

COMPARTIMOSS

Revista especializada en Tecnologías SharePoint

PROGRAMANDO LA RIBBON EN SHAREPOINT 2010

REST, WEBAPI 2 Y SHAREPOINT 2013 – WEBAPI Y ODATA

INTRODUCCIÓN AL SEARCH DRIVEN DEVELOPMENT EN SHAREPOINT ONLINE Y SHAREPOINT 2013

CREACIÓN DE UN CHAT PARA SHAREPOINT CON SIGNALR



02

COMPARTIMOSS

Revista especializada en Tecnologías SharePoint

Staff

CompartiMOSS es una publicación independiente de distribución libre en forma electrónica. Las opiniones aquí expresadas son de estricto orden personal, cada autor es completamente responsable de su propio contenido.

DIRECCIÓN GENERAL

- Gustavo Velez
- Juan Carlos Gonzalez
- Fabian Imaz
- Alberto Diaz

Contacte con nosotros

revista@compartimoss.com
 gustavo@gavd.net
 jcgonzalezmartin1978@hotmail.com
 fabian@siderys.com.uy
 adiazcan@hotmail.com

BLOGS

<http://www.gavd.net>
<http://geeks.ms/blogs/jcgonzalez>
<http://blog.siderys.com>
<http://geeks.ms/blogs/adiazmartin>

REDES SOCIALES

Facebook:

<http://www.facebook.com/group.php?gid=128911147140492>

LinkedIn:

<http://www.linkedin.com/groups/CompartiMOSS-3776291>

Twitter:

@CompartiMOSScom

Contenidos

03

Editorial

Hemos arrancado un nuevo año juntos y esperamos que, como siempre, hayan disfrutado todos los números anteriores.

04. Programando la Ribbon en SharePoint 2010

07. Las Aplicaciones High-Trust (II)

10. Integración Dynamics CRM 2015 con SharePoint

15

REST, WebAPI 2 y SharePoint 2013 – WebAPI y OData

21. Introducción al Search Driven Development en SharePoint Online y SharePoint 2013

24. Entrevista Jorge Castañeda

25. Administra mejor tus aplicaciones de servicio

30

Introducción a los Grupos de Office 365 – Trabaja como una red (II)

34. Búsqueda de datos empresariales en SharePoint 2013 (I)

40- Creación de un Chat para SharePoint con SignalR

42. Novedades en Power BI

45. Cálculo de costes en Microsoft Azure

48. La importancia de diseñar y aplicar un buen uso de Gobernanza en nuestro entorno SharePoint 2013



03

Editorial

Hemos arrancado un nuevo año juntos y esperamos que, como siempre, hayan disfrutado todos los números anteriores. Desde la dirección de la revista estamos continuamente pensando cómo mejorar número a número; 2015 supondrá también para CompartiMOSS un año de grandes retos en el que buscaremos adaptarnos y prepararnos a los cambios continuos en las tecnologías y plataformas con las que habitualmente trabajamos en la revista, y siguiendo la estela y el camino que Microsoft va marcando en su apuesta por la nube. Por este motivo, a partir el próximo número, empezaremos a publicar más artículos en torno a la plataforma Office 365 y los servicios y tecnologías que la sustentan: SharePoint Online, Microsoft Azure, Lync Online, Exchange Online y Dynamics CRM Online. Queremos invitar a todos nuestros lectores y autores que nos ayuden con este reto. Si bien hemos estado hablando de la plataforma de productividad en la nube de Microsoft, esta vez queremos darle un énfasis más determinante ya que sabemos que hay muchos más clientes año a año en esta versión de nuestro servidor favorito.

También queremos aprovechar para agradecer a todos los colaboradores y autores que incansablemente comparten sus experiencias y conocimientos para que podamos armar los distintos números que lanzamos, ¡muchas gracias!

EL EQUIPO EDITORIAL DE COMPARTIMOSS

Programando la Ribbon en SharePoint 2010

Es bastante común que se necesiten agregar botones a la barra de herramientas de SharePoint 2010 (RIBBON). Muchas veces esto se resuelve en forma declarativa o con SharePoint Designer, pero otras veces necesitamos ir un poco más allá, por ejemplo, ejecutar código C# luego de presionar un botón o lograr que el botón este asociado a una lista específica y no a un tipo de lista. En este artículo veremos un ejemplo de cómo lograr este tipo de programación con Visual Studio. ¡Que lo disfruten!

El objetivo

Nuestro objetivo es agregar tres botones a la página de edición de una lista en particular. Esos tres botones estarán agrupados bajo un mismo título tal como muestra la imagen 1:



Imagen 1.- Botones agregados a la Ribbon.

Al presionar el botón, se abre una página ASPX en C# que nos permite trabajar con la API de SharePoint tal como muestra la imagen 2:

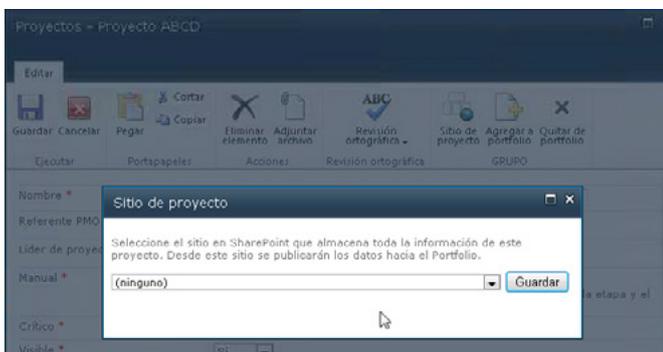


Imagen 2.- Página de aplicación llamada desde el botón de la Ribbon.

Es bastante común que se necesiten agregar botones a la barra de herramientas de SharePoint 2010

¿Cómo agregar los botones?

Los botones suelen agregarse en forma declarativa, pero en este caso lo haremos mediante código C# para poder agregarlos en una lista en particular, y no en un tipo de lista. En este caso lo haremos a través de un evento de activación de feature. Esto puede verse en esta parte del código:

```
public override void
FeatureActivated(SPFeatureReceiverProperties properties)
{
    using (SPWeb mi_web = (properties.Feature.Parent as
    SPSite).OpenWeb())
    {
        // Agrego botones personalizados para la edición del
        Proyecto
        mi_web.AllowUnsafeUpdates = true;
        SPList lista = mi_web.GetList("/Lists/Proyectos");
        var action = lista.UserCustomActions.Add();
        action.Location = "CommandUI.Ribbon.EditForm";
        action.Sequence = 10;
        action.Title = "SURPOINT";
        action.CommandUIExtension = "@"
    }
}
```

Observar especialmente como agregar los botones en la página de edición de la lista: "CommandUI.Ribbon.EditForm".

Dos puntos adicionales:

- Los botones estarán dentro de un grupo
- Al presionar el botón ejecutamos código JavaScript para llamar a una página ASPX.

Esa página ASPX se carga en una ventana modal. Además, se controla el resultado de esa ventana para decidir si se cierra o no la pantalla de Edición del ítem. Esto puede verse en esta línea de código:

```
CommandAction=""javascript:
eIID = _spGetQueryParam("id");
var dlg=SP.UI.ModalDialog.showModalDialog((url:
'{SiteUrl}/_Layouts/EMP_Procesos/VincularProyecto.
aspx?ProjectId='+eIID, dialogReturnValueCallback:function(res,
val) { if (res == 1) window.frameElement.
commonModalDialogClose(1, 0); }));
"" />
```

Los botones suelen agregarse en forma declarativa, pero en este caso lo haremos mediante código C#

El siguiente es el código completo para agregar los botones:

```
public override void
FeatureActivated(SPFeatureReceiverProperties properties)
{
    using (SPWeb mi_web = (properties.Feature.Parent as
SPSite).OpenWeb())
    {
        // Agrego botones personalizados para la edición del
        Proyecto
        mi_web.AllowUnsafeUpdates = true;
        SPList lista = mi_web.GetList("/Lists/Proyectos");
        var action = lista.UserCustomActions.Add();
        action.Location = "CommandUI.Ribbon.EditForm";
        action.Sequence = 10;
        action.Title = "SURPOINT";
        action.CommandUIExtension = "@"
        <CommandUIExtension>
            <CommandUIDefinitions>
                <CommandUIDefinition Location=""Ribbon.
Templates._children">
                    <GroupTemplate Id=""Ribbon.Templates.
CustomTemplate">
                        <Layout Title=""NewGroupInExistingTabOneLarge""
LayoutTitle=""NewGroupInExistingTabOneLarge">
                            <Section Alignment=""Top"" Type=""OneRow">
                                <Row>
                                    <ControlRef DisplayMode=""Large""
TemplateAlias=""Button1"" />
                                    <ControlRef DisplayMode=""Large""
TemplateAlias=""Button2"" />
                                    <ControlRef DisplayMode=""Large""
TemplateAlias=""Button3"" />
                                </Row>
                            </Section>
                        </Layout>
                    </GroupTemplate>
                </CommandUIDefinition>
                <CommandUIDefinition Location=""Ribbon.ListForm.
Edit.Scaling._children">
                    <MaxSize
                        Id=""Ribbon.ListForm.Edit.MaxSize""
                        Sequence=""15""
                        GroupId=""Ribbon.ListForm.Edit.Groups.
CustomGroup""
                        Size=""NewGroupInExistingTabOneLarge"" />
                </CommandUIDefinition>
                <CommandUIDefinition Location=""Ribbon.ListForm.
Edit.Groups._children">
                    <Group Id=""Ribbon.ListForm.Edit.Groups.
CustomGroup""
                        Sequence=""100"" Title=""SURPOINT""
                        Description=""SURPOINT""
                        Template=""Ribbon.Templates.CustomTemplate">
                <Controls Id=""Ribbon.ListForm.Edit.Groups.
CustomGroup.Controls">
                    <Button
                        Id=""Ribbon.ListForm.Edit.Groups.CustomGroup.
Controls.Button""
                        Alt=""Vincular a sitio de proyecto""
                        Sequence=""1""
                        Image32by32=""/_layouts/images/newweb32.
png""
                        Image16by16=""/_layouts/images/
SharePointFoundation10.png""
                        Command=""vincularSitio""
                        LabelText=""Sitio de proyecto""
                        TemplateAlias=""Button1""
                        CommandType=""General""/>
                    <Button
                        Id=""Ribbon.ListForm.Edit.Groups.CustomGroup.
Controls.Button2""
                        Alt=""Agregar a informe de portfolio""
                        Sequence=""2""
                        Image32by32=""/_layouts/images/uploaddoc.
png""
                        Image16by16=""/_layouts/images/
SharePointFoundation10.png""
                        Command=""agregarInforme""
                        LabelText=""Agregar a portfolio""
                        TemplateAlias=""Button2""
                        CommandType=""General""/>
                </Controls>
            </CommandUIExtension>
        }
    }
}
```

```
<Button
    Id=""Ribbon.ListForm.Edit.Groups.CustomGroup.
Controls.Button3""
    Alt=""Quitar de informe de portfolio""
    Sequence=""3""
    Image32by32=""/_layouts/images/mewa_backb.
gif""
    Image16by16=""/_layouts/images/
SharePointFoundation10.png""
    Command=""quitarInforme""
    LabelText=""Quitar de portfolio""
    TemplateAlias=""Button3""
    CommandType=""General""/>
</Controls>
</Group>
</CommandUIDefinition>
</CommandUIDefinitions>
<CommandUIHandlers>
<CommandUIHandler
    Command=""vincularSitio""
    CommandAction=""javascript:
    eIID = _spGetQueryParam('id');
    var dlg=SP.UI.ModalDialog.showModalDialog((url:
'{SiteUri}/_Layouts/EMP_Procesos/VincularProyecto.
aspx?ProjectId='+eIID, dialogReturnValueCallback:function(res,
val) {if (res == 1) window.frameElement.
commonModalDialogClose(1, 0);});
"" />
<CommandUIHandler
    Command=""agregarInforme""
    CommandAction=""javascript:
    eIID = _spGetQueryParam('id');
    var dlg=SP.UI.ModalDialog.showModalDialog((url:
'{SiteUri}/_Layouts/EMP_Procesos/proyectoInforme.
aspx?Accion=agregar&ProjectId='+eIID,
dialogReturnValueCallback:function(res, val) {if (res == 1)
window.frameElement.commonModalDialogClose(1, 0);});
"" />
<CommandUIHandler
    Command=""quitarInforme""
    CommandAction=""javascript:
    eIID = _spGetQueryParam('id');
    var dlg=SP.UI.ModalDialog.
showModalDialog((url: '{SiteUri}/_Layouts/EMP_Procesos/
proyectoInforme.aspx?Accion=quitar&ProjectId='+eIID,
dialogReturnValueCallback:function(res, val) {if (res == 1)
window.frameElement.commonModalDialogClose(1, 0);});
"" />
</CommandUIHandlers>
</CommandUIExtension>
";
action.Update();
mi_web.AllowUnsafeUpdates = false;
}
}
```

Las páginas de aplicación

Las páginas que son llamadas desde la RIBBON son páginas de aplicación típicas de SharePoint. Ejecutan el código C# que necesitamos. Sólo tenemos que tener en cuenta que deben encargarse de cerrar la pantalla modal. A modo ilustrativo, muestro parte del código de carga de la página:

```
protected void Page_Load(object sender, EventArgs e)
{
    if (!IsPostBack)
    {
        // Obtener el ID del proyecto recibido como parámetro
        string proyecto = "";
        proyecto = Page.Request.QueryString["ProjectId"];

        // Datos del actual sitio elegido
        string idSitio = "";
        using (SPSite sitio = new SPSite(SPContext.Current.Site.
Uri))
        (...)
```

Y lo más importante la parte final del evento OnClick del botón de nuestra página:

```
protected void vincularSitio(object sender, EventArgs e)
{
    ..(
        this.Page.Response.Clear();
        this.Page.Response.Write("<script type='text/
        javascript'>window.frameElement.commonModalDialogClose(1,
        1);</script>");
        this.Page.Response.End();
    )
}
```

Utilizamos **commonModalDialogClose** para cerrar la ventana.

Las páginas que son llamadas desde la RIBBON son páginas de aplicación típicas de SharePoint

Conclusión

En este breve artículo vimos que no es tan complejo agregar componentes avanzados en la RIBBON de SharePoint. La posibilidad de ejecutar código C# de servidor, nos abre un abanico de posibilidades para extender nuestros sitios. Y lo que es más importante, se hace de una forma que resulta bastante amigable para el usuario, ya que la RIBBON es una barra que se ha acostumbrado a utilizar.

¡Hasta la próxima!

JUAN PABLO PUSSACQ LABORDE

SharePoint MVP

Blog: <http://surpoint.blogspot.com/>



Las Aplicaciones High-Trust – II

En el último artículo para CompartiMOSS hablé sobre el concepto de las aplicaciones High-Trust o de alta confianza (también conocidas como S2S, server-to-server) en los entornos de SharePoint 2013. Pues bien, hoy vamos a meternos en la harina y hacer una de ellas.

Preparando el entorno para crear la app High-Trust

Para comenzar a crear una aplicación High-Trust, necesitamos configurar varias cosas en nuestro entorno local de SharePoint 2013. Las configuraciones no son muchas, pero es fácil olvidarse de una de ellas y luego tendremos problemas para investigar de donde viene el error.

Antes que nada, necesitamos que nuestro SharePoint 2013 tenga activo el servicio de perfiles de usuario y que además tenga indexados los perfiles de los usuarios que vamos a utilizar en la aplicación. Esto es necesario porque el servicio de autenticación de una app High-Trust necesita “encontrar” el usuario en el servicio de perfiles de SharePoint para poder ejecutar las consultas en su nombre. Si el perfil del usuario no está, la autenticación fallará.

Realmente, el token de acceso que la app envía hacia SharePoint contiene el identificador del usuario, y SharePoint se basa en él para saber si el usuario es válido o no, buscándolo en su base de datos de perfiles. El identificador suele ser el SID del usuario Windows, su UPN o nombre de usuario de Active Directory. Si usamos otros sistemas de autenticación como FBA o Claims, los identificadores serán otros. Es estrictamente necesario que el identificador del usuario esté presente en su perfil y que no haya repeticiones. Si os pica mucho la curiosidad, hay un excelente post de Steve Peschka al respecto (1).

Para poder firmar el token de la app, necesitamos un certificado SSL. Mientras desarrollamos, podemos usar un certificado de desarrollo firmado por nosotros mismos (self-signed certificate). Luego, en producción, usaremos un certificado real. Además, para que nuestra app se pueda comunicar con SharePoint de manera segura, necesitamos que la comunicación esté encriptada bajo HTTPS. Para ello, necesitaremos otro certificado SSL con la URL de la app. Esto no es necesario en desarrollo, donde podemos relajar la restricción y usar HTTP, pero en producción esto sería una imprudencia seria.

Para crear un certificado autofirmado, iremos a la consola de IIS y bajo el apartado “Server Certificates” y dentro de él la opción “Create Self-Signed Certificate”. Le daremos el nombre CertificadoHighTrust.

Al final, exportaremos el certificado incluyendo la clave privada. Como contraseña le pondremos “password”. Al final, tendremos un fichero PFX con el certificado digital que usaremos en nuestra app. Este fichero tiene que estar en una carpeta accesible desde Visual Studio. En nuestro caso, como estamos desarrollando en una máquina de SharePoint, no tenemos que mover el fichero y lo tendremos en la ruta C:\Certificates\CertificadoHighTrust.pfx.



Imagen 1.- Creando un certificado SSL autofirmado.

Necesitamos que nuestro SharePoint 2013 tenga activo el servicio de perfiles de usuario y que además tenga indexados los perfiles

También haremos una exportación del certificado sin la clave privada, para obtener el fichero CertificadoHighTrust.cer. Para ello, tenemos que ir a “Server Certificates” dentro del IIS, abrir el certificado y en la pestaña “Details” ir a la opción “Copy to file” indicando que no queremos la clave privada.



Imagen 2.- El certificado SSL con las partes públicas (.cer) y privadas (.pfx).

Ahora vamos a comprobar los permisos necesarios para que SharePoint pueda procesar nuestros certificados. Los requerimientos son dos:

- El application pool SecurityTokenServiceApplicationPool tiene que tener permisos de lectura sobre la carpeta de los certificados.

- El application pool de la aplicación web en la que instalaremos la app (en nuestro caso, la del puerto 80) tiene que tener permisos de lectura sobre la carpeta de los certificados.

En nuestro caso, son las cuentas SPFarm y SQLSvc. Les daremos los permisos correspondientes en la carpeta Certificates.

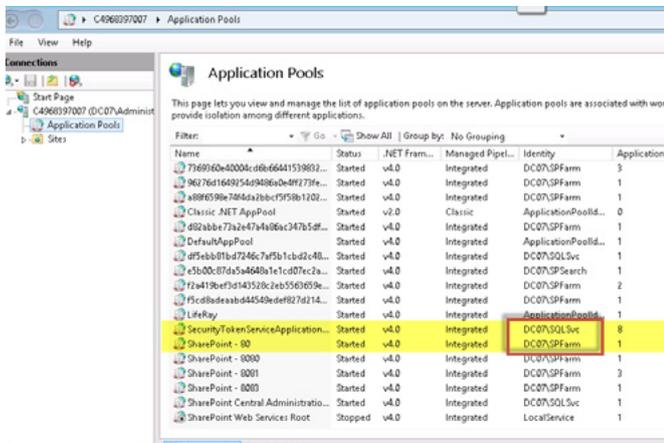


Imagen 3.- Los dos application pools a los que hay que dar acceso al certificado.

Ahora tenemos que hacer que SharePoint reconozca nuestro certificado. Abrimos una consola PowerShell de SharePoint y registramos el certificado como de confianza.

```
$publicCertPath = "C:\Certificates\CertificadoHighTrust.cer"
$certificate = New-Object System.Security.Cryptography.X509Certificates.X509Certificate2($publicCertPath)
New-SPTrustedRootAuthority -Name "CertificadoHighTrust" -Certificate $certificate
```

Una vez que SharePoint se fía de nuestro certificado, vamos a proceder a configurar lo que se conoce como un emisor de confianza (trusted issuer). Esto no es más que indicarle a SharePoint que los tokens firmados por un "emisor de confianza" son de fiar. Y, ¿cómo sabe SharePoint qué un emisor es de confianza? Primero, el ID del emisor (un GUID que va dentro del token) tiene que existir en la configuración de SharePoint. Segundo, el token tiene que estar firmado por un certificado del que SharePoint se "fía" porque tiene su parte pública. Como esta parte del certificado la hemos hecho ya, sólo falta decirle a SharePoint el ID de nuestro proveedor de confianza. Puede ser cualquier GUID, y aquí vamos a utilizar el aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee (si usamos letras en el GUID, tienen que ser en minúscula). Bonito y fácil de recordar, ¿verdad?

Para registrar nuestro emisor de confianza, hay que ejecutar el siguiente código en PowerShell, a continuación del script de importación del certificado:

```
$realm = Get-SPAuthenticationRealm
$specificIssuerId = "aaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee"
$fullIssuerIdentifier = $specificIssuerId + '@' + $realm
New-SPTrustedSecurityTokenIssuer -Name "CertificadoHighTrust" -Certificate $certificate -RegisteredIssuerName $fullIssuerIdentifier -IsTrustBroker iisreset
```

Ya podemos proceder a desarrollar la app, pero antes de esto vamos a permitir el uso del certificado autofirmado relajando los permisos de autenticación porque estamos en desarrollo.

```
$serviceConfig = Get-SPSecurityTokenServiceConfig
$serviceConfig.AllowOAuthOverHttp = $true
$serviceConfig.Update()
```

Desarrollando la app

La app necesitará el certificado SSL y conocer la contraseña de su parte privada. Además, la cuenta bajo la que se ejecutará la app (el application pool del IIS) tiene que tener permisos para acceder a la ubicación del certificado. Abrimos Visual Studio 2013 y creamos una app de SharePoint 2013. Al salir el asistente, le indicamos que queremos una app provider-hosted y que la identidad de la app se establecerá mediante certificado.

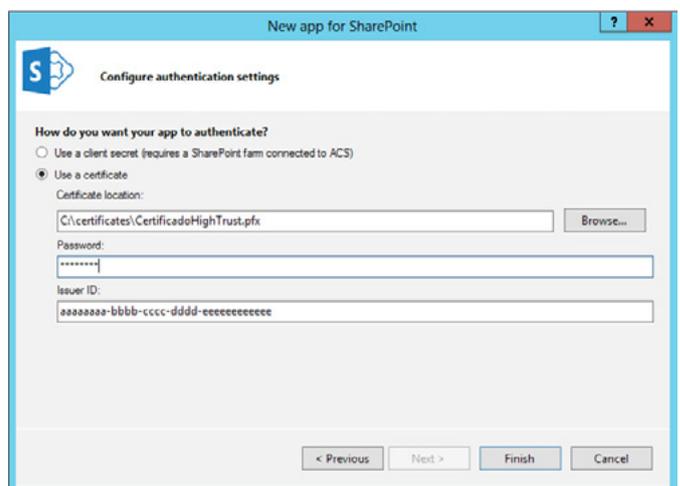


Imagen 4.- Diciendo a Visual Studio que cree una app High-Trust.

Ahora tendremos una aplicación (en mi ejemplo, creada con Web Forms) que muestra el nombre del sitio actual de SharePoint donde está instalada la app.

La app necesitará el certificado SSL y conocer la contraseña de su parte privada

El código que hace la llamada a SharePoint es muy sencillo:

```
protected void Page_Load(object sender, EventArgs e)
{
    Uri hostWeb = new Uri(Request.QueryString["SPHostUrl"]);

    using (var clientContext = TokenHelper.GetS2SClientContextWithWindowsIdentity(hostWeb, Request.LogonUserIdentity))
    {
        clientContext.Load(clientContext.Web, web => web.Title);
        clientContext.ExecuteQuery();
        Response.Write(clientContext.Web.Title);
    }
}
```

Como se puede ver, el contexto de SharePoint se establece usando la clase auxiliar TokenHelper con el método GetS2SClientContextWithWindowsIdentity. Esta llamada obtiene un contexto de High-Trust app usando la identidad del usuario Windows que está ejecutando la aplicación. Esta es la configuración por defecto, pero se puede modificar para usar la identidad federada, por ejemplo. Ejecutando la aplicación, nos sale el diálogo de otorgar permisos a la aplicación, y al aceptarlo, podemos ver el título del sitio de SharePoint, "Home".

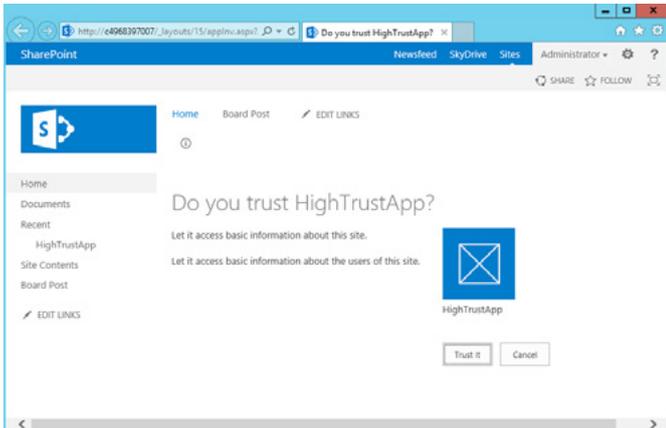


Imagen 5.- Dando permisos a la app High-Trust.

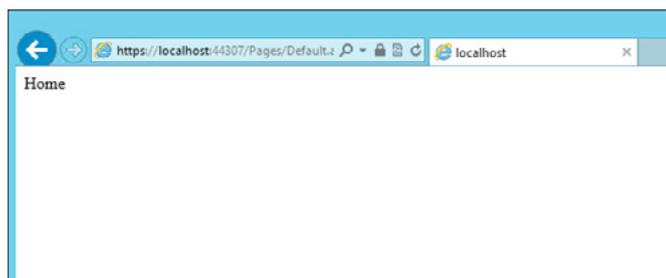


Imagen 6.- La app High-Trust en ejecución.

La prueba de Fiddler

Si abrimos Fiddler para ver el tráfico HTTP entre la aplicación y SharePoint, veremos que la aplicación hace una llamada a la API CSOM (/_vti_bin/client.svc/ProcessQuery). Si miramos la petición, en las cabeceras veremos un parámetro llamado Authentication con el valor "Bearer: " seguido de un texto codificado en Base64. Este es nuestro token de acceso.

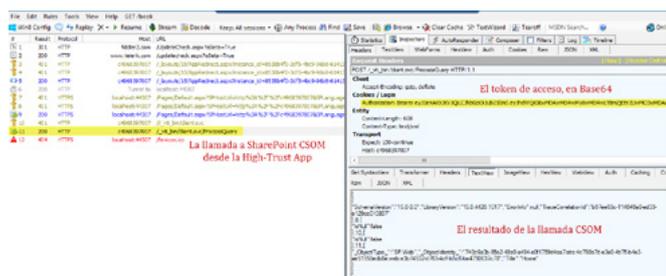


Imagen 7.- La base de datos de fotografías en SQL Server.

Para más información sobre la estructura del token, hay un magnífico post de Kirk Evans al respecto (2).

