

COMPARTIMOSS

Revista especializada en Tecnologías SharePoint

DICCIONARIOS PARA EL MOTOR DE BÚSQUEDA DE SHAREPOINT 2013

ADMINISTRACION DE SHAREPOINT ONLINE MEDIANTE POWERSHELL

PUBLICANDO UN SITIO DE SHAREPOINT EN CYBEROAM

AUTOMATIZACIONES PARA EDITAR EL IIS EN SHAREPOINT 2010/2013



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

Parece que tenemos SharePoint para rato, y hablamos tanto de la versión On-Premises como de la versión Online.

04. Arquitectura SP distribuida en Azure

10. Diseño Gráfico en SharePoint Foundation 2010 Parte 3

14. Gobernabilidad de Seguridad en SharePoint para la Plataforma

17

The Shift Toward Personalized and Intelligent Collaboration

19. Entrevista a Adrián Díaz

21. Nuevas API's para desarrolladores en Office 365 - Parte II

27. Diccionarios para el Motor de Búsqueda de SharePoint 2013

34

Publicando un sitio de SharePoint en Cyberoam

37. Introducción a las App Forms con ACESS

42. Administración de SharePoint Online mediante PowerShell

46. Automatizaciones para editar el IIS en SharePoint 2010/2013

52. Creación de Soluciones personalizadas y automatizadas con Nintex



03

Editorial

Parece que tenemos SharePoint para rato, y hablamos tanto de la versión On-Premises como de la versión Online. No tenemos claro cuál es el Roadmap, ni cuando dejará de haber versión On-Premises, pero lo que si podemos asegurar es que esta revista intentará estar ahí, con artículos de calidad sobre nuestro servidor favorito.

Mientras SharePoint Online se convierte en la opción adecuada, Microsoft y nuestros autores están haciendo muchos esfuerzos para que nos subamos a Azure e implementemos SharePoint en modo IAAS, y como no podía ser menos, estás leyendo una editorial que se encuentra a medio camino entre On-Premises y Azure.

Como siempre, los dejamos con un número cargado de artículos impresionantes, con autores habituales y nuevos autores que esperamos sigan publicando porque sin ellos la revista no llegaría cada tres meses a vuestros ordenadores. Gracias a todos y esperamos que la disfruten como nosotros disfrutamos haciéndola.

EL EQUIPO EDITORIAL DE COMPARTIMOSS

El autor destacado del Número 20



Un nuevo número y vuestras lecturas nos aportan salud a la revista y con ellas sabemos que los artículos os aportan beneficios, lo que nos complica la elección del artículo destacado.

Los artículos por capítulos suelen tener éxitos y esta vez tenemos a Miguel Tabera, por su artículo “Buenas prácticas en infraestructura en SharePoint 2013 Parte II” como el artículo destacado del número 20. Felicitaciones para Miguel por su continuidad en la revista y por los buenos artículos que nos escribe, y por supuesto, al resto de autores que aportan su granito de arena en este proyecto.

Arquitectura SP distribuida en Azure

Resumen

Muchas veces nos planteamos el cómo llevar nuestra arquitectura On-Premise a Azure, para aprovechar de forma conveniente las ventajas de las que disponemos. En este artículo vamos a ver como montar una arquitectura distribuida entre varias suscripciones de Azure para disponer de resistencia ante fallos (Failover) y antes desastres (Disaster Recovery), manteniendo unos costes contenidos

Introducción

Azure es una de las grandes apuestas actuales de Microsoft, y de cara al aprovechamiento de los recursos (sobre todo económicos) de una empresa, presenta grandes ventajas ante una arquitectura On-Premise, especialmente respecto a la facilidad que nos brinda a la hora de hacer crecer nuestra arquitectura conforme a los requisitos y/o disponer de más potencia ante picos puntuales. Del mismo modo, esto nos permite el disponer de plataformas “pre-configuradas” como recurso ante caídas de sistemas.

Azure se ha colocado en muy poco tiempo a la altura, si no por encima, del resto de grandes plataformas cloud, en cuanto a funcionalidad, y mantiene un ritmo de actualizaciones que permite visualizar en un futuro muy corto funcionalidades que no están disponibles en ninguna de las actuales, pero al igual que SharePoint en sus inicios es un entorno que requiere de pequeños “trucos” que nos hagan más fácil el utilizar sus recursos. El dar a conocer estos trucos y procedimientos es la idea principal de este artículo en forma de lecciones aprendidas.

Como ejemplo a seguir, vamos a montar una infraestructura en dos suscripciones. La arquitectura será la mostrada en el siguiente diagrama:

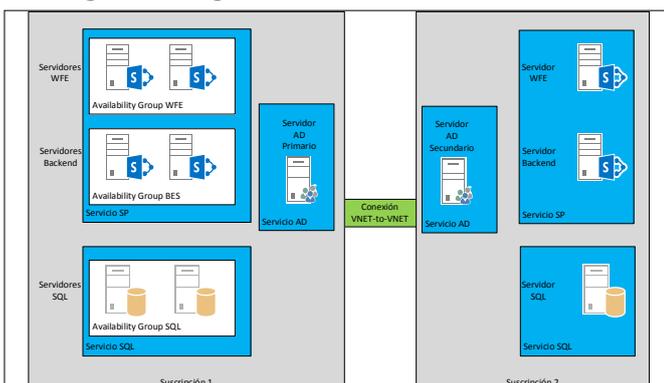


Imagen 1.- Diagrama de Arquitectura.

En esta arquitectura utilizamos varios de los recursos importantes de Azure:

- **Grupos de disponibilidad:** Al disponer varios servidores en un mismo grupo de disponibilidad, nos aseguramos que nunca estén sobre el mismo host físico, de manera que ante actualizaciones de la plataforma, nuestros servidores nunca serán rebotados de forma simultánea.
- **Conexiones VNET-to-VNET o Point-to-VNET:** Estas conexiones nos permiten la conexión entre nuestra suscripción y nuestra arquitectura On-Premise, o bien entre diferentes suscripciones
- **Servicios en la nube (Cloud Services):** Se considera el Cloud Service como cada una de las capas (tier) de una aplicación multicapa, agrupando los servidores que realizan una función definida (WebRole, WorkerRole. En nuestro ejemplo, SP, SQL y AD).

Por último, para sincronizar nuestras granjas, se establecerá una sincronización a nivel de SQL con Log Shipping para permitir que la granja secundaria actúe como Disaster Recovery. Esta configuración tiene la ventaja a nivel de

costes que es posible tener todas las máquinas apagadas (mientras no sea necesario instalar actualizaciones y/o ponerla en servicio) exceptuando el SQL. Para el ejemplo propuesto la granja secundaria esta simplificada, pero en el ejemplo real sería posible montar una estructura gemela a la de la granja primaria. El único coste que tendría esta arquitectura sería a nivel de almacenamiento (Storage) y de tráfico de red a través de la conexión entre suscripciones. En la práctica sería como tener el doble de servidores en nuestra arquitectura a un coste prácticamente idéntico. Como desventaja presenta el hecho de que al instalar actualizaciones/nuevos desarrollos en nuestra arquitectura primaria, también hay que realizar la misma acción en la secundaria, por lo que cobra especial relevancia el uso de un correcto [Plan de Gobierno](#). Esta desventaja puede ser mitigada usando scripts de arranque/parada de la granja secundaria y de despliegue.

Truco: Es posible acelerar nuestro despliegue realizando la generación de la red, servicios en la nube, grupos de disponibilidad y máquinas virtuales por medio de scripts PowerShell contra las conexiones de Azure y ejecutándolos (con los cambios pertinentes) contra ambas suscripciones

Prerrequisitos:

- Dos suscripciones a Azure (Suscripcion1 y Suscripcion2).
- Hemos descargado el certificado digital de Azure para poder acceder por PowerShell.
- Comprobar que desde el panel de suscripciones podemos cambiarnos de una a otra sin problemas.
- Dibujar nuestra arquitectura (esto no es imprescindible, pero ayuda).

Creación del grupo de Afinidad

Un grupo de afinidad se usa para agrupar todos los servicios en nube juntos, de forma que internamente se distribuyan en hosts cercanos, maximizando el rendimiento. Todo el escenario será generado por tanto dentro de tan solo un servicio de afinidad por suscripción.

Para generar nuestro nuevo grupo, seleccionaremos Settings o configuración, según el idioma seleccionado. Una vez dentro, seleccionar grupos de afinidad (Affinity Groups), y pulsar en +Add. Para cada grupo de afinidad, es necesario dar un nombre específico (en nuestro ejemplo suscripcion1) y seleccionar una región geográfica. A España le corresponde la región Europa Norte.

Esta operación será necesario replicarla en la segunda suscripción.

para sincronizar nuestras granjas, se establecerá una sincronización a nivel de SQL con Log Shipping

Creación del almacenamiento

Todas las unidades que tengamos dentro de nuestro ejemplo las crearemos dentro de una sola cuenta de almacenamiento, aunque en entornos reales el almacenamiento podemos distribuirlo dentro de diferentes cuentas de almacenamiento. En cada suscripción podemos disponer de hasta 5 cuentas de almacenamiento, y un máximo de 100 Tb en cada una. Conviene tener en cuenta que estos límites son revisados muy a menudo por Microsoft, y en muchos casos no son límites reales. Debido a esto, y si por la casuística de la implementación que estamos creando consideramos necesario ampliar alguno de los límites, la opción más adecuada consiste en ponerse en contacto con los administradores de Microsoft, pulsando en el menú de la suscripción, en la opción "Contact Microsoft Support".

Para crear la cuenta de almacenamiento, seleccionar en el menú lateral la opción Storage o almacenamiento, y pulsar sobre +New en la parte inferior. Aquí, seleccionamos

creación rápida, e introducimos:

- URL. Por ejemplo storage1.core.windows.net.
- Localización o Grupo de Afinidad. Seleccionamos el que hemos creado en el paso anterior.
- Tipo de redundancia. Las opciones son:
 - Localmente redundante. Tiene una copia de seguridad en el mismo CPD.
 - Geo redundante. Tiene una copia de seguridad en otro CPD de otra área geográfica.
 - GeoRedundante de lectura.
 - Redundancia de zona: Para nuestro ejemplo seleccionaremos redundancia local. Es conveniente estudiar con cuidado que opción de redundancia seleccionamos, puesto que tiene influencia en la facturación.

De nuevo, como en los pasos anteriores, este paso hay que hacerlo también en la segunda suscripción.

Creación de la infraestructura de red

Hay que tener en cuenta que ambas redes (suscripción uno y suscripción dos) no pueden estar en el mismo rango de red, por lo que vamos a crear ambas redes con subsegmentos distintos. Para ello:

- Pulsar sobre el  de la esquina inferior derecha. Seleccionar network services -> Virtual network --> Custom create. Darle un nombre (en nuestro caso Suscripcion1) y seleccionar un área geográfica (a España le corresponde North Europe).
- En el siguiente paso, dar un nombre al DNS, a pesar de no haberlo creado aún, y seleccionar la primera IP libre de nuestro rango. Hay que tener en cuenta que Microsoft reserva las tres primeras IPs de nuestro rango, por lo que habrá que seleccionar la 4ª. Por ejemplo, del rango 10.0.x.x, la IP del servidor DNS será la 10.0.0.4.
- Seleccionar el rango de IPs que utilizaremos. Para el ejemplo basta con disponer de 256 direcciones.

Al aceptar, se montará la red. Posteriormente repetir en la suscripción 2, teniendo en cuenta la precaución de cambiar el rango de IPs.

Importante

En caso de NO asociar la dirección del DNS en este paso, posteriormente nos puede dar problemas para asociarlo a la red.

Creación de los servicios en nubes (Cloud Services)

A continuación podemos crear los servicios en nube. Para ello, en cada suscripción, crearemos los tres servicios en nube que hemos dibujado. Para ello, en el menú lateral, seleccionar Cloud Services, y pulsar sobre el menú inferior en +new. El sistema abrirá la ventana para crear un nuevo servicio en nube.



Imagen 2.- Panel de Creación.

Como siempre, es recomendable seleccionar la creación custom. Pulsar sobre Custom Create, y seleccionar el grupo de afinidad, y la url de acceso. Todas las máquinas que están dentro de un mismo Cloud Services serán accesibles a través de una única URL, por lo que no pueden coexistir aplicaciones web, y se distingue entre ellas por medio del puerto asociado.

Creación de las máquinas

En este punto, y para el ejemplo mostrado, se modifican los pasos según el número de máquinas a crear. Para el ejemplo mostrado, debemos crear por suscripción:

Subscripción 1

- 1 máquina A1. Donde alojaremos nuestro AD.
- 6 máquinas A5, donde alojaremos nuestra granja SharePoint y nuestro SQL.

Subscripción 2

- 1 máquina A1 para el AD.
- 3 máquinas A5 para SharePoint y SQL.

La elección del tamaño A5 no es gratuito. Existe una limitación impuesta de 20 cores máximo por suscripción, por lo que el uso de máquinas A4, que en principio parecen más adecuadas para montar una infraestructura SharePoint, con 8 cores * 6 máquinas + 1 = 49 cores, por lo que no podríamos crearlas todas en esta suscripción, o tendríamos que solicitar el aumento del límite.

Truco: Aunque es posible crear las máquinas ya montadas, es recomendable la creación de máquinas básicas con el SO, y posteriormente realizar la instalación de cada sistema server manualmente, y con la licencia propia. De esta forma no se nos facturará por el uso de las licencias.

Para la creación de cada una de las máquinas, seleccionaremos en la parte inferior +New -> Compute -> Virtual Machine -> From Gallery

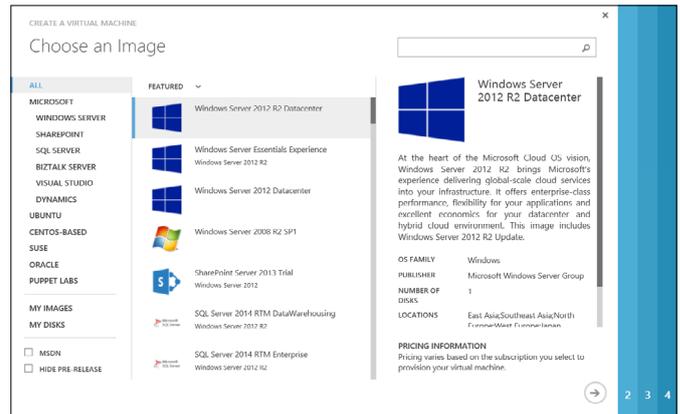


Imagen 3.- Galerías de Máquinas Virtuales.

En el segundo paso, hay que introducir:

- Release Date: Elegir siempre el último, excepto en casos excepcionales en los que no queramos específicamente un kb de una fecha. Suele haber al menos un par de snapshots.
- Nombre de la máquina virtual.
- Tier o Capa: Tipo de máquina virtual. Las básicas son más baratas pero a cambio tienen limitaciones como que no pueden hacer un escalado automático, lo cual es muy útil en algunas aplicaciones.
- Usuario y Password: Administrador local de la máquina creada. Existen nombres reservados como Administrador o Admin que NO pueden ser utilizados (se sobreentiende que es porque Microsoft los utiliza internamente).

En el tercer paso hay que seleccionar:

- Cloud Service: Hay que elegir uno de los creados anteriormente, aunque ofrece la opción de crear uno nuevo de forma automática.
- Opcional. Nombre de Cloud Service (en caso de no elegir uno de los creados).
- Opcional. Región/Grupo de Afinidad/Red virtual (en caso de no elegir uno de los creados).
- Storage Account: Elegir el que se creó en un paso anterior.
- Conjunto de disponibilidad: es extremadamente importante que, en la primera máquina creada de cada conjunto, lo creamos para evitar que ambas máquinas virtuales recaigan en el mismo host, y de este modo la granja estará disponible incluso en el caso de que el host sea reiniciado para mantenimiento.
- Nombre del Conjunto de disponibilidad: En el caso de seleccionar crear uno es necesario introducirlo.

