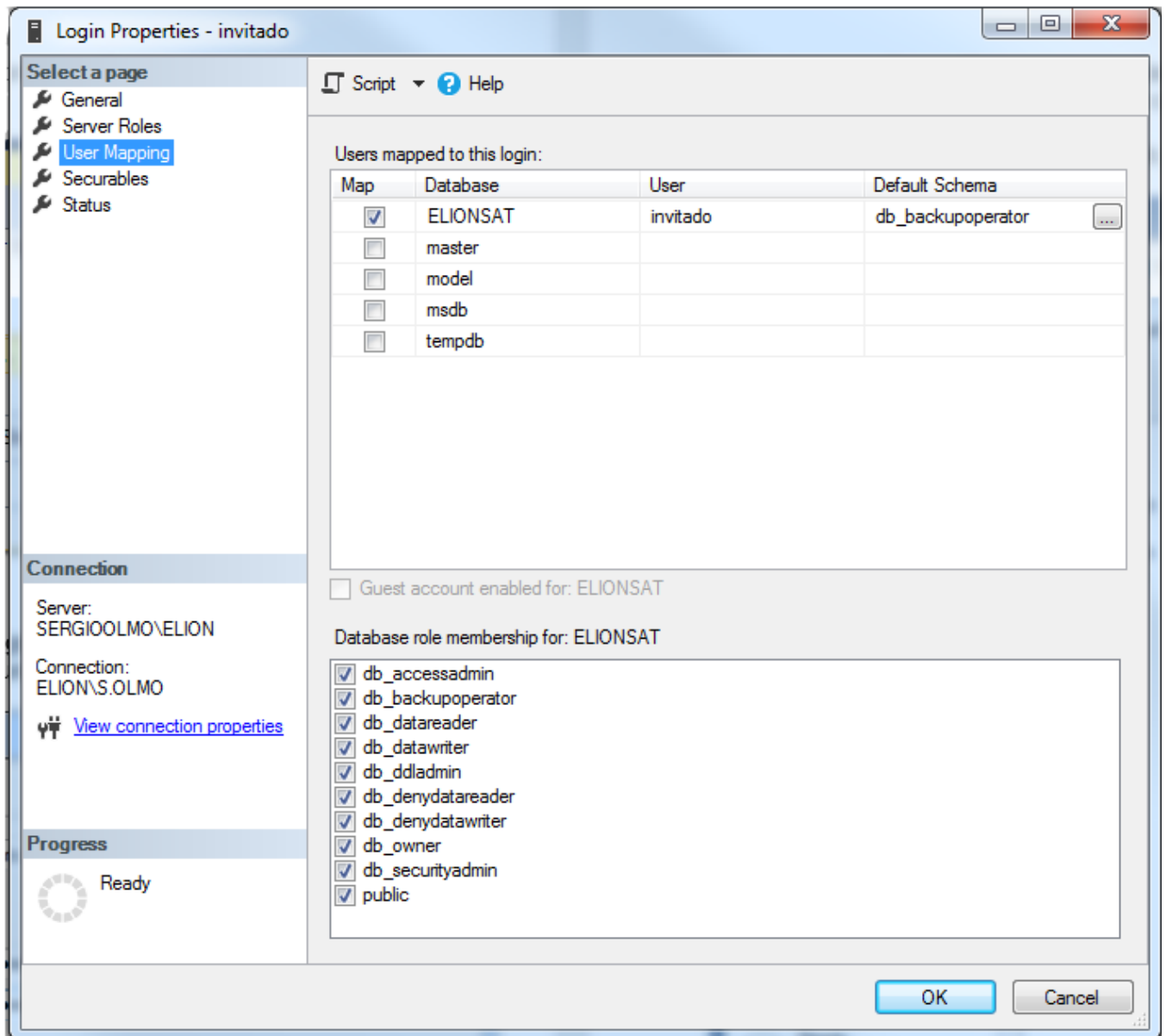


En la página de *Server Roles* podemos dar los permisos que tendrá este usuario sobre toda las bases de datos en conjunto, en caso de que se conecte con un SQL Server Management Studio desde otro ordenador.

Para estar seguros de su acceso absoluto, seleccionamos todas las casillas.