Conclusión

Espero haber desmitificado un poco el mundo de las aplicaciones High-Trust con este post. Como veréis, nos permite usar el modelo de apps sin tener que estar en la nube, lo que es un paso importante para poder adaptar nuestros desarrollos a los escenarios híbridos que parece que serán mucho más habituales en el futuro.

me parece fascinante de este modelo High-Trust es que podemos separar la identidad de nuestra aplicación de los usuarios que la van a utilizar

Seguro que muchos de vosotros tenéis dudas sobre si el modelo de las apps vale la pena para los desarrollos On-Premise. Yo también lo pienso muchas veces. Lo que me parece fascinante de este modelo High-Trust es que podemos separar la identidad de nuestra aplicación de los usuarios que la van a utilizar porque la app ya tiene su identidad y sus permisos dentro de SharePoint. Por el otro lado, ahora mismo sólo el modelo de apps ofrece la posibilidad de usar autenticación OAuth con SharePoint. Hasta ahora, si queríamos llamar a SharePoint desde fuera de él, teníamos que poner las credenciales del usuario. Incluso no hace falta seguir el modelo de apps en su totalidad, ya que podemos hacer una app que no tenga que iniciarse desde SharePoint, o una app que funciona como una aplicación de línea de comandos, servicio de Windows o un proceso de Node.js.

En el próximo artículo de esta serie veremos cómo usar un proveedor de identidad diferente al de Windows, como por ejemplo un usuario de Claims (SAML).

Bibliografía

- (1) <http://blogs.technet.com/b/speschka/archive/2012/08/15/oauth-and-the-rehydrated-user-in-sharepoint-2013-how-d-they-do-that-and-what-do-i-need-to-know.aspx>
- (2) <http://blogs.msdn.com/b/kaevans/archive/2013/04/05/inside-sharepoint-2013-oauth-context-tokens.aspx>

EDIN KAPIC
 Arquitecto SharePoint
 edin.kapic@spenta.es
 @ekapic
<http://www.spenta.es>

Integración Dynamics CRM 2015 con SharePoint

Dynamics CRM y Sharepoint

Dynamics CRM y SharePoint son soluciones de Microsoft que tienen una vida ya de más de diez años cada uno. Con el paso del tiempo, estas dos soluciones no solo fueron incrementando funcionalidades, sino que se han convertido en grandes plataformas donde poder implementar cualquier tipo de proceso de negocio. Si bien las posibilidades son ilimitadas en ambas plataformas, por su naturaleza cada una lo hace de otras maneras, y por lo tanto existen escenarios de negocio que se acercan más a una o a otra.

Tradicionalmente se asocia al CRM con las relaciones con los clientes (Ventas, Servicio al cliente y Marketing) y SharePoint se asocia más a escenarios de entornos de intranets, gestión de contenidos o documental. Ambas asociaciones son correctas, pero desde mi punto de vista son algo simplistas si se tiene en cuenta las posibilidades que ambas plataformas pueden ofrecer.

Desde el punto de vista funcional, SharePoint y Dynamics CRM tienen enfoques diferentes como se puede apreciar en la siguiente imagen:

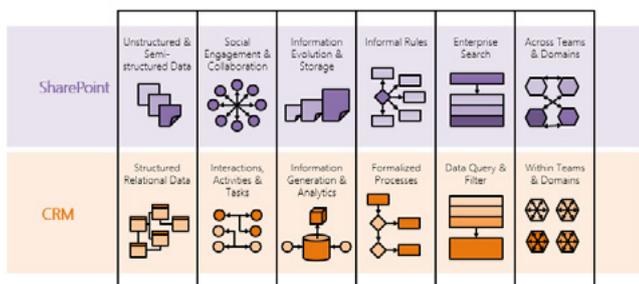


Imagen 1.- Enfoques de SharePoint y Dynamics CRM.

Este gráfico explica a alto nivel los enfoques la forma de trabajo en una y otra plataforma, que permiten enfocar los datos y los procesos relacionados con los mismos de diferente manera.

Por qué integrar Dynamics CRM con SharePoint

La integración de Dynamics CRM con SharePoint nace de la necesidad de cubrir un GAP funcional en Dynamics CRM en cuanto a la gestión documental que dispone SharePoint. En Dynamics CRM existe la posibilidad de añadir ficheros adjuntos, que son almacenados como "notas" relacionadas con las entidades de CRM. Pero estos ficheros

adjuntos solo se permiten guardar en CRM y nada más. En SharePoint en cambio podemos disponer de todo un abanico funcional y de colaboración mucho más rico.

Además del GAP funcional, existen un también un tema económico. En la "nube" se paga por uso y uno de estos conceptos es el espacio de almacenamiento. El almacenamiento de Dynamics CRM Online es mucho más caro que el almacenamiento de SharePoint. Esto es importante, ya que en caso de enfrentarnos a un escenario en Dynamics CRM Online con gran volumen de ficheros adjuntos, se debería evaluar la repercusión del precio de estos adjuntos en cuanto al almacenamiento en la nube.

Arquitectura de integración lado servidor

La integración de la versión actual de Dynamics CRM 2015 con SharePoint Online está bastante desarrollada, aunque existen algunos puntos que deberían ser mejorados para garantizar una completa integración. En versiones anteriores de Dynamics CRM la integración estaba basada en el lado cliente, en donde mediante un iframe apuntando a SharePoint, se podía acceder a documentos allí almacenados desde un contexto de un registro de CRM. Esto significa que desde el navegador cliente, es el que directamente accede y se autentica contra el SharePoint.

Desde Dynamics CRM 2013 Online, existe la posibilidad de integrar Dynamics CRM desde el lado servidor. Esto quiere decir que directamente será Dynamics CRM el que se encargue de conectar con SharePoint, consultando y actualizado los datos que allí se almacenan. Esto permite que, desde el navegador del cliente, solo se acceda a Dynamics CRM, haciendo la integración más transparente para los usuarios. A continuación, se enseñan estos dos tipos de escenarios:

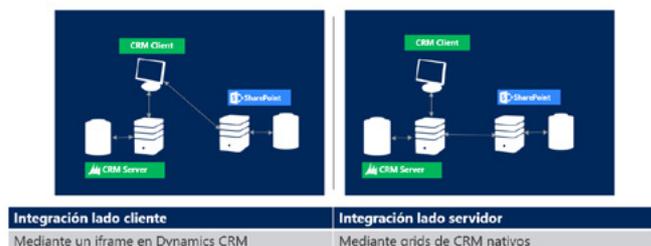


Imagen 2.- Escenarios de integración entre Dynamics CRM y SharePoint.

Configurando la Integración de documentos Dynamics CRM y SharePoint

Veamos un paso a paso como configurar Dynamics CRM 2015 Online con SharePoint Online. Para este ejemplo, he creado una trial de Office365, con una trial de SharePoint y de CRM.

- Lo primero que voy a hacer es crearme una biblioteca de documentos en SharePoint, donde quiero almacenar los documentos a almacenar provenientes de Dynamics CRM:

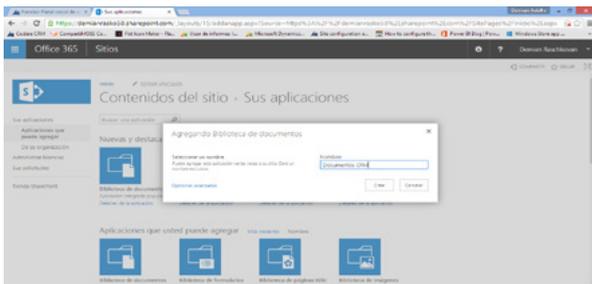


Imagen 3.- Creación de la biblioteca de documentos en SharePoint.

- Luego en Dynamics CRM en Configuración->Administración de documentos, podemos configurar toda la integración.



Imagen 4.- Acceso a la configuración de integración de SharePoint y Dynamics CRM.

- Y al habilitar la integración de SharePoint basada en servidor, introducimos la dirección URL del sitio de SharePoint:

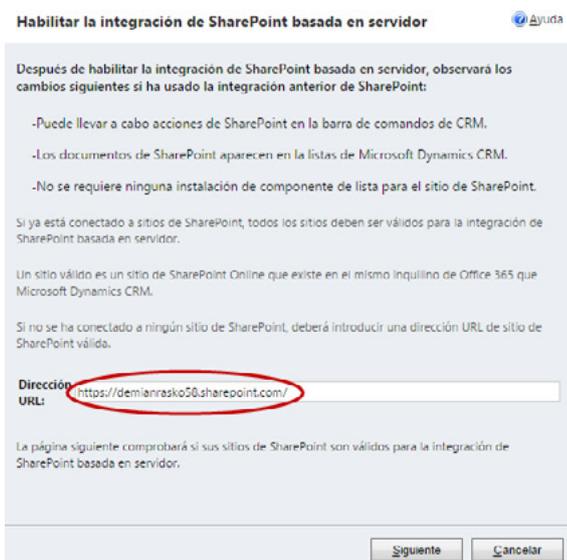


Imagen 5.- Configuración de la URL de SharePoint Online.

- A continuación, la URL de SharePoint Online es validada por el asistente:

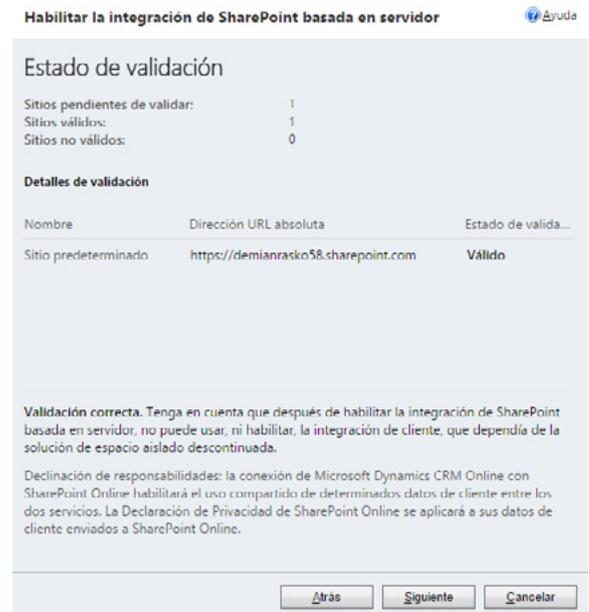


Imagen 6.- Validación de la URL de SharePoint Online.



Imagen 7.- Pantalla de conexión de Dynamics CRM y SharePoint Online.

- Luego abrimos la configuración de administración de documentos, donde introducimos la url de SharePoint y seleccionamos que entidades dispondrán de documentos:

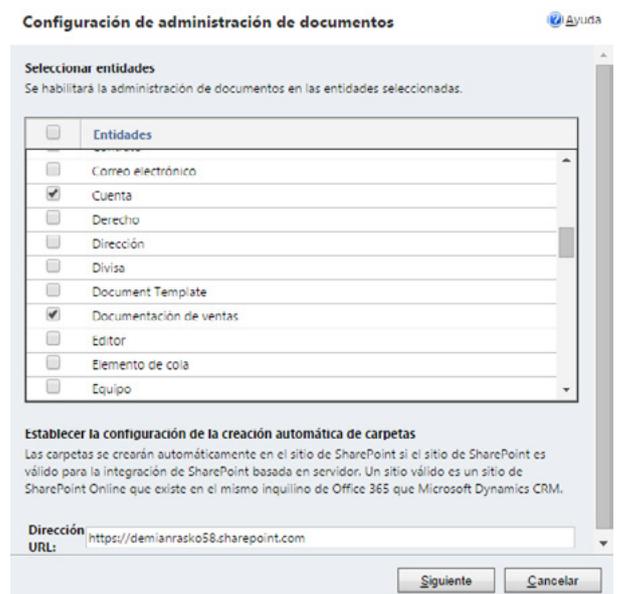


Imagen 8.- Selección de las entidades con documentos.

- A continuación, configuraremos la estructura de carpetas para cada entidad:





Imagen 9.- Selección de la estructura de carpetas.

- Una vez seleccionadas las entidades y la configuración de la estructura de carpetas, se muestra una ventana que indica el estado de creación de las bibliotecas de documentos en SharePoint.

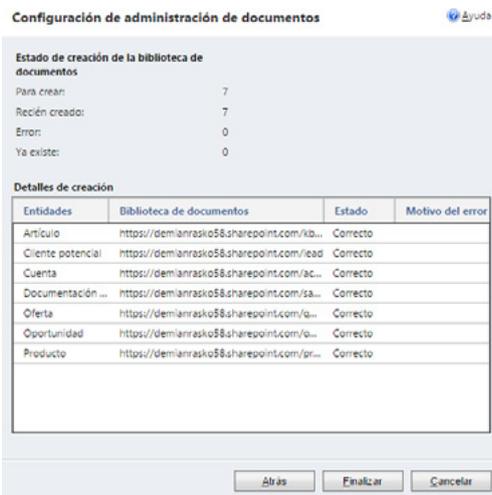


Imagen 9.- Estado de creación de las bibliotecas de documentos.

La integración de Dynamics CRM con SharePoint nace de la necesidad de cubrir un GAP funcional en Dynamics CRM en cuanto a la gestión documental

- Una vez configurada esta integración, desde un formulario de una Cuenta de CRM, podemos acceder a los Documentos relacionados:

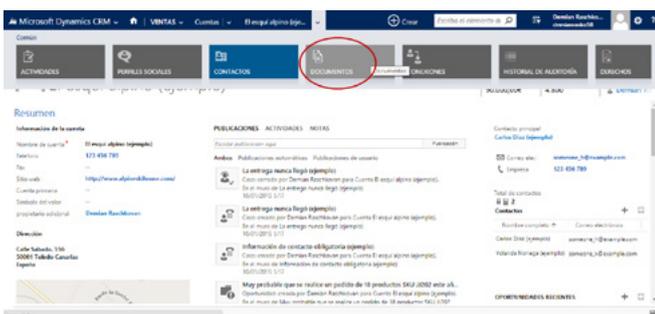


Imagen 10.- Acceso a los documentos desde una cuenta del CRM.

- Si es la primera vez que se accede a los documentos de esta Cuenta, aparece la siguiente ventana:

Crear una carpeta

La carpeta se creará en la ubicación:
[https://demianrasko58.sharepoint.com/account/A-\(ejemplo\)_2E96EE867F98E41180D5C4346BAD3068](https://demianrasko58.sharepoint.com/account/A-(ejemplo)_2E96EE867F98E41180D5C4346BAD3068)

Imagen 11.- Diálogo de confirmación de creación de carpeta en SharePoint.

- Al confirmar esta acción, se crea la carpeta en SharePoint, relacionada con el registro de la Cuenta de CRM. A partir de este momento, los documentos pueden añadirse en CRM, mediante formularios nativos, y estos documentos son almacenados en SharePoint.

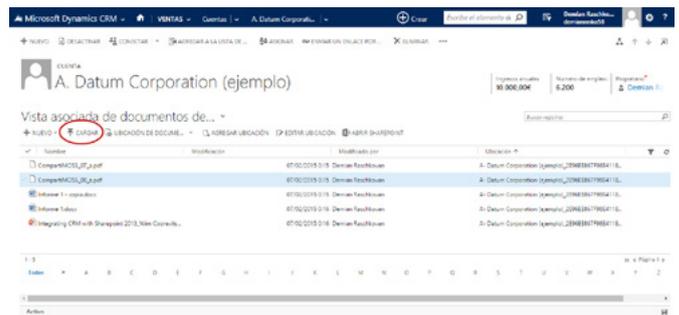


Imagen 12.- Carga de documentos desde Dynamics CRM en SharePoint.

Dynamics CRM y SharePoint son soluciones de Microsoft que tienen una vida ya de más de diez años cada uno

Integración de metadatos de SharePoint con Dynamics CRM

Otra funcionalidad interesante relacionada con la integración de Dynamics CRM con SharePoint, es la utilización de metadatos. La creación de un documento en SharePoint, permite que otros usuarios, incluso no usuarios de CRM, puedan interactuar con un documento. Una opción podría ser un flujo de aprobación o revisión de un documento. A continuación, se detalla cómo se pueden crear metadatos en SharePoint, que pueden ser consultados directamente en CRM.

- Primero creamos una columna con un desplegable de estados, que quiero que se vean en SharePoint

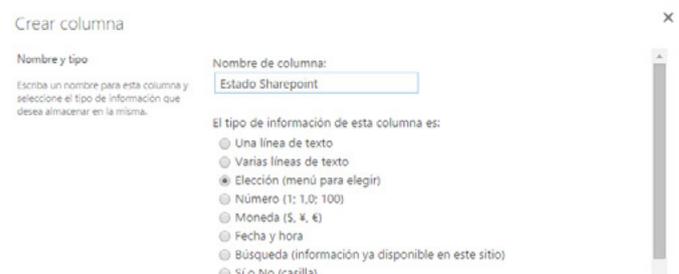


Imagen 13.- Definición de la columna de tipo Elección.

Escriba cada opción en una línea distinta:

Borrador
 Aprobado
 Rechazado

Mostrar opciones con:

- Menú desplegable
- Botones de opción
- Casillas (permitir varias selecciones)

Imagen 14.- Valores para la columna.

- Así se ven estos registros que se han creado en CRM en SharePoint:

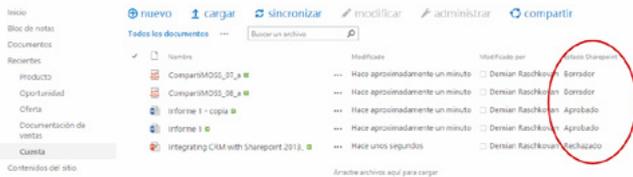


Imagen 15.- Visualización de los valores de la columna en la vista de la lista.

- La entidad que utiliza Dynamics CRM para integrar con SharePoint se llama “Documentos de SharePoint”, y si creamos en la misma un campo con el mismo nombre que el metadato creado en SharePoint, Dynamics CRM se encarga de forma automática de integrar los mismos:

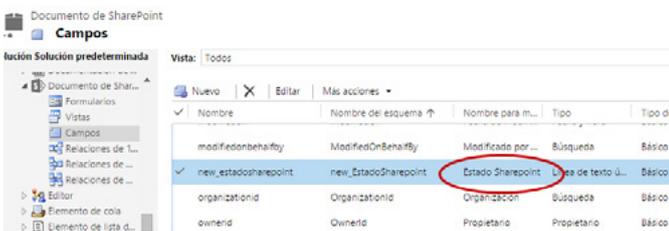


Imagen 16.- Definición del campo Estado SharePoint en Dynamics CRM.

- También añadimos esta columna en las vistas de la entidad, para que sea visible. Publicamos y ya tenemos disponible este campo en el listado de documentos:

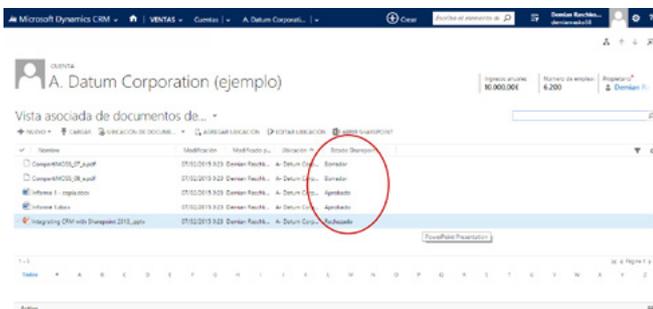


Imagen 17.- Incorporación de la columna a la vista de documentos en Dynamics CRM.

Esta integración de atributos es muy interesante, pero hay que tener en cuenta que los tipos de datos que hay en Dynamics CRM y SharePoint no son iguales y se debe revisar qué tipo de metadato se puede mapear con qué tipo de campo con CRM. En este ejemplo se ha utilizado un atributo de tipo texto, que puede ser mapeado con cualquier metadato de SharePoint.

Escenarios avanzados de integración

Lo explicado anteriormente ofrece una serie de opciones configurables, que están muy bien, pero en muchos escenarios de negocio, las integraciones deben ser algo más complejas. Ya sea por requerimientos funcionales no cubiertos o por requerimientos de seguridad. Es por esto, que se abre un abanico de nuevas posibilidades que los propios entornos de Dynamics CRM y SharePoint ofrecen. Como se comentaba al inicio, la madurez de los productos nos garantiza que “casi” todo requerimiento funcional se pueda conseguir, aunque a veces hay que valorar si los esfuerzos de desarrollo a dedicar para este tipo de extensiones funcionales son los adecuados.

El primero de los escenarios es el siguiente:

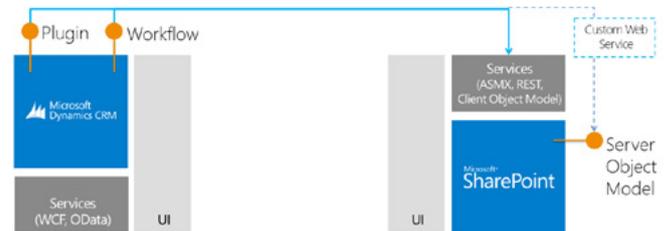


Imagen 18.- Escenario de integración con Flujos de Trabajo o Plugins de Dynamics.

Desde Dynamics CRM, se pueden crear Plugins o actividades de Workflow, que hagan llamadas directamente a SharePoint, o a un web service intermedio que conecte con SharePoint. De la misma manera, podría existir un camino inverso desde SharePoint, hacia Dynamics CRM:

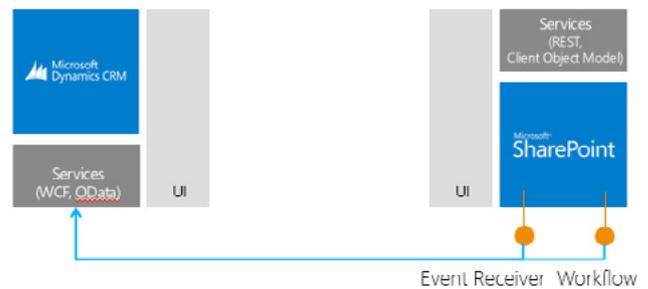


Imagen 19.- Escenario de integración con Flujos de Trabajo o Event Receivers de SharePoint.

Desde el punto de vista de la interfaz de usuario, en Dynamics CRM se podría añadir un iframe en formularios o Dashboards que apunten a URLs de SharePoint, o bien crear Recursos Web (HTML, Silverlight), que conecten directamente al punto REST y consulten datos de SharePoint y se muestren.

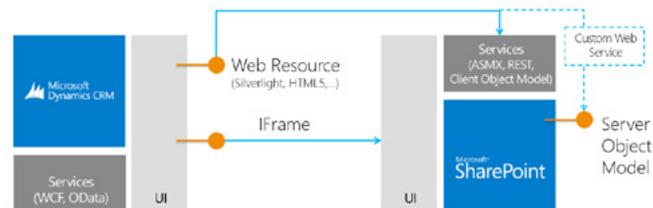


Imagen 20.- Escenario de integración con iFrames o Recursos Web desde Dynamics.

Igualmente, el camino inverso en SharePoint se podría añadir un iframe con una URL de cualquier registro de

CRM, o desarrollar una App o un WebPart, que conecte con los servicios de Dynamics CRM.

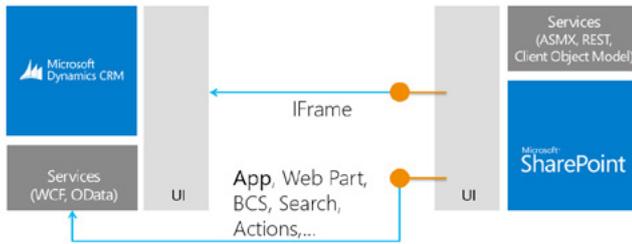


Imagen 21.- Escenario de integración con artefactos (Apps, WebParts, BCS, etc) de SharePoint.

Esta integración de atributos es muy interesante, pero hay que tener en cuenta que los tipos de datos que hay en Dynamics CRM y SharePoint no son iguales

Conclusiones

Como puede verse, la integración de Dynamics CRM con SharePoint, es sencilla de montar y configurar, pero la

funcionalidad que ofrece es limitada. De todos modos, la madurez de estas dos plataformas permite que se pueda extender a los requerimientos de negocio. La integración de la gestión documental de SharePoint en Dynamics CRM cubre un GAP funcional a CRM, y ofrece a SharePoint la posibilidad de trabajo de documentos que ya vienen relacionados con entidades de negocio de Dynamics CRM. De los temas que quedarían pendientes de montar, es el tema de la seguridad. En Dynamics CRM la seguridad es implícita, basada en roles de seguridad y en la ubicación jerárquica de los usuarios. En cambio, en SharePoint la seguridad es explícita en cada registro, y por lo tanto se debería tener que desarrollar algo para permitir que esta integración sea más sólida.

DEMIAN ADOLFO RASCHKOVAN

Microsoft MVP | Dynamics CRM

Director Técnico | Infoaván Soluciones

mvp@demianrasko.com

[@demian_rasko](https://twitter.com/demian_rasko)

<http://www.demianrasko.com>

[@ComunidadCRM](https://twitter.com/ComunidadCRM)

<http://www.comunidadcrm.com>



Ya están disponibles en Amazon los manuales de la serie sobre administración y desarrollo de SharePoint

www.compartimoss.com/guias

REST, WebAPI 2 y SharePoint 2013 – WebAPI y OData

SharePoint siempre ha sido un sistema abierto a aceptar modificaciones por medio de programación y que expone todos sus componentes e información al mundo exterior por medio de un API (Application Programming Interface) muy poderoso. Ese es, probablemente, uno de los factores más importantes para su aceptación empresarial. Aunque SharePoint siempre ha tenido una API remota (basada en Servicios Web), es decir, que permite crear software que interactúa con el servidor sin que sea indispensable instalar los programas localmente, solamente hasta la versión 2010 Microsoft le incluyó un API remoto basado en REST (Representational State Transfer).

REST, como se explicó en el artículo anterior, es un protocolo abierto y público que permite intercambiar información entre dos sistemas informáticos remotamente. En los últimos años, REST se ha convertido en el estándar para intercambio de información por excelencia, siendo utilizado por todos los grandes fabricantes de software (Amazon, Google, etc.), incluyendo a Microsoft. REST, fuera de ser un protocolo práctico y fácil de utilizar, soluciona los problemas técnicos de rapidez de ejecución bajo carga y transparencia a través de Internet que otros protocolos de Servicios Web, como SOAP, WSCL y BEEP, presentan.

A su vez, Microsoft ha establecido un Framework de programación para crear servicios REST llamado WebAPI, basado en ASP.NET e integrado en Visual Studio. De esta forma, es fácil construir servicios que accedan una gran cantidad de clientes, incluyendo navegadores y dispositivos móviles.

Nota: El primer artículo de la serie (<http://www.compartimoss.com/revistas/numero-22/rest-webapi-2-sharepoint-2013-introduccion>) discute los principios fundamentales de REST y WebAPI.

Otros conceptos de WebAPI

En el primer artículo de la serie se utilizaron las características esenciales del WebAPI, tal y como las crea la plantilla de que dispone Visual Studio 2013 por defecto. Pero el WebAPI permite controlar muchos otros aspectos de un Servicio REST.

Ruteo (Routing). El Controlador es el encargado de manejar las consultas de HTTP y sus métodos públicos son llamados «Acciones». Cuando un Servicio REST creado con el WebAPI recibe una consulta, el Framework rutea la con-

sulta hacia una Acción determinada. El Framework contiene una «Tabla de Ruteo» que indica cómo debe hacer el ruteo. Abriendo el archivo «App_Start/WebApiConfig.cs» se puede observar la tabla creada por defecto:

```
public static void Register(HttpConfiguration config)
{
    config.MapHttpAttributeRoutes();

    config.Routes.MapHttpRoute(
        name: "DefaultApi",
        routeTemplate: "api/{controller}/{id}",
        defaults: new { id = RouteParameter.Optional }
    );
}
```

Esta configuración indica que una consulta hecha al Servicio REST debe ser utilizando el URL `http://[servidor]:[NumeroPuerto]/api/[NombreControlador]`; por ejemplo, en el programa de prueba que se creó en el primer artículo sería `http://localhost:[NumeroPuerto]/api/elementos` para consultar todos los elementos y `http://localhost:[NumeroPuerto]/api/elementos/1` para pedir los datos del primer elemento. Si es necesario, este ruteo por defecto ("api/") se puede modificar, modificando, a su vez, la forma de llamar las Acciones

Fuera de poder rutear todo el Servicio, es también posible rutear solamente una Acción, o cada Acción con un prefijo diferente. Para hacerlo, solamente es necesario decorar la declaración de la clase del Controlador definiendo el nuevo prefijo, por ejemplo:

```
namespace SpRest01.Controllers
{
    [RoutePrefix("api/poc")]
    public class ElementosController : ApiController
    {
```

Lo que rutea la consulta de tal forma que la consulta debe ser hecha por medio del URL `http://localhost:[NumeroPuerto]/api/poc/elementos` y `http://localhost:[NumeroPuerto]/api/poc/elementos/1` en el Servicio diseñado en el artículo. Esto permite crear una ruta específica para caso de Pruebas de Concepto ("poc") en el ejemplo, pero el prefijo puede ser cualquier cadena deseada.