En el cuarto paso, tan solo es necesario asegurarse que está marcada la opción de instalar el agente, y es posible instalar otros agentes, o antivirus, directamente en

la máquina. Aceptar y tras unos minutos estarán nuestras máquinas disponibles.

En este paso es recomendable NO PERSONALIZAR LAS MÁQUINAS. Tan solo es necesario crearlas conforme a nuestro diagrama original. En los siguientes pasos configuraremos cada una de nuestras máquinas.

Para instalar correctamente el controlador secundario, localizado en la subscripción 2, repetimos los pasos realizados para la creación del controlador primario

Asignación de IPs fijas (Opcional)

De forma nativa, Azure asigna las IP por un DHCP, de forma que en caso de parar las máquinas, estas vuelve a coger la IP según el orden de arranque. En caso de reinicio las máquinas mantienen su IP, pero si se paran y arrancan de nuevo, pueden coger una IP incorrecta.

Para evitar los errores que pueden surgir a partir de estos cambios, es posible “designar” la IP que debe coger cada máquina, pero en caso de aplicar esta opción, es necesario aplicarla en todas las máquinas de cada grupo de afinidad/red virtual. En caso de que al intentar arrancar la IP estuviese cogida, el sistema no podría arrancar la MV.

La reserva de IP privada no es posible realizarlo por medio de la interfaz, por lo que es necesario realizarla por medio de PowerShell, con la siguiente orden:

```
Set-AzureStaticVNetIP -VM $staticVM -IPAddress 10.7.115.55 |
Update-AzureVM
```

Creación del controlador de AD primario

Para crear el controlador primario, es necesario acceder a la máquina virtual, e instalar el rol de controlador de dominio. Una vez instalado, promocionar el servidor y configurar el dominio según deseemos.

Conexión entre las subscripciones

Esta es la parte más “divertida” de este artículo, y es como crear una conexión entre nuestras dos subscripciones. Recapitulando, tenemos creadas las redes virtuales, los servicios de nube, los conjuntos de disponibilidad, y las máquinas virtuales. Para que ambas granjas puedan ser sincronizadas, y podamos utilizar una como respaldo de la otra, es necesario que ambos controladores pertenezcan al mismo dominio, y por lo tanto tienen que tener conexión directa entre ellos. Para ello, vamos a crear una conexión

VNET-to-VNET entre ambas subscripciones.

Antes de comenzar, en la subscripción 2 vamos a registrar el servidor DNS de la subscripción primaria. Esto nos permitirá en el siguiente paso establecer nuestro controlador de AD de la subscripción 2 como controlador secundario del mismo dominio, y por ende nos dará resistencia ante caídas de la granja primaria. Para ello, nos iremos a Redes (Networks), pulsaremos sobre nuestra red creada (recordemos que estamos en la subscripción 2) y pulsaremos configuración (Settings). A media página disponemos de un apartado “Servidores DNS” donde podremos grabar el controlador ya creado en la subscripción 1 (por IP).

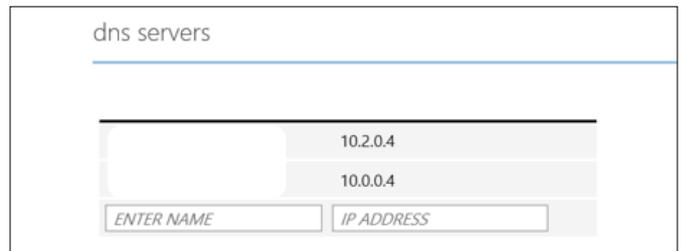


Imagen 4.- Registro de DNS.

Una vez realizado esto, es necesario añadir cada red como red local en la otra subscripción. Para ello:

1. En la parte inferior de la pantalla, pulsar en Nuevo. Seleccionar Servicio de Red (Network Service) y posteriormente Red Virtual (Virtual Network). Seleccionar añadir Red Local.
2. En la siguiente página, introducir el nombre que queramos dar a la nueva red. Para el ejemplo la denominaremos VNET1 y VNET2.

Como Dirección IP para el dispositivo VPN, deberemos utilizar la IP del Gateway, pero como en este momento aún no hemos creado el Gateway, podemos poner cualquier dirección IP, y posteriormente retornar a esta ventana para asignar la IP correspondiente cuando Azure la genere.

3. En el apartado de especificar el direccionamiento, deberemos poner exactamente el mismo direccionamiento que configuramos originalmente para la red a la que queremos conectar. Esto es, cuando estemos configurando la conexión desde la Subscripción 1, deberemos poner el direccionamiento configurado en la red de la Subscripción 2
4. Repetir los pasos para configurar como red local en la otra subscripción
5. Una vez realizado el paso anterior, podemos ir a la red creada y pulsar en la página de configuración. Bajo el apartado conectividad sitio a sitio (site-to-site connectivity) seleccionar conectar a una red local, y seleccionar la red que configuramos anteriormente (VNET2 en la subscripción 1, VNET1 en la subscripción 2).

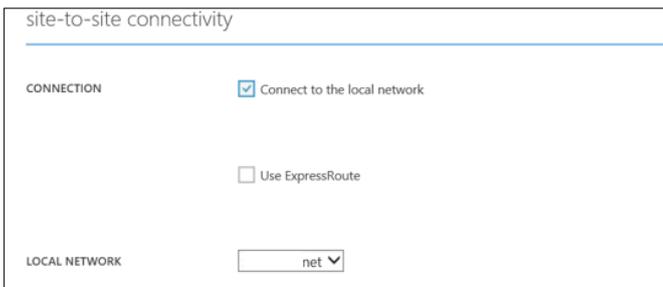


Imagen 5.- Configuración de las conexiones.

- En el siguiente apartado, denominado espacios de direcciones de red virtual (virtual network address spaces) pulsar sobre añadir subred de Gateway, y pulsar salvar al final de la página.



Imagen 6.- Sección de espacios de redes virtuales.

- Repetir los mismos pasos en la otra suscripción

En este punto, aún no hemos conectado ambas redes, puesto que no hemos establecido el Gateway de conexión. Para ello, desde el nombre del servicio de red, en la portada (Dashboard), deberemos pulsar en la parte inferior, deberemos pulsar sobre la opción Crear puerta de enlace Puerta de enlace dinámica. Al pulsar, el sistema necesitará un periodo aproximado de unos 15 minutos para terminar de crearlo, y el sistema mostrará un icono de conexión en amarillo. A continuación, es posible repetir la creación de la puerta de enlace en la segunda suscripción. No es necesario esperar a que finalice la creación de una para empezar la creación de la otra.

En algún momento, el panel de control de la red mostrará una IP de la puerta de enlace. Apuntar esta IP puesto que la usaremos posteriormente, indicando claramente a que suscripción pertenece.

Una vez tengamos ambas IPs, deberemos volver a la pantalla de configuración de la dirección IP de la VPN. Para ello, pulsando en el menú lateral networks, pestaña “Redes Locales”, seleccionar la red que creamos como local (VNET2 en la suscripción 2) y añadir la dirección IP del **gateway de la red remota que queremos conectar**. Esto es, las direcciones que cogimos anteriormente tenemos que colocarlas cruzadas (la de Gateway 2 en la suscripción 1 y la del Gateway 1 en la suscripción 2).

Por último, deberemos establecer la misma clave compartida a ambos gateways. Para ello, la forma más fácil es mediante PowerShell. En este ejemplo, lanzamos la configuración para nuestras redes creadas:

```
Set-AzureVNetGatewayKey -VNetName VNet2 -LocalNetwork-SiteName Suscripcion1 -SharedKey A1b2C3D4
Set-AzureVNetGatewayKey -VNetName VNet1 -LocalNetwork-SiteName Suscripcion2 -SharedKey A1b2C3D4
```

Otra opción es cambiar la clave compartida pulsando en redes, seleccionamos nuestra red (suscripción1) y pulsando en la parte inferior en gestionar clave (Manage Key).

Si todo ha ido bien, el sistema mostrará la conexión entre redes en verde, y empezará a mostrar el traspaso de datos entre ambas subredes.

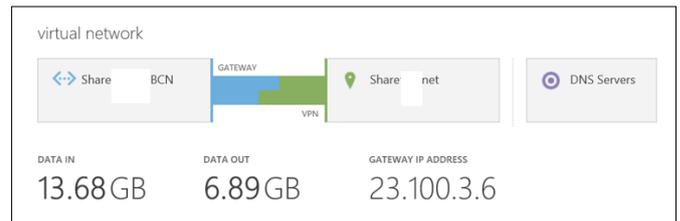


Imagen 7.- Redes virtuales creadas.

En caso de que la conexión quede mostrándose en amarillo, es recomendable parar y volver a arrancar todas las máquinas de la granja. Una buena prueba del funcionamiento es intentar hacer ping desde cualquiera de las máquinas de la infraestructura hacia la IP de una de las máquinas de la otra suscripción.

Creación del controlador de AD Secundario

Como ahora tenemos conexión entre ambas suscripciones, podemos configurar nuestra máquina controladora de AD de la suscripción 2 como controlador secundario para el dominio que establecimos anteriormente. Para instalar correctamente el controlador secundario, localizado en la suscripción 2, repetimos los pasos realizados para la creación del controlador primario, pero indicando que el servidor será un controlador secundario de dominio, indicando que el servidor primario es el creado en la suscripción 1.

Creación de las granjas SharePoint

Cae fuera del alcance de este artículo el indicar como realizar la instalación de una granja SharePoint, puesto que los pasos son los mismos que realizaríamos en una instalación On-Premise, pero queremos dejar dos recomendaciones:

- Utilizar siempre un procedimiento automatizado de instalación, bien mediante scripts PowerShell o bien mediante herramientas como AutoSPInstaller.
- Instalar de forma estricta los servicios que se vayan a necesitar, y **siempre utilizando la política de mínimos privilegios, que nos garantice la seguridad de nuestra granja**.

Para profundizar en el diseño e implementación de una



granja SharePoint en Azure, yo recomiendo el whitepaper [SharePoint 2013 on Windows Azure Infrastructure Services](#)

Sincronización entre las granjas

En este punto del ejemplo, nos quedaría sincronizar ambas granjas de forma que, en caso de caída de la granja primaria, podamos levantar la secundaria de una forma ágil para seguir prestando servicio a nuestros usuarios, y especialmente con la menor pérdida de datos posible. Para ello utilizaremos la sincronización por **Log Shipping** a nivel de SQL.

Desgraciadamente, el espacio del que disponemos en cada revista es limitado, por lo que esta configuración, y los próximos pasos los ejecutaremos en el siguiente número.

Próximos pasos

El escenario propuesto necesitaría de entre una y tres jornadas para montarlo completamente. Una vez establecido y funcionando, los siguientes pasos que se podrían realizar con ella dependen de los objetivos que nos propongamos:

- Podemos establecer un EndPoint público para

permitir que la granja sirva un sitio hacia internet, o bien establecer otra conexión vnet-to-vnet para conectarla a nuestra red interna de forma que sirva como intranet.

- Podemos gestionar el tráfico con un traffic manager.

A partir de aquí, ¡vuestra imaginación y los requisitos de los usuarios marcan el límite!! Espero que no se os haya hecho muy pesado, y si alguien necesita ayuda adicional, puede contactarme a través de mis perfiles o correo electrónico. ¡¡Gracias a todos los que hayáis aguantado hasta el final!

FABIÁN CALVO

Team leader

fcalvo@encamina.com

[@fcvspain](#)

<http://www.encamina.com>

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



Diseño Gráfico en SharePoint Foundation 2010 Parte 3

Resumen

En artículos anteriores hicimos una introducción acerca de cómo incorporar en SharePoint 2010 los estilos creados por un diseñador gráfico. Nos basamos en algunas hipótesis como el uso de SharePoint Foundation, páginas maestras, hojas de estilo y modelo de objetos de cliente para JavaScript. Se recomienda leer la [parte 1](#) y la [parte 2](#) antes de continuar. También les recomiendo leer los [ejemplos de CAML con Client Object Model](#), porque los utilizo para obtener datos de listas.

Introducción

En esta tercera etapa, vamos a trabajar con dos casos puntuales:

- Botonera gráfica en reemplazo del Inicio Rápido.
- Tablero de control con indicadores gráficos.

Botonera gráfica en reemplazo del Inicio Rápido

Lo primero que vamos a mostrar es cómo agregar una botonera con imágenes que reemplace al tradicional Inicio Rápido. Esta botonera se verá en todas las páginas. Para ello modificamos la página maestra y agregamos algunos estilos. El resultado final se ve en esta imagen:



Imagen 1.- Botonera gráfica en reemplazo de Inicio Rápido

Lo primero que vamos a mostrar es cómo agregar una botonera con imágenes que reemplace al tradicional Inicio Rápido

El siguiente código muestra el agregado de la botonera en la página maestra:

```
<div id="botonera_izquierda">
  <div id="menu">
    <ul id="nav">
      <li><a href="/"></a></li>
      <li><a href="/Lists/Noticias"></a></li>
      <li><a href="/Lists/Personal"></a></li>
      <li><a href="/Politicas"></a></li>
      <li><a href="/Formularios"></a></li>
      <li><a href="/Lists/Busquedas"></a></li>
      <li><a href="/Lists/Menu"></a></li>
      <li><a href="/Revistas"></a></li>
      <li><a href="/"></a></li>
      <li><a href="/Capacitaciones"></a></li>
    </ul>
  </div>
</div>
```

El código debe ser agregado justo antes de esta línea:

```
<div id="s4-leftpanel-content">
```

Por otro lado, agregamos los siguientes estilos en nuestra hoja de estilos:

```
#s4-leftpanel { width: 95px !important;}
#s4-leftpanel-content { display: none;}
.s4-ca { margin-left: 95px;}
```

Un detalle más, queremos que los botones cambien a otra imagen en el momento en que pasamos el mouse por sobre ellos (hover), tal como se ve en la siguiente imagen:



Imagen 2.- Efecto Hover en la botonera-

Con ese fin agregamos en la página maestra, el siguiente código jQuery:

```
<script type="text/javascript">
    $('#nav a img').hover(
        function() {
            $(this).attr('src', $(this).attr('src').replace('static', 'over'));
        },
        function() {
            $(this).attr('src', $(this).attr('src').replace('over', 'static'));
        }
    );
</script>
```

Queremos que los botones cambien a otra imagen en el momento en que pasamos el mouse por sobre ellos (hover),

Tablero de control con indicadores gráficos

Cómo segundo ejemplo del día, vamos a agregar unos indicadores gráficos que se vean como los de la siguiente imagen:

CALENDARIO 95% <small>Tiempo del proyecto.</small>	CALIDAD 1324 <small>Indicador de calidad.</small>	COSTO - \$78.000 <small>Evaluación de presupuesto.</small>	FACTURACION 70% <small>Estado de facturas.</small>
AL CANCE 20% <small>Complejidad de requerimientos.</small>	COMPRAS \$ 500.000 <small>Terminación.</small>	RIESGOS 0 <small>Nivel de riesgo.</small>	PROCESO 70% <small>Completamiento de los procesos.</small>

Imagen 3.- Ejemplo de Tablero de Control.