En la siguiente página *User Mapping* asignaremos las bases de datos a las cuales tendrá acceso y qué cantidad total de permisos tendrá. Lectura, escritura, denegado de acceso a otros usuarios, etc. Para estar seguros de su acceso absoluto, seleccionamos todas las casillas.



d. Configuración acceso (SqlProxy)

Haciendo ahora click en *OK* finalizaremos la creación de este usuario y sus permisos, que podrán ser modificados a la postre.

Volviendo ahora a la carpeta en C:\Users\Public\SqlProxy, con el botón derecho del ratón, actuamos sobre el archivo .ini y lo abrimos con nuestro editor de texto deseado, para el caso, Notepad++.

```
1 [Settings]
2 DBConnection=Data Source=DIRECCION_IP;Initial Catalog=BASE_DE_DATOS;User ID=USUARIO;Password=CONTRASEÑA
3
```

Deberemos cambiar **únicamente**:

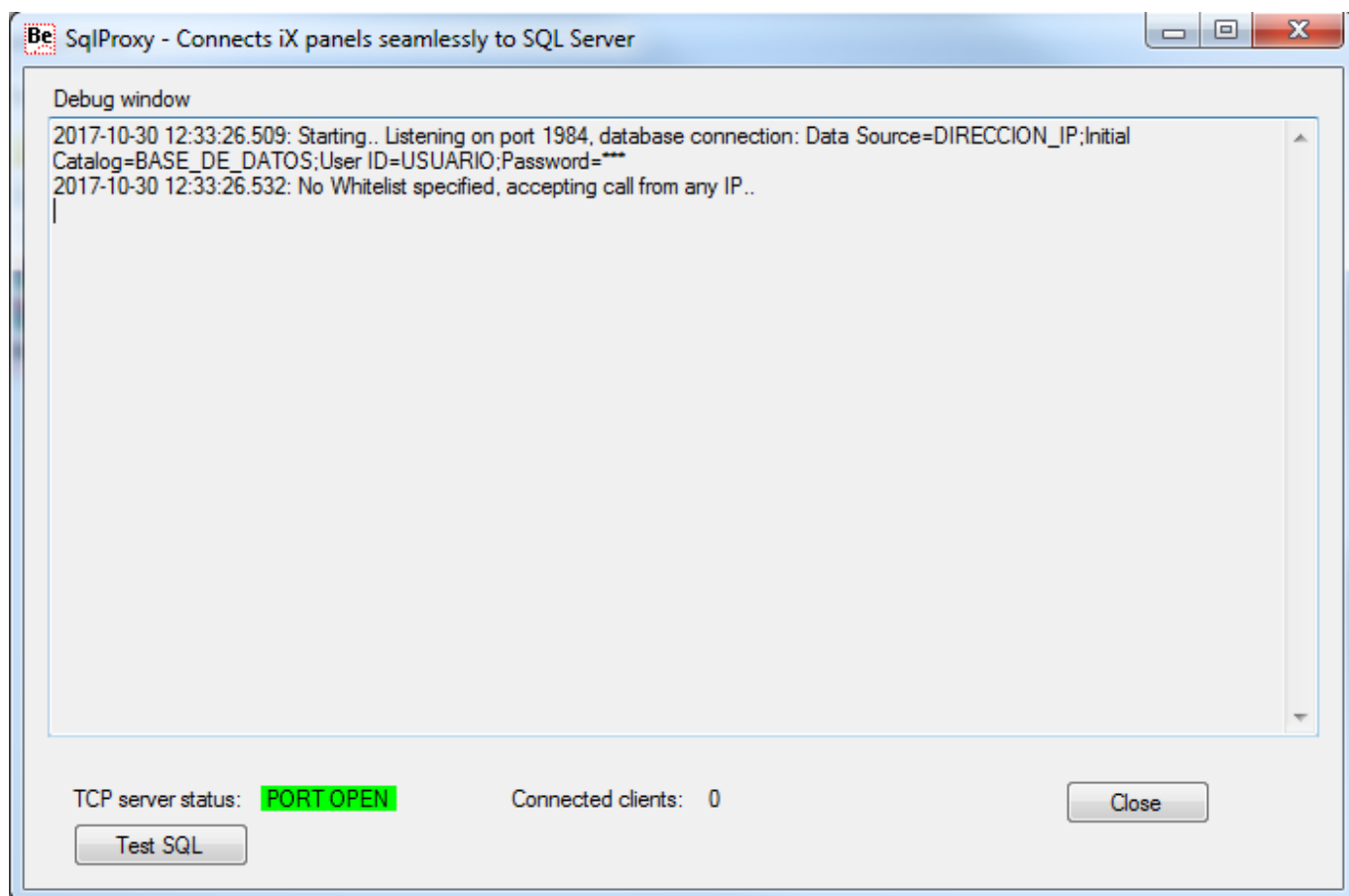
- DIRECCION_IP
- BASE_DE_DATOS
- USUARIO
- CONTRASEÑA