También es posible modificar el ruteo por defecto a las Acciones por medio de decoración del método. Por ejemplo,

la siguiente decoración del método “DarElementos” indica que es el método para utilizar el verbo GET:

```
[HttpGet]
public IEnumerable<Elemento> DarElementos()
{
    return myElementos;
}
```

La Acción se puede modificar por medio de una decoración especial. El siguiente método tiene una decoración indicando el nombre que se debe utilizar para hacer la consulta:

```
[HttpGet]
[ActionName(“DarElementos”)]
public IEnumerable<Elemento> GetAllElementos()
{
    return myElementos;
}
```

De tal forma que la consulta sea “http://localhost:[NumeroPuerto]/api/elementos/DarElementos”.

Finalmente es posible crear métodos públicos que no se deben rutear por medio de la decoración NonAction, de la siguiente forma:

```
[NonAction]
public IEnumerable<Elemento> GetAllElementos()
{
    return myElementos;
}
```

En este caso, el framework no ruteara la consulta GET al método GetAllElementos.

Manejo de excepciones. Cuando un Servicio Web creado con el WebAPI genera una excepción que no ha sido manejada apropiadamente en código, retorna un código 500 de HTTP (Internal Server Error). Pero el framework permite devolver códigos de estado HTTP específicos si el desarrollador lo considera necesario. Por ejemplo, el siguiente método devuelve un error 404 (Not Found) si el elemento buscado no existe:

```
public IHttpActionResult GetElemento(int id)
{
    Elemento unElemento = myElementos.Where(e => e.Id == id).
    First();
    if (unElemento == null)
    {
        throw new HttpResponseException(HttpStatusCode.
        NotFound);
        //return NotFound();
    }
    return Ok(unElemento);
}
```

En el código se utilizan dos métodos diferentes: uno creando una excepción del tipo HttpResponseException, y el segundo (comentariado) retornando directamente un método NotFound. Si es necesario entregar más información al

usuario, se puede utilizar la forma verbosa del método:

```
public IHttpActionResult GetElemento(int id)
{
    Elemento unElemento = myElementos.Where(e => e.Id == id).
    First();
    if (unElemento == null)
    {
        var resp = new HttpResponseMessage(HttpStatusCode.
        NotFound)
        {
            Content = new StringContent(string.Format(“Ningun
            Elemento con id = {0}”, id)),
            ReasonPhrase = “El elemento no se encontro”
        };
        throw new HttpResponseException(resp);
        //return NotFound();
    }
    return Ok(unElemento);
}
```

Otras características: WebAPI ofrece otras características altamente especializadas que se escapan al alcance de este artículo por, entre otras, ser poco utilizadas. Por ejemplo, es posible crear proveedores propios para el manejo de errores que pueden enviar mensajes personalizados que no existen en la lista de errores de HTTP por defecto, y es posible crear formateadores especiales que envían la respuesta del servicio fuera de los dos formatos por defecto (JSON y XML), por ejemplo, en formato csv.

SharePoint siempre ha sido un sistema abierto a aceptar modificaciones por medio de programación

Creación de un Servicio REST integrado en SharePoint 2013

Desafortunadamente Microsoft no admite la creación de Servicios REST personalizados que funcionen simultáneamente con los Servicios REST nativos del servidor, de la misma forma que si se puede hacer con los Servicios Web tradicionales. El principal problema es fundamentalmente la forma en que SharePoint maneja el contexto. Aunque Visual Studio 2013 introdujo una plantilla especial para crear Aplicaciones de SharePoint utilizando MVC, no contiene un proveedor de autenticación y, además, los Controladores definidos para MVC no funcionan correctamente con los Controladores del WebAPI (ApiController) pues los primeros funcionan óptimamente para devolver vistas (del modelo MVC) como HTML, mientras que los segundos han sido diseñados para devolver datos estructurados como XML o JSON. Como se ha indicado, el manejo del contexto es totalmente diferente en aplicaciones MVC y WebAPI, lo que no es de extrañar teniendo en cuenta que el WebAPI está hecha para trabajar con REST que, a su vez, es sin estado, es decir, no mantiene, por defecto, el estado de las consultas ni del servidor que las realiza:

- Un Controlador de MVC hereda de System.Web.Mvc.Controller que, entre otras propiedades, mantiene un registro de la consulta (HttpRequestBase) y de su contexto, basado en el cual se crea el contexto de SharePoint, el que se guarda en la propiedad HttpContext.Session.
- El ApiController de WebAPI hereda de System.Web.Http.ApiController que no contiene un HttpContext sino un HttpContext que, a su vez, contiene una propiedad para la Session. Por otro lado, las propiedades para Request y RequestContext son del tipo HttpRequestMessage y HttpRequestMessageContext, no del tipo HttpContext.

En Internet se pueden encontrar varios intentos para hacer funcionar el WebAPI dentro de los Servicios REST nativos de SharePoint, pero todos no son más que parches de funcionamiento, y no son soportados oficialmente por Microsoft. Las soluciones que se pueden encontrar se basan principalmente en:

- Agregar un objeto de Session al API Controller (<http://www.strathweb.com/2012/11/adding-session-support-to-asp-net-web-api/>). Esta solución básicamente elimina el requisito de sin estado de REST, haciendo que no cumpla el estándar aceptado universalmente.
- Alterar la clase SharePointContextProvider para que acepte el ApiController (<http://blog.baslijten.com/getting-sharepoint-2013-apps-and-webapi-to-work/>). Fuera de ser una solución técnicamente bastante compleja y difícil de implementar (crear una clase HttpRequestBase propia, reemplazar el uso de la clase HttpContext por HttpContext, modificación de los tokens del ApiController), modifica clases por defecto de Microsoft, violando por definición la garantía del producto

En conclusión:

- No hay una solución aceptable por el momento para integrar nuevos Servicios REST creados con el WebAPI con los Servicios REST nativos y por defecto de SharePoint.
- Aunque técnicamente es posible crear Aplicaciones de SharePoint que funcionen como un Servicio REST, esta forma de trabajo no está soportada por el momento por Microsoft, por el estándar de REST y/o son complejas de implementar y de alto riesgo técnico.
- La opción más viable es crear Servicios REST que funcionen paralelamente a SharePoint y que:
 - Accedan a la información contenida en SharePoint por medio de su Modelo de Objetos de Servidor o de Cliente.
 - Utilizan su propio Website de IIS, ya sea utilizando un servidor separado a los utilizados por la granja de SharePoint o un

puerto específico (o un Host Name) si los Servicios tienen que funcionar dentro de los mismos servidores que la granja de SharePoint.

Extensión del Servicio REST de ejemplo

El siguiente ejemplo ha sido pensado para que sea lo más sencillo posible y continuar con el código creado en el primer artículo, de tal forma que se pueda, de nuevo, demostrar lo fácil que es crear un servicio REST con el WebAPI que interactúe con SharePoint. Por simplicidad, el ejemplo solamente lee datos de una Lista Personalizada de SharePoint utilizando el Modelo de Objetos de Servidor, pero el ejemplo se puede ampliar fácilmente para hacer todas las operaciones CRUD y se puede modificar para que utilice el Modelo de Objetos de Cliente. El ejemplo y las imágenes han sido creados utilizando Visual Studio 2013 Update 4 y CSharp, pero cualquier versión de Visual Studio 2013 (y Visual Basic) producirá los mismos resultados.

El ejemplo va a interactuar con una Lista Personalizada de SharePoint (Servidor o Foundation) llamada "ListaTestRest" que, adicionalmente a los campos de "Titulo" e "Id" creados por defecto, tiene otros dos campos de los siguientes tipos:

- Nombre: "Categoria", tipo "Una línea de texto".
- Nombre: "Precio", tipo "Moneda".

Como se puede ver en la definición del Modelo "Elemento", estos cuatro campos corresponden con la entidad "Elemento" configurada para el Servicio.

1. Descargue el código del proyecto creado en el primer artículo, o siga las instrucciones dadas en el para crearlo desde el principio con Visual Studio.
2. Abra el proyecto con Visual Studio. Porque se va a utilizar el Modelo de Objetos de servidor de SharePoint, es necesario agregar una referencia a Microsoft.SharePoint y una directiva using a Microsoft.SharePoint al inicio del archivo de código del Controlador.
3. El código de SharePoint necesita ejecutarse bajo la plataforma de 64 bits, mientras que un proyecto de WebAPI creado con la plantilla de Visual Studio utiliza por defecto código de 32 bits. Por lo tanto, es necesario modificar la compilación de «Any CPU» a «x64». Desde el menú «Compilar» de Visual Studio, seleccione «Administración de configuración» y en el elemento del proyecto seleccione (o cree una nueva) entrada en la columna de «Plataforma» que señale a «x64».

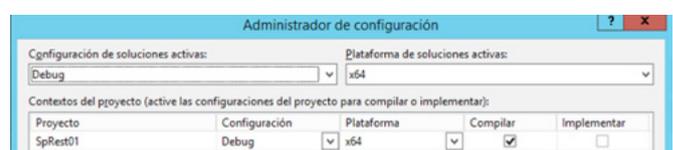


Imagen 1.- Configuración de "x64" como Plataforma de compilación.

- Como Visual Studio utiliza IIS Express como servidor Web estándar en modo de 32 bits, es necesario cambiar la configuración a 64 bits. Desde el menú de “Herramientas” – “Opciones” de Visual Studio, seleccione “Proyectos y soluciones” – “Proyectos web” y seleccione la casilla “Usar la versión de 64 bits de IIS Express para sitios Web y proyectos” (puede que sea necesario reiniciar totalmente el servidor para que la configuración sea efectiva).

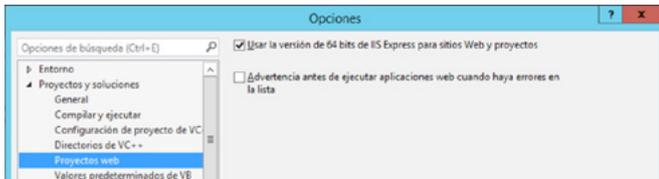


Imagen 3.- Configuración de “x64” para IIS Express.

Note que Visual Studio 2013 utiliza IIS Express versión 8.5. En versiones anteriores de Visual Studio se utilizaba IIS Express 7, que no dispone de una versión de 64 bits. También, la configuración mostrada en la Figura 2 es exclusiva de las últimas versiones de Visual Studio. En versiones anteriores es necesario instalar IIS Express 8 o superior y configurar a Visual Studio para que utilice la versión de 64x manualmente

- Cree una rutina privada en el Controlador para devolver todos los elementos de la Lista de SharePoint. La rutina del ejemplo utiliza el Modelo de Objetos de Servidor de SharePoint, pero la misma rutina se puede crear utilizando el Modelo de Objetos de Cliente si el Servicio utiliza un servidor que no sea uno de los de la granja de SharePoint. La rutina del ejemplo es de la forma:

```
private List<Elemento> DarTodosLosElementos()
{
    List<Elemento> TodosLosElementos = new List<Elemento>();

    SPSecurity.RunWithElevatedPrivileges(delegate()
    {
        using (SPSite mySite = new SPSite("http://servidor"))
        {
            using (SPWeb myWeb = mySite.OpenWeb())
            {
                SPList myList = myWeb.Lists["ListaTestRest"];
                foreach (SPItem unItem in myList.Items)
                {
                    Elemento unElemento = new Elemento();
                    unElemento.Id = unItem.ID;
                    unElemento.Nombre = unItem["Titulo"].ToString();
                    unElemento.Categoria = unItem["Categoria"];
                    ToString();
                    double myPrecio;
                    double.TryParse(unItem["Precio"].ToString(), out
                    myPrecio);
                    unElemento.Precio = myPrecio;
                    TodosLosElementos.Add(unElemento);
                }
            }
        }
    });

    return TodosLosElementos;
}
```

Esta rutina tiene un par de puntos importantes que hay

que remarcar. Primero que todo, utiliza Privilegios Elevados para acceder la Lista de SharePoint. Esto es necesario pues la consulta no está enviando las credenciales del usuario al Servicio, de tal forma que SharePoint va a denegar el acceso. Utilizar Privilegios Elevados no es una buena práctica. En el último artículo de esta serie se discutirán los aspectos de autenticación y autorización de Servicios REST, y se explicará cómo enviar las credenciales del usuario para que SharePoint puede devolver los datos apropiados.

es posible crear proveedores propios para el manejo de errores que pueden enviar mensajes personalizados

El segundo punto importante es que la consulta de la Lista no está creada para que de un buen rendimiento pues utiliza un bucle sencillo para leer todos los elementos. Además, la misma rutina se utilizará para la consulta que lee todos los elementos de la Lista y para la consulta que devuelve los datos de uno solo, lo que la hace terriblemente ineficiente si la Lista contiene altas cantidades de elementos. Como se explicó en el principio, el código solamente es de ejemplo y no está pensado para ser utilizado en un Servicio de producción.

Note también que no se ha hecho ningún esfuerzo para enviar el URL del sitio (ni el nombre de la Lista ni de sus campos) de forma dinámica, todo está codificado directamente. Estos parámetros, especialmente el URL del Sitio en donde se encuentra la Lista, se deberían enviar como parámetros de entrada de la consulta (el primer artículo indica como enviar estos parámetros).

- Finalmente, las dos rutinas públicas para las consultas GET de todos los elementos y de uno solo se han modificado ligeramente para que utilicen la rutina de acceso general:

```
public IEnumerable<Elemento> GetAllElementos()
{
    return DarTodosLosElementos();
}

public IHttpActionResult GetElemento(int id)
{
    List<Elemento> myElementos = DarTodosLosElementos();
    Elemento unElemento = myElementos.Where(e => e.Id == id).
    First();
    if (unElemento == null)
    {
        return NotFound();
    }
    return Ok(unElemento);
}
```

Las dos rutinas simplemente hacen una llamada a la rutina de acceso general para encontrar todos los elementos de la Lista (en el primer caso), y para filtrarlos posteriormente (en el segundo caso).

- Los métodos para crear, eliminar y modificar un

elemento de la Lista siguen exactamente el mismo patrón para acceder a la Lista de SharePoint y para que sean utilizados por el Servicio REST, tal como se especificó en el primer artículo, y no se muestran en este ejemplo.

8. Compile y ejecute el Servicio. Para testearlo se puede utilizar el navegador directamente, o inyectar las consultas y ver las respuestas por medio de una herramienta como Fiddler (ver el primer artículo de la serie).

Utilización del Servicio REST

Un Servicio REST puede ser utilizado desde múltiple tipos de sistemas y equipos. Como ejemplos, los siguientes párrafos indican como crear clientes que utilizan código manejado y JavaScript para llamar al Servicio.

Aplicación de Consola que utiliza código manejado

9. Cree un nuevo proyecto de Visual Studio del tipo “Escritorio de Windows” – “Aplicación de consola”. Asígnele un nombre (“SpRest01_ClienteConsola” en el ejemplo).

Nota: no cree el nuevo proyecto dentro de la misma solución de Visual Studio en donde se encuentra el Servicio REST pues dificulta considerablemente el trabajo de depuración. Cree una solución separada para cada proyecto y, cuando se necesite depurar, lance cada solución por separado desde diferentes instancias de Visual Studio.

10. Agregue una referencia a System.Net y su correspondiente directiva using.
11. Modifique la clase Main para que incluya el siguiente código fuente:

```
static void Main(string[] args)
{
    string restURL = "http://localhost:33497/api/elementos";

    HttpWebRequest myRequest = (HttpWebRequest)
    HttpWebRequest.Create(restURL);
    myRequest.Credentials = CredentialCache.DefaultCredentials;
    myRequest.Method = "GET";
    myRequest.Accept = "application/xml";
    Console.WriteLine("URL - " + restURL);

    HttpResponseMessage myResponse = (HttpResponse)
    myRequest.GetResponse();
    XmlDocument myDoc = XmlDocument.Load(myResponse.
    GetResponseStream());
    Console.WriteLine(myDoc);
}
```

La dirección del Servicio REST tiene que ser conocida de antelación, lo mismo que la consulta que se desea realizar (variable “restURL”). Inicialmente se crea la consulta utilizando la clase “HttpWebRequest”, se le indican las credenciales que debe utilizar, el método y el tipo de respuesta que se espera (“xml”). La consulta se envía por medio del método “GetResponse” y la respuesta se atrapa en una

instancia de la clase “HttpWebResponse”. La respuesta se puede procesar de diferentes formas. El ejemplo la convierte en un documento XML, que se puede leer utilizando los métodos tradicionales de .NET.

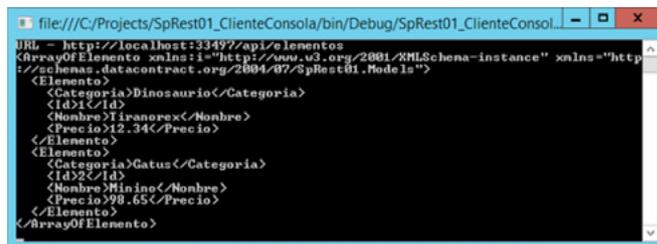


Imagen 3.- Aplicación de Consola llamando el Servicio REST.

Aplicación Web que utiliza JavaScript

12. Cree un nuevo proyecto de Visual Studio del tipo “Web” – “Aplicación web ASP.NET”, asígnele un nombre (“SpRest01_ClienteWeb” en el ejemplo) y seleccione “Empty” como plantilla.
13. Agréguele al proyecto un elemento del tipo “Página HTML” asignándole un nombre.
14. En la sección de “body” del código de la página HTML añada un botón que llama una función de JavaScript cuando el usuario lo aprieta, de la siguiente forma:

```
<body>
<input id="btnLlamarServicio" type="button" value="Llamar
Servicio" onclick="LlamarServicio0" />
</body>
```

15. Cree la función de JavaScript en la sección “head” del código de la página:

```
<script src="http://ajax.aspnetcdn.com/ajax/jquery/jquery-
1.9.0.js"></script>
<script language="javascript">

function LlamarServicio0 {
    jQuery(function () {
        jQuery.ajax({
            url: "http://localhost:33497/api/elementos",
            type: "GET",
            headers: {
                "accept": "application/json",
            },
            success: function (data, status, jqXHR) {
                var myResultado;
                $.each(data, function (key, value) {
                    $.each(value, function (llave, valor) {
                        myResultado += " - " + valor;
                    });
                });
                alert(myResultado);
            },
            error: function (jqXHR, status, message) {
                alert(JSON.stringify(jqXHR));
            }
        });
    });
}
</script>
```

El primer script simplemente agrega el código de jQuery que se va a utilizar posteriormente. La función “LlamarServicio” tiene tres secciones principales:

- La cabecera, en donde se define el URL del Servicio, junto con la consulta que se va a realizar.
- La sección que ejecuta cuando el Servicio devuelve una respuesta sin problemas (“success”). Los datos se encuentran en la variable “data” en forma de un objeto JSON. Por medio de dos bucles “each” se extrae la información y se muestra al usuario por medio de una Alerta de JavaScript
- Si el Servicio devuelve un error, la sección de “error” comienza a funcionar, mostrando el mensaje que el Servicio ha enviado por medio de otra Alerta

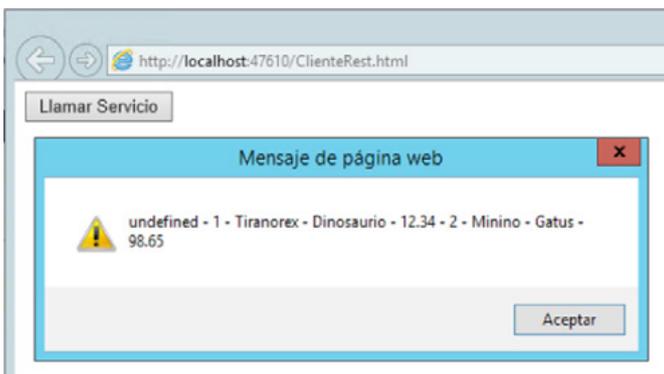


Imagen 4.- Aplicación Web (HTML) llamando el Servicio REST.

Un Servicio REST puede ser utilizado desde múltiple tipos de sistemas y equipos

Conclusión

SharePoint 2013 y SharePoint Online en Office 365 utilizan extensivamente REST como medio de comunicación para manipular información. La creación de Servicios REST se ha simplificado considerablemente desde la aparición del Framework WebAPI de Microsoft, que se integra perfectamente en Visual Studio 2013 y permite la creación de servicios REST rápida e intuitivamente. Desafortunadamente no es posible integrar Servicios REST personalizados con los Servicios nativos de SharePoint 2013, pero eso no impide que se puedan crear (y utilizar) por separado, ya sea usando código del Modelo de Objetos de Servidor, o del Modelo de Objetos de Cliente.

GUSTAVO VELEZ
MVP SharePoint
gustavo@gavd.net
<http://www.gavd.net>

KWizCom Forms

Formularios Nativos para SharePoint y Soluciones Móviles

Cree fácilmente formularios personalizados utilizando los formularios de las listas existentes. Sin necesidad de despliegue ni aprender a utilizar nuevas herramientas!



www.kwizcom.com/forms

Introducción a Search Driven Development en SharePoint Online y SharePoint 2013

El presente artículo muestra un ejemplo de cómo aprovechar algunas de las funcionalidades que ofrecen las búsquedas para implementar un requisito habitual en muchos de nuestros sitios, tanto escenarios Intranet como de internet. Se va a crear una vista de un catálogo y detalles de elementos de ese catálogo, como podría ser por ejemplo un catálogo de noticias y el detalle de cada una de ellas, y lo haremos además usando URLs amigables. Los conceptos del presente artículo pueden aplicarse en escenarios de BCS o Cross Site Publishing. Con el objetivo de no hacer el artículo demasiado largo y centrarnos en el motivo del mismo, se desarrollará todo en el mismo sitio y con una lista personalizada que se creará para tal fin.

una de las mejoras más importantes que incorporó la versión 2013 de SharePoint tiene que ver con las nuevas funcionalidades relacionadas con las búsquedas

Preparación de nuestro sitio

En primer lugar se creará la lista personalizada que contendrá el catálogo, que recibirá el nombre de “Noticias”. Además, se habilitará la gestión de tipos de contenido para esta lista y se creará un “tipo de contenido” para la misma que represente a cada elemento de nuestro catálogo y que será clave en este ejemplo. El tipo de contenido se llamará Noticias y puede tener una estructura como la que vemos en la Imagen 1.



Imagen 1.- Tipo de contenido Noticias.

Creación del origen de resultados

Uno de los aspectos más importantes para lograr nuestro fin es la definición de un “Origen de resultados”. Esta opción permite crear búsquedas limitadas a ciertos contenidos o a un subconjunto de resultados de búsqueda. En nuestro caso, vamos a definir un origen de resultados que nos devuelva las “Noticias” a partir del Tipo de Contenido que hemos definido en el apartado anterior. El Origen de resultados se puede definir a distintos niveles: a nivel de administración del servicio de búsqueda, a nivel de colección de sitios, o bien, a nivel de sitio. Para el ejemplo que se está mostrando, vamos a definir el “origen de resultados” al nivel del sitio en el que se está trabajando. Desde la configuración del sitio, vamos al apartado de búsqueda y fuentes de resultado, y desde ahí a “Nuevo origen de resultado”, Imagen 2.

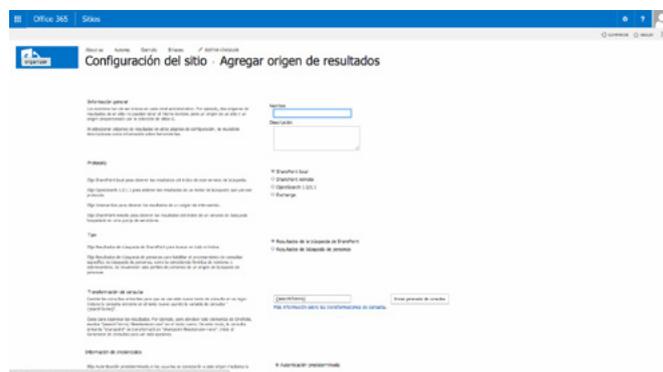


Imagen 2.- Creación de un origen de resultados.

Tras indicar el nombre y la descripción, se señala como protocolo “SharePoint local” y el tipo de búsqueda como “Resultados de la búsqueda de SharePoint” para seleccionar los elementos de todo el índice. Para configurar la consulta se entra en “Iniciar generador de consultas”.

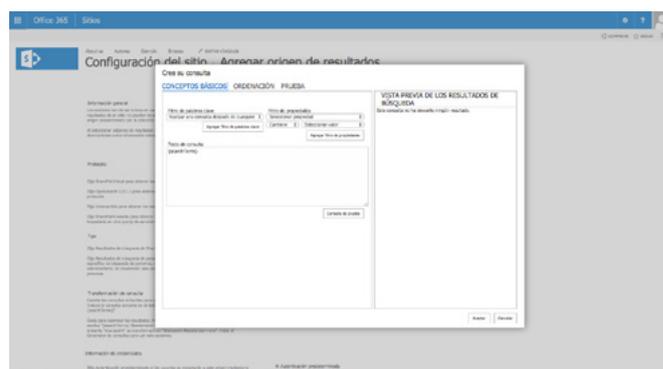


Imagen 3.- Pantalla de configuración de la consulta.

La configuración de nuestra consulta se realiza en la pantalla que aparece, como puede apreciarse en la Imagen 3. Los filtros de la consulta pueden definirse mediante filtros de palabras clave o de propiedades, donde se pueden establecer filtros a partir de las propiedades administradas que hay identificadas en nuestro sistema. Para este caso, se va a añadir un filtro de propiedad en el que se indicará que los elementos devueltos sean del tipo de contenido creado en la sección anterior, tipo de contenido “Noticias”, como vemos en la Imagen 4.

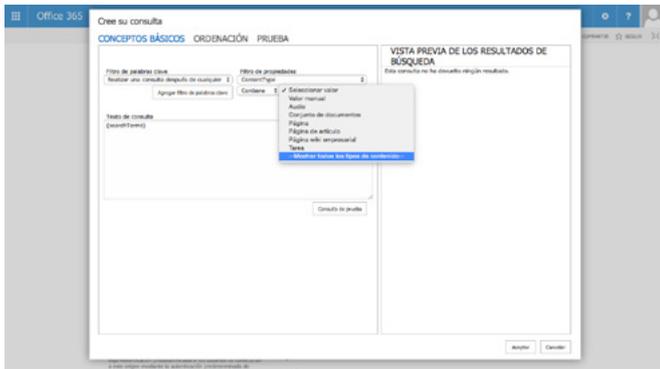


Imagen 4.- Añadiendo un filtro para el tipo de contenido del resultado,

Una vez añadido el filtro, se puede efectuar la comprobación de que los resultados son los correctos en la opción «Consulta de prueba». Para comprobar que la consulta configurada devuelve los resultados correctos, es necesario asegurar que se ha re-indexado la lista después de haber insertado los datos, ya que, de otro modo, al usar la opción de “Consulta de prueba”, podrían no obtenerse los resultados esperados. Una vez comprobado, se guarda la fuente de resultados que será usada más adelante en el Content Search WebPart.