Estos indicadores serán generados a partir de información en listas usando CAML. En este caso usaremos una página de aplicación en C# con código de servidor por un tema de rendimiento, pero también podría ser JavaScript con Client Object Model. Dentro de nuestra página ASP.Net llamaremos a una hoja de estilos además de crear el DIV "contenido":

```
<asp:Content ID="Main" runat="server" contentplaceholderid="PlaceholderMain" >
    <div runat="server" ID="contenido" />
    <link href="dashboard.css" type="text/css" rel="stylesheet" id="stylesheet" />
</asp:Content>
<asp:Content ID="PageTitle" runat="server" contentplaceholderid="PlaceholderPageTitle" >
    Panel de Control
</asp:Content>
<asp:Content ID="PageTitleInTitleArea" runat="server" contentplaceholderid="PlaceholderPageTitleInTitleArea" >
    Panel de Control
</asp:Content>
```

Leeremos la información vía CAML. No voy a mostrar en el código la lectura de información porque está muy relacionada con la lógica del negocio y no es más que lecturas típicas en CAML. Lo que si mostraré es cómo incorporar esos valores dentro de dos DIVs pensados para estructurar la información:

- DIV dash01 para el valor del indicador 1.
- DIV dash01_detalle para el detalle del indicador 1.
- Y así sucesivamente.

```
contenido.InnerHtml = "<div class='semaforos'>";
contenido.InnerHtml += "<div id='dash01' class=''" +
    luz_indicador_1 + " dash'><p class='titulo'>CALENDARIO</p><p class='valor'>" + x_equipos_sin_linea + "</p><p class='leyenda'>Tiempos del Proyecto</p></div>";
contenido.InnerHtml += "<div id='dash02' class=''" +
```

Continuar el armado de DIVs para el resto de los indicadores.

```
contenido.InnerHtml += "</div>";
contenido.InnerHtml += "<div id='dash01_detalle' class='detalle'>" + detalle_indicador_1 + "</div>";
contenido.InnerHtml += "<div id='dash02_detalle' class='detalle'>" +
```

Continuar el armado de DIVs de detalle para el resto de los indicadores. Agregaremos además algo de código jQuery para que al presionar sobre un indicador, nos muestre la información de detalle. Para ello llamamos a los scripts correspondientes de ASPX:

```
Page.ClientScript.RegisterClientScriptInclude("dash_jquery",
    ResolveUrl("./jquery-1.8.2.min.js"));
Page.ClientScript.RegisterClientScriptInclude("dash_script",
    ResolveUrl("./dashboard.js"));
Page.ClientScript.RegisterStartupScript(this.GetType(), "dash_script", "asignar_botones0;", true);
```

Y este es nuestro código jQuery “**dashboard.js**”:

```
function asignar_botones () {
  $( document ).ready(function () {
    $(".dash").click(function () {
      var panel = "#"+$(this).attr("id")+"_detalle";
      $(".detalle").css("display","none");
      $(panel).slideDown("slow").css("overflow","visible");
      $(".dash").removeClass("dash_selected");
      $(this).addClass("dash_selected");
    });
  });
}
```

Por último nos queda incorporar nuestra hoja de estilo, cuidando que el diseño sea responsive y se adapte al ancho de la pantalla, como se ve en esta imagen (en una pantalla más angosta que la anterior):



Imagen 4.- Ejemplo de Tablero de Control con un ancho de pantalla más angosto.

```
.dash {
  border: solid white 2px;
}

.dash_selected {
  border: solid #333333 2px;
}

.semaforos div {
  float: left;
  width: 22%;
  margin: 3px;
  padding: 15px 15px 10px 15px;
}

.semaforos {
  font-family: tahoma;
  text-align: center;
  cursor: pointer;
}

.verde {
  background-color: #8CC152;
}

.rojo {
  background-color: #E9573F;
}

.amarillo {
  background-color: #F6BB42;
}

.gris {
  background-color: #C8C8C;
}
```

```
.valor {
  color: #fff;
  font-weight: 500;
  font-size: 36px;
  margin-top: 20px;
  margin-bottom: 10px;
}

.titulo {
  color: #fff;
  font-size: 17px;
  font-weight: 500;
}

.leyenda {
  border-top: solid white 5px;
  padding-top: 12px;
  color: #fff;
  font-size: 12px;
  height: 65px;
}

.verde .leyenda {
  border-top-color: #76AD3E;
}

.rojo .leyenda {
  border-top-color: #D1462C;
}

.amarillo .leyenda {
  border-top-color: #E0A52C;
}

.gris .leyenda {
  border-top-color: #737373;
}

.detalle {
  border-left: solid #E6E9ED 10px;
  width: 95%;
  height: 200px;
  margin: 5px;
  padding: 5px;
  font-size: 11px;
  font-family: tahoma;
  clear: both;
  display: none;
}

.detalle span {
  font-size: 11px;
  font-family: tahoma;
}

.detalle a {
  padding: 5px;
  font-size: 11px;
}
```

Conclusión

Esta fue la tercera entrega de esta colección de artículos dedicada a demostrar cómo se puede trabajar en Share-Point junto a un diseñador gráfico profesional. En mi opinión, esto es muy valorado por los clientes. Finalmente, les presento otra imagen del resultado final, ya vieron una en el artículo anterior. Espero como siempre que haya sido útil y cualquier duda me consultan.

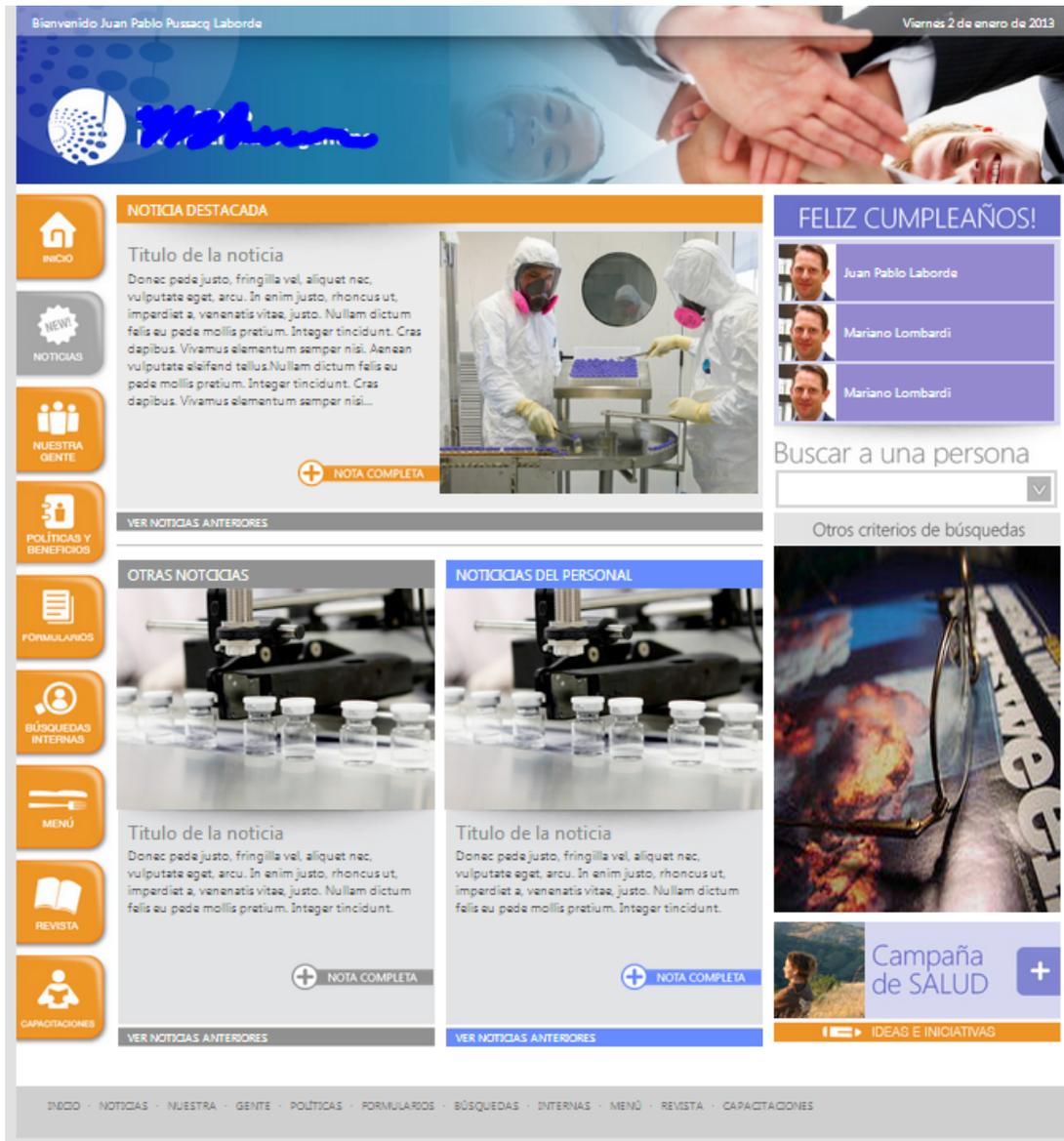


Imagen 5.- Diseño Gráfico aplicado a la página de inicio de SharePoint (caso 2).

JUAN PABLO PUSSACQ LABORDE

SharePoint MVP

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

Facebook: <http://facebook.com/surpointblog/>

Twitter: <http://twitter.com/jpussacq/>

Gobernabilidad de Seguridad en SharePoint para la Plataforma

Resumen

La Gobernabilidad es un concepto amplio que ya hemos tocado en nuestra revista. Sin embargo, la Gobernabilidad de Seguridad, es un foco particular que es transversal a la Gestión de la Infraestructura, Gestión de la Información y Gestión del Desarrollo. En este artículo, daremos una breve reseña, los controles que podemos definir desde la Gestión de la Infraestructura, pero que impactan en los otros dos ejes.

Introducción

Gobernabilidad es uno de los pocos conceptos de la administración de SharePoint que parecen tener muchos adeptos, pero pocos prácticos. A pesar que las organizaciones están seguras que un correcto gobierno de la plataforma, ayudará a su real potencial, son muy pocas las que logran llevar a un nivel madurez adecuado. A pesar que los tópicos del Gobierno en SharePoint nos acercan a tres ejes fundamentales (Gestión de la Información, Gestión del Desarrollo y Gestión de la Infraestructura), es innegable que una forma de acercarnos a este “gobierno adecuado”, es a través de la gestión de la seguridad, la que a su vez puede ser transversal a los ejes de gobierno antes mencionados.

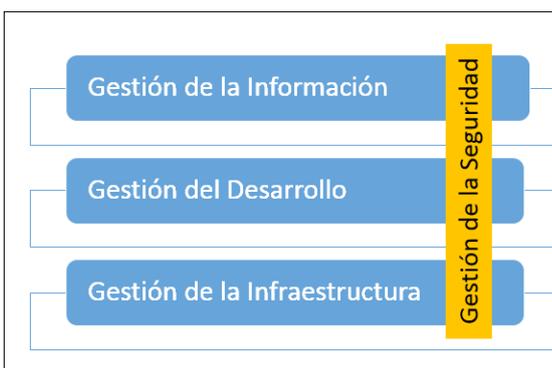


Imagen 1.- Gobernabilidad y Seguridad.

Si realizamos un seguimiento a la Gobernabilidad de Seguridad nos encontraremos con diferentes controles que debemos considerar, donde la clave es definir: Políticas y Roles.

La política establece el control explícito que debemos realizar según las necesidades de nuestra organización, además considerando las buenas prácticas del producto. En resumen, son puntos de focos a considerar para “Gobernar”, y atribuidos a roles específicos en la organización.

En el conjunto tendremos una plataforma segura, estable, y gobernada en cuanto a la seguridad.



Imagen 2.- Esquema de Gobernabilidad de Seguridad

A continuación generamos un repaso de algunos elementos funcionales y configuraciones que debiésemos considerar en un Gobierno Seguridad de SharePoint, y en la generación de Políticas y Roles.

Gobierno Seguridad TI

- El rol de SQL Server, en esta parte es fundamental. Entre los elementos a considerar en un Gobierno de Seguridad están.
 - Bloqueo de Puertos estándar. Implica el cambio de TCP 1433 en un puerto definido por los administradores de SQL.
 - Configuración de SQL Alias. Además de administrar la seguridad, esta política nos permite flexibilizar futuros movimientos de bases de datos a instancias distintas.

Esta es una buena forma de independizar el papel que tienen el administrador de SQL, de los administradores de SharePoint.

- “Pass Phrase” de Instalación. Los administradores se deben asegurar que el “santo y seña” de Instalación quede salvaguardado, sólo por los administradores de la Granja, independizando de esa forma el ámbito que tienen los administradores de bases de datos.

Contenido.

Contar con un Plan de Cuentas adecuado, que cumplan con las definiciones establecidos como requisitos en la documentación del producto. Estas cuentas deben estar correctamente definidas e identificadas en el escenario de Administración Central. Esta misma característica permitirá la habilitación del cambio de contraseña automático por parte de la propia Farm, esto en función de las programaciones de restablecimiento de contraseñas configuradas de forma individual por cada una de las cuentas de servicio. Esta funcionalidad de cambio de contraseña, permite no romper ninguna política corporativa relacionada con la caducidad de cuentas, y mantiene la posibilidad de recambio de password.

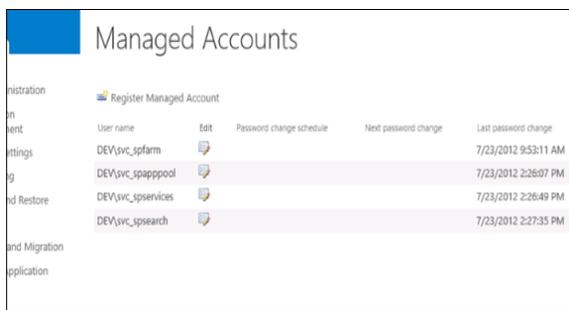


Imagen 3.- Pantalla de Administración de Cuentas Administradas.

- Delegación a Personas, de la Administración de la Granja. Es fundamental que los usuarios administradores de la granja, sean declarados en roles de "personas", y no necesariamente se utilicen las cuentas de servicios o administradas. Por ese motivo es que tenemos la posibilidad de administrar a este grupo selecto de personas, en un grupo independiente de "Administradores de la Granja".

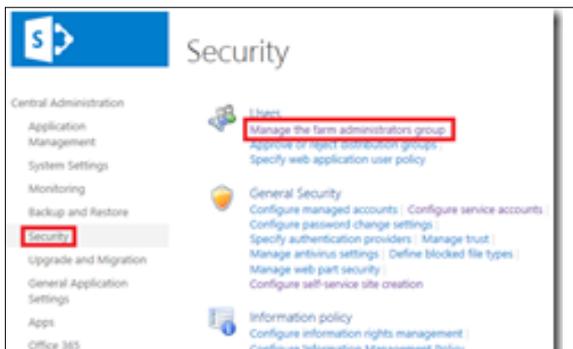


Imagen 4.- Acceso a administración de Grupo de Administradores de la Granja.

A pesar que los Administradores de Granja, con Control Total, pueden recaer en pocas personas, siempre existe la necesidad que un equipo más completo tenga privilegios de operación sobre la Administración Central. De todas maneras, vale la pena indicar, que pertenecer al Grupo de Administradores de Granja, opera, pero no necesariamente implica permisos sobre operaciones con PowerShell o en la creación de Bases de Datos de

- Administración delegada de aplicaciones de servicio: Otra de las políticas posibles de definir es definir en forma pormenorizada a los administradores de las aplicaciones de servicio. Esto permitirá que los administradores de la granja puedan delegar ciertas operaciones en cada una de las aplicaciones de servicio, y así independizar la posibilidad de configurar cada una de las necesidades de servicio.

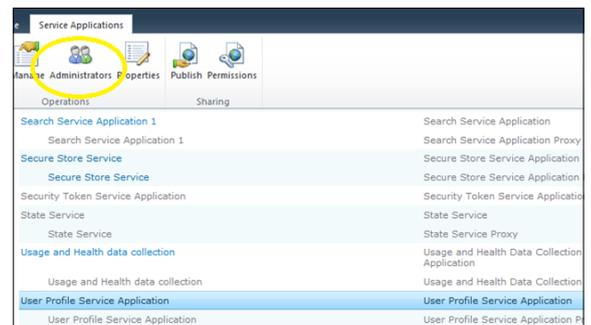


Imagen 5.- Opción de Ribbon que permite acceder a definición de Administradores por Aplicaciones de Servicio.

En ocasiones el grupo de TI, necesitará del apoyo o complemento de otras áreas que integran sistemas. Así es el caso de la Administración de BCS o Secure Store Services, donde administradores distintos a SharePoint aportarán datos importantes para la conexión a datos externos.