Donde, dirección ip será el nombre de conexión/server, la IP de la red Wireless o la IP de la red local por cable, según la necesidad y capacidad de conexión.

Base de datos, será el nombre de la BBDD creada dentro del servidor a la cual queremos hacer referencia tanto para escritura como para lectura **no confundir con la tabla**.

Usuario y contraseña, son las credenciales creadas en esta misma guía.

Con este paso ya realizado, podremos abrir el archivo SqlProxy.exe que también debería estar ubicado en esa carpeta siguiendo los pasos de la guía.



Si podemos ver que los datos modificados en el archivo .ini se ven reflejados aquí correctamente, podemos hacer una prueba de comunicación más con el botón “*Test SQL*” y haciendo uso de alguna función/query SQL Server.

3. Configuración Ix Developer

Ya tenemos la parte de Base de Datos, Servidor y Accesos configurados; así que vamos a configurar la parte de pantalla desde el Software IX Developer, para el caso, IX Developer 2.30. Podremos descargar el programa de ejemplo desde el siguiente enlace:

[Ejemplo en IX](#)

Cuando abrimos el programa, nos encontramos en la pantalla “Screen1” en la cual, podemos acceder a la parte de Script y modificar la query de SQL para que, de forma automática, al abrir esta pantalla aparezca esta query, que no quiere decir que se ejecute al abrir, simplemente la deja preparada; según este ejemplo.

```
public partial class Screen1
{
    void Screen1_Opened(System.Object sender, System.EventArgs e)
    {
        txtSQL.Text = "Select * From BASE_DE_DATOS";
    }

    void btnSQL_Click(System.Object sender, System.EventArgs e)
    {
        //DateTime dt;

        string sql = txtSQL.Text.Trim();
        //Globals.scmSecurity.Login(s);

        txtResponse.Text= "";

        DateTime dtStart = DateTime.Now;

        string xml = Globals.scmSqlProxy.runSQL(sql);

        TimeSpan span = DateTime.Now - dtStart;
        int ms = (int)span.TotalMilliseconds;
        txtResponseTime.Text = "Tiempo de respuesta: " + ms.ToString() + " ms";

        if(Globals.scmSqlProxy.isDataSet(xml)){
            DataTable dt = Globals.scmSqlProxy.FromXml(xml);
            //Proceso de DataGridView aquí.
            //Devolverá el archivo xml con los datos que contiene la BBDD y el tiempo de respuesta con la operación de arriba.
        }

        txtResponse.Text = xml;
    }
}
```

En este apartado de Script, también vemos que devuelve el valor del tiempo de respuesta que ha tardado en establecer la comunicación, enviar la petición y recibir la confirmación con la operación ejecutada, o por contrario, la negativa con el error.

Este Script nos devuelve el resultado de dicha petición en formato xml para programadores, será el mismo resultado visual que el obtenido en el programa SqlProxy.exe en caso de hacer “Test SQL”.

Es recomendable no tocar ningún otro parámetro, a no ser que tengamos más conocimientos.

Cambiando ahora a la Screen 2 en el propio Script, sí que debemos entrar para parametrizar dicho Script ya que se ejecutará sin un control por interfaz.



```

// Devuelve toda la tabla BASE_DE_DATOS mostrada en el DataGrid
//
void Update()
{
    try{
        string sql="Select * from BASE_DE_DATOS";
        string xml = Globals.scmSqlProxy.runSQL(sql);

        if(Globals.scmSqlProxy.isDataSet(xml)){
            DataTable dt = Globals.scmSqlProxy.FromXml(xml);
            //Aquí es el punto que rellena el DataGrid con los valores de SQL Server.
            Globals.scmSqlProxy.InitializeGrid(DataGrid, dt);
        }else{
            MessageBox.Show(Globals.scmSqlProxy.Error(xml), "A ocurrido un error", MessageBoxButtons.YesNo, MessageBoxIcon.Exclamation, MessageBoxDefaultButton.Button1);
        }
    }
    }catch{};
}

```

La función “Update” se ejecuta automáticamente al abrir la Screen 2 y al hacer click al botón de Refrescar.