Los filtros de la consulta pueden definirse mediante filtros de palabras clave o de propiedades

Creación de las páginas

El siguiente paso será crear las páginas que se van a utilizar y crear el término de la navegación basada en metadatos para nuestro catálogo. La navegación basada en metadatos es otra de las interesantes funcionalidades que incorpora SharePoint 2013 y que permite utilizar nuestro almacén de términos para crear URLs amigables. Aunque el objetivo de este artículo no es cómo activar esta opción, si se puede mencionar que, al menos, es necesario tener activada la Feature «Infraestructura de publicación de SharePoint Server» de la colección de sitios y activar la navegación basada en metadatos del sitio para poder crear URLs amigables. A continuación, se crearán dos páginas en nuestro sitio en la librería de páginas, una para el catálogo y otra para los elementos del catálogo.

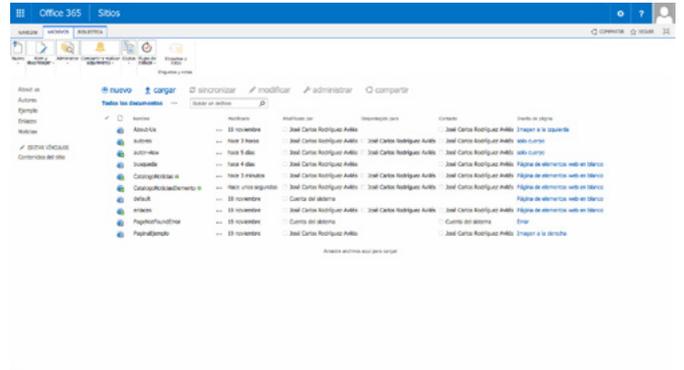


Imagen 5.- Biblioteca de páginas con las dos nuevas páginas creadas.

Después, se accede al almacén de términos de nuestro sitio y al conjunto de términos de navegación del mismo para crear un nuevo término, que llamaremos “Noticias” y que creará un nuevo elemento en el menú de navegación y una URL amigable de la forma “<site>/noticias”. Una vez creado, en la configuración de este nuevo término, nos dirigimos a la pestaña «Páginas basadas en términos» y activamos en la sección de Configuración de la página de destino las dos casillas, seleccionando para ambas la página que hemos creado para el catálogo. Por último, en la sección que aparece más abajo sobre Configuración de la página de elemento de catálogo, activamos igualmente las dos casillas y para ambas, se selecciona la página que se ha creado para los elementos de catálogo.

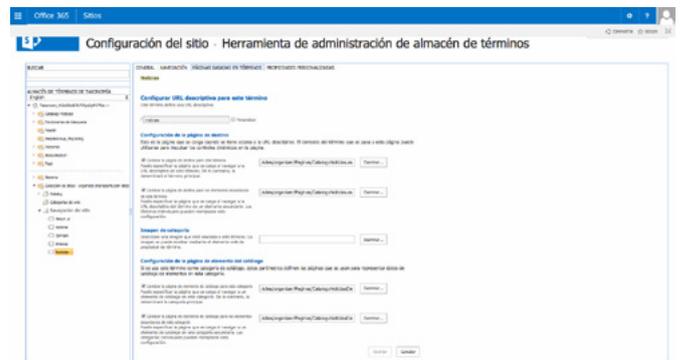


Imagen 6.- Configuración del término de navegación basada en metadatos.

Configuración del Content Search WebPart

El último paso para tener listas las páginas que se han creado, es añadir y configurar en cada una de ellas el Content Search WebPart, tanto para el catálogo como para los elementos del mismo. En primer lugar, se va a trabajar sobre la página del catálogo, editándola para añadir el WebPart. Posteriormente se accede a las propiedades y hacemos click sobre «Cambiar Consulta», lo que permitirá modificar la consulta y configurar los datos que se desea que devuelva el WebPart.

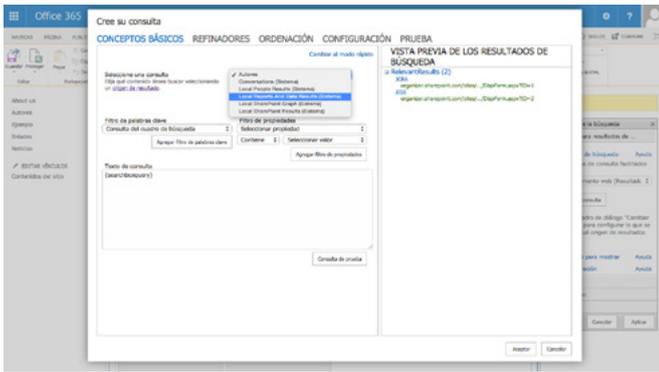


Imagen 7.- Pantalla de configuración de la consulta del Content Search WebPart.

En la pestaña de conceptos básicos, seleccionamos el Origen de resultados que había sido creado previamente, Imagen 7. En este caso, como se está editando la página del catálogo y se pretende que sea devuelto el catálogo completo, solo seleccionamos el origen de resultados, por lo que para continuar, se acepta y guarda la configuración del WebPart. El siguiente paso es modificar la página que se ha creado para los elementos del catálogo y, de la misma manera, añadir el WebPart. Se acceden a las propiedades del mismo, y de nuevo a la opción de Cambiar consulta. En este caso, una vez seleccionado el Origen de resultados se añade un filtro para que devuelva solo el elemento del catálogo al que se está accediendo.

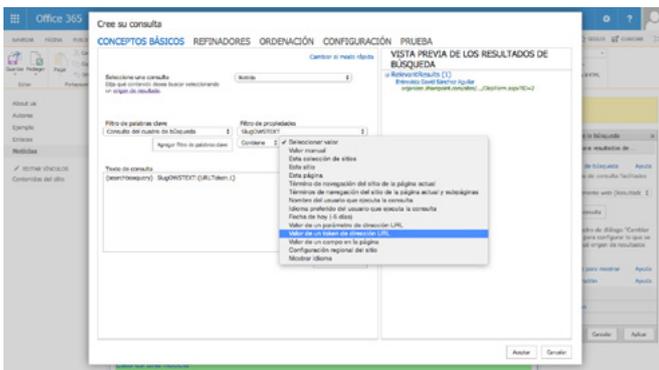


Imagen 8.- Configuración de la consulta en el Content Search WebPart.

En nuestro caso, se añade un filtro para la propiedad Slug que tenía nuestro tipo de contenido, que será lo que acompañe a la URL del catálogo para acceder a cada elemento. Posteriormente se selecciona la opción de «Valor de un token de la dirección URL», para la propiedad administrada relacionada con el campo al que se desea asociar y se

añade el filtro. Con este procedimiento, el WebPart para la página de elementos del catálogo habrá sido configurado. Desde ahora se puede acceder a una noticia de nuestro catálogo a través de la URL “/noticias/slug-de-la-noticia-de-ejemplo”.

Una vez preparadas, la página de nuestro catálogo y de detalle de ese catálogo, se puede combinar con los Display Templates para proporcionar un diseño más atractivo a los resultados que nos da el Content Search WebPart. Este tema ha sido objeto de un artículo en la revista (Número 17, Introducción a las plantillas de elementos de contenido (Display templates), Santiago Porrás Rodríguez), por lo que se puede acudir a él y, junto con lo que hemos podido ver en éste, comenzar a explotar todas las posibilidades que nos ofrecen estas features.

con las nuevas herramientas que nos proporcionan las búsquedas, se pueden generar de una forma sencilla funcionalidades de nuestros sitios

Como se ha podido comprobar, sin la necesidad de escribir código y por medio de las nuevas herramientas que nos proporcionan las búsquedas, se pueden generar de una forma muy sencilla funcionalidades muy habituales en nuestros sitios. Me gustaría concluir destacando que las búsquedas en SharePoint 2013 nos ofrecen una amplia variedad de nuevas oportunidades que nos llevarán, no solo a tenerlas en cuenta para construir un mero buscador con refinamiento, sino también a considerarlas como una opción válida a la hora de plantear la resolución de muchos de los requerimientos que se nos presentan cuando abordamos un proyecto con SharePoint 2013 o SharePoint Online.

JOSÉ CARLOS RODRÍGUEZ AVILÉS

Analista Programador en Soluciones de SharePoint

josecarlos.rodriguez@uci.com

@jcraov

<http://elblogdelprogramador.wordpress.com>





24

Entrevista Jorge Castañeda

Mi nombre es Jorge Castañeda Cano, vivo en Lima, Perú.

Tengo más de 10 años trabajando con tecnología Microsoft. Me gusta trabajar con la plataforma y los servicios de Microsoft especialmente con los servicios Cloud actualmente, su enfoque a la innovación y mejoras de los servicios brindados.

Ingresé al servicio de Office 365 por el 2012 y tuve la oportunidad realizar mi primera migración de los servicios de correo en la empresa para la que trabajaba en aquel entonces. Desde ahí quede conectado teniendo esa visión de que grandes cosas en temas Cloud se venían en camino y comencé a especializarme desde aquel entonces en el servicio y ver evolucionarlo desde ese momento es grandioso. De la misma forma, comencé a publicar en mi blog sobre los servicios de Office 365 ayudando a otros en la



plataforma colaborando sobre el uso y configuración de los servicios y también compartiendo con las comunidades técnicas. Obtuve el reconocimiento de Microsoft Most Valuable Professional (MVP) Office 365 en el año 2014 y tuve la oportunidad de recibir la renovación el pasado mes de enero del 2015

¿Por qué y cómo empezaste en el mundo de la tecnología?

Me inicié desde los 11 años de edad cuando ingresé a estudiar carreras técnicas cortas en las que, gracias a la visión y apoyo de mi padre, tuve la oportunidad de desenvolverme y aprender muchas cosas. Más adelante, cuando terminé mi secundaria decidí estudiar una carrera de sistemas, estando allí me uní a una comunidad tecnológica Microsoft de estudiantes realizando eventos y charlas, hasta la fecha lo vengo haciendo, pero ahora para la audiencia profesional.

¿Cuáles son tus principales actividades tecnológicas hoy en día?

En este mundo de tecnología siempre tienes que estar actualizado, me mantengo al día enlazando mis blogs preferidos al servicio de correo que voy revisando durante el día, comparto los temas de interés en mis redes sociales y revisar las novedades internas con los encargados del producto de Office 365. En el área profesional he montado mi empresa dedicada a brindar los servicios de Microsoft y entrenamientos, enfocando los temas Cloud Office 365, Azure e Intune.

¿Cuáles son tus principales activi-

dades NO tecnológicas hoy en día?

Pasar tiempo con mis seres queridos, salir al cine y con mis amigos.

¿Cuáles son tus hobbies?

Me gusta ver series de televisión, escuchar música del género rock y jugar al fútbol.

¿Cuál es tu visión de futuro en la tecnología de acá a los próximos años?

Las empresas perderán el temor de alojar sus servicios en Cloud, estarán apostando a tener los servicios siempre disponibles desde cualquier lugar, todos estarán conectados a sus dispositivos electrónicos. Donde se está viendo en estos últimos años y donde se estaría apuntando el internet de las cosas.

JORGE CASTAÑEDA

Microsoft MVP Office 365

Blog: <http://office365technical.support/>

Correo: jorge.castaneda@outlook.com Twitter: @xorxe

Administra mejor tus aplicaciones de servicio

En SharePoint 2010 y 2013, el modelo de arquitectura de servicios ofrece un marco en el que implementar y administrar servicios dentro de una granja o entre varias. Las llamadas aplicaciones de servicio representan a una instancia de un servicio que se puede ofrecer en SharePoint para ser consumido y que se puede configurar de forma centralizada. Las aplicaciones de servicio nos permiten ofrecer servicios en nuestra granja de SharePoint que pueden ser compartidos y consumidos por distintas aplicaciones web o incluso por distintas granjas (en infraestructuras muy grandes).

Las aplicaciones de servicio disponibles en SharePoint 2010 y 2013 por defecto son las siguientes:

Nombre de la aplicación de servicio	SP 2013	SP 2010
Servicios de Access	X	
Servicios de Access 2010	X	X
Servicio de administración de apps	X	
Equilibrio de carga y detección de aplicaciones	X	
Conectividad a datos empresariales (BCS)	X	X
Servicios de Excel	X	X
Conector a Lotus Notes	X	X
Servicio de traducción automática	X	
Servicio de metadatos administrados	X	X
Servicio de configuración de suscripción de Microsoft SharePoint Foundation	X	X
Servicios de PerformancePoint	X	X
Conversión de PowerPoint	X	X
Servicio de búsqueda	X	X
Servicio de almacenamiento seguro	X	X
Aplicación de servicio de token de seguridad	X	X
Servicio de estado	X	X
Recolección de datos de mantenimiento y uso	X	X
Servicio de perfil del usuario	X	X
Servicio de gráficos de Visio	X	X
Word Automation Services	X	X
Administración del trabajo	X	

Web Analytics		X
Aplicación de servicio de flujo de trabajo (sólo si hay un servicio externo de flujo de trabajo conectado)	X	

Además de estas aplicaciones de servicio, existen otras que pueden ser creadas por otros productos que se integran con SharePoint (Project Server, Office Web Apps 2010, SQL Reporting Services, PowerPivot...) o pertenecer a productos de terceros. Para conseguir un correcto funcionamiento, mantenimiento y escalabilidad de nuestros entornos, es necesaria una correcta configuración de nuestras aplicaciones de servicio. En este artículo repasaremos los puntos clave a tener en cuenta para organizar las aplicaciones de servicio, configurar su seguridad, organizar sus bases de datos y planear el escalado de éstas.

Planear las cuentas de seguridad

El primer paso a tener en cuenta a la hora de planear las aplicaciones de servicio son las cuentas a utilizar. Para ello nos basaremos en las cuentas recomendadas que expusimos en el nº 20 de CompartiMOSS (“Buenas prácticas en infraestructura en SharePoint 2013 - Parte II”). Las aplicaciones de servicio residen, como cualquier otra aplicación web, en el IIS del servidor. Es por ello que es necesario al menos un grupo de aplicación (application pool) para ejecutarlas. Para este grupo de aplicaciones utilizaremos una cuenta de seguridad que llamaremos spservices.

El servicio de búsquedas requiere varias cuentas para funcionar correctamente. Es necesaria una cuenta de acceso al contenido, que tendrás permisos de lectura en todo el contenido y será utilizada por el rastreador (crawler) de las búsquedas. La llamaremos spcrawl. También es necesaria una cuenta independiente para la administración de búsqueda, que se asignará a todas las aplicaciones de servicio de búsqueda de la granja. Además, se recomienda independizar el grupo de aplicación (application pool) del servicio de búsquedas y disponer de una cuenta para este servicio. Microsoft especifica que puede ser la misma cuenta de administración u otra cuenta distinta. Nosotros utilizaremos la misma: spsearch.

El servicio de sincronización de perfiles necesita una cuenta para acceder al Active Directory y obtener datos de los usuarios. Debe tener permisos de “Replicating Directory

Changes” en el AD. La llamaremos spprofile.

Los servicios de Excel, Visio y PerformancePoint necesitan utilizar una cuenta en nombre de usuarios autorizados para proporcionar acceso a orígenes de datos externos. Estas cuentas, llamadas cuentas “desatendidas” se configuran en el servicio de almacenamiento seguro. Las llamaremos spvisio, spexcel y spperfpoin.

Planear los grupos de aplicación (application pools)

Como se ha mencionado en el apartado anterior, las aplicaciones de servicio se deben distribuir en uno o más grupos de aplicación (application pool). Microsoft indica en TechNet que se puede implementar aplicaciones de servicio en distintos grupos de aplicaciones para aislar procesos. Sin embargo, también se indica que el rendimiento de una granja se optimiza si todos los servicios están dentro de un mismo grupo de aplicaciones. Aun así, se recomienda la ejecución del servicio de búsqueda en un grupo de aplicación independiente por razones de escalabilidad y de seguridad. También se recomienda ejecutar los servicios de token de seguridad y de equilibrio de carga con las credenciales del administrador de granja y cuenta de acceso a base de datos (spfarm).

Buscando la convergencia entre la recomendación de juntar servicios en un mismo grupo de aplicación y estas configuraciones especiales, el resultado sería la creación de cuatro grupos de aplicación en el IIS para las aplicaciones de servicio de SharePoint:

- SecurityTokenServiceApplicationPool [cuenta: spfarm].
 - Servicio de token de seguridad.
- SharePoint Hosted Services [cuenta: spservices].
 - Servicio de administración de apps.
 - Conectividad a datos empresariales (BCS).
 - Servicios de Excel.
 - Servicio de traducción automática.
 - Servicio de metadatos administrados.
 - Servicios de PerformancePoint.
 - Conversión de PowerPoint.
 - Servicio de almacenamiento seguro.
 - Servicio de configuración de suscripción.
 - Servicio de perfil del usuario.
 - Servicio de gráficos de Visio.
 - Word Automation Services.
- SharePoint Search Application Pool [cuenta: sp-search].
 - Aplicación de servicio de búsqueda.

- Search Administration Web Service for Search Service Application.
- SharePoint Web Services System [cuenta: spfarm].
 - Equilibrio de carga y detección de aplicaciones.

Nota: se ha utilizado la nomenclatura usada en muchos artículos y en proyectos como AutoSPInstaller.

Planear las bases de datos

Hay determinadas aplicaciones de servicio que requieren el uso de una o varias bases de datos. Es muy recomendable tener una nomenclatura de nombres muy clara a la hora de nombrar estas bases de datos, ya que ayudará a la gestión y al mantenimiento, permitiéndonos identificarlas rápidamente. La recomendación es utilizar un prefijo único que las identifique como bases de datos de servicio (para verlas todas juntas en SQL Server) seguida de un nombre descriptivo. Para el artículo, indicamos los nombres más utilizados por la comunidad y usados en proyectos como AutoSPInstaller:

Aplicación de servicio	Nombre de BD recomendado
Servicio de administración de apps	Services_AppManagement
Conectividad a datos empresariales (BCS)	Services_BusinessDataCatalog
Servicio de traducción automática	Services_TransaltionService
Servicio de metadatos administrados	Services_MetaData
Servicio de configuración de suscripción	Services_SubscriptionSettings
Servicios de PerformancePoint	Services_PerformancePoint
Servicio de búsqueda	Base de datos de administración de búsquedas: <ul style="list-style-type: none"> - Services_Search Base de datos de informes de Analytics <ul style="list-style-type: none"> - Services_AnalyticsReportingStore Base de datos de rastreo <ul style="list-style-type: none"> - Services_CrawlStore Base de datos de vínculo <ul style="list-style-type: none"> - Services_LinksStore
Servicio de almacenamiento seguro	Services_SecureStore
Servicio de estado	Services_StateService
Recolección de datos de mantenimiento y uso	Services_UsageAndHealth

Servicio de perfil del usuario	<p>Base de datos de información sobre los perfiles:</p> <ul style="list-style-type: none"> - Services_Profile <p>Base de datos de etiquetas sociales y notas de los usuarios:</p> <ul style="list-style-type: none"> - Services_Social <p>Base de datos con información provisional para la sincronización de perfiles:</p> <ul style="list-style-type: none"> - Services_Sync
Word Automation Services	Services_WordAutomation

Los permisos de las bases de datos se asignan al crear la aplicación de servicio o al actualizar la seguridad desde la Administración Central. Es recomendable no editarlos desde SQL Server para evitar problemas.

Creación de las aplicaciones de servicio

Existen tres formas de crear las aplicaciones de servicio:

- Todas a la vez mediante el asistente de configuración de la granja.
- De una en una desde la página de administración de aplicaciones de servicio (Administración Central de SharePoint).
- De una en una mediante Windows PowerShell.

El Asistente de Configuración de la Granja permite crear todas las aplicaciones de servicio a la vez y si tener que preocuparse por nombres, seguridad ni otras configuraciones (Imagen 1).

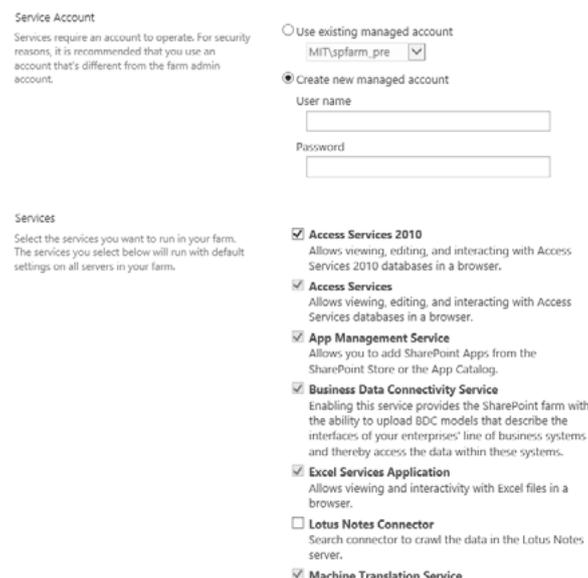


Imagen 1.- El asistente de configuración de la granja permite crear aplicaciones de servicio.

Precisamente por esta facilidad es por lo que no se recomienda el uso del asistente, no permite tener el control de la configuración y eso genera multitud de inconvenientes tales como:

- Todas las aplicaciones de servicio se implementan en el mismo grupo de aplicaciones (application pool).
- Se usa una única cuenta para todos los servicios. Se puede cambiar más adelante, pero es complicado.
- Los nombres de las bases de datos se generan automáticamente, con GUIDs largos e incómodos de recordar. No es sencillo identificarlas rápidamente al administrar SQL Server.
- No nos permite tener una convención de nomenclatura para las aplicaciones de servicio y para sus bases de datos.
- Se aplica la configuración predeterminada en cada aplicación de servicio. Se puede cambiar más adelante.

En cambio, la creación manual (mediante Administración Central o PowerShell) permite tener el control sobre nombres, conexiones, seguridad y configuración (Imagen 2). Una vez se hayan decidido la nomenclatura de nombres, las cuentas y grupos de aplicación, el servidor de base de datos y los nombres de bases de datos, es muy sencillo crear una aplicación de servicio. En la Imagen 2 puede verse que los datos para la aplicación de servicio de administración de apps son estos datos de los que hemos estado hablando.

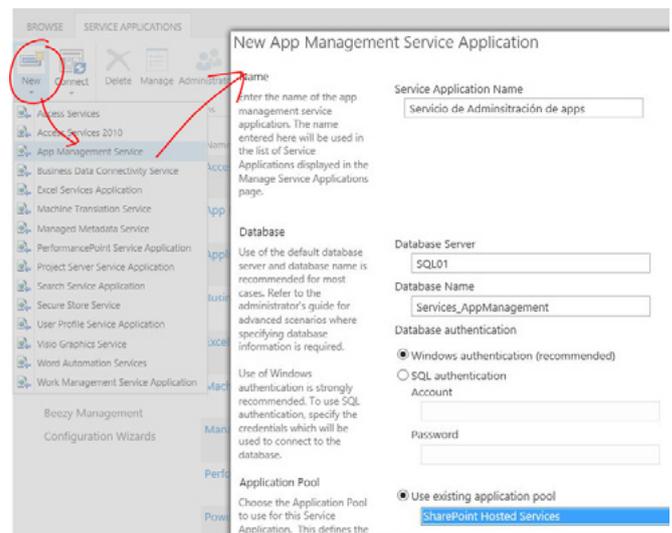


Imagen 2.- La creación manual de una aplicación de servicio nos permite tener el control.

Además de estos parámetros, algunas aplicaciones de servicio requieren configuración adicional. Por ejemplo, los servicios de perfiles y de búsqueda necesitan varias bases de datos, el servicio de perfiles necesita varios parámetros para configurar Mi Sitio, el servicio de almacenamiento seguro tiene opciones de auditoría, etc. Para las opciones específicas de cada servicio, se puede consultar esta guía de TechNet: <https://technet.microsoft.com/es-es/library/ee794878.aspx>

Seguridad

Una vez se han creado las aplicaciones de servicio, es necesario configurar una serie de parámetros de seguridad para asegurarnos que los servicios, usuarios o aplicaciones

que las consuman no tengan problemas de acceso. Para explicar este punto, utilizaremos las cuentas mencionadas en este artículo, a las que le añadiremos:

- spfarm: cuenta administrador de granja y de conexión a base de datos. Se impersonal como Cuenta del Sistema y es cuenta del grupo de aplicación (application pool) de la Administración Central.
- spsetup: cuenta de instalación, configurador de SharePoint y PowerShell.
- sppool1, sppool2, sppool3...: cuentas del grupo de aplicación (application pool) de cada aplicación de contenido de SharePoint.
- spmysite: cuenta del grupo de aplicación (application pool) de la aplicación de Mi Sitio.

Por un lado, se pueden configurar uno o varios administradores en cada aplicación de servicio y establecerles diferentes niveles de permiso, tal y como se ve en la Imagen 3.

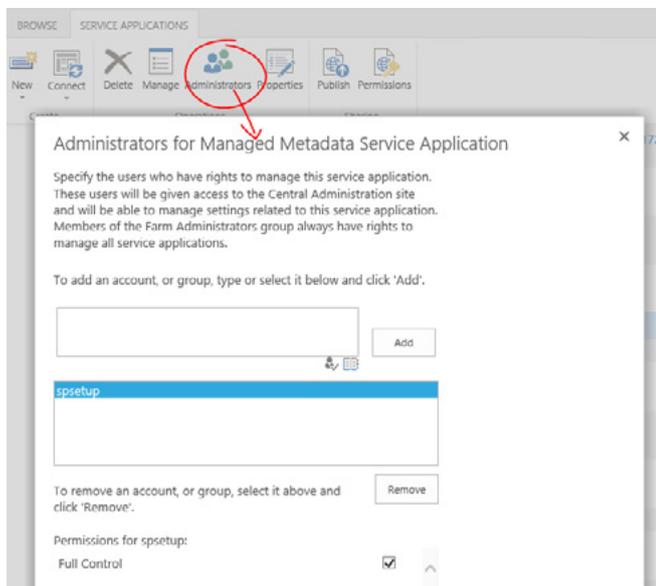


Imagen 3.- Es posible asignar administradores a las aplicaciones de servicio.

De forma predeterminada, los miembros del grupo de administradores de la granja de servidores tienen permisos para administrar las aplicaciones de servicio. Aun así, es necesario hacer algún ajuste adicional en las siguientes:

- Servicio de metadatos administrados:
 - spsetup: full control.
- Servicio de perfiles de usuario:
 - spsetup: full control.
 - spmysite: full control.
 - sppool1, sppool2, sppool3...: full control.
 - sprawl: retrieve people data for search crawlers.
 - spfarm: full control.

De esta forma, las aplicaciones de contenido no tendrán problemas para acceder al servicio de perfiles y la cuenta de rastreo de búsquedas podrá rastrear personas. También evitaremos problemas accediendo desde PowerShell al servicio de metadatos. Si se desea administrar alguna

de las aplicaciones de servicio con otra cuenta, es posible asignarles permisos desde aquí. Esta asignación provocará que se le otorguen permisos internamente en las bases de datos de las aplicaciones de servicio.

Por otra parte, existe la posibilidad de compartir las aplicaciones de servicio con las cuentas de usuario que se considere necesarias, como se muestra en la Imagen 4. El objetivo de esta funcionalidad es poder compartir determinadas aplicaciones de servicio con otras granjas, aunque también es necesario para que algunas funcionalidades y servicios de la granja puedan utilizar alguna de las aplicaciones de servicio.