- Manejar una correcta administración de las Políticas de Seguridad aplicada a los usuarios para la Aplicación Web es fundamental, para entender las restricciones o atribuciones que podemos realizar a usuarios en Aplicaciones Web determinadas, y sus zonas de publicación. Este tipo de funcionalidad permite, por ejemplo de Restringir operaciones funcionales a usuarios, en ciertas zonas de una ubicación web (intranet, extranet, u otro), a pesar que en las colecciones de sitios el permiso esté otorgado. Contrariamente, permite otorgar obligatoriamente, los permisos de diferente nivel, a usuarios que no lo tienen en las colecciones de sitios en su interior.

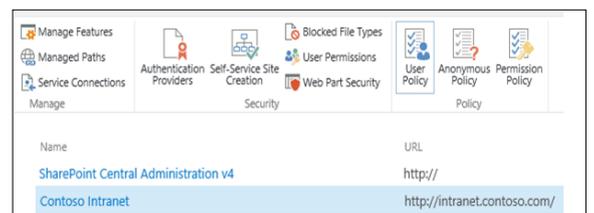


Imagen 6.- Opción de Ribbon en Administración Central, que permite generar Políticas específicas en las Aplicaciones Web para el acceso a usuarios.

Esta funcionalidad permitirá crear diferentes políticas, además de personalizar las ya existentes por defecto.

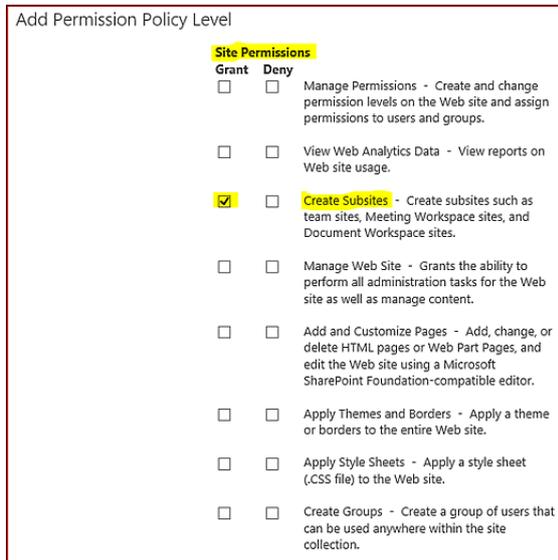


Imagen 7.- Detalles en la configuración de una Política particular.

La utilización de esta funcionalidad, es fundamental, por ejemplo; para otorgar privilegios de "Full Read" a la cuenta de servicio encargada del Rastreo de la Cuenta de Servicio en el servicio de Búsqueda.

- Definición de Administradores de la Colección de Sitios. Este quizás es uno de los elementos funcionales más utilizados en la entrega regular de la creación de Colecciones de Sitios. Sin embargo, que política definir en cuanto de la forma de entrega, es una discusión que en muchas organizaciones no ocurre. En este escenario, los administradores de la Granja, deben definir Administradores Principal y Secundario, pero la pregunta es ¿Quiénes deben ser esos administradores? Hay 3 combinaciones claras, que dependen del nivel de restricción que queremos definir en las Soluciones de los usuarios. Como una analogía simple, finalmente aquí determinamos quien obtiene "las llaves" de esta nueva casa.
 - Opción A: Los dos administradores son delegados en el área de TI. Eso significa restringir el papel de administración de la colección a los usuarios del "negocio", que finalmente sólo tendrán control como "Propietarios" de los sitios.
 - Opción B: Estas dos "llaves" son entregadas al área Usuaría, dejando completamente en manos de los usuarios el control funcional (Activación de Características), y el control de Contenidos (A un usuario Administrador de Colección de Sitio, no es posible restringir permisos, a menos que sea por una Política de Aplicación Web). Esto restringe el acceso al área TI, y los deja con poca capacidad de soporte funcional sobre los sitios.

- Opción C: Una de las más utilizadas, con-



Imagen 8.- Escenario de otorgación de Rol de Administración de Colección de Sitios a usuarios.

- Antivirus para SharePoint. A pesar de que Usuarios y Servidores pueden tener una buena protección con antivirus de Estaciones de Trabajo y Sistemas Operativos, no podemos desconocer el papel que significa contar con AntiVirus para SharePoint. Estos softwares tienen la habilidad de entender el modelo de objeto de SharePoint, de tal manera que puedan detectar el "movimiento" de archivos infectados hacia o desde los usuarios. Sin embargo, para utilizarlos, SharePoint necesariamente debe "autorizar" su operación y es por esto, que necesariamente, además de la instalación de estos productos en todos los servidores SharePoint de la Granja, se deben definir un conjunto de configuraciones.

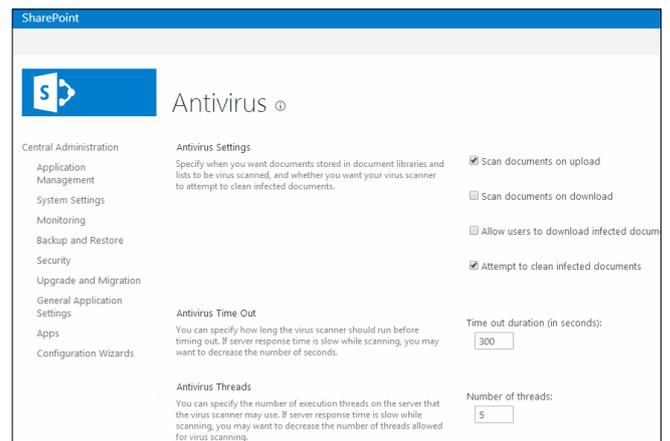


Imagen 9.- Pantalla de configuración de la acción de los antivirus "for SharePoint" en la Granja.

Los administradores de la Granja deben definir, entre otros:

- La posibilidad de escanear los archivos en la subida a SharePoint.
- La posibilidad de escanear los archivos en la bajada a los usuarios.
- Permitir a los usuarios bajar documentos infectados
- Permitir que los documentos infectados sean limpiados por el software.

JUAN ANDRÉS VALENZUELA
 MVP de SharePoint Server
jandresvalenzuela@outlook.com
[@jandresval](https://www.linkedin.com/in/jandresval)
<http://jandresval.wordpress.com>



17

The Shift Toward Personalized and Intelligent Collaboration

Abstract

The collaboration needs of end users are maturing, and simple document sharing and team sites no longer satisfy business requirements. The future of collaboration includes highly personalized solutions, catered to the specific needs of the individual, but plugged into the larger social collective of the organization, and with access to vast data sources. Organizations that recognize this shift in how teams are collaborating, and develop strategies around the future-looking knowledge management and business intelligence systems, will have a distinct competitive advantage over organizations that do not take action.

Article

Over the past decade, much of the focus of SharePoint has been around the capture and storage of content. For many companies using the platform, that focus will change -- in fact, that change is well underway, moving more and more toward the presentation layer where personalization, social interactions, and business intelligence are becoming a business priority. Even within Microsoft's latest collaboration messaging to "work like a network" the focus has moved away from document management-related features toward "seamless social experiences" that emphasize the user experience: listen to conversations that matter, adapt to make smarter decisions, and grow your business.

This is not just smart marketing, but recognition that the customer landscape has changed, and the user experience is the key to long-term collaboration success within the enterprise. While there are many drivers to the change in focus within Microsoft, there are two primary drivers, in my view, for this strategic shift: the acquisition of Yammer, and the persistent customer feedback surrounding SharePoint's adoption and engagement gap with end users.

Analyst firm IDC predicts that a majority of enterprises will implement some sort of social solution by the end of 2017, and according to Forrester, these same enterprises are increasingly placing a high priority on business intelligence (BI) platforms, in many circumstances adding this capability to established collaboration environments such as SharePoint. Both social and BI solutions have the ability to abstract complex data and business activities into conversations and visual representations, respectively, making content and business processes more consumable and

manageable by the masses.

For example, rather than rely on a workflow to route necessary documentation through stakeholders, social tools help participants interact, answer questions as they arise, and focus on the nuances of each customer request. As part of that social interaction, those stakeholders may use real-time geographical heat mapping to illustrate the size and impact of customer issues, allowing managers and front line workers an easier way to review massive amounts of customer data and quickly flag any "hot spots" which may be the result of regional issues. Increasingly, these kinds of dynamic interactions and data visualizations are being used within established collaboration platforms to help data owners quickly convey important information, and to allow people to interact and share that data as incidents occur.

SharePoint and Office 365 can provide governance, risk, and compliance capabilities similar to what is available offline

The mantra of software companies who sell social solutions is for "open collaboration" with as few regulations or end user controls as possible, which encourages adoption and, they argue, innovation. While BI solutions have historically catered to security-minded customers, much of the real innovation in the space is being directed toward the cloud. Customers with strong regulatory, compliance, or discovery requirements have been slow to broadly adopt these new solutions -- even though these new technologies may hold the answer to many known end user adoption or engagement issues.

While Microsoft's public announcements and roadmaps point toward a technology direction that will converge social, BI and personalization capabilities within their Office 365 platform, and across their market-leading Office web applications, for many enterprises there is still a "wait and see" attitude toward moving on premises production systems to pure-cloud solutions, which is where Microsoft's Office Graph and Delve interface (formerly codenamed Oslo) are available for customers. Until SharePoint and Office 365 can provide governance, risk, and compliance capabilities similar to what is available offline, these social

and BI features will continue to be a “nice to have” feature set for many CIOs. In short, most enterprises need to have some degree of visibility and control over their collaboration platforms before they move past “dipping their toes in the water” and roll them out enterprise-wide.

As more and more organizations begin to realize the business benefits of collaboration, the need for accurate measurements and analytics increases: How are people collaborating? Was my social collaboration deployment successful? Is it driving more business, better customer service, improved team communication, or any other business value metrics? The latest Forrester survey results indicate that enterprises are investing more now than ever in business intelligence (BI) platforms -- but are these initiatives successful in helping improve collaboration overall?

social collaboration is an effective method for getting teams to talk, to share, and innovate

Governance and management concerns aside, personalized experiences -- with a tight coupling of social interactions and data visualizations -- are the secret to improving end user engagement, pure and simple. Organizations are increasingly turning to web-based BI and social tools, mostly developed for the consumer market, to meet their collaborative needs. This is both exciting and scary -- especially if you don't have the tools in place to monitor and respond to these end user interactions. However, within the enterprise, it's not enough to simply see what people are discussing or how they are collaborating. What is needed is the ability to guide and direct end users toward secure and compliant activities -- without interrupting or discouraging their collaboration. Successful collaboration is not just about passive observance, but in taking the data and doing something with it.

In my many presentations on the topic, I try to address this shifting customer direction and provide advice with a pragmatic approach: move forward with both eyes open. My advice is to recognize these three truths:

- Companies are increasingly using social and BI tools, because social collaboration is an effective method for getting teams to talk, to share, and innovate. Likewise, BI tools have moved out of the realm of the highly technical business analyst or power user and into the hands of the business

stakeholder. You can call it “the consumerization of IT” or the maturity of the business user, but the fact remains that end users are not waiting to be handed the tools they need to get their work done.

- Companies are ill-prepared for this shift, because the leading platforms are focused almost entirely on delivering features to encourage collaboration, and are not focusing on the governance, risk, and compliance requirements that an enterprise requires to fully adopt and deploy these solutions. Most vendors view these requirements as being restrictive to end user collaboration. There is a widening gap between what end users want and what enterprise governance standards will allow.
- The organizations that can close this gap will have a distinctive competitive advantage over those who cannot, allowing them to offer their end users unfettered collaboration within secure and compliant environments.

Different people consume data in different ways -- some prefer spreadsheets, pivot charts, and tried-and-true database management techniques to capture, manipulate, and analyze complex data. For organizations that rely heavily on structured collaboration platforms like SharePoint, or even unstructured collaboration tools like Yammer, it is becoming increasingly important for power users, administrators, and leadership teams alike to be able to capture, track, and represent what is happening within their collaboration platforms.

Social interactions and BI dashboards and data visualizations are increasingly the “norm” in how we share complex data. As SharePoint evolves to meet the needs of shifting customer demands, the goal for organizations is to make business processes and data more user-friendly and consumable for stakeholders, focusing less on the platform and more on business productivity. Organizations that can make this shift will find themselves to be more nimble and responsive to changing customer needs, and will have a distinct advantage over their competitors in their ability to meet market demands.

CHRISTIAN BUCKLEY

Chief Evangelist at Metalogix

SharePoint Server MVP

cbuck@metalogix.com



19

Entrevista Adrián Díaz

Mi nombre es Adrián Díaz Cervera, vivo en el Puerto de Sagunto, una pequeña ciudad a las afueras de Valencia rodeado de una de las mejores playas de España.

Llevo más de 10 años trabajando con tecnología Microsoft, comenzando con Visual Basic versión 5. Casualidades del destino encontré mi sitio laboral en mi pueblo cuando empecé a dar los primeros pasos con ese gran producto llamado SharePoint. Desde ese momento mi interés sobre el mismo ha ido creciendo, aprendiendo de los expertos e intentando ayudar en todo lo que pueda, de la misma forma que a mí me han ayudado gente como Juan Carlos, Gustavo, Fabian o Alberto.

Actualmente trabajo en Encamina, una empresa especializada en SharePoint, donde tengo la responsabilidad de diseñar soluciones para extender y aprovechar las capacidades de SharePoint.

El pasado mes de enero, para comenzar el año con buen pie tuve la fortuna de ser reconocido como MVP de SharePoint.



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

La verdad es que empecé relativamente pronto mi idilio con la informática, parte de la culpa de ello la tiene mi padre que se aficionó a los ordenadores cuando yo tenía 8 años. Aunque él lo utilizaba para trabajar, yo tenía que buscarme la vida para poder jugar. Como no le gustaba darme las cosas sencillas tenía que aprender a poder arrancar los juegos y para ello aprendí a montar diskettes de 5 1/4 y lanzar ejecutables. Que tiempos! Ya un poco más mayorcito a los 15 años ya tenía claro que iba a estudiar informática, empecé con lenguajes como Turbo Pascal o Turbo C, para pasarme a Visual Basic/ Delphi en la carrera universitaria, pero siempre tenía claro que me encantaba desarrollar y de ahí que hiciera diversos pinitos con Java, PowerBuilder, Access para volver a Visual Studio. Aunque tengo que reconocer que la parte IT nunca la he abandonado ya que siempre me ha gustado poder ser autosuficiente y no depender de nadie para poder tener mis entornos de desarrollo.

¿Cuáles son tus principales actividades tecnológicas hoy en día?

Dado que hace ya unos cuantos años, debido a que la monotonía del lugar donde trabajaba me hizo estancarme profesionalmente, trato de mantenerme lo más actualizado posible desde el punto de vista tecnológico. Para ello lo primero que realizo por las mañanas es leer los RSS con los blogs preferidos. Además nada más sale un nuevo producto, ya estoy probándolo e intentando ver en qué parte puedo utilizar.

Otra de las cosas que realizo es analizar los productos/tecnologías competencia a .NET/SharePoint para poder ver en qué aspectos se puede mejorar el producto y por lo tanto podemos ofrecer una solución SharePoint que se caracterice por maximizar los beneficios y disimule los puntos débiles de SharePoint.

Como actividades tecnológicas (aunque puede considerarse como hobbies) me gusta dedicarme a aspectos de comunidad, hablando con expertos o profesionales para poder tener cierta opinión sobre aspectos tecnológicos.

Ayudar en los foros y sobre todo participar en todos los eventos que pueda, ya sea organizando y/o participando en los mismos.

¿Cuáles son tus principales actividades NO tecnológicas hoy en día?