Esta función, hace referencia, en este caso, a la misma que teníamos en la Screen 1 pero de forma que rellena el DataGridView de la Screen 2 con todos los datos asociados dentro de la BBDD del SQL Server.

De forma que, en vez de obtener el resultado en formato .xml de código, obtendremos los datos de la base de datos en formato columnas.

A continuación, abriremos el script module y deberemos cambiar la “DIRECCION_IP” por la que estemos utilizando o queramos utilizar para la comunicación.

```

public string runSQL(string sql)
{
    string xml = "";

    try
    {
        //Ajuste dirección IP, local, WiFi, etc.
        IPAddress IP = IPAddress.Parse("DIRECCION_IP");

        port = 1984;
        clientEndpoint = new IPEndPoint(IP, port);

        Socket clientSocket = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);
        info.socket = clientSocket;
        try
        {
            info.socket.Connect(clientEndpoint);
            info.PortOpened = true;

            //Thread.Sleep(500);
        }
    }
}

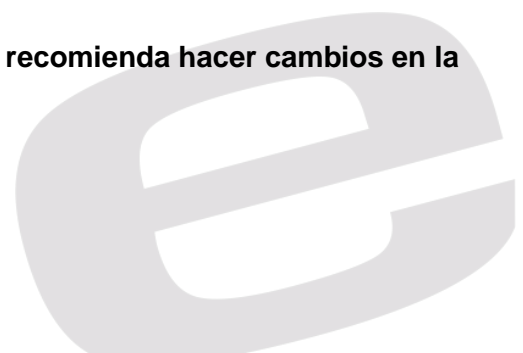
```

La línea de código, ubicada un poco más abajo que hace mención a:

“dgView.Font = new Font("Tahoma", 11, FontStyle.Regular); “

Da lugar al formato y tamaño de letra, la cual es de fácil comprensión.

Únicamente, en caso de un mayor y avanzado conocimiento se recomienda hacer cambios en la parte de código que sigue en este ScriptModule.



sqlProxy -

Screen1 Screen2 CPU 0 % RAM 498584 kB free 30/10/2017 15:22:30

Select * From BASE_DE_DATOS

Run SQL Tiempo de respuesta: 3644 ms

```
<?xml version="1.0" encoding="utf-8"?>
<DataSet>
  <xs:schema id="NewDataSet" xmlns=""
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:msdata="urn:schemas-
microsoft-com:xml-msdata">
  <xs:element name="NewDataSet" msdata:IsDataSet="true"
msdata:UseCurrentLocale="true">
  <xs:complexType>
  <xs:choice minOccurs="0" maxOccurs="unbounded">
  <xs:element name="Table">
  <xs:complexType>
  <xs:sequence>
  <xs:element name="Apellido" type="xs:string" minOccurs="0" />
  <xs:element name="Nombre" type="xs:string" minOccurs="0" />
  <xs:element name="Extension" type="xs:string" minOccurs="0" />
```

Resultado de la Screen 1

sqlProxy -

Screen1 Screen2 CPU 0 % RAM 451892 kB free 30/10/2017 15:25:19

Apellido	Nombre	Extension	ngeografico	movil
S.A.T.	.	2240	932982040	671092189
OLMO	Sergi	2265	932982040	671483314
REVUELTA	Iván	2252	932982052	671092198
SOLÉ	Josep	2261	.	607892341
SORINAS	Jordi	2241	932982041	671092757

Refresh Up Down

Resultado obtenido de la Screen 2

DELEGACIONES:

Cataluña:

Tel. 932 982 000
elion@elion.es

Centro:

Tel. 913 835 709
elionmad@elion.es

Sur:

Tel. 955 943 441
egiraldez@elion.es

Norte:

Tel. 943 217 200
imorales@elion.es



ELION, S.A.

Farell, 5
08014 Barcelona
Tel. 932 982 000
Fax 934 311 800
elion@elion.es
www.elion.es

Servicio Asistencia Técnica

Farell, 5
0814 Barcelona
servicio.tecnico@elion.es