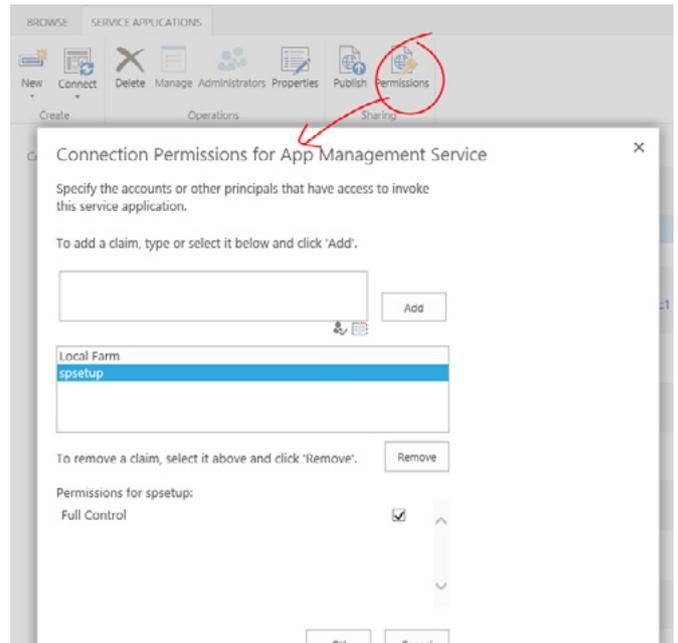


Imagen 4.- Podemos compartir las aplicaciones de servicio con otras cuentas.

Por ejemplo, la navegación en los sitios de publicación necesita acceder internamente al servicio de perfiles para obtener información de audiencias. Y la cuenta del grupo de aplicación (sppooln) de una aplicación web necesita acceso de lectura al almacén de términos del servicio de metadatos para, entre otras cosas, proporcionar la funcionalidad de "I like it" en SP 2010 y de los tag sociales. Por razones como estas (y muchas otras), se recomienda configurar los siguientes permisos:

- Todas las aplicaciones de servicio:
 - Local farm: full control (configurado por defecto).
- Servicio de metadatos administrados:
 - Local farm: Full Access to Term Store.
 - All Authenticated Users: Read Access to Term Store.
 - spsetup: Full Access to Term Store.
- Servicio de perfiles de usuario:
 - spsetup: full control.
 - spmysite: full control.
 - sppool: full control.
 - spfarm: full control.

Nótese que se otorga acceso de lectura al almacén de términos a todas las cuentas. Esto permitirá que cualquier usuario pueda utilizar y crear términos si así se desea y evitará problemas de acceso al almacén de términos desde los sitios. Finalmente, es necesario otorgar ciertos permisos en las aplicaciones de contenido mediante políticas para la correcta interacción con las aplicaciones de servicio.

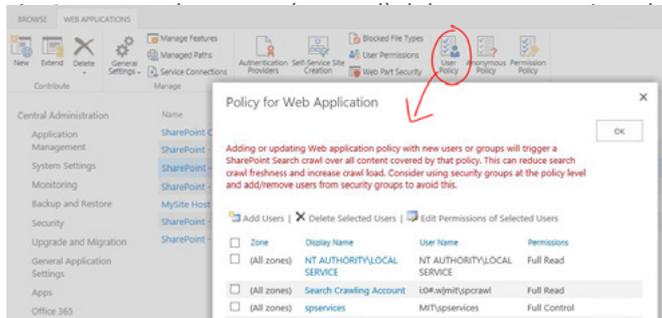


Imagen 5.- Podemos configurar políticas para la correcta interacción con las aplicaciones de servicio.

Se recuerda que es necesario configurar la cuenta spcrawl como cuenta de rastreo en la configuración del servicio de búsquedas.

Una vez se han creado las aplicaciones de servicio, es necesario configurar una serie de parámetros de seguridad para asegurarnos que los servicios, usuarios o aplicaciones

Configurar las cuentas desatendidas

Las aplicaciones de servicio de Excel, Visio y PerformancePoint necesitan utilizar una cuenta en nombre de usuarios autorizados para proporcionar acceso a orígenes de datos externos. Estas cuentas, llamadas cuentas “desatendidas” se configuran en el servicio de almacenamiento seguro. Las cuentas tendrán acceso a los orígenes de datos que

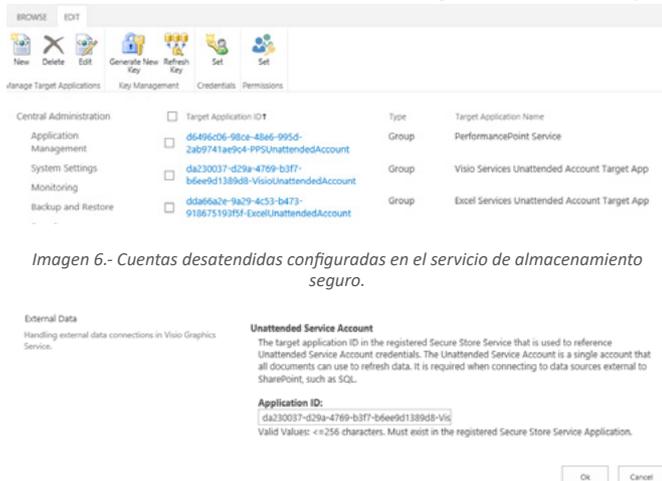


Imagen 6.- Cuentas desatendidas configuradas en el servicio de almacenamiento seguro.

Imagen 7.- Configuración de cuenta desatendida en los servicios de Visio.

Asociar servicios

Tras haber creado las aplicaciones de servicio y configurado su seguridad, es posible asignarlas a las aplicaciones de contenido (en la Administración Central) para que éstas puedan usarlas. El botón de Conexiones de Servicio permite realizar esta asociación. Si se mantiene el combo en el grupo “default”, todas las predeterminadas se asignarán. Si se cambia a un valor personalizado, podremos elegir las aplicaciones de servicio que necesitamos, tal y como se muestra en la Imagen 8.

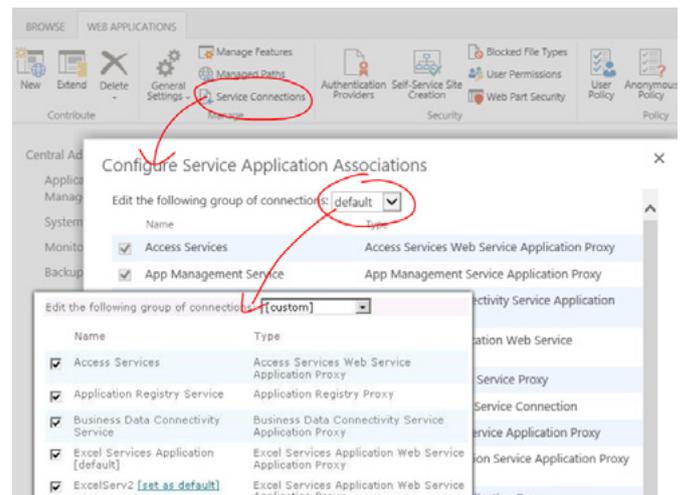


Imagen 8.- Conexión de una aplicación de contenido a las aplicaciones de servicio.

Es posible incluso tener más de una aplicación de servicio de un tipo. En la Imagen 8 se muestran dos aplicaciones de servicio de Excel y se permite decidir cuál es la predeterminada.

Bibliografía recomendada

Si se desea profundizar más en la gestión y configuración de las aplicaciones de servicio, recomiendo los siguientes enlaces:

- Configuración de servicios y aplicaciones de servicio:
 - <https://technet.microsoft.com/es-es/library/ee794878.aspx>
- Servicios de SharePoint 2013 (diagrama técnico muy recomendado):
 - <http://go.microsoft.com/fwlink/?LinkId=259425>
- Aplicaciones de servicio en SharePoint 2013: Introducción y funcionalidad (Gustavo Vélez, Juan Carlos González, Fabián Imaz):
 - <https://msdn.microsoft.com/es-es/library/dn195886.aspx>

MIGUEL TABERA PACHECO

SharePoint Lead en Plain Concepts

MVP de SharePoint Server

miguel.tabera@outlook.com

@migueltabera

Introducción a los Grupos de Office 365 – Trabaja como una red (Parte II)

La funcionalidad de Grupos de Office 365 facilita el trabajo colaborativo entre personas de una organización dotándoles de funcionalidad de valor añadido como muro conversacional, experiencia de usuario completamente integrada en Outlook Web App (OWA) en Office 365 y capacidades adicionales como calendario de grupo, sitio de SharePoint Online dónde ir dejando los documentos del grupo, etc. Adicionalmente, los Grupos de Office 365 cuentan con opciones adicionales de configuración como se verá en el presente artículo.

Opciones de configuración para Grupos de Office 365

Desde el punto de vista de configuración, un Grupo de Office 365 se puede manejar de tres formas diferenciadas:

- Desde la interfaz de usuario del Grupo en OWA.
- Desde la sección Grupos del portal de Administración de Office 365, ya que un Grupo de Office 365 viene a ser un tipo especial de grupo de directorio activo definido bajo el concepto de “Colaboración en grupo”.
- Desde las opciones de administración del servicio de Azure AD asociado a Office 365 que nos permite administrar el Directorio Activo vinculado a cada tenant de Office 365.
- La interfaz de línea de comandos proporcionada por PowerShell.

Posibilidades de configuración en el propio Grupo

Desde el acceso a un Grupo en OWA, un usuario administrador del mismo puede acceder a todas las opciones de configuración disponibles como por ejemplo:

- Acceder al listado de miembros suscritos al Grupo así como a la información de los administradores del Grupo. Desde la página de miembros del Grupo se pueden añadir nuevos miembros.
- Ver los detalles del Grupo y editarlo.

Un Grupo se define como un espacio de trabajo compartido para e-mail, conversaciones, archivos y eventos

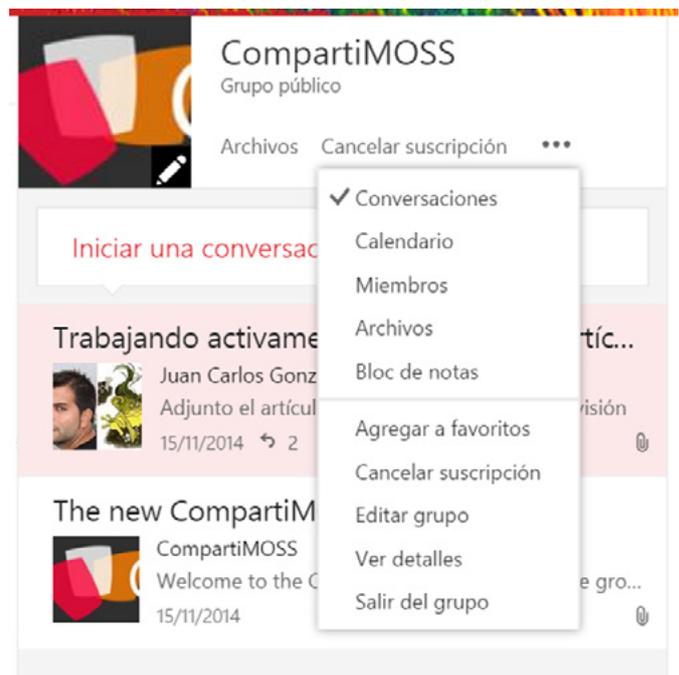


Imagen 1.- Acceso a las opciones de configuración del Grupo.

Si vemos estas opciones en detalle:

- Al editar el Grupo, tendremos la posibilidad de cambiar toda la información que originalmente se indicó en su creación: nombre, descripción, idioma para el grupo, o el idioma del grupo. En cambio, no podremos cambiar el estado del grupo de “Publico” a “Privado”.
- La página de “Miembros” nos muestra los integrantes actuales del Grupo a la vez que nos permite realizar operaciones de administración típicas como añadir nuevos miembros al Grupo, eliminar un miembro del Grupo o hacer a un miembro existente Administrador del Grupo.

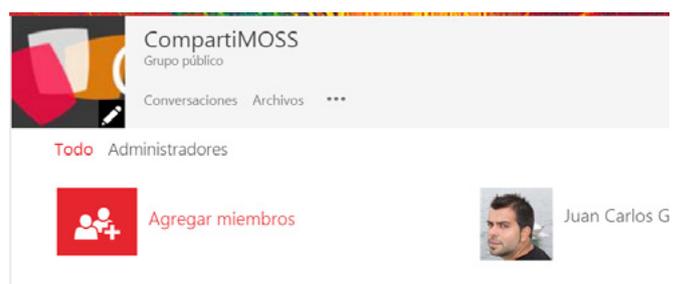


Imagen 2.- Página de detalle de los miembros del Grupo y operaciones disponibles para un miembro.

Posibilidades de configuración a través del portal de Administración de Office 365

Las mismas opciones que se han visto en la sección anterior están disponibles a través del portal de Administración de Office 365. En concreto, la opción “GRUPOS” del menú vertical del portal da acceso a todos los Grupos que se han creado en el tenant, lo que incluye los Grupos relativos a colaboración en grupo objeto del presente artículo:

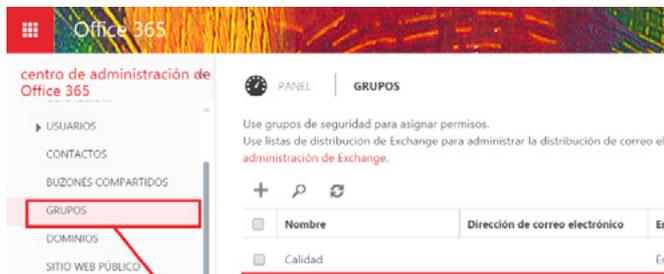


Imagen 3.- Detalle de un Grupo desde el portal de Administración de Office 365.

La página de “Miembros” nos muestra los integrantes actuales del Grupo

Como se puede apreciar en la Imagen 3, al seleccionar un Grupo definido para colaboración en grupo se habilita un panel desde el que se pueden hacer las mismas operaciones comentadas:

- Editar el nombre y descripción del Grupo.
- Acceder a la información de los miembros del Grupo, así como añadir nuevos miembros, actualizar la información de miembros existentes (incluyendo su rol) o bien eliminar miembros del grupo.

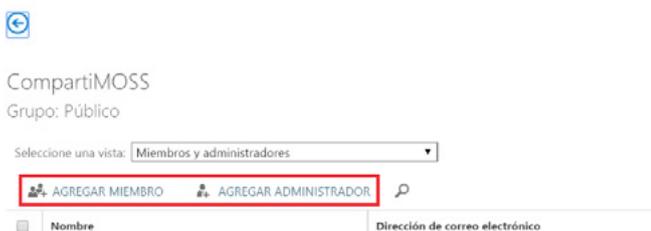


Imagen 4.- Listado de los miembros del Grupo desde el portal de Administración de Office 365.

Posibilidades de configuración a través del portal de Microsoft Azure

Finalmente, la última posibilidad para administrar y configurar Grupos de Office 365 haciendo uso de la interfaz de usuario viene dada por el portal de Microsoft Azure ya que desde el pasado mes de diciembre es posible administrar el directorio activo (DA) de Azure asociado a un tenant de Office 365 haciendo uso del portal de Microsoft Azure:

- Para acceder al DA de Azure asociado al tenant,

sólo hay que hacer clic en el enlace Azure AD del menú vertical del portal de Administración de Office 365 de forma que a continuación se muestra el portal de Microsoft Azure que permite administrar la suscripción correspondiente (y gratuita) vinculada al tenant de Office 365 y desde dónde se puede gestionar el servicio de Azure AD del tenant.



Imagen 5.- Detalle del servicio de Azure AD en el portal de Microsoft Azure.

- Como se aprecia en la Imagen 5, la suscripción de Microsoft Azure asociada al tenant de Office 365 únicamente cuenta con el servicio de Azure AD en el que por defecto se encuentra definido el DA del tenant de Office 365 y dónde es posible agregar Directorios Activos adicionales en caso de que sea necesario.
- Si se hace clic en el AD asociado al tenant de Office 365, de nuevo se puede acceder (como en los apartados anteriores) a la información relativa a los Grupos creados en el tenant.

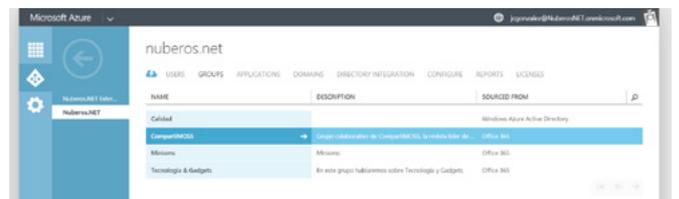


Imagen 6.- Detalle de los Grupos de Office 365 en Azure AD.

- En el caso de Azure AD, dado un Grupo definido para trabajo colaborativo no es posible realizar los mismos tipos de configuración vistos hasta ahora: editar / eliminar miembros existentes y añadir nuevos miembro. Sólo es posible visualizar la información del Grupo y la de los miembros que forman parte del Grupo.

Office 365 permite manejar los Grupos haciendo uso de PowerShell y los comandos por defecto definidos por Microsoft

Posibilidades de configuración a través de PowerShell

Como alternativa a la gestión y configuración de Grupos con la interfaz de usuario, Office 365 permite manejar los Grupos haciendo uso de PowerShell y los comandos por defecto definidos por Microsoft que se pueden encontrar

en el siguiente artículo de soporte: <https://support.office.com/es-es/article/Usar-PowerShell-para-administrar-grupos-aeb669aa-1770-4537-9de2-a82ac11b0540>. Además de estos comandos específicos para Grupos, es posible utilizar comandos estándar PowerShell para Office 365 para trabajar con cualquier tipo de grupo como los comandos Get-MsolGroup y Get-MsolGroupMember. El siguiente script PowerShell muestra cómo obtener todos los Grupos de un tenant de Office 365 y los miembros de un Grupo particular del tenant:

```
#Conexión a Office 365
$UserName="<Usuario_Office365>"
$message="Introduce your Office 365 Credentials"
$msolcred = get-credential -UserName $UserName -Message $message
connect-msolservice -credential $msolcred
#Comandos PowerShell específicos para trabajar con Grupos de Office 365
Get-MsolGroup
$gCompartimosGroup=Get-MsolGroup | Where {$_.DisplayName -eq "Compartimos"}
Get-MsolGroupMember -GroupObjectId $gCompartimosGroup.ObjectId | Format-Table DisplayName,EmailAddress
```

La salida por pantalla correspondiente es la que se muestra en la Imagen 7.



Imagen 7.- Grupos y miembros de un grupo obtenidos con los comandos Get-MsolGroup y Get-MsolGroupMember.

Conclusiones

Además de las posibilidades de colaboración que los Grupos añaden a una organización, es posible administrarlos y configurarlos de diversas formas haciendo uso de la interfaz de usuario o de PowerShell. A día de hoy, las posibilidades de gestión que brinda la interfaz de usuario son mucho más sencillas que las que proporciona PowerShell.

JUAN CARLOS GONZÁLEZ MARTÍN

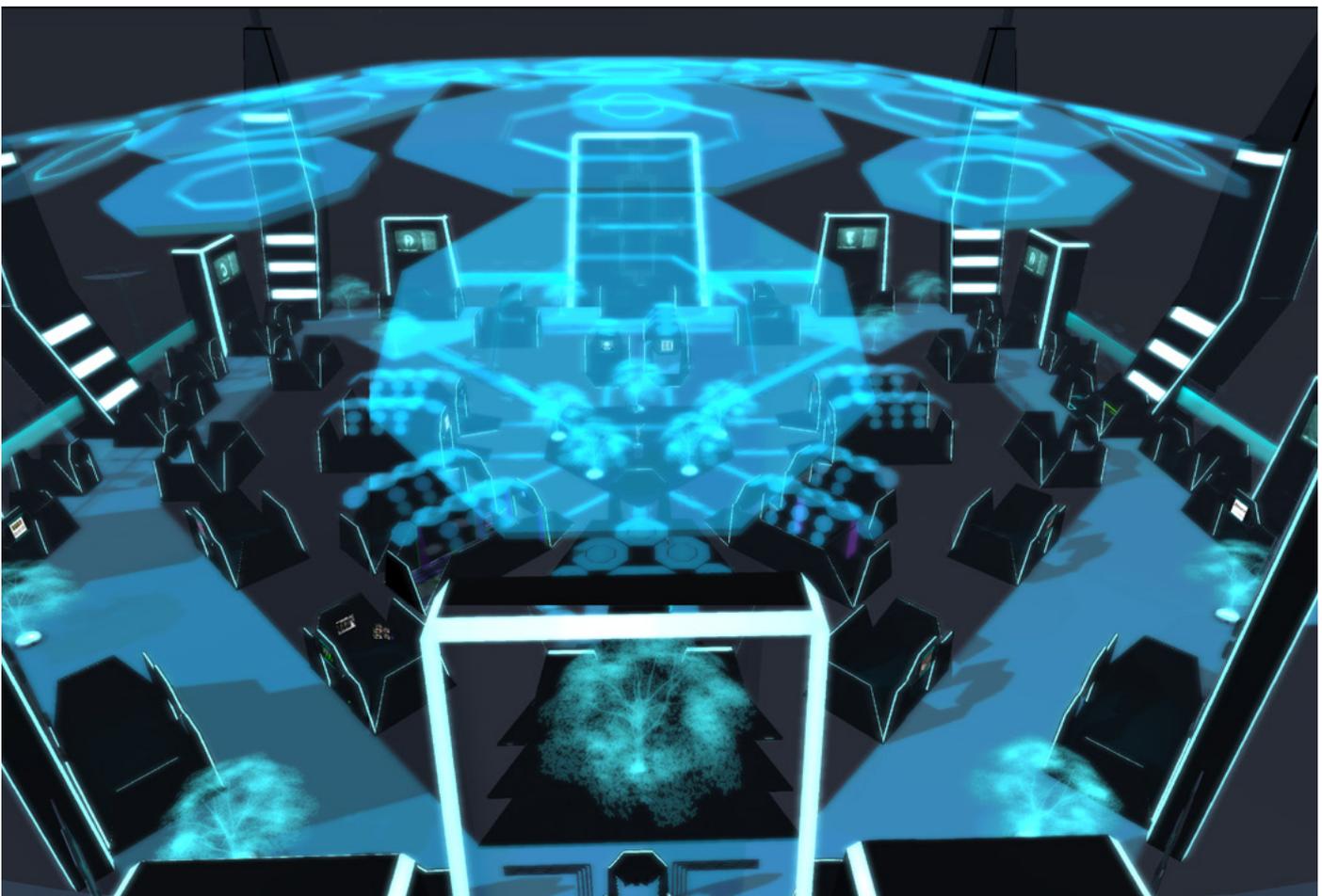
MVP SharePoint de SharePoint Server | Cloud & Productivity Advisor en MVP CLUSTER

E-Mail de Contacto: juancarlos.gonzalez@fiveshareit.es

Twitter: @jcg1978

Blog: <https://jcgonzalezmartin.wordpress.com/>

Web: www.lks.es



Mentoring

Comparti MOSS

Un servicio experto alrededor de su SharePoint



CompartiMOSS le puede ayudar a través de su programa de Mentoring!

Contacte con nosotros y le enviaremos los planes de mentoring que tenemos disponibles para SharePoint.



Búsqueda de datos empresariales en SharePoint 2013 – Parte I

En muchas ocasiones necesitamos presentar en nuestro sitio de búsqueda datos empresariales de otros orígenes (fuera de SharePoint) como pueden ser base de datos (SQL Server, Oracle, MySQL), ERP (Dynamics, SAP, ECI, NetSuite), CRM (Dinamics, SAP,Sugar), etc.

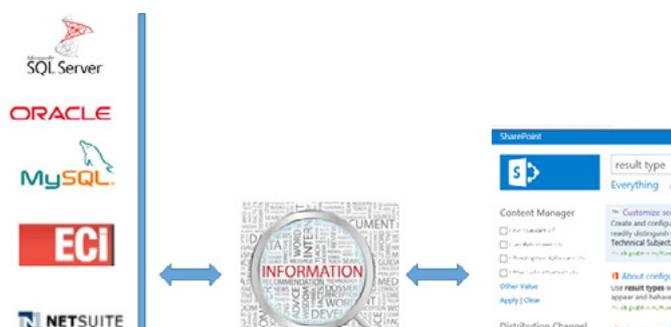


Imagen 1.- Muestra un gráfico ilustrativo para presentar datos de orígenes externos en SharePoint.

Esto es posible mediante la búsqueda de datos externos utilizando los servicios de conectividad empresarial, a continuación vamos a ver paso a paso como realizar dicha integración utilizado como conector un servicio web.

Paso 0 – Requisitos necesarios.

Para nuestro ejemplo es necesario disponer de las siguientes aplicaciones de servicio creadas en nuestro servidor.

- **Búsqueda.** Para más información de cómo crear dicho aplicación [http://technet.microsoft.com/es-es/library/gg502597\(v=office.15\).aspx](http://technet.microsoft.com/es-es/library/gg502597(v=office.15).aspx)
- **Almacenamiento Seguro.** Para más información de cómo crear dicho aplicación [http://technet.microsoft.com/es-es/library/ee806866\(v=office.15\).aspx](http://technet.microsoft.com/es-es/library/ee806866(v=office.15).aspx)
- **Conectividad a datos empresariales:**
Para más información de cómo crear dicha aplicación [http://technet.microsoft.com/es-es/library/jj683108\(v=office.15\).aspx](http://technet.microsoft.com/es-es/library/jj683108(v=office.15).aspx)

También es necesario tener instalado Microsoft SharePoint Designer 2013 que lo podemos descargar del <http://www.microsoft.com/es-es/download/details.aspx?id=35491>

En muchas ocasiones necesitamos presentar en nuestro sitio de búsqueda datos empresariales de otros orígenes

Paso 1 - Preparación del servicio web de proveedores.

Disponemos en el código fuente del artículo un servicio web que devuelve un conjunto de proveedores codificado pero suficiente para nuestro ejemplo. Aunque existen muchas técnicas para publicar el servicio web, la más fácil en nuestro caso es copiar todo el contenido del fichero comprimido “WSPproveedores.zip” en “C:\inetpub\wwwroot”, para después crear el directorio virtual en IIS, para ello seguimos los siguientes puntos.

- Copiamos el servicio web: Abrimos el fichero “WSP-Proveedores.zip” con Winrar o equivalente y extraemos en la ubicación anteriormente mencionada.

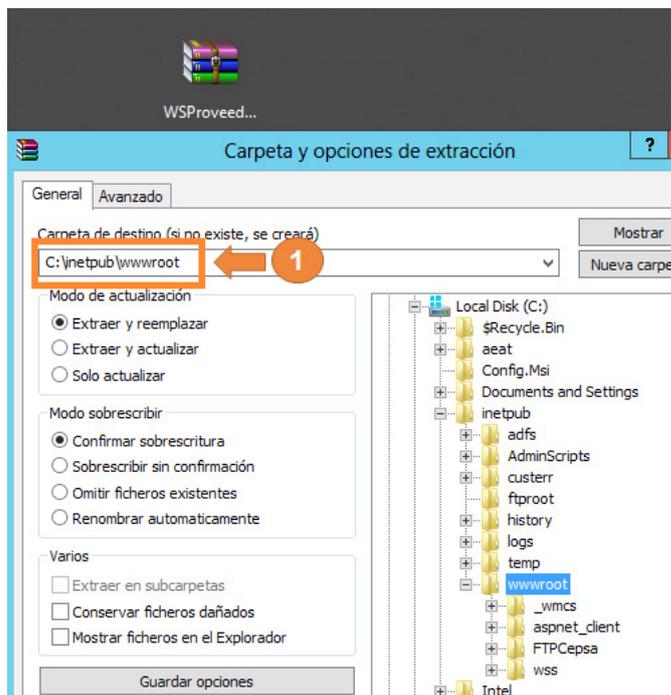
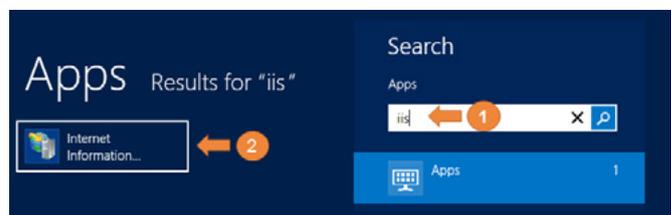


Imagen 2.- Muestra del contenido del fichero comprimido del servicio web de proveedores.