Principalmente me gusta mucho estar rodeado de las personas que quiero, mi mujer Silvia, mis soletes Leyre y Marc, y mis amigos. Soy muy familiar e intento pasar todo el tiempo que pueda con ellos. Disfrutar de todos los momentos que te dan los niños ya que cada día es una sorpresa lo cual te da una vitalidad y ganas de disfrutar que no está escrito en ningún sitio.

¿Cuáles son tus hobbies?

La informática es uno de mis hobbies (y tengo la suerte que trabajo en lo que me gusta), pero aparte me encanta mucho los deportes. Principalmente me gusta mucho el fútbol, pero como decía mi abuela te gustan hasta las carreras de burros. También me gusta mucho jugar a la videoconsola (aunque la tengo un poco abandonada) Además me encantan las películas de superhéroes (Superman y Spiderman son mis favoritos). Me gusta bastante escuchar Rock And Roll y principalmente ponerme música de Héroes de Silencio.

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

Creo que la informática va a sufrir un gran cambio en los últimos años, hoy en día cada vez es más normal estar plagados de dispositivos móviles, lo que está haciendo que cada vez el usuario de las aplicaciones sea muy distinto. Todo este cambio se ve reflejando poco a poco en nuestra rutina diaria hasta que dentro de unos años sin habernos dado cuenta estén más presente dentro de nuestra vida diaria.

A nivel "profesional" creo que todo el cambio hacia el cloud, invita a nuevas arquitecturas, nuevos conceptos y múltiples desafíos para poder aprovechar todas las ventajas de la nube. Todo este cambio llevará un gran cambio en la forma en la que actualmente planteamos los proyectos IT lo que para mi gusto hace de la informática una profesión en la que no te aburres.

ADRIÁN DIAZ CERVERA

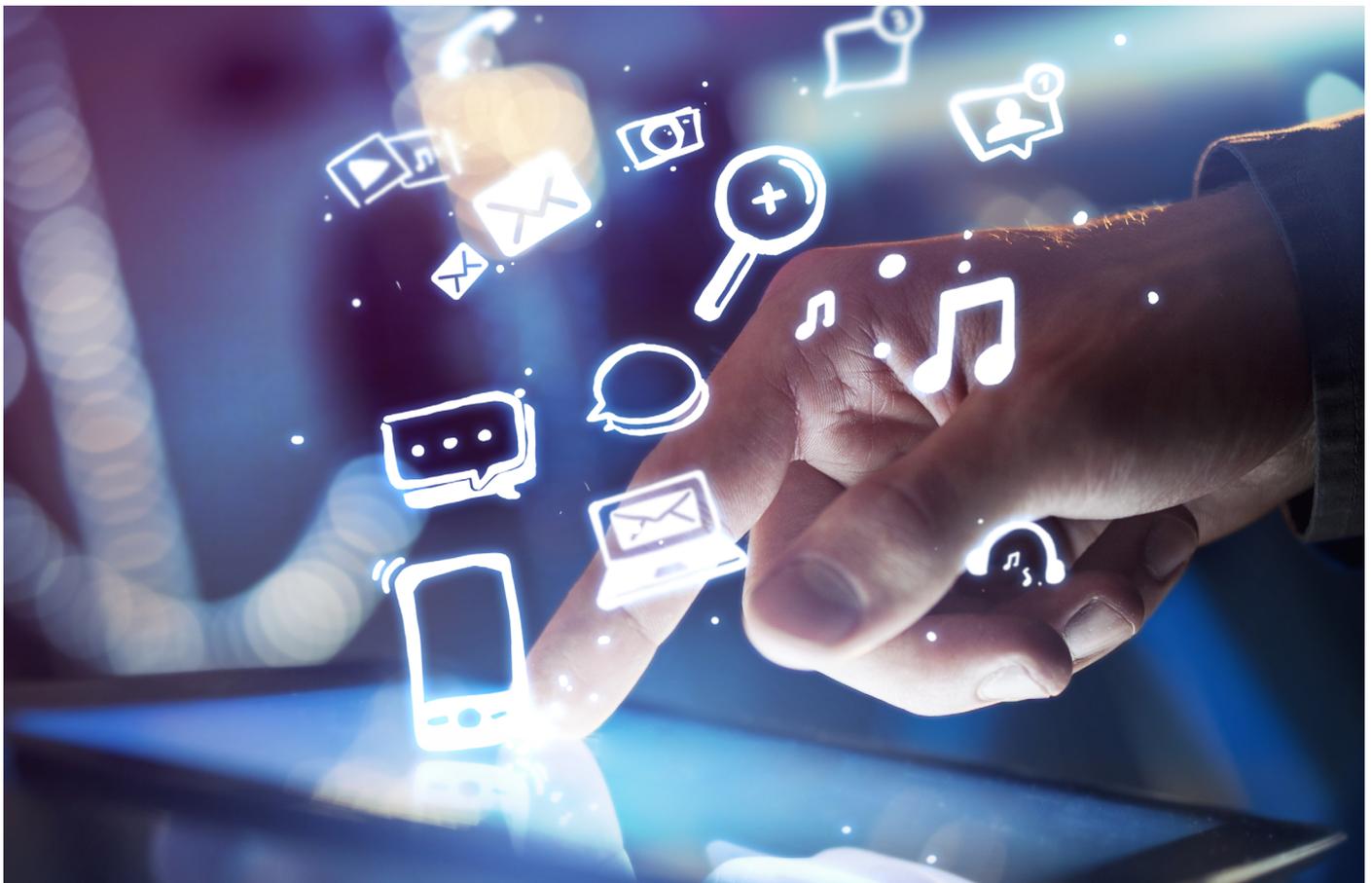
SharePoint Architect at Encamina

MVP SharePoint Server

<http://blogs.encamina.com/desarrollandosobresharepoint>

<http://geeks.ms/blogs/adiazcervera>

adiaz@encamina.com @AdrianDiaz81





21

Nuevas API's para desarrolladores en Office 365 Parte II

Resumen

Continuando con el artículo del número anterior, vamos a seguir desgranando las nuevas API's en Office 365. Vamos a describir las opciones disponibles en la API de SharePoint para el acceso a ficheros y la API de Active Directory.

Artículo

Nota: Estas APIs están actualmente en fase de Preview y pueden sufrir modificaciones cuando se lance la versión definitiva. Naturalmente, no se recomienda el uso de ninguna de estas APIs en un entorno de producción.

Introducción

Tal y como comentamos en el anterior número de CompartiMOSS, estas nuevas API's de Office 365 tienen como finalidad facilitarnos nuestros desarrollos para la plataforma de productividad de Microsoft en la nube. Y de la misma forma que se ha desarrollado API REST en SharePoint, se van a implementar servicios REST para acercar Office365 a nuestros desarrollos.

Desgranando las API de Exchange Online

La API REST de Ficheros representa un rediseño del almacenamiento de archivos y gestión de ficheros en SharePoint. La API de Archivos permite acceder y manipular tanto el contenido de los documentos de Office como los documentos almacenados en mensajes de correo electrónico, calendario, contactos y datos de SharePoint. La API de ficheros tiene como principal utilidad establecer una forma simple y sencilla de acceder a los ficheros y carpetas independientemente de la plataforma donde este utilizada. Esta API contiene acciones sobre dos elementos ficheros y carpetas, para facilitar su uso han establecido 3 entidades: FileSystemItem, File y Folder.

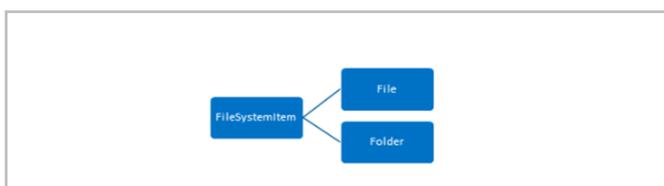


Imagen 1.- Relación entre las entidades de la API

- FileSystemItem — Clase base que encapsula las

operaciones básicas y propiedades de las entidades del sistema de archivos.

- File — Tipo de entidad que define los archivos del sistema de archivos. Hereda de FileSystemItem.
 - Folder — Tipo de entidad que define las carpetas del sistema de archivos. Hereda de FileSystemItem.
- La definición de las entidades es la siguiente:
- FileSystemItem:

Nombre	Tipo	Descripción	Creación	Update	Null
Id	Guid	Id del elemento	No	No	No
ETag	String	ETag del elemento	No	No	No
Name	String	Nombre del elemento	Requerido	Sí	No
Url	String	Path absoluto	No	No	No
TimeLastModified	Fecha / Hora	Hora de modificado	No	No	No
CreatedBy	Usuario	Usuario	No	No	No
TimeCreated	Fecha / Hora	Fecha de creación	No	No	No
LastModifiedBy	Usuario	Usuario que ha modificado	No	No	No
Size	Entero	Número de bits del documento	No	No	No ⁹

- File: Tiene las mismas propiedades de FileSystemItem.
- Folder:

Nombre	Tipo	Descripción	Creación	Update	Null
Children	Colección de FileSystemItem	Ficheros y carpetas que están incluidas dentro de esta carpeta	No	Sí	Sí
ChildrenCount	Entero	Número de ficheros que tiene esa carpeta	No	No	No

Operaciones con ficheros

Tenemos las siguientes endpoints disponibles:

- **Obtener todos los ficheros y carpetas en la biblioteca por defecto:**

```
GET ../_api/files
```

- **Obtener el contenido del fichero:**

```
GET ../_api/files(<file_path>)/download
```

La devolución de esta petición es el propio fichero

- **Obtener las propiedades del fichero:**

```
GET ../_api/files(<file_path>)
```

- **Obtener la propiedad del fichero:**

```
GET ../_api/files(<file_path>)/<property_name>
```

- **Crear un fichero:** A la hora de creación de un fichero tenemos dos opciones: crear un fichero en blanco o crear un fichero con contenido. Para crear un fichero en blanco tendremos que enviar la siguiente estructura JSON en el body de la petición Ajax.

```
{
  'metadata':
  {
    'type': 'MS.FileServices.File'
  },
  Name : '<FileName>'
}
```

Realizaremos la siguiente petición:

```
POST ../_api/files
```

Para crear un fichero con contenido tendremos que enviar en el cuerpo de la solicitud enviaremos un vector de bits y realizar la siguiente petición:

```
POST ../_api/files/Add(string name, boolean overwrite)
```

Si ha ido todo de forma correcta la API nos devuelve las propiedades del fichero recién creado.

- **Actualizar el contenido de un fichero:** Para actualizar el contenido de un fichero, basta con realizar la petición anterior, pero en el body enviamos el contenido modificado.

- **Eliminar un fichero**

```
DELETE ../_api/files(<file_path>)
```

Operaciones con carpetas

Disponemos de las siguientes opciones:

- **Obtenemos las propiedades de la carpeta:**

```
GET ../_api/files(<folder_path>)
```

- **Obtener una propiedad de una carpeta:**

```
GET ../_api/files(<folder_path>)/<property_name>
```

- **Obtener el contenido de una subcarpeta:**

```
GET ../_api/files(<folder_path>)/SubCarpeta
```

- **Crear una carpeta:**

```
POST ../_api/files
```

- **Eliminar una carpeta:**

```
DELETE ../_api/files(<folder_path>)
```

- **Mover un fichero:**

```
POST ../_api/files(<file_path>)/MoveTo(<file_path>, overwrite=true)
```

- **Copiar un fichero:**

```
POST ../_api/files(<file_path>)/CopyTo(<file_path>, overwrite=true)
```

Nota: Para las opciones Mover un fichero y Copiar un fichero, la biblioteca destino no puede ser distinta que la biblioteca donde está el fichero.

Operaciones en la Consulta

Tanto en las operaciones de ficheros como en las carpetas podemos añadir diversos parámetros en la consulta a realizar como son indicar el número de ítems que nos devuelve la consulta, como establecer el orden en el que se devuelve la petición. Para ello realizaremos las peticiones de la siguiente forma:

- **\$orderby:**

```
GET ../_api/files?$orderby=<orderbyfield>
```

- **\$top:**

```
GET ../_api/files?&top=<row_limit>
```

Desgranando la API Graph de Azure Active Directory

La API Graph de Azure Active Directory proporciona acceso programático a Azure AD a través de los extremos de la API de REST. Las aplicaciones pueden usar la API Graph para hacer operaciones de creación, lectura, actualización y eliminación (CRUD) en objetos y datos de directorio. Por ejemplo, la API Graph permite realizar las siguientes operaciones comunes en un objeto de usuario:

- Crear un usuario.
- Obtener las propiedades detalladas de un usuario.
- Actualizar las propiedades de un usuario, como su ubicación, número de teléfono, o cambiar su contraseña.
- Consultar la pertenencia a grupos de un usuario.
- Deshabilitar la cuenta de un usuario o eliminarla por completo.

Operaciones con usuarios

• Crear usuario:

Para la creación de un usuario realizaremos la siguiente petición POST:

```
https://graph.windows.net/mytenantdomain/users?api-version=2013-04-05
```

Donde tendremos que sustituir *mytenantdomain* por el nombre de nuestro dominio. En esa petición tendremos que enviar en el cuerpo de la solicitud un formato JSON como el siguiente:

```
{
  "accountEnabled": true,
  "displayName": "Adrian Díaz",
  "mailNickname": "adiaz",
  "passwordProfile": { "password" : "xxxxx", "forceChangePasswordNextLogin": false },
  "userPrincipalName": "adiaz@encamina.com"
}
```

• **Obtener usuario:** Para obtener el usuario podremos utilizar cualquiera de las siguientes uris, el parámetro de la de la cadena de consulta de api-version es obligatorio.

```
https://graph.windows.net/mytenantdomain/users?api-version=2013-04-05
```

```
https://graph.windows.net/mytenantdomain/users/<objectId>?api-version=2013-04-05
```

```
https://graph.windows.net/mytenantdomain/users/<userPrincipalName>?api-version=2013-04-05
```

Nota: El URI sin una especificación de usuario devolverá todos los usuarios en el directorio. Para obtener un usuario específico, especifica la propiedad *objectId* o *userPrincipalName* del usuario en el URI de solicitud.

• **Actualizar usuario:** Para actualizar un usuario disponemos de dos opciones bien indicando el *ObjettID* o el *User Principal Name*.

```
PATCH https://graph.windows.net/mytenantdomain/users/<objectId>?api-version=2013-04-05
```

```
PATCH https://graph.windows.net/mytenantdomain/users/<userPrincipalName>?api-version=2013-04-05
```

En el cuerpo de la solicitud enviaremos un JSON con las propiedades que deseamos modificar. Por ejemplo para modificar el departamento y la ubicación del usuario sería de la siguiente forma:

```
{
  "department": "Development",
  "usageLocation": "ES"
}
```

• Eliminar usuario

```
DELETE https://graph.windows.net/mytenantdomain/users/<objectId>?api-version=2013-04-05
DELETE https://graph.windows.net/mytenantdomain/users/<userPrincipalName>?api-version=2013-04-05
```

Si la operación ha ido correctamente la API REST nos devolverá un código de estado 204 Sin Contenido

• Obtener grupos a los que pertenece el usuario

```
GET https://graph.windows.net/mytenantdomain/users/<objectId| |userPrincipalName>/$links/directReports?api-version=2013-04-05
```

Para recuperar las personas que dependen directamente de un contacto, especifica "contacts" como el conjunto de recursos. También puedes especificar "directoryObjects" como el conjunto de recursos en el URI de solicitud; por ejemplo, <https://graph.windows.net/contoso.onmicrosoft.com/directoryObjects/5e624f44-d38d-4943-b07c-2bad078f52ff/directReports?api-version=2013-04-05>. Solo se puede usar *objectId* cuando el conjunto de recursos es "contacts" o "directoryObjects".

• **Resetear la contraseña del usuario.** Tendremos que realizar la siguiente petición:

```
PATCH https://graph.windows.net/mytenantdomain/users/<objectId>?api-version=2013-04-05
```

PATCH

```
https://graph.windows.net/mytenantdomain/users/<userPrincipalName>?api-version=2013-04-05
```

```
{
  "passwordProfile":
  {
    "password": "Test123456",
    "forceChangePasswordNextLogin": false
  }
}
```

Nota: En todas las peticiones hay que enviar dentro de la cabecera la Autenticación, que no es más que un token de portador emitido por el Control de acceso de Azure. Este token se genera en cuanto la aplicación que estamos implementando se autentifique contra el Active Directory.