- Creamos el directorio web. En el buscador del sistema operativo introducimos el texto “IIS” y seleccionamos el icono de “Internet Information Server”.



Nos posicionamos en el nodo de la lista de sitios del panel de navegación y pulsamos el botón derecho de ratón para seleccionar añadir sitio web (Add WebSite), después introducimos los siguientes datos en el formulario nombre del sitio (Site Name) "Web Services Proveedores", en pool de aplicaciones (.Net 2.0), ruta física (Physical Path) "C:\inetpub\wwwroot", puerto (Port) 9797 y pulsamos en aceptar (OK) para crear el sitio web.

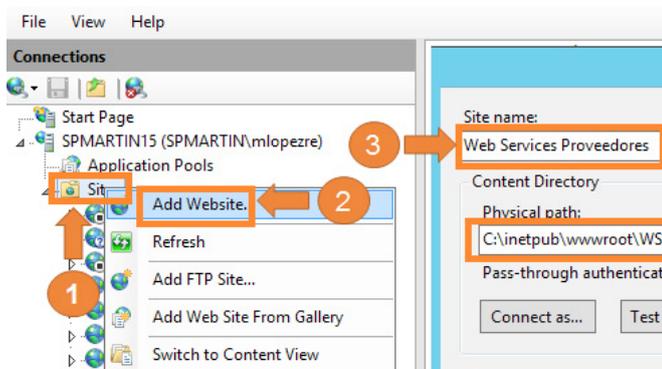


Imagen 3.- Muestra la ventana para crear un nuevo directorio virtual en IIS.

- Validamos el servicio. En el navegador introducimos la url del servicio web, en nuestro caso "http://spmartin15:9797/service.aspx"

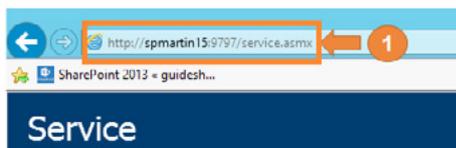


Imagen 4.- Muestra el servicio web de proveedores.

Paso 2 - Añadir credenciales al servicio de almacén seguro

Aunque podemos proteger nuestras conexiones con sistemas externos de muchas maneras, uno de los métodos recomendados es utilizar la aplicación de servicios de almacén seguros, para ello seguimos los siguientes puntos:

- Abrimos la Administración central de SharePoint 2013. En el buscador del sistema operativo introducimos el texto "SharePoint 2013" y seleccionamos el icono de "SharePoint 2013".



Imagen 5.- Muestra el resultado de la búsqueda realizada de SharePoint 2013.

- Abrimos el formulario de la aplicación del servicio de almacén seguro. Seleccionamos la opción de gestión de servicios de aplicaciones (Manage services applications) para posteriormente seleccionar nuestra aplicación de servicio de almacén seguro (Secure Store Services Application).

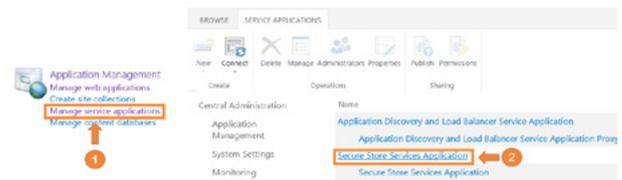


Imagen 6.- Muestra la página de las aplicaciones del servicio en la administración central.

- Creamos una nueva aplicación destino. Seleccionamos el botón de nuevo (New) en la cinta de opciones.

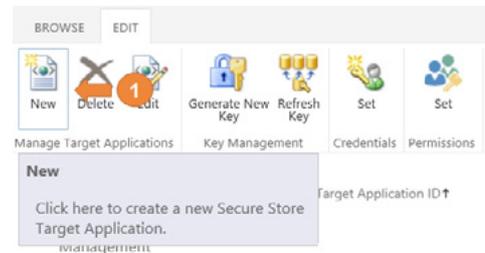


Imagen 7.- Muestra la cinta de opciones de la aplicación de servicios de almacén seguro (New).

Introducimos los datos solicitados. En nuestro caso en el identificador de la aplicación de servicio (Target Application ID), en el nombre a mostrar (Display Name) introducimos "ServiciosProveedores", en el campo de email de contacto (Contact E-Email) introducimos martin@spmartin15.com, dejamos los demás campos con los valores por defecto y pulsamos el botón de siguiente (Next).

Create New Secure Store Target Application

Target Application Settings

The Secure Store Target Application ID is a unique identifier. You cannot change this property after you create the Target Application.

The display name is used for display purposes only.

The contact e-mail should be a valid e-mail address of the primary contact for this Target Application.

The Target Application type determines whether this application uses a group mapping or individual mapping. Ticketing indicates whether tickets are used for this Target Application. You cannot change this property after you create the Target Application.

The Target Application page URL can be used to set the values for the credential fields for the Target Application by individual users.

Target Application ID: ServiciosProveedores (1)

Display Name: ServiciosProveedores (2)

Contact E-mail: martin@spmartin15.com (3)

Target Application Type: Individual

Target Application Page URL: Use default page

Next (4)

Imagen 8.- Muestra el primer formulario para crear una aplicación de destino.

Marcamos el tipo de campo de nombre de usuario de Windows (Windows User Name) y pulsamos sobre el botón de siguiente (Next).

Create New Secure Store Target Application

Field Name: Windows User Name, Field Type: Windows User Name, Masked: (1)

Field Name: Windows Password, Field Type: Windows Password, Masked:

Important: The field names and field types cannot be edited later.

Next (2)

Imagen 9.- Muestra el segundo formulario para crear una aplicación de destino.

Introducimos la relación de usuarios que tienen permisos para la gestión de la configuración de la aplicación destino, en nuestro caso "Martin Luis Lopez" y pulsamos sobre el botón de aceptar (OK)

Create New Secure Store Target Application



Imagen 10.- Muestra el formulario para asignar las credenciales.

- Asignamos las credenciales a la aplicación de destino. Seleccionamos nuestra aplicación de destino “ServicioProveedores” y en la cinta de opciones pulsamos sobre el botón de asignar credenciales (Set).

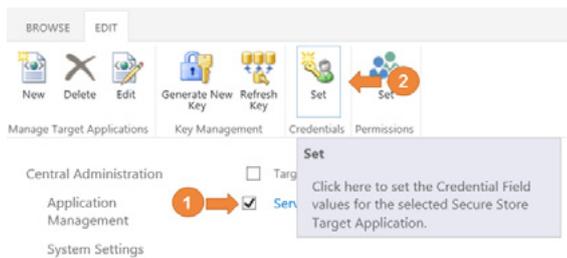


Imagen 11.- Muestra la cinta de opciones de la aplicación de servicios de almacén seguro (Set).

Introducimos los datos solicitados. En nuestro caso el propietario (Credential Owner) “Martin Luis Lopez”, usuario de Windows “spmartin15\mlopezre”, contraseña de Windows “*****”, y por últimos pulsamos sobre el botón de aceptar (OK)

Set Credentials for Secure Store Target Application (Individual) ✕

Warning: this page is not encrypted for secure communication. User names, passwords, and any other information will be sent in clear text. For more information, contact your administrator.

Target Application Name: ServiciosProveedores
 Target Application ID: ServiciosProveedores

Credential Owner: 1

Name: Value: 2

Confirm Windows User Name: 3

Windows Password: 4

Confirm Windows Password: 5

Note: Once the credentials are set, they cannot be retrieved by the administrator. Any existing credentials for this credential owner will be overwritten.

6

Imagen 12.- Muestra el formulario para asignar las credenciales en el almacén seguro.

Aunque podemos crear todas las operaciones CRUD, en las búsquedas sólo necesitamos las operaciones de lecturas

Paso 3 - Creación del tipo de contenido externo.

Aunque podemos crear el tipo de contenido externo mediante código vamos a utilizar Microsoft SharePoint Designer 2013 por su facilidad, para ello seguimos los siguientes puntos:

- Abrimos SharePoint Designer 2013. En el buscador

del sistema operativo introducimos el texto de “Designer” y seleccionamos el icono de “SharePoint Designer 2013”

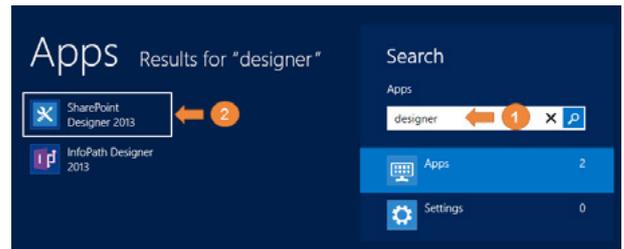


Imagen 13.- Muestra el resultado de la búsqueda realizada de Designer.

- Abrimos nuestro sitio de búsqueda. Para ello pulsamos sobre el botón de abrir sitio (Open Site) introducimos la Url en el nombre del sitio (Site Name) en nuestro caso “http://spmartin15:39910/” y pulsamos sobre el botón abrir (Open).

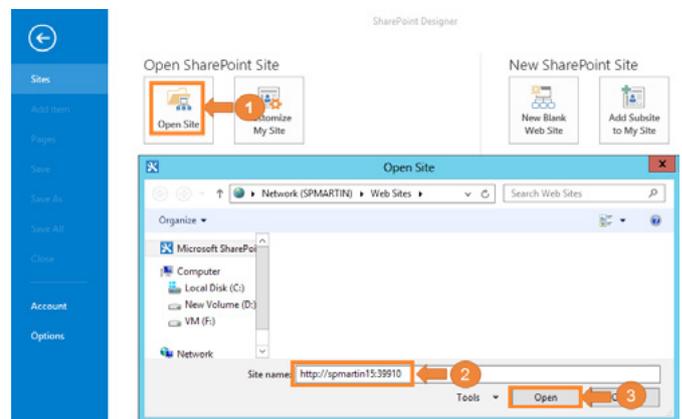


Imagen 14.- Muestra el formulario principal de Microsoft SharePoint Designer 2013.

- Creamos el nuevo tipo de contenido externo. En el panel de navegación seleccionamos el objeto de tipos de contenido externo (External Content Type), aparecerán los iconos correspondiente en la cinta de opciones y seleccionamos el primero de la categoría nuevo (External Content Type).

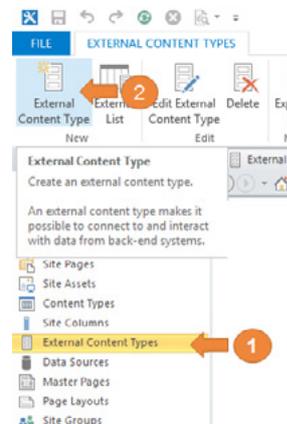


Imagen 15.- Muestra la cinta de opciones del objeto de tipos de contenido externo.

- Datos básicos. Introducimos el valor “Proveedores” en los campos nombre (Name) y nombre a mostrar (Display Name), después seleccionamos el enlace del sistema Externo (External System).

Si combinamos el motor de búsqueda con Business Connectivity Services (BCS) nos brinda grandes posibilidades de integración con datos empresarial

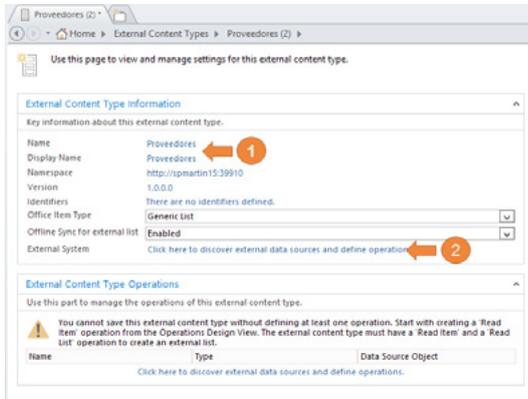


Imagen 16.- Muestra el formulario del nuevo tipo de contenido externo.

- Creamos la conexión. Pulsamos sobre el botón de añadir conexión (Add Connection) y seleccionamos el tipo de origen de datos, en nuestro caso “WCF Services” y aceptamos pulsando el botón aceptar (OK), posteriormente introducimos los datos de la conexión, en nuestro caso URL Servicio de metadatos (Metadata Connection Model) “http://spmartin15:9797/service.asmx?WSDL”, punto de enlace del servicio (Service EndPoint URI) http://spmartin15:9797/service.asmx, nombre (Name) “ServiciosProveedores”, utilizamos la credenciales almacenada en nuestro almacén seguro (Connect with impersonated Custom Identity) creado en el apartado anterior “ServiciosProveedor” y pulsamos sobre el botón de aceptar (OK).

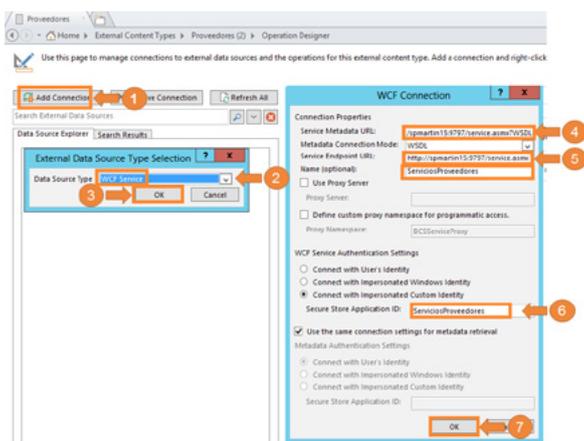


Imagen 17.- Muestra el formulario para crear una nueva conexión de datos externo.

- Creamos las operaciones. Aunque podemos crear todas las operaciones CRUD, en las búsquedas sólo necesitamos las operaciones de lecturas. Sobre el método “GetSupplier” pulsamos con el botón derecho del ratón y seleccionamos nueva operación de Lectura de Lista (New Read List Operation).

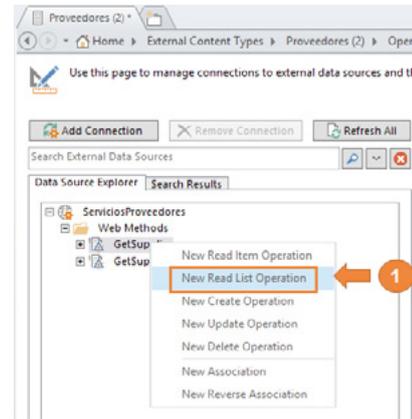


Imagen 18.- Muestra las operaciones de los métodos web (New Read List Operation).

En la primera ventana del formulario (Operation Properties) dejamos los valores por defecto y pulsamos el botón de siguiente (Next).

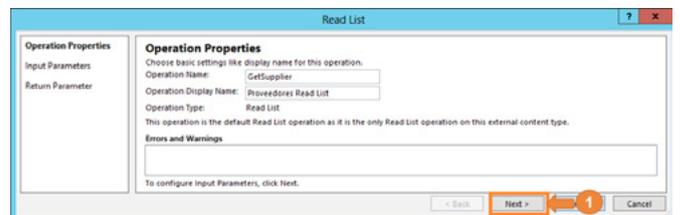


Imagen 19.- Muestra el formulario de propiedades de las operaciones de lectura de la lista.

Aunque se recomienda por buenas prácticas introducir parámetros de entrada para limitar el número de elementos devuelto, en nuestro caso de ejemplo pasamos por alto esta advertencia pulsamos sobre el botón siguiente (Next).

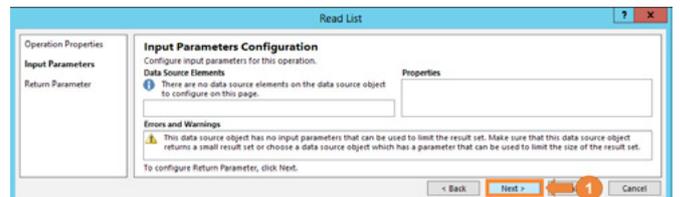


Imagen 20.- Muestra el formulario de parámetros de entradas en las operaciones de lectura de la lista.

En los parámetros de retorno Seleccionamos el campo “code”, indicamos que es mapa a identificador (Map to identifier), también lo indicamos como identificador (Identifier), indicamos que lo muestre en un recogedor (Picker), y por último pulsamos el botón de finalizar (Finish).

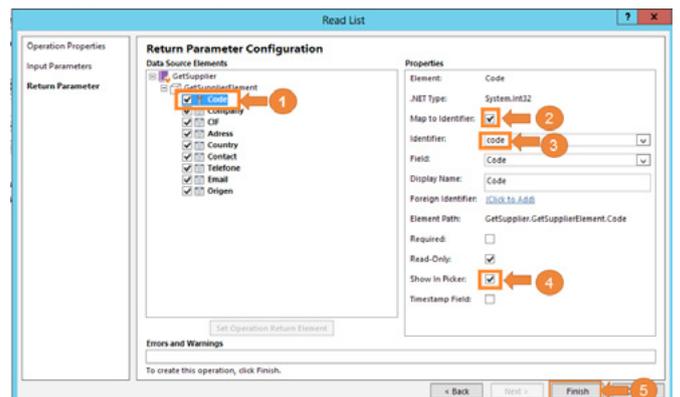


Imagen 21.- Muestra el formulario de parámetros de retorno en las operaciones de lectura de la lista.

Ahora creamos la operación de lectura de elemento.

Sobre el método "GetSupplierById" pulsamos con el botón derecho del ratón y seleccionamos nueva Operación de Lectura de elemento (New Read Item Operation).

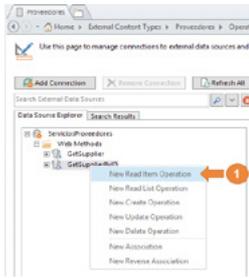


Imagen 22.- Muestra las operaciones de los métodos web (New Read Item Operation).

En la primera ventana del formulario (Operation Properties) dejamos los valores por defecto y pulsamos el botón de siguiente (Next).

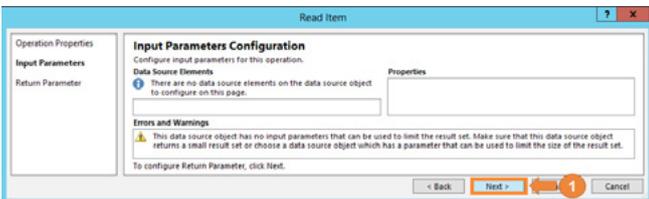


Imagen 23.- Muestra el formulario de propiedades de las operaciones de lectura de la lista.

En los parámetros de entrada seleccionamos "code" como mapa a identificar (Map to identifier) y pulsamos en el botón de siguiente (Next).

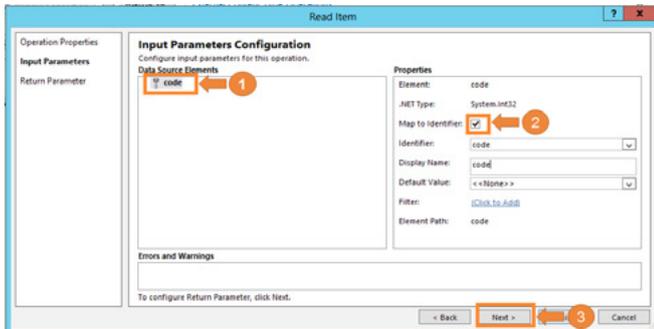


Imagen 24.- Muestra el formulario de parámetros de entradas en las operaciones de lectura de elemento.

En los parámetros de retorno Seleccionamos el campo "code", indicamos que es mapa a identificador (Map to identifier), también lo indicamos como identificador (Identifier) y por último pulsamos el botón de finalizar (Finish).

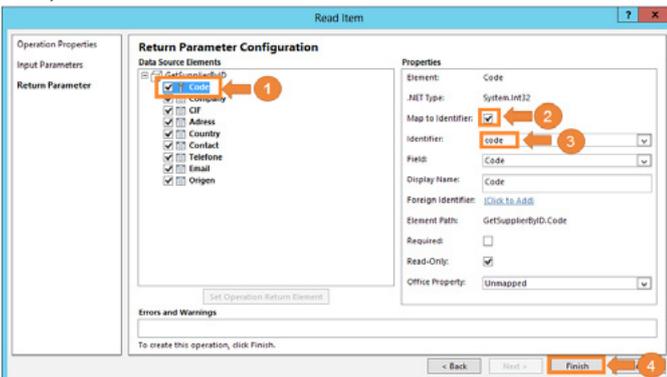


Imagen 25.- Muestra el formulario de parámetros de retorno en las operaciones de lectura de un elemento.

- Guardamos el nuevo tipo de contenido externo. Por último guardamos las acciones realizadas, para ello pulsamos sobre el botón representado con una imagen de un disquete situado en parte superior del formulario principal o pulsamos la combinación de teclas "Ctrl+S".

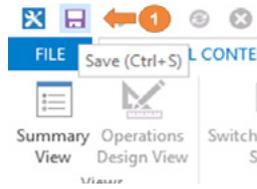


Imagen 26.- Muestra la opciones de guardado del SharePoint Designer.

Paso 4 - Configuración y Seguridad.

Para presentar los datos es necesario disponer de las páginas de perfil de los tipos de contenidos externo de proveedores y quien tiene acceso a ellos. Para ello seguimos los siguientes puntos:

- Abrimos la administración central de SharePoint 2013. En el buscador del sistema operativo introducimos el texto "SharePoint 2013" y seleccionamos el icono de "SharePoint 2013".



Imagen 27.- Muestra el resultado de la búsqueda realizada de SharePoint 2013.

- Abrimos el formulario de la aplicación de servicio conectividad a datos empresariales. En el caso de no estar en la página inicial de la Administración Central pulsamos sobre la imagen de SharePoint, seleccionamos la opción de gestión de servicios de aplicaciones (Manage services Applications) para posteriormente seleccionar nuestra aplicación de servicio de conectividad a datos empresariales (Business Data Connectivity Services Application).

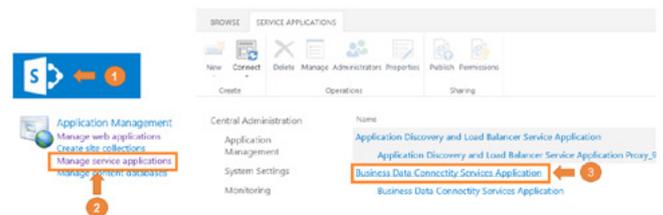


Imagen 28.- Muestra la página de las aplicaciones del servicio en la administración central.

- Activamos la creación de la página de perfil del tipo de contenido externo. Seleccionamos nuestro contenido externo creado anteriormente "Proveedores" y pulsamos sobre el botón de configuración "Configure" de la cinta de opciones.

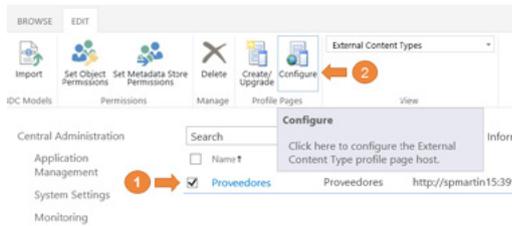


Imagen 29.- Muestra los tipos de contenidos externo.

En el formulario de configuración seleccionamos la casilla de activar la creación de página de perfil (Enable Profile Page Creation) e introducimos la URL correspondiente del sitio de búsqueda en nuestro caso <http://spmartin15:39910> y aceptamos los valores.



Imagen 30.- Muestra el formulario de configuración del origen de datos.

- Creamos la página de perfil del tipo de contenido externo. Volvemos a SharePoint Designer 2013 concretamente a nuestro tipo de contenido externo “Proveedores” en caso de no estar abierto ver los pasos en el punto 3. En la cinta de opciones pulsamos sobre crear página de perfil (Create Profile Page).

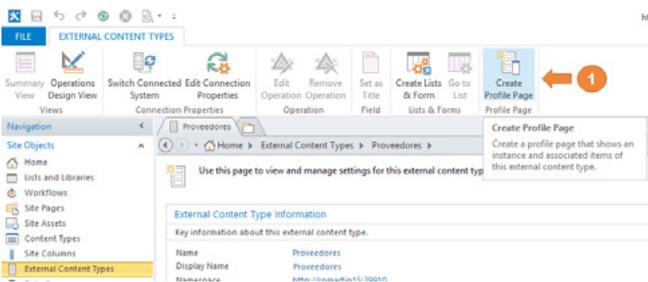


Imagen 31.- Muestra el formulario de tipo de contenido externo de proveedores.

- Introducimos los permisos para nuestro tipo de contenido externo. Volvemos a la Administración central, concretamente al formulario del servicio de conectividad a datos empresariales, seleccionamos el contenido de origen “Proveedores” y pulsamos sobre el botón

coger los permisos del objeto (Set Object Permission), en el formulario asignamos los permisos necesarios y guardamos la información introducida.

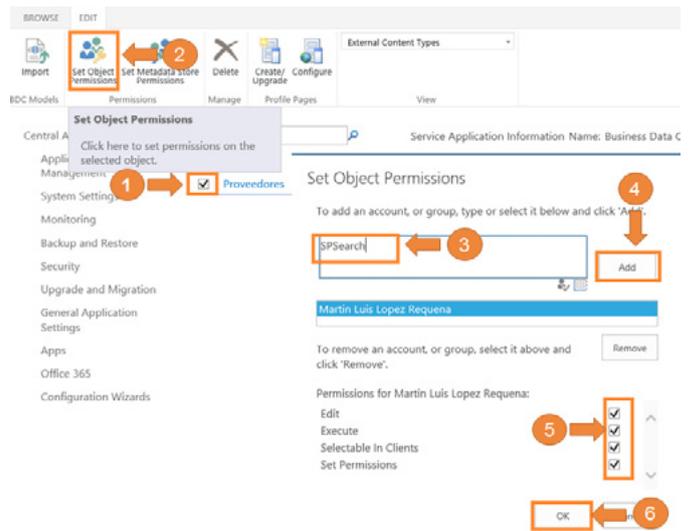


Imagen 32.- Muestra el formulario de seguridad de tipo de contenido externo de proveedores.

Conclusiones

Si combinamos el motor de búsqueda con Business Connectivity Services (BCS) nos brinda grandes posibilidades de integración con datos empresarial sin ningún tipo de desarrollo personalizado en la mayoría de los casos además de una gran facilidad y flexibilidad en su configuración. En el siguiente artículo de la serie, continuaremos realizando las configuraciones necesarias para aprovechar estas características de integración.

MARTIN LUIS LOPEZ REQUENA

SharePoint Solution Architect & Trainer at everis

martinluislopez@hotmail.com



Creación de un Chat para SharePoint con SignalR

En este artículo vamos a explicar cómo hemos desarrollado un chat para SharePoint usando SignalR, pero, ¿qué es SignalR? SignalR es una librería Open Source que forma parte de la familia de ASP.Net. Mediante SignalR podemos añadir funcionalidad de tipo comunicación en tiempo real entre cliente y servidor, teniendo así una comunicación bidireccional.

Todas las conexiones son administradas por SignalR y permite transmitir los cambios a todos los clientes conectados o en caso de que sea necesario podemos notificar a un cliente específico.

Lo que nos ayuda a resolver SignalR es que en lugar de tener una aplicación que desde el lado del cliente periódicamente este preguntando al servidor si hay “algo nuevo” (lo denominado “Polling”), sea el servidor el que notifique al cliente cuándo se ha producido “algo nuevo”.

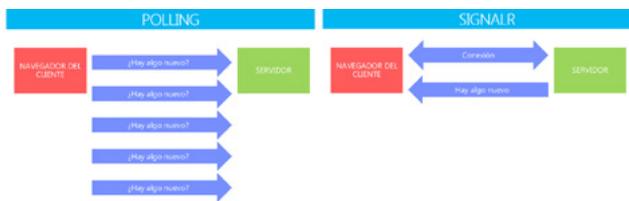


Imagen 1.- Arquitectura de comunicaciones de SignalR.

Gracias a esta arquitectura podremos mandar mensajes 1 a N (por ejemplo cuando un usuario se conecta o se desconecta para informar al resto de clientes) o 1 a 1 (cuando un cliente manda un mensaje a otro cliente) todo ello de una forma ligera dado que el servidor únicamente actúa cuando tiene que realizar una tarea, lo mismo en la parte cliente dado que no se realiza el llamado “Polling” mostrada en la imagen anterior.

En este artículo explicaremos qué es SignalR y cómo nos puede ayudar a crear funcionalidades en tiempo real

Para entender un poco cómo funciona todo esto vamos a mostrar algunos ejemplos de código. Cuando un cliente se conecta al chat, lo primero que hace es establecer la conexión con el servidor, esto se hace de la siguiente manera:

```
<script src="/signalr/hubs" type="text/javascript"></script>
<script type="text/javascript">
$(document).ready(function () {
    var chatHub = $.connection.chatHub;
    registerClientMethods(chatHub);
    // Start Hub
    $.connection.hub.start()
        .done(function () { registerEvents(chatHub); })
        .fail(function () { alert("El servicio de chat no está disponible en este momento."); });
});
</script>
```

Donde se incluye una referencia a “/signalr/hubs”, ubicación donde el propio SignalR genera de forma automática las librerías de JavaScript necesarias para llamar a las funciones de la parte servidora. Gracias a esta referencia tenemos acceso a “\$.connection” y de ahí realizar llamadas como por ejemplo a la función “\$.connection.hub.start()”, que abrirá la conexión con el servidor.

Una vez establecida la conexión con el servidor, podremos llamar fácilmente a funciones del servidor mediante JavaScript. A continuación se muestra un ejemplo de cómo se enviaría un mensaje privado a otro usuario:

```
<script type="text/javascript">
$div.find("#btnSendMessage").click(function () {
    chatHub.server.enviarMensajePrivado(userId, toUserId, msg);
});
</script>
```

Al pulsar el botón de enviar, realizaremos una llamada a la función “enviarMensajePrivado”, llamada que se realiza en JavaScript pero que será respondida por una función servidora que se muestra a continuación:

```
public void EnviarMensajePrivado(string fromUserId, string toUserId, string message)
{
    //Previamente al envío del mensaje se haría la persistencia del mensaje en base de datos, se ha quitado del ejemplo por simplicidad
    foreach (var user in toUsers)
    {
        //Con este foreach lo que hacemos es recorrer todas las sesiones que el usuario receptor tiene abiertas (porque puede tener varias pestañas, varios navegadores...) y enviar el mensaje a cada una de ellas
        Clients.Client(user.ConnectionId).sendPrivateMessage(fromUserId, message);
    }
    foreach (var user in fromUsers)
    {
        //Se recorren todas las sesiones que el usuario emisor tiene abiertas (puede tener varias pestañas, varios navegadores...) y se envía el mensaje a cada una de ellas
        Clients.Client(user.ConnectionId).sendPrivateMessage(toUserId, message);
    }
}
```

Al llamar a la función “sendPrivateMessage” lo que hacemos es llamar a la función de JavaScript que estará incluida en cada navegador del cliente (del usuario emisor o receptor) y enviará el id del usuario y el mensaje para que cada uno de estos navegadores clientes refleje el mensaje en pantalla.

```
<script type="text/javascript">
chatHub.client.sendPrivateMensaje = function (userId,
message) {
  if ($('#' + userId).length == 0)
  {
    OpenPrivateChatWindow(chatHub, userId);
  }
  else
  {
    $('#' + userId).append("<div class='message'>" + message +
"</div>");
  }
};
</script>
```

Al recibir el mensaje en la parte cliente se comprueba si (“#’ + userId”) existe, en caso de no existir significa que el usuario no tiene abierta la ventana de conversación con el usuario que le está enviando el mensaje, de modo que se llamará a la función “OpenPrivateChatWindow” para mostrar la ventana emergente. En caso de ya existir la ventana de la conversación lo que se hace es mostrar el nuevo mensaje.

Lo que nos ayuda a resolver SignalR es que en lugar de tener una aplicación que desde el lado del cliente periódicamente este preguntando al servidor

A continuación se muestra una captura de pantalla de cómo se visualiza el chat dentro de SharePoint:

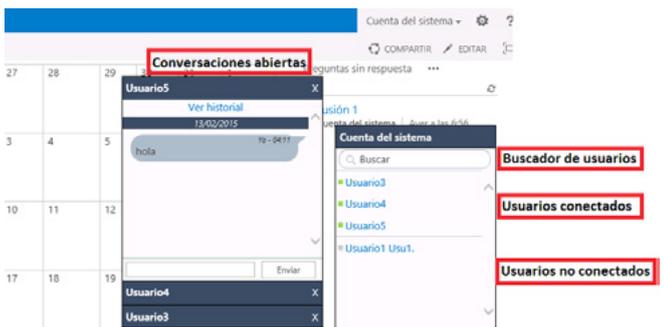


Imagen 2.- Ejemplo de visualización del chat en SharePoint.

Y de la visión que tendría el usuario 5 al recibir el mensaje:

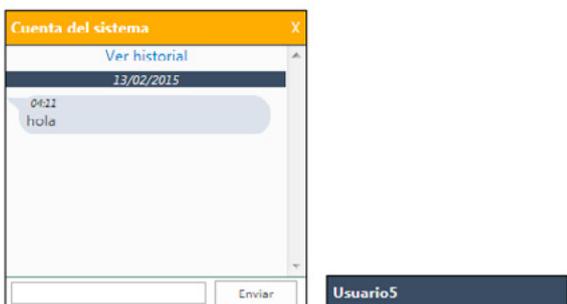


Imagen 3.- Recepción del mensaje por parte del destinatario.

Entre las características principales que nos ofrece el chat cabe destacar:

- Información de conexión de los usuarios en tiempo real, de modo que sabemos en todo momento quién está conectado.
- Los mensajes son entregados correctamente a sus destinatarios aun cuando no estén conectados. El servicio proporciona la entrega fuera de línea de mensajes personales. Dispone además de un histórico que podremos consultar para ver conversaciones pasadas.
- Dispone de un buscador de usuarios para encontrar rápidamente un usuario en concreto y ver su disponibilidad.
- El diseño está totalmente integrado con la UI de SharePoint y es fácilmente personalizable dado que se basa en una CSS que se puede modificar.
- Mejora la comunicación interna y garantiza la confidencialidad de la conversación.
- Permite un número ilimitado de usuarios usando para ello pocos recursos del sistema y de una forma ligera gracias a la arquitectura de SignalR.

SharePoint es una plataforma extensible por naturaleza y la incorporación de tecnologías como SignalR facilitan habilitar más opciones de extensibilidad

Conclusiones

SharePoint es una plataforma extensible por naturaleza y la incorporación de tecnologías como SignalR facilitan habilitar más opciones de extensibilidad de la misma dotando de funcionalidad adicional como la descrita en este artículo: proporcionar una funcionalidad de chat de forma rápida y sencilla sin necesidad de recurrir a productos de terceros o a la integración nativa de la plataforma con Lync.

RICARDO GIL GONZÁLEZ

MVP CLUSTER | Especialista en SharePoint & Office 365

francisco.gil@fiveshareit.es

LinkedIn

<http://www.mvpcluster.es>

JOSÉ ANTONIO FRAGA SÁNCHEZ

MVP CLUSTER | SharePoint Team Leader

jose.fraga@fiveshareit.es

LinkedIn

<http://www.mvpcluster.es>



Novedades en Power BI

En los últimos días puede que algunos de vosotros hayáis oído que se acercan novedades a Office 365 en lo que respecta a Power BI, y con el objetivo de daros un poco de luz sobre estos rumores aquí os dejo algunas pinceladas de lo que nos vamos a encontrar. Como en toda preview quiero recordaros que todo lo que hoy tenemos disponible es factible de cambio y/o eliminación, por lo que tomar este artículo como una antesala de la realidad a corto plazo.

¿Algo nuevo?

¡Pues sí, muchas cosas! Tal vez uno de los puntos más importantes es que Power BI ya no es una asociación a los Power* que teníamos en Excel como complementos, es decir, ha dejado ser un mero visualizador de reportes creados con PowerView, PowerMap o PowerQuery dando un paso adelante con su propio diseñador.

Power BI Designer

Aunque ya se vislumbra como la herramienta de diseño de Dashboards por excelencia, todavía sigue siendo una herramienta de escritorio que debes descargar. A decir verdad, y aunque imagino que en breve veremos su homóloga web, debo reconocer que este modelo híbrido (aplicación de escritorio donde diseñamos los informes en local, pero están especialmente orientados a ser visualizados en la nube) resulta bastante cómoda, aunque obviamente, para gustos los colores....

Una de las cosas más importantes a tener en cuenta con esta "separación" y aunque por intrínseco parece obvio es que este nuevo funcionamiento separa a Power BI de Excel, lo que es trasladable a Excel Services, lo que quiere decir que... ¡exacto! Existe una separación patente entre SharePoint Online y Power BI, lo que además de ofrecer un modelo de licenciamiento mucho más abierto y modular, abre un abanico a nuevos tipos de conexión hasta ahora limitados, o por lo menos influenciados por Excel.

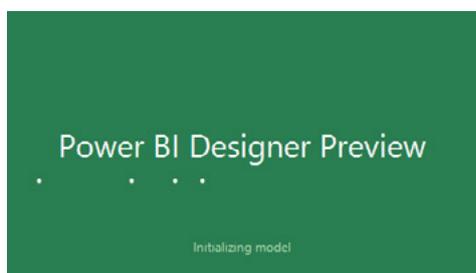


Imagen 1.- Inicio de Power BI.

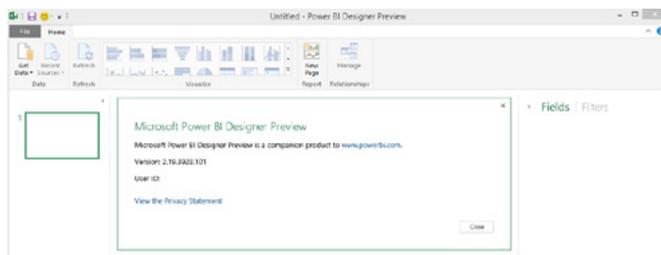


Imagen 2.- Diseñador de Power BI.

El acceso a los datos

Otro de los puntos a destacar que impactan a primera vista del Designer es la nueva capacidad de conectividad a nuevas fuentes de datos predefinidas, algo que hasta ahora teníamos que hacer vía oData, pero lo mejor de esto es que se abre un abanico para agregar nuevos conectores fácilmente.

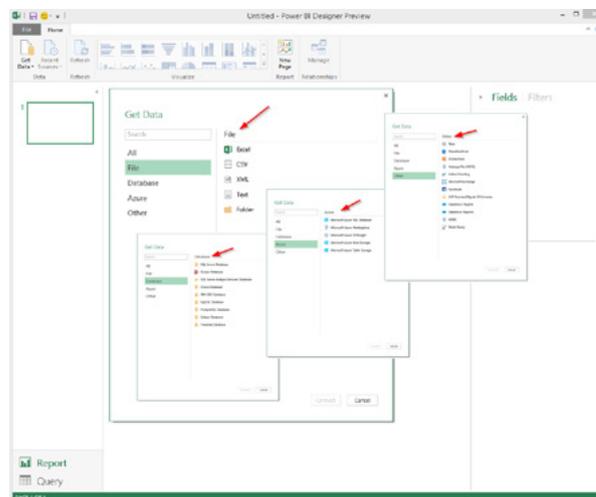


Imagen 3.- Fuentes de datos disponibles para Power BI.

Power BI ya no es una asociación a los Power* que teníamos en Excel como complementos, es decir, ha dejado ser un mero visualizador

Incluso (por ahora desde el interfaz web) tenemos conexiones predefinidas a determinadas soluciones y plataformas que otorgan dashboards predefinidos con los informes más populares.

dan a viejos conocidos para los que hemos podido trabajar hace unos años con Proclarity (precursor de Performance-Point). Por mencionar algunos: Tree Map (donde cada cuadrado representa el peso de ese grupo dentro del total), Gauge (especialmente pensado para grandes indicadores), dos tipos de mapas (conectan a Bing), Funnel.

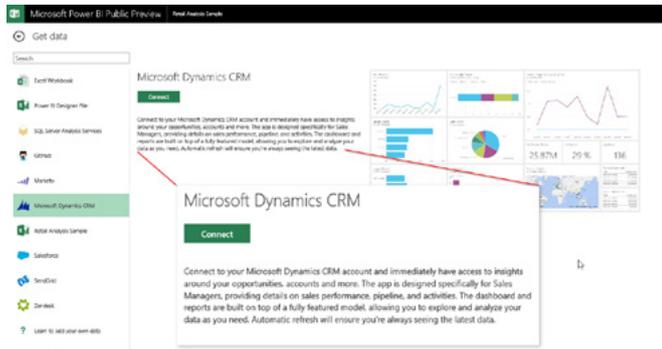


Imagen 4.- Conexiones pre-definidas en Power BI.

La creación de conexiones a las distintas fuentes de información es sencilla y guiada, y si ya has utilizado PowerPivot y PowerQuery encontrarás el interfaz sumamente familiar, ya que la forma de editar columnas, cambiar tipos, enlazar con tablas relacionadas, etc. es muy similar. La forma en que almacena los cambios que realizas sobre los orígenes de datos se basa en almacenar cada modificación en pasos individuales, cosa que ya conocíamos en PowerQuery como lenguaje “M”.

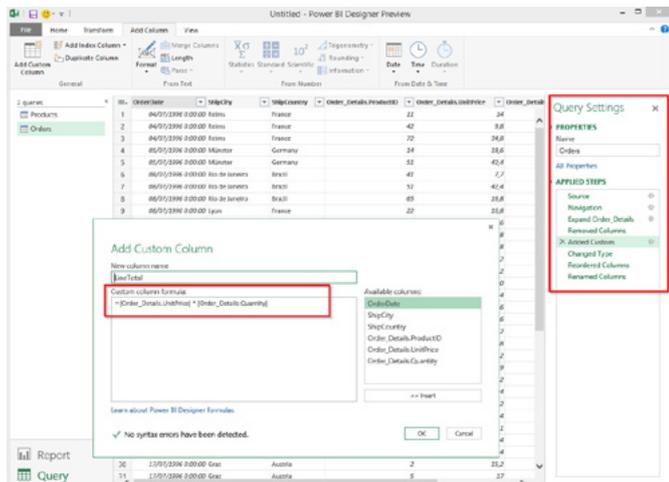


Imagen 5.- Edición del modelo de datos.

La edición del modelo de datos es intuitiva y recuerda a la edición en PowerPivot y PowerQuery.

estas novedades que incorpora Power BI auguran una solución mucho más atractiva, modular y abierta a la integración con toda clase de plataformas

Diseñando Informes

El diseño de los informes es muy similar a lo que ya habíamos visto en Power View. Se basa en ir arrastrando los valores que deseamos, momento en el cual el Designer construye automáticamente el gráfico más adecuado a ese valor. A partir de ese punto podemos agregar filtros, slicers, otros ejes, editar las leyendas, etc. Es importante mencionar los nuevos tipos de gráficos para los informes que antes no eran tan fáciles de construir y que recuer-

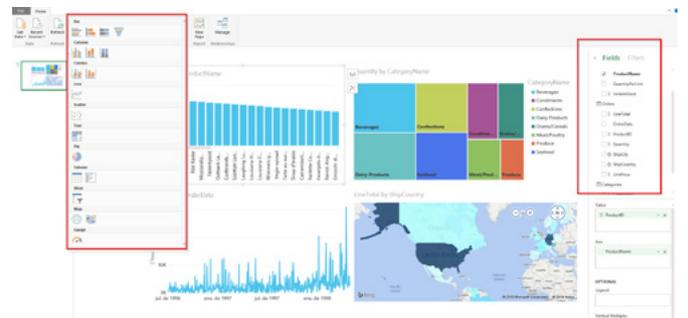


Imagen 6.- Diseñador de informes.

El Dashboard web: punto final

Aunque ya hemos visto bastantes novedades la integración la encontramos en el dashboard que nos ofrece el interfaz web. Este dashboard, compuesto de Tiles, nos permite vincular a varios orígenes de datos y mostrar la información de varias formas:

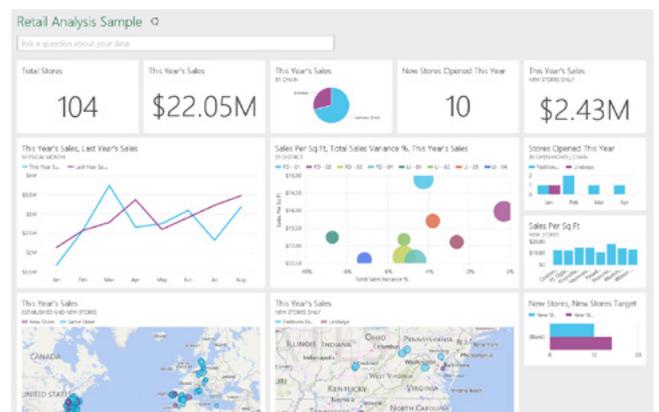


Imagen 7.- Dashboard web.

Cabe destacar que los tiles que se visualizan en el dashboard pueden provenir de 3 orígenes distintos: de un reporte, de un dataset o como respuesta a una pregunta, pero lo que debemos tener en cuenta es que un Tile es siempre una foto de la información que representa.

A nuestros usuarios les llamará la atención la potencia del Q&A, pero debemos tener en cuenta que para que funcione correctamente necesita un origen de información bien estructurado en un fichero Excel y al mismo tiempo que todos los campos, relaciones, valores y elementos calculados tengan una definición clara y directa en inglés.

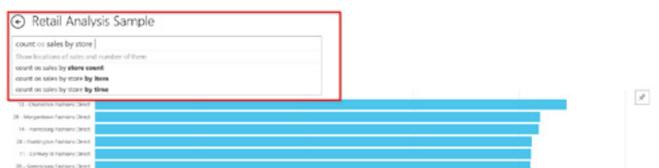


Imagen 8.- Q & A del Dashboard Web.

*Existe una separación patente entre
SharePoint Online y PowerBI*

En resumen

Para terminar, mencionar que ya se están liberando también las aplicaciones móviles para consumir Power BI desde plataformas de otros fabricantes, lo que nos deja claro una vez más, la “apertura de mente” por parte de Microsoft hacia otros dispositivos y soluciones. No cabe duda

que estas novedades que incorpora Power BI auguran una solución mucho más atractiva, modular y abierta a la integración con toda clase de plataformas, marcando un claro objetivo a corto plazo: el aprovechamiento de la nube a todo nivel, desde lo pequeño que nos pueda aportar un fichero plano hasta lo enorme que podamos encontrar en HDInsight.

JAVIER MENÉNDEZ PALLO

Director Oficina Centro – España y Portugal
 ENCAMINA

jmenendez@encamina.com



Cálculo de costes en Microsoft Azure

La nueva dinámica empresarial y tecnológica de movimiento hacia la nube normalmente viene seguida de la típica pregunta ¿Y cuánto me va a costar mes a mes? Este cálculo no es nada sencillo de hacer, por cuanto depende mucho la granja, las prestaciones requeridas, las consideraciones de resistencia a fallos, y la misma relación de la empresa con Microsoft para hacer la cuenta, pero ese tampoco es el objetivo del presente artículo.



Imagen 1.- Portal de Microsoft Azure.

En este artículo voy a permitirme la licencia de diseñar una granja desde cero, intentando conseguir un “estado del arte” de lo que se puede montar a día de hoy. Esta consideración es importante, porque para cualquier persona que comienza a trabajar con Azure es impactante la velocidad con la que evoluciona la plataforma, y esto hace que sea complicado el mantenerse al paso de los nuevos servicios e infraestructura en Preview y GA (Globally Available). Eso sí, siempre teniendo en cuenta la relación inversión/beneficio. Sería muy fácil montar decenas de máquinas con cientos de Gb de RAM, procesadores gigantescos, y enormes discos (o al menos muchos), pero no siempre son la respuesta adecuada, y su coste sería desmesurado. Por ello, mi objetivo es proponer pequeñas ideas y/o trucos que pueden servir para tener una infraestructura óptima, y a un coste moderado.

Consideraciones de infraestructura

Para ello, nada mejor que recordar una regla importante en Azure: De un nivel al siguiente, el coste se dobla. Esta regla implica que dos máquinas A5 costarán, aproximadamente, lo mismo que una máquina A6 de forma mensual. En una infraestructura SharePoint, habitualmente es preferible una estrategia Scale Out (Añadir máquinas a la

granja) que una estrategia Scale Up (añadir recursos a las máquinas). Por muchos recursos que añadamos, llegado a un determinado nivel no vamos a conseguir ganar velocidad a la granja, y tendremos que recurrir a otras estrategias, como añadir servidores de cacheado en capas adicionales, para mejorar el rendimiento. La mejor analogía que se puede hacer es como potenciar un coche... podemos ponerle 200 caballos más, pero una vez llega al corte en 5ª o 6ª, no va a correr más.

La nueva dinámica empresarial y tecnológica de movimiento hacia la nube normalmente viene seguida de la típica pregunta ¿Y cuánto me va a costar mes a mes?

Otro punto a tener en cuenta es que por lo general las máquinas de SharePoint suelen tener grandes requerimientos a nivel de discos duros, y en Azure existen limitaciones de IOPs por disco añadido. Ante esta situación, existen dos formas de atajar la situación. La primera es disponer de máquinas muy grandes, que permiten añadir muchos discos, y hacer stripping sobre ellos de forma que aparezcan como una sola unidad con muchos IOPs, y además disponen de mucha memoria lo que reducen los requerimientos de acceso a disco. Esta era la única estrategia de la que se disponía hasta hace poco, pero a día de hoy (al final vamos a tener que inventar un acrónimo para esta expresión cuando hablamos de Azure) disponemos de una solución más óptima. Azure Premium Storage permite la definición de unidades con los límites de tamaño y velocidad que se requiera hasta unos límites bastante impresionantes. 32 Tb, 50.000 IOPs y una latencia < 1 ms... Bastante impresionante, ¿no? Esto no solo nos daría cobertura a los requisitos de una granja de SharePoint, sino que abre las posibilidades para subir a la nube a sistemas de alta demanda como un Dynamics, una instalación SQL grande, o un sistema SAP.

Se ha tomado como premisa que el cliente dispone de las licencias bases de software y el contrato de Software Assurance necesario para utilizar las licencias en la nube. En caso contrario, para las máquinas base de SQL debería tomarse el coste de las máquinas de tipo SQL que ya incluye

el coste de licenciamiento incluido en el precio por hora

Los costes totales se han tomado con los precios P-a-G (Pay as you Go) publicados en <http://azure.microsoft.com> a día 10/02/2014.

Máquinas virtuales

Para una granja estándar, se puede proponer una configuración 2x2x2:

- Dos servidores WFE de tipo D12.
- Dos servidores BE de tipo D12.
- 1 cluster de SQL formado por dos máquinas de tipo D13.

A estas máquinas se recomienda anexar también una réplica del AD local en la nube. Para este cometido puede usarse una máquina sencilla A1. El porqué de utilizar máquinas de tipo D12 es debido a que, salvo que los requisitos de la aplicación destino indique lo contrario, apostamos a nivel de coste más por invertir en memoria que en procesador.

Mis recomendaciones personales para las máquinas de SharePoint serían:

Apartado	Cantidad
Procesador	4 cores, 2,0 GHz mínimo
Memoria	16 GB mínimo.
Discos	Dos unidades: <ul style="list-style-type: none"> ✓ C:\ 80 GB mínimo. ✓ E:\ 80 GB mínimo ✓ Unidades localmente redundantes (LRS).
S.O.	Windows Server 2012.
Otros	Microsoft SharePoint Server 2013 + CALs.

Para los SQL

Apartado	Cantidad
Procesador	4 cores, 2,0 GHz mínimo.
Memoria	32 GB mínimo.
Discos	Dos unidades: <ul style="list-style-type: none"> ✓ C:\ 80 GB mínimo. ✓ Logs:\ 512 GB mínimo. ✓ Datos:\ 1 TB mínimo. ✓ MasterDB\ 160 GB mínimo.
S.O.	Windows Server 2012.
Otros	SQL 2012 Server Standard Edition.

Las máquinas especificadas tienen las siguientes características:

Máquina	Cores	Memoria	Tamaño Discos	Coste
D12	4	28 GB	200 GB	0,6271 €/h
D13	8	56 GB	400 GB	1,129 €/h

Una opción para reducir coste en estas máquinas sería utilizar máquinas de tipo A, pero estas máquinas no permiten el uso de discos de Azure Premium. Tomando estos datos de partida, el total en coste de máquinas sería:

Máquina	Máquinas	Coste/hora	Horas	Total/Mes
SP				
D12	4	0,6271 €	744	1866,25 €
SQL				
D13	2	1,129 €	744	1679,952 €
AD				
A1	1	0,0671 €	744	49,9224 €
Total Máquinas				3596,124 €

Almacenamiento

Para el almacenamiento, vamos a instalar nuestro SharePoint sobre la unidad Premium, y vamos a utilizar la unidad temporal para alojar el archivo de swap, de forma que utilicemos la velocidad de esta unidad, que es estándar para las máquinas D. Por lo tanto, necesitaremos al menos de una unidad Premium por máquina WFE o BE.

En el caso de los SQL, vamos a añadirle tres unidades. El tamaño de cada una dependerá de la aplicación, por lo que nos basaremos en el cálculo en los tamaños estándar.

Para Azure Premium, las configuraciones estándar son:

Máquina	Tamaño	Precio	IOPs	Proceso
P10	128 GB	8,0726 €	500	100 MB/s
P20	512 GB	29,99 €	2300	150 MB/s
P30	1024 GB	55,37 €	5000	200 MB/s

Actualmente esta funcionalidad se encuentra en vista previa, por lo que los precios tienen una rebaja con respecto a su precio final. Es muy probable que cuando se hagan GA, se dispongan de más tamaños alternativos. El total de almacenamiento sería:

	Unidades	Máquinas	Coste mensual	Capacidad	
SP					
P20	1	4	29,99 €	512 GB	119,96 €
SQL					
P20	2	2	29,99 €	512 GB	119,96 €
P30	1	2	55,37 €		110,74 €
Total Almacenamiento					350,96 €

Para cualquier persona que comienza a trabajar con Azure es impactante la velocidad con la que evoluciona la plataforma

Otros

Además de los gastos correspondientes a MV y almacenamiento, considerados como principales, también hay que tener en cuenta otros gastos que se pasan a enumerar:

- Conexión VNET-to-VNET: El tráfico de entrada a la nube no se tarifica. En cambio, el de salida sí que es tarificado. El coste es 0,0261€ por GB. Para un total de 1 Tb al mes: 26,7264 €\mes
- Puertas de enlace: 0,0269 por puerta de enlace. Aprox 21€\mes.
- Los datos de salida de Azure: En Zona 1 (europa) los primeros 5 Gb al mes son gratuitos. De 5 a 10 Tb se facturará a 0,0648 GB.
- Direccionamiento: Las 5 primeras reservas de IP interna son gratuitas.

Por último, y aunque no es necesario tenerlo en cuenta, es conveniente considerar utilizar el Load Balancer configurado delante de los servidores frontales, y aprovechar su funcionalidad de balanceo con persistencia de sesión. Esta funcionalidad, añadida hace poco, evita la necesidad de configurar cualquier otro tipo de balanceo, sea este externo en nube, como IMPERVA, o físico del tipo F5.