Ejemplo: Utilizar estas API's en una aplicación Windows 8.1

Requisitos Previos

- Visual Studio 2013.
- Descargar Office 365 API Tools bien directamente desde esta dirección <http://visualstudiogallery.msdn.microsoft.com/7e947621-ef93-4de7-93d3-d796c43ba34f> o directamente desde Nuget.

Manos a la obra

Una vez tenemos instalada las Office 365 API Tools, abrimos Visual Studio, creamos un proyecto Aplicaciones de la Tienda-> Aplicaciones Windows-> Aplicación Vacía. A continuación, a nuestro proyecto le vamos a añadir la conexión a los servicios REST de API 365. Para añadirlos, nos dirigimos al Explorador de soluciones, Agregar Servicio conectado tal y como se muestra en la siguiente imagen:

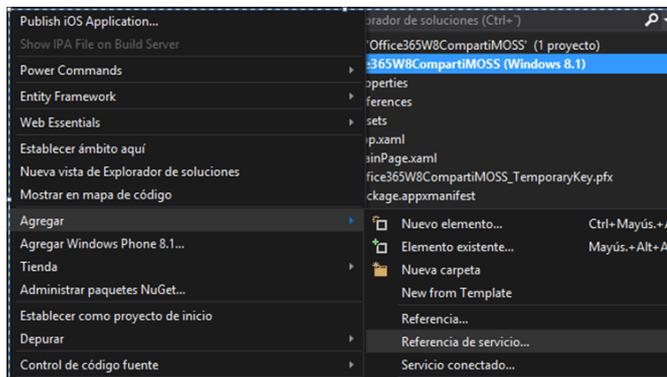


Imagen 2.- Añadir Servicio Conectado a nuestra APP MVC

Aparecerá el Administrador de servicios. Selecciona Office365 e introduce el login de tu cuenta. Una vez autenticado contra el Tenant de Office365 podrás ver una lista de 6 servicios: Calendar, Contacts, Mail, My Files, Sites y Users/Groupst. Inicialmente, la columna Permisos a la derecha de cada servicio estará vacía. En nuestro ejemplo vamos a seleccionar primero My Files y posteriormente Users/Groups. Una vez seleccionado pulsamos Aceptar (Imagen 2).

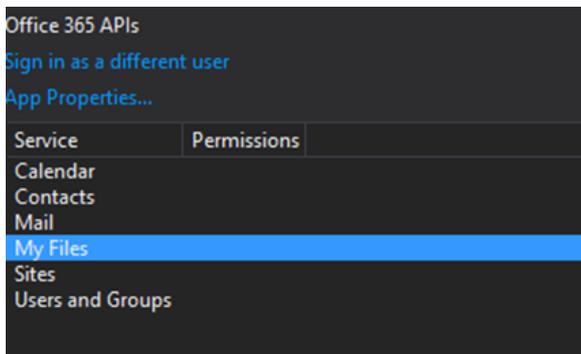


Imagen 3.- Administrador de Servicios.

Dado que no están establecidos los permisos, al tratarse de una versión preview, estos se solicitan por pantalla. En nuestro caso, vamos a otorgar todos los permisos para poder ver todas las opciones que vienen incluidas y que hemos visto en este artículo.

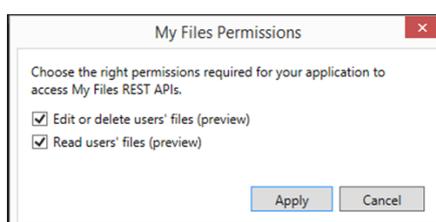


Imagen 4.- Permisos sobre las API's de My Files.

Una vez hemos añadido la API a nuestro proyecto revisamos la estructura de nuestro proyecto que quedará de la siguiente forma:

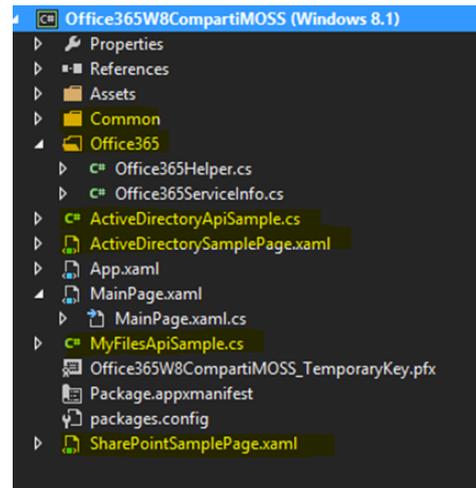


Imagen 5.- Solución Proyecto Office365W8CompartiMOSS.

Tal y como hemos remarcado en la imagen, se han añadido dos carpetas: Common y Office365. Además de una serie de archivos XAML donde están las pantallas para poder comprobar la potencia de estas APIs. Si analizamos los ficheros que se han añadido:

- **Office365ServiceInfo:** En esta clase disponemos del modelo que se utiliza para obtener la información para cada una de las APIs, para el caso de Active Directory tendremos el siguiente método:

```
public static async Task<Office365ServiceInfo> GetActiveDirectoryServiceInfoAsync()
```

Devuelve información sobre el servicio de Active Directory, incluida su token de acceso. En caso de error, este método se mostrará un mensaje de error al usuario, y devolverá una instancia de Office365ServiceInfo cuya propiedad HasValidAccessToken se establece en "false".

En el caso de SharePoint tendremos el siguiente método:

```
public static async Task<Office365ServiceInfo> GetSharePointOneDriveServiceInfoAsync()
```

Este método devuelve información sobre el servicio de SharePoint, incluyendo su token de acceso en caché. Para cada SharePoint, el ID de recurso y la API será diferente para cada inquilino. En caso de error, al igual que en el servicio de Active Directory se muestra un mensaje de error y devolverá una instancia de Office365ServiceInfo instancia cuya propiedad HasValidAccessToken se establece en "false".

- **Office365Helper:** Implementa una clase para ayudarnos en nuestros desarrollos. Entre las funciones que están implementadas: ClearCache, GetToken, SaveInCache, etc.
- **ActiveDirectoryApiSample** y **MyFilesApiSample:** En estas

clases están las funciones que llaman a la API REST correspondiente.

- **Ficheros XAML:** En estos ficheros están la representación de los datos que nos envían las API's correspondientes.

Una vez analizado el código, en la página principal de nuestra aplicación (MainPage.xaml) tendremos que añadir la funcionalidad para poder ir a las ventanas que nos ha descargado las herramientas tools de Office 365.

En primer lugar modificaremos el XAML de MainPage.xaml añadiendo el siguiente código dentro del GRID que hay por defecto:

```
<TextBlock Grid.Row="0" x:Name="pageTitle" Text="Demo Office 365" Style="{StaticResource HeaderTextBlockStyle}" IsHitTestVisible="false" TextWrapping="NoWrap" VerticalAlignment="Bottom" Margin="50,0,30,40"/>

<StackPanel Grid.Row="1">
    <TextBlock TextWrapping="Wrap" Margin="30,30,30,50" FontSize="20" HorizontalAlignment="Center">
        APIS Office 365 by CompartIMOSS
    </TextBlock>

    <Grid Margin="30,0,10,30">
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="*" />
            <ColumnDefinition Width="*" />
            <ColumnDefinition Width="*" />
        </Grid.ColumnDefinitions>

        <StackPanel Grid.Column="0" Margin="20,20,20,20">
            <Button x:Name="buttonPeople" FontSize="30" Click="ButtonPeople_OnClick">People</Button>
            <TextBlock TextWrapping="Wrap" Margin="0,30,0,0" FontSize="20" Text="Obtenemos los contactos del Active Directory." />
        </StackPanel>

        <StackPanel Grid.Column="1" Margin="20,20,20,20">
            <Button x:Name="buttonDocuments" FontSize="30" Click="ButtonDocuments_OnClick">Documents</Button>
            <TextBlock TextWrapping="Wrap" Margin="0,30,0,0" FontSize="20" Text="Obtenemos los ficheros de SharePoint." />
        </StackPanel>
    </Grid>
</StackPanel>
```

A continuación implementamos la funcionalidad de los dos botones. La funcionalidad de estos botones es muy simple, navegará a la opción de la API seleccionada.

```
private void ButtonPeople_OnClick(object sender, RoutedEventArgs e)
{
    this.Frame.Navigate(typeof(ActiveDirectorySamplePage));
}

private void ButtonDocuments_OnClick(object sender, RoutedEventArgs e)
{
    this.Frame.Navigate(typeof(SharePointSamplePage));
}
```

Una vez añadido estas modificaciones, compilamos y si todo ha ido bien visualizaremos la siguiente pantalla:



Imagen 6.- Ejemplo de la Aplicación Implementada

Al pulsar sobre cada opción: People o Documents, se mostrarán las distintas vistas añadidas donde visualizamos la información solicitada a la API correspondiente. Una vez pulsamos sobre cualquier opción, en primer lugar, nos pedirá autenticación sobre Office365 para lo cual se muestra una pantalla de login como la siguiente:

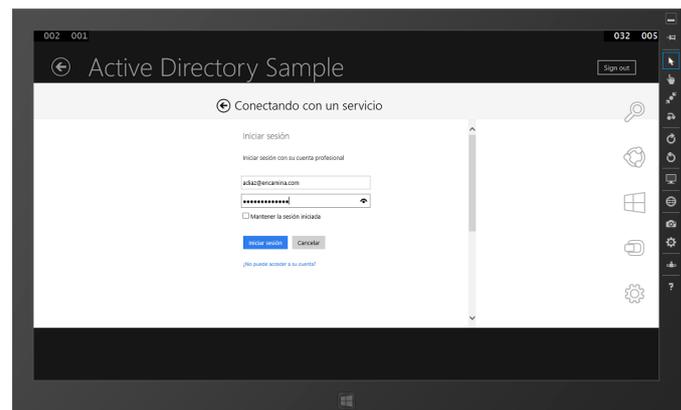


Imagen 7.- Autenticación con Office 365.

Una vez autenticado visualizaremos los datos solicitados a la API correspondiente.

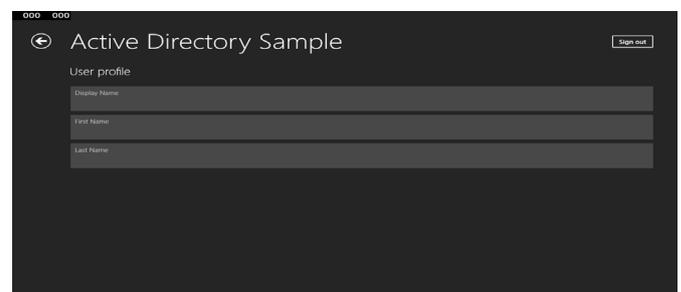


Imagen 8.- Resultado del Perfil de usuario.

Conclusión

Office 365 es un producto clave dentro de la estrategia que está trazando Satya Nadella para hacer competencia a Google y Apple. Dentro de esta estrategia Microsoft está haciendo especial hincapié en el acceso desde cualquier dispositivo y en la productividad. Y en estos dos puntos estas nuevas API's tienen gran importancia, primero porque



Office 365 es la principal herramienta de productividad y segundo porque están implementando herramientas para poder hacer uso de Office 365 desde cualquier plataforma/dispositivo. Otro de las grandes ventajas que llevan estas API's es que facilitan entornos híbridos. El poder hacer uso de esta API en nuestros desarrollos pueden facilitar a las empresas a migrar determinados servicios al Cloud.

Referencias:

- [http://msdn.microsoft.com/EN-US/library/office/dn605900\(v=office.15\).aspx](http://msdn.microsoft.com/EN-US/library/office/dn605900(v=office.15).aspx) <http://blogs.msdn.com/b/officeapps/archive/2014/03/12/announcing-office-365-api-tools-for-visual-studio-preview.aspx>

- <http://msdn.microsoft.com/es-es/library/azure/hh974476.aspx>

ADRIÁN DIAZ CERVERA

Sharepoint Architect at Encamina

MVP de SharePoint Server

<http://blogs.encamina.com/desarrollandosobresharepoint>

<http://geeks.ms/blogs/adiazcervera>

adiaz@encamina.com @AdrianDiaz81





27

Diccionarios para el Motor de Búsqueda de SharePoint 2013

Resumen

Aunque estrictamente hablando, los diccionarios no permiten modificar la relevancia en el Motor de Búsqueda de SharePoint 2013, si es posible mejorar la experiencia de uso de las búsquedas influyendo indirectamente en la relevancia. Por ejemplo, definiendo diccionarios de sinónimos se puede lograr que los resultados se ajusten mejor a lo esperado.

Introducción

Este artículo es la continuación de “Ajuste de la Relevancia en el Motor de Búsqueda de SharePoint 2013” publicado en CompartiMOSS No. 20 (<http://www.compartimoss.com/revistas/numero-20/ajuste-de-la-relevancia-en-el-motor-de-busqueda-de-sharepoint-2013>).

Aunque estrictamente hablando, los diccionarios no permiten modificar la relevancia en el Motor de Búsqueda de SharePoint 2013, si es posible mejorar la experiencia de uso de las búsquedas influyendo indirectamente en la relevancia. Por ejemplo, definiendo diccionarios de sinónimos se puede lograr que los resultados se ajusten mejor a lo esperado.

Un Diccionario personalizado (Custom Dictionary) es un archivo o configuración creado por un Administrador de SharePoint para especificar palabras que el interpretador de un determinado idioma en el Motor de Búsqueda debe tratar como indivisibles al momento de indexación y consulta. En general, los Diccionarios del Motor de Búsqueda de SharePoint no son creados por defecto cuando se instala el servidor, y deben ser establecidos para cada lenguaje en el que se necesite que la Búsqueda se comporte de forma particular.

SharePoint 2013 utiliza diferentes diccionarios para ajustar los resultados de una consulta, convirtiéndose de hecho en una forma para modificar la relevancia de los resultados producidos por el Motor de Búsqueda. En SharePoint 2013 se pueden utilizar diccionarios para Sinónimos, Extracción de Nombres de Compañías, Correcciones de Ortografía, Sugerencias y Extractores de Entidades personalizados.

Diccionario de Sinónimos

Por medio de la utilización de Diccionarios de Sinónimos,

un Administrador puede especificar palabras o frases que se pueden utilizar en lugar de las que utiliza el usuario en la consulta. Un Sinónimo se puede utilizar comúnmente para expandir acrónimos (siglas que se pronuncian como una palabra, y que por el uso acaba por incorporarse al léxico habitual), pero también se pueden usar para incluir automáticamente variaciones de un término de búsqueda utilizado por la terminología específica de una organización.

El diccionario de Sinónimos permite realizar dos labores:

- Reemplazo de palabras o frases: Se pueden designar una o más palabras o frases que reemplazan una palabra o frase particular. Por ejemplo, si el término “USA” aparece en una búsqueda, el sistema puede buscar también por resultados que incluyan “Estados Unidos”. Para realizar esto, es necesario insertar un “Grupo de Reemplazo”(Replacement set) en el archivo de Sinónimos.
- Sinónimos para palabras o frases: Un administrador puede especificar palabras o frases que tengan el mismo significado de una palabra o frase utilizada como término de búsqueda. Por ejemplo, se puede definir que “casa” y “hogar” son sinónimos el uno del otro. Por lo tanto, una consulta que utilice cualquiera de los dos, devolverá resultados que contienen ambos. Sinónimos se realizan insertando “Grupos de Expansión” (Expansion set) en el diccionario.