Costes totales

Para el ejemplo descrito, y tomando en cuenta que los precios pueden modificarse en función de las decisiones y/o licenciamiento efectuado, el total de coste sería:

Concepto	Coste mes
SP	
Maquinas	3596,12 €
Almacenamiento	350,96 €
Conexión	26,73 €
Trafico salida	21,00 €
Total	3.994,81 €

Y esto nos da la siguiente distribución de costes:



Imagen 2.- Distribución de Costes.

Nada mejor que recordar una regla importante en Azure: De un nivel al siguiente, el coste se dobla

Conclusiones finales

A lo largo del artículo, hemos visto como ir elaborando el coste para una granja de SharePoint, y como resultado de esta inversión disfrutaríamos de una granja de rendimiento muy robusto, y que podría dar servicio a un gran número de usuarios de forma concurrente. Aunque a priori pueda resultar un coste no muy reducido, una vez que se compara el valor con el coste de provisionar estas máquinas, tiempo para su disposición, posibilidades de fallo, robustez de la configuración y seguridad de acceso, el coste de esta solución se puede ver desde otra perspectiva. Como se observa de forma gráfica, el grueso del coste para una solución de SharePoint en la nube corresponde con el coste de las máquinas virtuales y el almacenamiento. De cara a bajar el coste de la solución propuesta, por lo tanto, podremos jugar tanto con dimensionamiento de máquinas, como con el tamaño de las mismas. Es perfectamente válido el disponer de máquinas de un tamaño en el frontal, y otras de tamaño diferente en la capa de backend y/o otras capas intermedias, como las de cacheo, o máquinas para la instalación de las OWA.

Espero que os haya gustado, y que tengáis una respuesta preparada la próxima vez que un cliente os pregunte ¿Cuánto me va a costar mi granja en Azure?

FABIÁN CALVO

Team leader

fcalvo@encamina.com

[@fcvspain](#)

<http://www.encamina.com>

La importancia de diseñar y aplicar un buen uso de Gobernanza en nuestro entorno SharePoint 2013

A través de una serie de artículos trataré de exponer, con casos prácticos extraídos de situaciones reales, paradigmas con el objetivo de que, si te encuentras con un caso similar, sepas enfocarlo y resolverlo.

Que es la gobernanza

A mi entender, la gobernanza es un conjunto de acciones, comportamientos y compromisos que se relacionan con un servicio de SharePoint, y contribuye a un conjunto intencionado de procesos establecidos de operaciones y procedimientos, funciones y responsabilidades, y toma de decisiones protocolarias.



Imagen 1.- Conceptos generales asociados a la gobernanza.

- Una acción es la primera iniciativa o una respuesta a una oportunidad.
- Un comportamiento es un conjunto de prácticas que se convierte en un hábito.
- Un compromiso es una dedicación que se desarrolla en un propósito.

La aptitud que adoptemos frente a estas tres figuras determinará el camino correcto hacia el éxito, una vez puestas en práctica, de nuestro entorno SharePoint. “Las acciones hablan más que las palabras,” como dice el refrán. En este contexto, las acciones son lo primero en esta gobernanza tripartita. Me gusta pensar en esto como respuesta a la oportunidad que se presenta o tomar la iniciativa para llegar a la cima del objetivo.

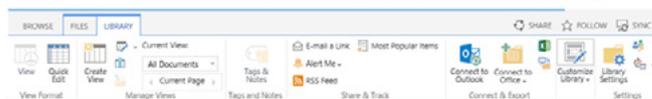
La gobernabilidad de un país no puede depender de personas a los que el país no les importa

Las acciones pueden ser cosas como insuflar a los usuarios energía y ofrecerles orientación para ayudar a maximizar su experiencia en el uso de SharePoint mientras también podemos orientarlos hacia el uso óptimo del mismo. Las acciones, a veces, son la parte más difícil, porque implican a menudo el primer paso hacia algo, y esos primeros pasos, al igual que sucede en cualquier inicio, suelen ser inseguros y sin impulso.

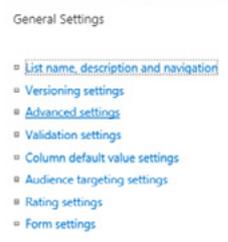
Los comportamientos se definen como aquellas acciones que, puestas en prácticas, se convierten en parte de una rutina. Cuando una acción podría implicar la investigación de algo diferente, o la realización de un análisis, los comportamientos incorporar las actividades más comunes con una reiteración de forma constante. Éstas pueden ser cosas como procedimientos operativos, el mantenimiento de la granja o el proceso de creación de una colección de sitios. Pueden ser la forma en que interactuamos con SharePoint, como si tiene recursos dedicados a la gestión del servicio de SharePoint o si sus responsabilidades principalmente están en otra parte, pero también son necesarios para mantener los servicios disponibles que nos ofrece SharePoint en base al uso de buenas prácticas.

Set the default behavior for document library

Browse to a SharePoint document library and select Library tab on the ribbon and select Library Settings.



Select Advanced Settings in Document Library settings.



In Advanced Settings go to the Opening Documents in the Browser section choose the default behavior. To open documents in the client application, select Open in the client application. To open in web browser, select Open in the browser. The default setting is "Use the server default".

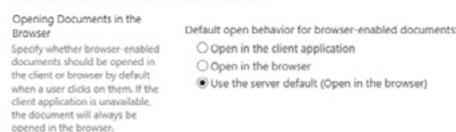


Imagen 2.- Configuración típica de uso de una biblioteca de documentos.

Los compromisos engloban la actitud y dedicación hacia el servicio de SharePoint: ¿Es SharePoint servicio de alta prioridad o por el contrario sólo le prestamos atención

cuando tenemos tiempo? Dependiendo de nuestras circunstancias, situación individual, y los objetivos de nuestra organización, cualquiera de ellos puede ser válida. Compromisos relacionados a su disciplina en la adopción de medidas y el mantenimiento de sus comportamientos. También realizamos compromisos con nuestros clientes, nuestro compromiso de ofrecer un servicio a un determinado nivel de servicio en el que se pueda confiar.

No pretendo fijar conceptos de gobernanza para una determinada situación o proceso específico. Mi objetivo es exponer estos conceptos basados en una forma lo suficientemente genérica para que pueda adoptar tal y como son, o adaptarse para que se ajusten con cualquier marco que se utiliza. Mi enfoque es específico de SharePoint con las consideraciones que deben regir tu entorno de SharePoint al compartir mi experiencia en la administración de SharePoint y consultar con una amplia variedad de clientes en la vida real.

También hemos de ser objetivos y ser conscientes de que muchas veces, por diferentes motivos como pueden ser falta de recursos, carencia de autoridad para implementar muchos de los procesos necesarios, pueden convertir el objetivo que voy a proponer en una tarea complicada, pero debemos de ser conscientes de que si somos capaces de asimilar nuevos conceptos y técnicas de mejoras podremos, sin duda alguna, tener un mayor éxito en la experiencia y uso de nuestro entorno de SharePoint. Consejo comenzar el planteamiento y diseño de nuestra gobernanza desde un punto de partida pequeño, enfocar los primeros pasos en un dilema menor, y como el efecto de una bola de nieve, tomar impulso e ir abarcando ámbitos mayores para construir nuestro plan de gobernanza, ya que no siempre hay una solución rápida y sería complicado y abrumador tratar de adoptar todo a la vez. En lugar de pensar en lo mucho que la gobernanza necesita, centrémonos mejor en donde deseamos iniciarla.

Las acciones pueden ser cosas como insuflar a los usuarios energía y ofrecerles orientación para ayudar

Por dónde empezar

El mejor lugar para comenzar es definir en qué consiste el servicio de SharePoint. En esta descripción, también estamos definiendo lo que no hace. Esta descripción da el enfoque de servicio, por lo que explícitamente de intención, en lugar de reaccionario. Una descripción de esta manera establece las bases clave para cualquier otra iniciativa de gobierno a depender y mejorar. Este ha sido un componente crucial para mí, y la base sobre la que utilizo para construir todas las demás tareas de gobierno e iniciativas. Por esa razón, la primera cosa que hacemos frente a

como comenzamos nuestro viaje hacia la gobernanza de SharePoint. Te animo a explorar algunas de estas ideas y perspectivas, muchos de los cuales serán referenciadas en la serie de artículos que se publicarán en la revista, y que recomiendo utilizar.

Hago referencia a la implementación de SharePoint como los servicios de SharePoint que ofrecemos. Esto es significativo para mí, ya que crea el derecho de pensar que desplegamos SharePoint para cumplir necesidades que algunos necesitan, para proporcionar algún servicio a los que lo usan.

Tomémonos nuestro tiempo: la gobernanza puede significar un montón de cambios de acciones, comportamientos y compromisos. Las organizaciones son muy lentas en adoptar los cambios de esta naturaleza, ya que pueden afectar a la cultura de la base de una organización. La gente todavía puede resistirse el cambio por diferentes razones, no importa lo posible, los cambios pueden parecer o la cantidad de los cambios puede beneficiar a las personas a largo plazo. Algunas personas se frustran cuando pasan de hacer algo cómodo a algo con lo que no están familiarizados con la sensación de un nuevo proceso, y como resultado, pueden reaccionar para resistirse a lo cambios. Dependiendo del enfoque que se le dé podremos tener mayor número de adeptos o detractores, fijémonos en ejemplos que pueden sernos de utilidad a la hora de poner en práctica nuestro plan, uno de ellos podemos enfocarlo en el número generalizado de dispositivos de táctiles y la naturaleza ubicua de las redes sociales como se han integrado en la vida de las personas - ambos elementos fueron en gran medida poco comunes para las masas, hace tan sólo una década más o menos

El cambio puede ser bueno, y algunos de los cambios que pueda requerir para las ideas de gobierno pueden ser para mejor también. Permanecer abiertos a las posibilidades y saber que si coloca desde la perspectiva de cómo los usuarios se beneficiarán, serán menos propensos a resistir el cambio que queremos introducir. A todo el mundo, por lo general, le gustan las cosas que le benefician. Las herramientas proporcionadas en estos artículos te ayudarán a realizar este cambio, si los usuarios se resisten o terminan abrazándolo.

Voy a compartir mis ideas para que consideres y pienses acerca de cómo puedes utilizarlas para empezar con tu propia Gobernabilidad de SharePoint. Sin embargo, si todavía no puedes decidir, a menudo el mejor lugar para empezar es por el principio: comenzar con la definición de su servicio de SharePoint.

Gobernanza y SharePoint 2013

SharePoint 2013 añade nuevas capacidades y mejora algunas ya existentes que ayudan en la consecución de diferentes objetivos de gobierno, haciendo de este un lanzamiento muy interesante para las necesidades de gobierno. Yo escogería eDiscovery como una nueva capacidad en

SharePoint 2013, que proporciona funciones ricas de gobierno, ya que proporciona la infraestructura para administrar y gobernar el contenido de elementos individuales con colecciones de sitios completas. SharePoint 2013 eDiscovery añade sofisticación desde una perspectiva de gestión de documentos y gestión de la información, y mejorando estas capacidades añaden madurez a SharePoint como un sistema de gestión de contenido empresarial.

Con el uso de eDiscovery en SharePoint 2013, podemos gobernar la retención de contenido y otros tipos de políticas, a nivel mundial y entre granjas. También podemos gobernar los requisitos legales y reglamentarios, así como realizar seguimientos e informar sobre nuestro cumplimiento.

Los Propietarios o las políticas de sitio pueden establecer sitios a un estado cerrado o congelado en lugar de simplemente eliminarlos, habilitando estrategias de gobernabilidad para hacer frente a retirar el contenido archivado en un proceso gradual de entre un estado de contenido que se está convirtiendo de on-line a off-line.

Las Aplicaciones para SharePoint y la tienda de SharePoint permiten escenarios tales como permitir a los usuarios comprar y adquirir su propia funcionalidad sin modificar o afectar la granja subyacente. Una organización también puede ofrecer un catálogo interno de aplicaciones para SharePoint que los usuarios pueden consumir y utilizar en su sitio, lo que permite un catálogo centralizado y único punto de acceso para proporcionar aplicaciones personalizadas y funciones de la organización. Este simplifica el proceso de implementación y administración de soluciones a medida, tanto para el departamento de IT a la hora de proporcionar las soluciones como para los usuarios finales a la hora de añadir la solución a su sitio.

SharePoint 2013 mejora la función de creación de sitios sin intervención del servicio, de modo que ahora se puede recopilar más información, información como el tiempo que el sitio está activo y otro tipo de información valiosa sobre el sitio. También puedes personalizar este proceso para agregar lógica adicional para ayudar a gobernar y administrar sitios. Puedes utilizar la función de cualquiera de las colecciones de sitio o nuevas redes dentro de una colección de sitios, que es una característica muy útil sobre todo para la aplicación de las políticas sobre la colección de sitios o nuevas webs dentro de la colección de sitios.

Las solicitudes de acceso del sitio es otra de las características de SharePoint 2013 ha mejorado. Estas mejoran en el proceso de petición en sitios, realizando una gestión de permisos y más sencilla para los usuarios comunes, y esto como resultado, ayuda a que la gobernanza del control de acceso sea más directo. Por un lado, existe una vía de auditoría de la actividad en la solicitud de permisos, por lo que puede rastrear quien concedió, qué permiso y cuando.

Otra característica muy útil para la gobernabilidad es la página de Gestión de solicitudes (Request Management

Page), donde peticiones de acceso pendientes y un historial de peticiones son visibles para los administradores del sitio. Las solicitudes también tienen un lugar para comentarios, por lo que los administradores del sitio pueden hacer preguntas a los solicitantes para entender del por qué necesitan los permisos que están solicitando.

Todo esto pensado para ayudar a reducir el número de permisos excesivos e innecesarios otorgados a los usuarios, permisos concedidos simplemente porque en el pasado los permisos o solicitudes no eran lo suficientemente claras para ser entendidas por el administrador del sitio.

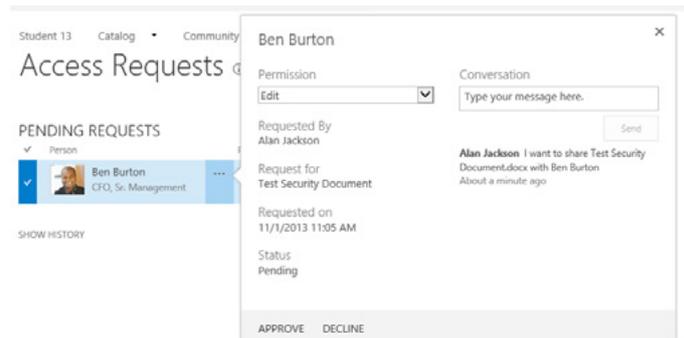


Imagen 3.- Lista de Solicitudes de Acceso en SharePoint 2013.

Algunos cambios sutiles en la forma de gestionar el Branding (marca) en SharePoint 2013 también puede ayudar a simplificar la forma en que gobernar personalizaciones de Branding. Por un lado, se puede crear un paquete de Branding mediante la creación de nuestros elementos de interfaz, estilos, imágenes, etc., todos en un sitio de grupo de SharePoint y luego exportarlos como un paquete de Branding. También podemos compartir este paquete con otros sitios. SharePoint 2013 ofrece las normas de Branding en HTML 5, lo que significa que nuestros diseñadores gráficos se enfrentarán a menos obstáculos a la hora de personalizar el aspecto y la sensación de un Sitio de SharePoint. Lo más importante para nosotros teniendo en cuenta la gobernabilidad, esto también significa que nosotros o los diseñadores de sitios pueden personalizar más fácilmente elementos de la interfaz de usuario en los sitios de una manera estándar y consistente, y esto nos ayudará a realizar un mantenimiento más fácil de gestionar.

El Rastreo continuo (Continuous crawling) en el motor de búsqueda de SharePoint 2013 nos ofrece la posibilidad de que el índice de contenido permanezca actualizado de forma continua. Por lo tanto, para aquellas fuentes de contenido que habilite dicha acción, no requerirá de intervención por parte del administrador para que el índice esté actualizado. Esto simplifica programar y coordinar el servicio de búsqueda. Hay implicaciones de rendimiento, y en algunos casos, es posible que todavía tengamos que planificar una frecuencia de rastreo incremental, para ello aún disponemos de los rastreos incrementales, tal vez por



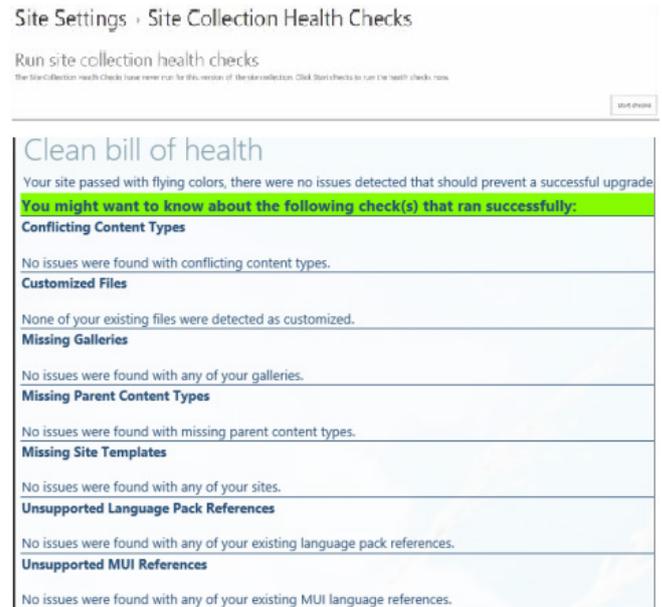
las acciones de archivos de red o archivo los medios de comunicación que no cambia con frecuencia y no necesita un índice de “dulce” disponible en el motor de búsqueda empresarial.

SharePoint 2013 introduce la gestión de navegación por metadatos (Managed Navigation), una característica que la navegación del sitio asocia con un plazo establecido en el Servicio de metadatos administrados. Ahora la navegación de un portal puede ser fácilmente manejada y mantenerse coherente en muchas colecciones de sitios. El uso de la navegación basada en la estructura integrada fue una de las objeciones más comunes contra la implementación de una arquitectura de la información que constaba de varias colecciones de sitios, pero ahora que SharePoint 2013 ofrece tanto una estructura y metadatos impulsados de navegación hay menos dificultad para poder realizar un diseño en múltiples colecciones de sitios con mayor escalabilidad.

El mejor lugar para comenzar es definir en qué consiste el servicio de SharePoint

Office 2013 ahora establece el archivo por defecto como ubicación de almacenamiento para los usuarios de la empresa MySites. Aunque el cuadro de diálogo Guardar archivo ofreció MySites como una opción de ubicación para varias versiones anteriores de Microsoft Office, nunca fue el valor predeterminado. Este cambio es importante porque le ayuda dar un gran salto hacia un almacenamiento de contenido centralizado. Desde la perspectiva de Gobernanza, habiendo contenidos almacenados centralizados nos proporciona más oportunidades de administrar y gobernar el contenido, por ejemplo, cuando los usuarios lo almacenan en sus MySites en lugar de en sus escritorios. SharePoint 2013 también hace gestionar y compartir contenido en Mysites de un usuario más fácil y con una experiencia de usuario mejorada. MySites sólo tienen una única biblioteca de documentos personales en SharePoint 2013, por lo que los usuarios ya no tienen que negociar entre el personal y el documento público de las bibliotecas como se hacía en las anteriores versiones. Ahora los usuarios pueden utilizar su biblioteca de documentos personal para almacenar los documentos a los que sólo ellos tienen acceso, y comparten con otros usuarios.

SharePoint 2013 también hace extensible los Health Checks, que, en versiones anteriores, sólo estaban accesible desde la Administración Central. Ahora también están disponibles en las colecciones de sitios para realizar un chequeo que valida el sitio contra de las reglas definidas. Los resultados de la verificación, proporcionan un informe visual en las áreas que necesitan atención para alertar a los administradores de las colecciones de sitio de los posibles problemas a futuro cuando actualicen su sitio o aplicaciones dentro del sitio.



Site Settings · Site Collection Health Checks

Run site collection health checks

The Site Collection Health Checks have never run for this version of the site collection. Click Start checks to run the health checks now.

Clean bill of health

Your site passed with flying colors, there were no issues detected that should prevent a successful upgrade.

You might want to know about the following check(s) that ran successfully:

- Conflicting Content Types**
No issues were found with conflicting content types.
- Customized Files**
None of your existing files were detected as customized.
- Missing Galleries**
No issues were found with any of your galleries.
- Missing Parent Content Types**
No issues were found with missing parent content types.
- Missing Site Templates**
No issues were found with any of your sites.
- Unsupported Language Pack References**
No issues were found with any of your existing language pack references.
- Unsupported MUI References**
No issues were found with any of your existing MUI language references.

Conclusiones

El diseño de una buena gobernanza en despliegues SharePoint y su aplicación es fundamental por un lado para facilitar la adopción y uso de la plataforma, y por otro para garantizar su crecimiento ordenado y bajo una serie de parámetros controlados por la organización. SharePoint 2013 incorpora de serie unas características y capacidades que facilitan el diseño y uso de estrategias de gobernanza adecuadas para una organización.

FRANCISCO RICARDO GIL GONZÁLEZ
MVP CLUSTER | Especialista en SharePoint & Office 365
francisco.gil@fiveshareit.es
[Linkedin](#)
<http://www.mvpcluster.es>





Alberto Díaz

Alberto Díaz es SharePoint Team Lead en Encamina, liderando el desarrollo de software con tecnología Microsoft. Para la comunidad, ha fundado TenerifeDev (www.tenerifedev.com) con otros colaboradores, un grupo de usuarios de .NET en Tenerife, y coordinador de SUGES (Grupo de Usuarios de SharePoint de España, www.suges.es) y colaborador con otras comunidades de usuarios. Microsoft MVP de SharePoint Server desde el año 2011 y asiduo conferenciante en webcast y conferencias de tecnología de habla hispana.

Sitio Web: <http://blogs.encamina.com/negocios-sharepoint/>

Email: adiazcan@hotmail.com

Blogs: <http://geeks.ms/blogs/adiazmartin>

Twitter: [@adiazcan](https://twitter.com/adiazcan)



Fabián Imaz

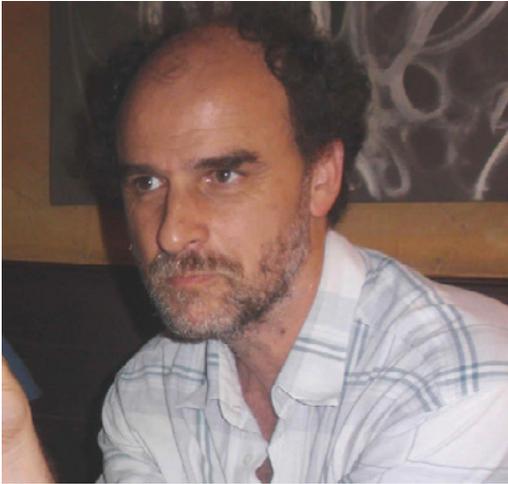
Fabián Imaz, MVP de SharePoint Server trabaja en el mundo del desarrollo de software desde hace más de 10 años, teniendo la suerte de trabajar en distintas arquitecturas y tecnologías Microsoft. Pertenece a la firma Siderys, <http://www.siderys.com> empresa de desarrollo de Software especializada en SharePoint 2007/2010/2013 y en desarrollo de soluciones inteligentes. Desde los comienzos Fabián ha trabajado en distintas comunidades donde organiza y promueve eventos locales para la difusión de tecnología dentro de los miembros de las mismas. Es director de la carrera SharePoint 2010 y SharePoint 2013 en Microsoft Virtual Academy, <http://www.mslatam.com/latam/technet/mva2/Home.aspx> y cuenta con un sitio en CodePlex con varios desarrollos <http://siderys.codeplex.com>.

Sitio Web: <http://www.siderysbsn.com>

Email: fabiani@siderys.com.uy

Blogs: <http://blog.siderys.com>

Twitter: [@fabianimaz](https://twitter.com/fabianimaz)



Gustavo Velez

Gustavo Velez es Ingeniero Mecánico y Electrónico; trabaja en el diseño e implementación de sistemas de IT basados en tecnologías de Microsoft, especialmente SharePoint, para Avanade (<http://www.avanade.com>), una compañía multinacional de IT. Propietario del sitio especializado en información sobre SharePoint en español <http://www.gavd.net> y autor de seis libros sobre SharePoint y sus tecnologías.

Sitio Web: <http://www.gavd.net>

Email: gustavo@gavd.net

Blogs: <http://geeks.ms/blogs/gvelez/>



Juan Carlos González Martín

Juan Carlos González, es Ingeniero de Telecomunicaciones por la Universidad de Valladolid y Diplomado en Ciencias Empresariales por la Universidad Oberta de Catalunya (UOC). Cuenta con más 10 años de experiencia en tecnologías y plataformas de Microsoft diversas (SQL Server, Visual Studio, .NET Framework, etc.), aunque su trabajo diario gira en torno a las plataformas SharePoint y Office 365. Juan Carlos es MVP de SharePoint Server desde el año 2008, coordinador del grupo de usuarios .NET de Cantabria (Nuberos.Net, www.nuberos.es) y co-fundador del Grupo de Usuarios de SharePoint de España (SUGES, www.suges.es), del Grupo de Usuarios de Cloud Computing de España (CLOUDES) y de la Comunidad de Office 365. Desde el año 2011 participa junto con Gustavo Vélez, Fabián Imaz y Alberto Díaz en la dirección de CompartiMOSS. Hasta la fecha, ha publicado cuatro libros sobre SharePoint y varios artículos en castellano y en inglés sobre la plataforma.

Email: jcgonzalezmartin1978@hotmail.com

Blogs: <http://geeks.ms/blogs/jcgonzalez> &

<http://jcgonzalezmartin.wordpress.com/>

¿Desea colaborar con CompartiMOSS?



La subsistencia del magazine depende de los aportes en contenido de todos. Por ser una revista dedicada a información sobre SharePoint en español, todo el contenido deberá ser directamente relacionado con Microsoft SharePoint y escrito en castellano. No hay limitaciones sobre el tipo de artículo o contenido, lo mismo que sobre el tipo de versión. Si desea publicar algo, por favor, utilice uno de los siguientes formatos:

- Artículos de fondo: tratan sobre un tema en profundidad. Normalmente entre 2000 y 3000 palabras y alrededor de 4 o 5 figuras. El tema puede ser puramente técnico, tanto de programación como sobre infraestructura, o sobre implementación o utilización.
- Artículos cortos: Máximo 1000 palabras y 1 o 2 figuras. Describen rápidamente una aplicación especial de SharePoint, o explica algún punto poco conocido o tratado. Experiencias de aplicación de SharePoint en empresas o instituciones puede ser un tipo de artículo ideal en esta categoría.
- Ideas, tips y trucos: Algunos cientos de palabras máximo. Experiencias sobre la utilización de SharePoint, problemas encontrados y como solucionarlos, ideas y trucos de utilización, etc.

Los formatos son para darle una idea sobre cómo organizar su información, y son una manera para que los editores le den forma al magazine, pero no son obligatorios. Los artículos deben ser enviados en formato Word (.doc o .docx) y las figuras por separado en un formato de alta resolución (.tif), todo comprimido en un archivo (.zip o .rar) con el nombre del autor y del artículo.

Si desea escribir un artículo de fondo o corto, preferiblemente envíe una proposición antes de escribirlo, indicando el tema, aproximada longitud y número de figuras. De esta manera evitaremos temas repetidos y permitirá planear el contenido de una forma efectiva.

Envíe sus proposiciones, artículos, ideas y comentarios a la siguiente dirección:

revista@compartimoss.com

adiazcan@hotmail.com

fabiani@siderys.com.uy

jcgonzalezmartin1978@hotmail.com

gustavo@gavd.net