Un Diccionario de sinónimos es un archivo separado por comas (.csv) que define los términos y sus sinónimos correspondientes. Opcionalmente es posible definir a que lenguaje aplica el sinónimo. Para definir más de un sinónimo para un término es necesario crear entradas múltiples en el Diccionario. También, si un sinónimo debe funcionar en ambas direcciones; por ejemplo, si “hogar” también debe devolver resultados que se refieren a “casa”, es necesario crear dos entradas en el Diccionario.

Todos los caracteres de Unicode se pueden utilizar como términos para sinónimos, lo mismo que subrayado (“_”), guiones (“-”) y apóstrofes porque ellos son tratados como caracteres. No es posible utilizar caracteres que no sean Unicode como “#” y “/”. El mapeo entre llaves de entrada en el Diccionario y términos de búsqueda no es sensible a mayúsculas/minúsculas.

El archivo .csv tiene que tener columnas definidas para "Key", "Synonym" y "Language" (en inglés) y ser separado por comas. Si el archivo contiene caracteres que estén en la tabla ASCII limitada, tales como diacríticas, deben ser codificados como UTF-8. Las columnas a utilizar son:

- "Key": Contiene el termino (una o varias palabras) que originan el sinónimo. No debe haber espacios al principio o al final del término en todas las columnas.
- "Synonym": Es el sinónimo (una o varias palabras) que el motor de búsqueda debe utilizar para entregar sus resultados.
- "Language" (opcional): Indica la abreviación utilizada para indicar el lenguaje al que aplica el sinónimo. Si la columna está vacía, se aplica a todos los idiomas.

Cree el archivo con un editor de texto ASCII (Notepad) en un sitio local en un front-end de la granja de SharePoint. Un ejemplo de archivo de Diccionario de sinónimos se indica en la Imagen 1.

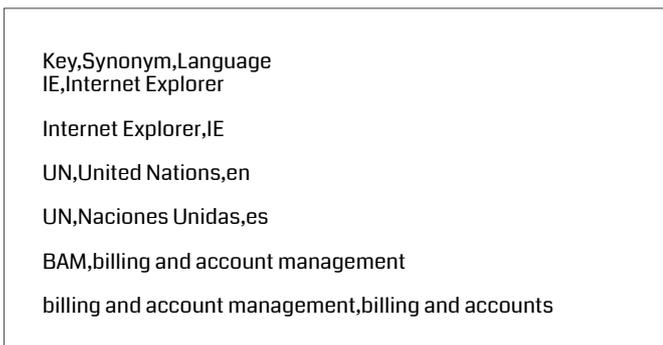


Imagen 1.- Ejemplo de Diccionario de sinónimos.

No es posible exportar el Diccionario de sinónimos utilizado por SharePoint 2013. Si es necesario hacer cambios en el, es necesario utilizar el Diccionario original, modificarlo y desplegarlo de nuevo, por lo que es indispensable guardar el diccionario utilizado originalmente. Un administrador de SharePoint puede desplegar el diccionario utilizando el siguiente cmdlet desde una aplicación de consola de PowerShell para SharePoint:

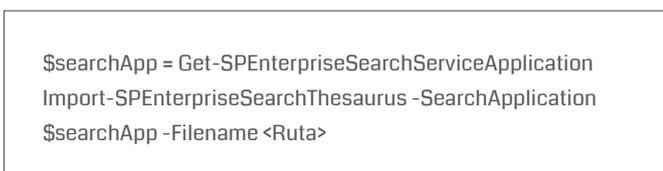


Imagen 2.- Cmdlet de PowerShell para desplegar el Diccionario de sinónimos.

"<Ruta>" indica la ruta completa hacia el archivo .csv del Diccionario.

Extracción de Nombres de Compañías

El sistema de búsqueda de SharePoint puede extraer nom-

bres de compañías desde el contenido indexado. Por ejemplo, si el nombre de una compañía se encuentra en un documento y el sistema ha sido configurado para Extracción de Nombres de Compañías, el nombre es mapeado en la Propiedad Manejada llamada "companies". Esta propiedad se puede utilizar para crear refinadores basados en nombres de compañías en la WebPart de refinadores de la página de resultados.

SharePoint dispone de un Diccionario prefabricado por Microsoft que incluye un número muy grande de compañías en el mundo. Es posible agregar nombres adicionales para que sean extraídos o prevenir que algunos nombres lo hagan utilizando listas de Inclusión y Exclusión de Compañías (Company Inclusion y Company Exclusion).

El Motor de Búsqueda puede extraer nombres de compañías si se cumplen las siguientes condiciones:

- La configuración "Extracción de nombre de compañía" debe estar chequeada en las Propiedades Manejadas del Motor de Búsqueda (no lo está por defecto). Los Nombres de compañías se extraen del contenido completo con el que la Propiedad está asociada, inclusive si el contenido las tiene marcadas con la etiqueta de "<no index>". La propiedad mostrada en la Imagen 3 se puede encontrar a nivel de Administración Central de SharePoint - Administración de Búsquedas - Esquema de búsqueda - abra la Propiedad administrada llamada «companies», o a nivel de Administración de cada Colección de Sitios - Esquema (sección de «Buscar»)

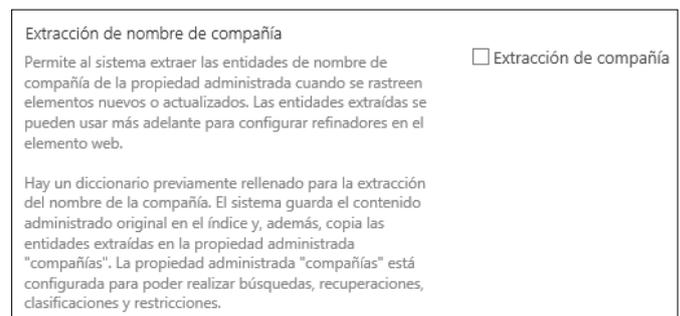


Imagen 3.- Configuración de la Extracción de Nombre de Compañías.

- El nombre de la compañía tiene que existir en el Diccionario original proporcionado por Microsoft, o en la lista de Inclusiones de Compañías.
- Se debe ejecutar un rastreo completo de la información.

El Diccionario con nombres de compañías Incluidas y Excluidas se maneja desde el Servicio de Metadatos Administrados de SharePoint 2013. La lista de compañías incluida originalmente no se puede exportar ni modificar de ninguna manera. La configuración la debe realizar un Administrador de SharePoint que tenga suficientes derechos en el Servicio de Metadatos Manejados. Desde la Administración Central, abra la ventana de administración del Servicio de

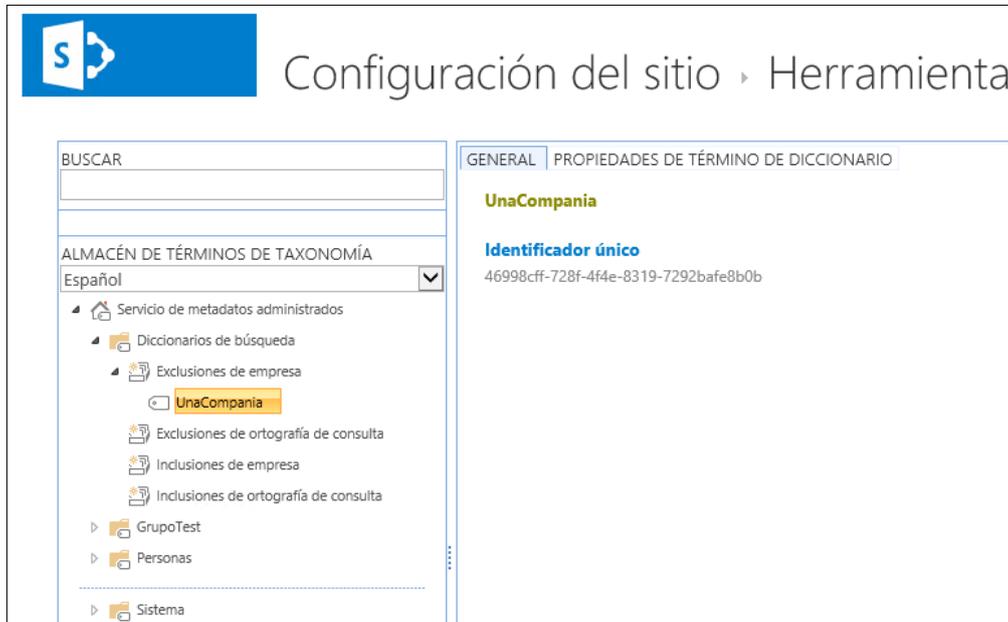


Imagen 4.- Configuración de Inclusión y Exclusión de Nombres de Compañías.

Metadatos Administrados y en el Grupo “Diccionarios de búsqueda” incluya y/o excluya los nombres necesarios en forma de Términos, como muestra la Imagen 4.

Corrección de ortografía en consultas

Si un usuario utiliza una consulta que aparentemente tiene algún error de escritura, la página de resultados puede incluir correcciones (“Se refiere a?”). Por ejemplo, si se utiliza el término “idiosincracia”, el sistema puede sugerir como corrección “idiosincrasia”. El Motor de Búsqueda crea correcciones automáticamente para una consulta cuando los usuarios han hecho clic una o más veces sobre el resultado

Un Diccionario personalizado (Custom Dictionary) es un archivo o configuración creado por un Administrador de SharePoint

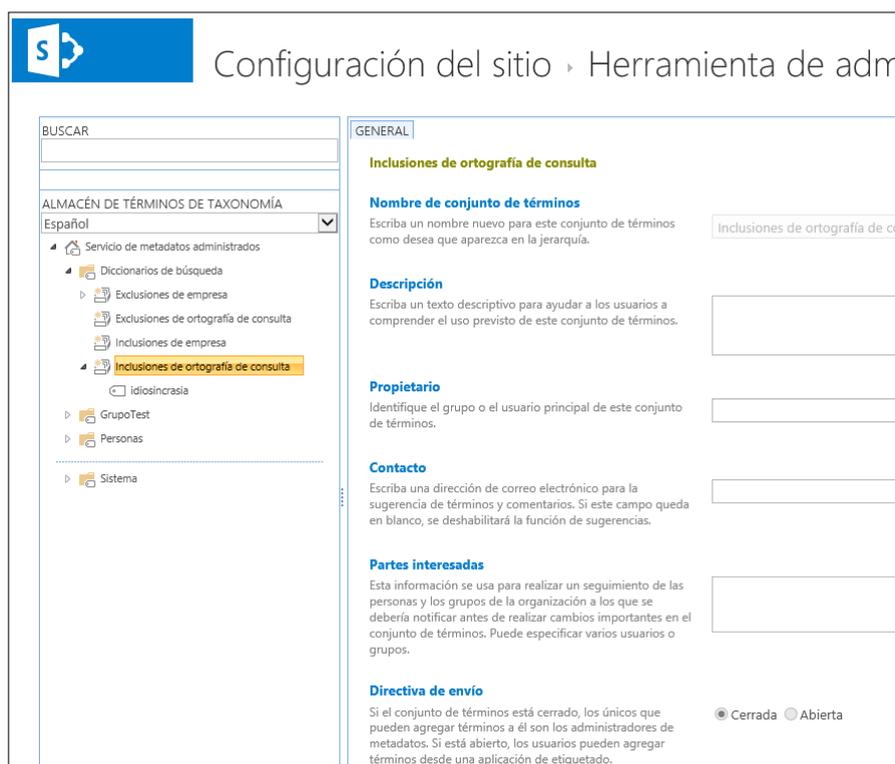


Imagen 5.- Configuración de los Diccionarios para las correcciones de ortografía.

generado para una consulta por lo menos seis veces. Las correcciones son generadas diariamente para cada Fuente de Resultados y para cada Colección de Sitios, por lo tanto, pueden ser diferentes para cada una de ellas.

El Diccionario que configura la Inclusión y Exclusión de correcciones se almacena también en el Servicio de Metadatos Administrados. La configuración debe ser realizada por un Administrador con suficientes derechos en el Servicio. Desde el Grupo “Diccionarios de búsqueda”, utilice “Inclusiones de ortografía de consulta” y “Exclusiones de ortografía de consulta”. Los Términos utilizados dependen también del idioma seleccionado en el Servicio; note que la “Directiva de envío” está cerrada por defecto y no se puede cambiar.

Para términos en la lista de “Exclusiones de ortografía de consulta” nunca se mostraran correcciones. Aunque las listas en el Servicio de Metadatos se pueden modificar, las correcciones generadas automáticamente por el sistema ni se pueden exportar ni se pueden modificar. Cuando se incluye algún término, puede demorar hasta 10 minutos para que las correcciones comiencen a funcionar.

Sugerencias de consultas

Las Sugerencias de consultas son frases que el sistema de búsqueda utiliza para guiar al usuario cuando comienza a escribir una consulta. Estas sugerencias pueden ser desactivadas y modificadas en SharePoint 2013. Las Sugerencias ayudan a los usuarios a encontrar información rápidamente mostrándole consultas que han sido realizadas anteriormente. Por ejemplo, cuando el usuario escribe “ventas”, puede escoger entre varias consultas anteriores que han realizado otros usuarios relacionadas con “ventas”.

El Motor de Búsqueda crea las Sugerencias automáticamente para una consulta cuando otros usuarios han hecho clic sobre uno o más resultados por lo menos por seis veces. Las Sugerencias son generadas diariamente para cada Origen de Resultados y cada Colección de Sitios, por lo que las Sugerencias generadas pueden ser diferentes en cada parte de SharePoint.

La Sugerencias están activadas por defecto en SharePoint, pero se pueden desactivar desde la Administración Central - Administración de búsquedas - Sugerencias de consulta (incluyendo el idioma que se prefiere para las sugerencias):



Imagen 6.- Configuración de Sugerencias.

Se pueden configurar dos tipos de Sugerencias: “Sugerir

frases siempre” y “Nunca sugerir frases”, que, como sus nombres indican, permiten agregar frases que siempre o nunca se van a mostrar. Las frases configuradas se aplican a todos los Orígenes de Datos y todas las Colecciones de Sitios de forma unificada.

La configuración se basa en uno o más archivos de textos que contienen las frases, y que luego se podrán importar en SharePoint. Es necesario tener un archivo separado para las frases que se sugieren siempre y otro para las que nunca se sugieren. También es necesario tener un archivo separado para cada idioma utilizado. El lenguaje determina la forma en que las Sugerencias son procesadas por el Motor de Búsqueda internamente, pero todas las Sugerencias son siempre mostradas o contenidas para todos los lenguajes cuando el usuario entra la consulta. Cada frase se debe crear en una línea separada en el archivo de texto y guardada utilizando codificación UTF-8. Los archivos se pueden importar en el Motor de Búsqueda desde la Administración Central - Administración de búsquedas - Sugerencias de consulta.

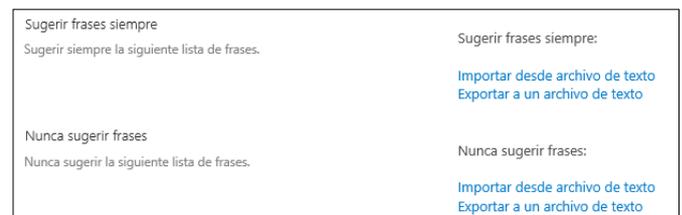


Figura 7. Importar y exportar archivos de frases de Sugerencias.

Cuando se importa un archivo con frases de Sugerencias, se sobrescriben todas las Sugerencias ya existentes, pero no las que se han creado automáticamente, que persisten en el sistema. Para editar las Sugerencias existentes, se puede exportar el archivo, modificarlo, y subirlo de nuevo a SharePoint.

Un Administrador de SharePoint puede utilizar también el cmdlet de PowerShell para SharePoint “Import-SPEnterpriseSearchPopularQueries” (<http://technet.microsoft.com/en-us/library/jj219665.aspx>) para realizar la misma configuración que con la Interfaz de Usuario, aunque el cmdlet permite algo más de control, pudiéndose especificar por ejemplo a que Origen de Contenido y Colección de Sitio se debe aplicar el archivo de Sugerencias.

Con PowerShell también se pueden agregar frases una por una. El cmdlet “New-SPEnterpriseSearchLanguageResourcePhrase” (<http://technet.microsoft.com/en-us/library/ff608062.aspx>) permite añadir una frase al tiempo. Adicionalmente, el cmdlet tiene un parámetro “QuerySuggestionSubstitution” que se puede utilizar para el mapeo de términos. Por ejemplo, si la frase es “automóvil” es posible agregar un mapeo a “carro o automóvil”, de tal forma que la frase “carro” se mostrara como una Sugerencia, pero cuando el usuario selecciona esta Sugerencia, la consulta final será “carro o automóvil”.

Extractores de Entidades Personalizados

Para utilizar entidades personalizadas (palabras o frases) como refinadores de búsqueda, es necesario crear primero un Diccionario de Entidades de Extracción Personalizadas y desplegarlo. De esta forma, se puede crear una Propiedad Administrada para utilizar las Entidades Personalizadas y ejecutar un rastreo completo. Cuando el rastreo termina, la WebPart de navegadores en la página de resultados se

puede configurar para utilizar las Entidades Personalizadas como refinador.

A los archivos de Entidades de Extracción Personalizadas se les da mantenimiento fuera de SharePoint, y cuando están listos, se importan en el sistema para hacerlos disponibles para el Motor de Búsqueda. Hay diferentes tipos de Diccionarios que se pueden crear: Word, Word Part, Word Exact o Word Part Exact, como la siguiente tabla (extraída de MSDN) indica:

Custom entity extractor / custom entity extractor dictionary	Description	Example	Dictionary name to use in Windows PowerShell	Managed property that will contain the extracted entity
Word Extraction	Case-insensitive, dictionary entries matching tokenized content, maximum 5 dictionaries.	The entry "anchor" matches "anchor" and "Anchor," but not "anchorage"	Microsoft.UserDictionaries.EntityExtraction.Custom.Word.n [where n = 1,2,3,4 or 5]	WordCustomRefiner1 WordCustomRefiner2 WordCustomRefiner3 WordCustomRefiner4 WordCustomRefiner5
Word Part Extraction	Case-insensitive, dictionary entries matching un-tokenized content, maximum 5 dictionaries.	The entry "anchor" matches "anchor," "Anchor" and "anchorage"	Microsoft.UserDictionaries.EntityExtraction.Custom.WordPart.n [where n = 1,2,3,4 or 5]	WordPartCustomRefiner1 WordPartCustomRefiner2 WordPartCustomRefiner3 WordPartCustomRefiner4 WordPartCustomRefiner5
Word Exact Extraction	Case-sensitive, dictionary entries matching tokenized content, maximum 1 dictionary.	The entry "anchor" matches "anchor," but not "Anchor" or "Anchorage"	Microsoft.UserDictionaries.EntityExtraction.Custom.ExactWord.1	WordExactCustomRefiner
Word Part Exact Extraction	Case-sensitive, dictionary entries matching un-tokenized content, maximum 1 dictionary.	The entry "anchor" matches "anchor" and "anchorage," but not "Anchor"	Microsoft.UserDictionaries.EntityExtraction.Custom.ExactWordPart.1	WordPartExactCustomRefiner

Tabla 1.- Extractores de Entidades Personalizados para el Motor de Búsqueda de SharePoint 2013.

Para crear un Extractor de Entidades es necesario crear un archivo .csv que contenga las columnas "Key" y "Display Form". Asegúrese de utilizar comas como el separador de las columnas. Si el archivo contiene caracteres que no sean ASCII tales como diacríticas, tienen que ser codificados como UTF-8. Guarde el archivo en uno de los servidores de Front-end de la granja de SharePoint. Las columnas a utilizar deben ser:

- Columna "Key" - Contiene el termino (una o varias palabras) a ser incluidas como Entidades Personalizadas. Se puede utilizar más de una llave por línea. Asegúrese de no dejar espacios en blanco antes o después de la llave
- Columna "Display Form" (Opcional) - Contiene el nombre del refinador de búsqueda. Si la columna no tiene contenido, el término extraído se mostrará como el refinador en la misma forma en la que aparece en el contenido original. La columna de Display Form permite controlar y estandarizar la forma en que los refinadores se muestran en la Interfaz

extraer estas Entidades de tal forma que puedan ser utilizadas como refinadores en la página de resultados de búsqueda. Sin importar si las palabras aparecen en minúsculas o mayúsculas en el contenido original, los refinadores se deben mostrar tal y como la empresa los ha especificado. Para este ejemplo, el archivo del Diccionario de Extracción de Entidades sería similar al siguiente:

```
Key,Display Form
ABC Principiante,ABC Principiante
ABC BI,ABC Principiante
ABC profesional,ABC Profesional
abc prof,ABC Profesional
ABC Experto,ABC Experto
```

Un Administrador del Motor de Búsqueda de SharePoint puede utilizar PowerShell para SharePoint para ejecutar el siguiente comando e instalar el archivo de Extracción de Entidades:

```
$searchApp = Get-SPEnterpriseSearchServiceApplication

Import-SPEnterpriseSearchCustomExtractionDictionary
-SearchApplication $searchApp -Filename <Ruta> -DictionaryName <NombreDiccionario>
```

Como ejemplo, una organización llamada "ABC" tiene un sistema de certificación con tres niveles: ABC Principiante, ABC Profesional y ABC Experto. La empresa necesita



En donde:

- <Ruta> especifica la ruta completa al archivo .csv conteniendo el Diccionario a ser importado.
- <NombreDiccionario> es el nombre del tipo de Diccionario de Extracción tal como indica la Tabla 1.

Límites de los Diccionarios

Para garantizar el uso óptimo de la memoria de los servidores de la granja de SharePoint, la eficiencia del proceso

de búsqueda y rapidez de los resultados, no se deben sobrepasar los siguientes límites en los Diccionarios del Motor de Búsqueda de SharePoint:

Las Sugerencias de consultas son frases que el sistema de búsqueda utiliza para guiar al usuario cuando comienza a escribir una consulta

Límite	Valor Máximo	Tipo de Límite	Notas
Número de entradas de sinónimos	1 millón	Soportado	Exceder este límite puede resultar en uso incremental de memoria y subidas en el tiempo de respuesta.
Número de entradas en un Diccionario de Extracción de Entidades	1 millón	Soportado	Exceder este límite puede resultar en uso incremental de memoria y subidas en el tiempo de respuesta.
Número de entradas en un Diccionario de búsqueda (Metadatos Administrados)	5,000 términos por tenant	Límite fijo	Límite del número de términos permitidos para Diccionarios de inclusiones y exclusión para la corrección de ortografía y Extracción de Nombres de Compañías. Es posible utilizar más términos que este número en el Servicio de Metadatos Administrados, pero el sistema de búsqueda puede utilizar solamente 5000 términos por tenant.

Tabla 2.- Límites de los diccionarios de búsqueda.

Conclusión

Los Diccionarios del Motor de Búsqueda permiten adaptar los resultados de las consultas de forma fácil y rápida. Existen varios tipos de Diccionarios que se pueden utilizar en SharePoint 2013, incluyendo Sugerencias de consulta y Sinónimos. Los Diccionarios consisten en archivos que se pueden subir al sistema por medio de PowerShell, y Diccionarios predefinidos en el Servicio de Metadatos Administrados.

GUSTAVO VELEZ

MVP de SharePoint Server

gustavo@gavd.net

<http://www.gavd.net>

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.





34

Publicando un sitio de SharePoint en Cyberoam

Resumen

Espero que este tema resulte de interés y de ayuda a alguien. En este artículo les voy a contar como publicar un sitio web de SharePoint en internet usando un Firewall Cyberoam con la característica de WAF (Web Application Filter) activada.

Desde hace algunas semanas mis compañeros de desarrollo están trabajando sobre el nuevo sitio web público de uno de nuestros clientes y llega la hora ¡Que esto lo vea el Mundo!

Introducción

Mi parte de este proyecto ha sido preparar la infraestructura de SharePoint de nuestro cliente para que el equipo de desarrollo tenga una buena infraestructura donde trabajar y nuestro cliente tenga además de su sitio Web Público una intranet corporativa. Esto hace que debamos hilar fino con la infraestructura, un sitio publicado en Internet y los datos internos del cliente en los mismos servidores, sin servidores de DMZ y otras buenas prácticas que deberíamos cuidar al hacer esto, pero bueno... la crisis es así y por suerte SharePoint nos ayuda bastante a poder hacer esto posible.

Como complemento a esto tenemos un Cluster de Firewalls Next Generation de Cyberoam con la característica WAF que nos permitirá publicar nuestro site y aportar seguridad, QoS, IPS y otras medidas de seguridad.

Preparemos SharePoint

Lo primero es instalar SharePoint 2013, crear Service Accounts, Web Applications, Contents Databases, yo a veces soy un poco "paranoico" así que intento tomar todas las medidas de seguridad que sean posibles, prácticas y aplicables. Entre otras cosas la intranet tiene su Web Application, que la ejecuta un usuario específico, con su usuario Owner de Site Collection propio y la misma receta para el sitio público.

Evidentemente toda la parte que corresponde a servicios de SharePoint, Service Application Pools, etc. se ejecutan con usuarios diferentes al de la intranet y el sitio público.

Ya tenemos nuestra site collection en marcha con una

plantilla de SharePoint para Publishing, responde internamente perfectamente en <http://www.cliente.local>, el cliente y el equipo de desarrollo han trabajado sobre el site desde la red del cliente y ha llegado la hora de la verdad, hay que ponerlo en internet y que responda a <http://www.cliente.com> ¿cómo lo hacemos?

Lo primero es desde SharePoint extender nuestra Web Application para habilitar el acceso anónimo al site, desde dentro los usuario han estado accediendo validados pero lo último que necesitamos es tan siquiera la posibilidad de que salte un cartel de validación en internet, primero porque quedaría muy feo y según así no tenemos ni la tentación de un ataque de fuerza bruta o por diccionario.

Extendemos nuestra Web Application desde la Administración Central de SharePoint, definimos el nombre de nuestro nuevo Sitio Web en el IIS, Le damos un nombre, puerto de escucha (80) y Host Header (www.cliente.com) gracias al Host Header podremos hacer un par de cosas cuanto menos curiosas, una de ellas es que <http://www.cliente.local> y <http://www.cliente.com> estén en el puerto 80 y el IIS nos haga el favor de mostrarnos la web. Pero para mí lo mejor es que cuando publique en internet bien desde un Proxy Inverso (ISA Server, Forefront TMG, etc.) o un Firewall con filtrado de Header podremos usar el puerto 80 para múltiples sites en una única IP Pública y aseguraremos que nuestros IIS o servidores web solo respondan a la petición del puerto 80 pero además con el nombre publico adecuado, Solo veremos nuestro sitio web de SharePoint si la petición llega a www.cliente.com en el puerto 80, no vale la IP y el puerto, por valer no vale ni <http://localhost>.

En esta definición, especificaremos que corresponde este Site de IIS a la Zona de Internet y habilitaremos el acceso anónimo, todo esto nos permitirá definir un comportamiento distinto para el acceso interno del cliente para cargar contenido, así como el acceso desde internet para visualizarlo.

Lo primero es instalar SharePoint 2013, crear Service Accounts, Web Applications, Contents Databases

Pasos a realizar

- Vamos a Administración Central --> Manage web applications:

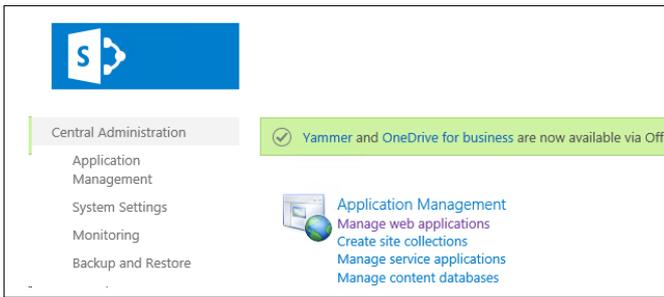


Imagen 1.- Acceso a la Administración Central de SharePoint 2013.

- Tenemos nuestra lista de Web applications:

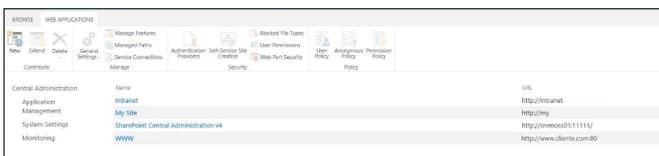


Imagen 2.- Listado de Aplicaciones Web.

- Clicamos nuestra Web applications y luego en Extend:

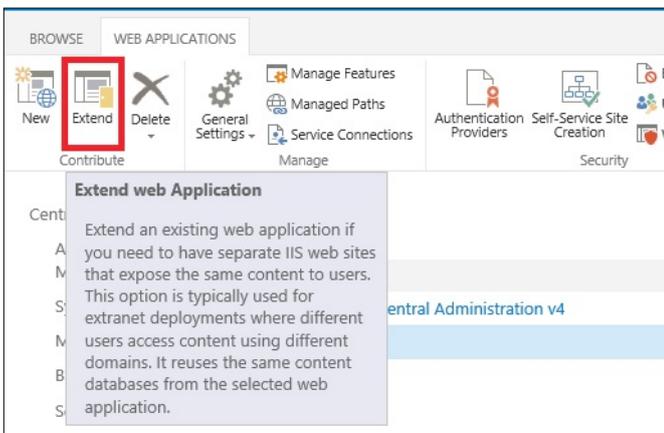


Imagen 3.- Extensión de una aplicación web existente.

Es posible con un despliegue mínimo de infraestructura crear un entorno bastante robusto y seguro

Ahora debemos introducir los datos del nuevo sitio en el IIS, lo más importante es definir puerto, host header, permitir el acceso anónimo, deshabilitar la autenticación integrada, url y seleccionar la zona que usaremos para esta url.

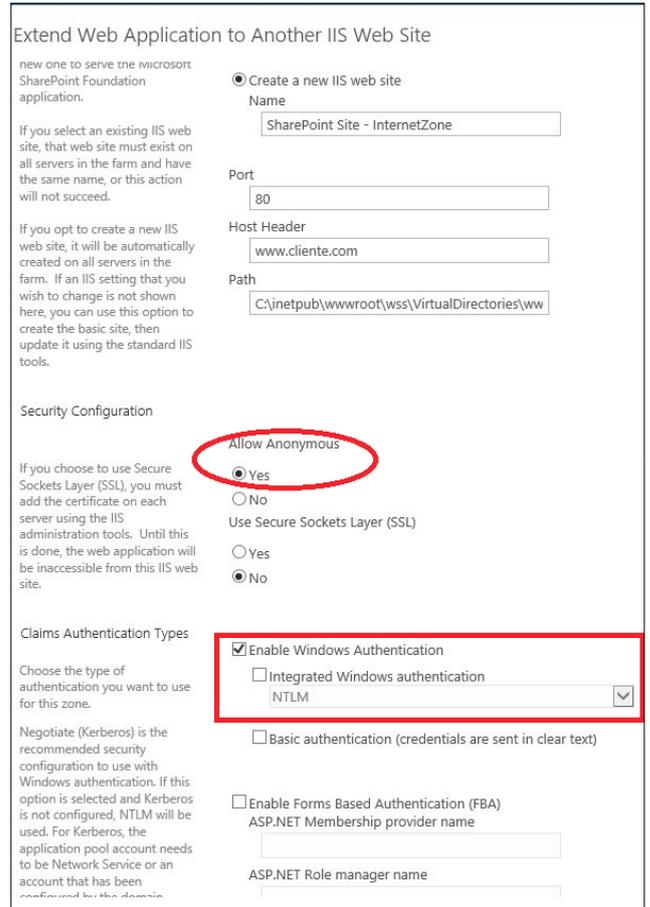


Imagen 4.- Configuración de la aplicación web extendida.

A continuación se muestran el resto de configuraciones necesarias al extender la aplicación web:

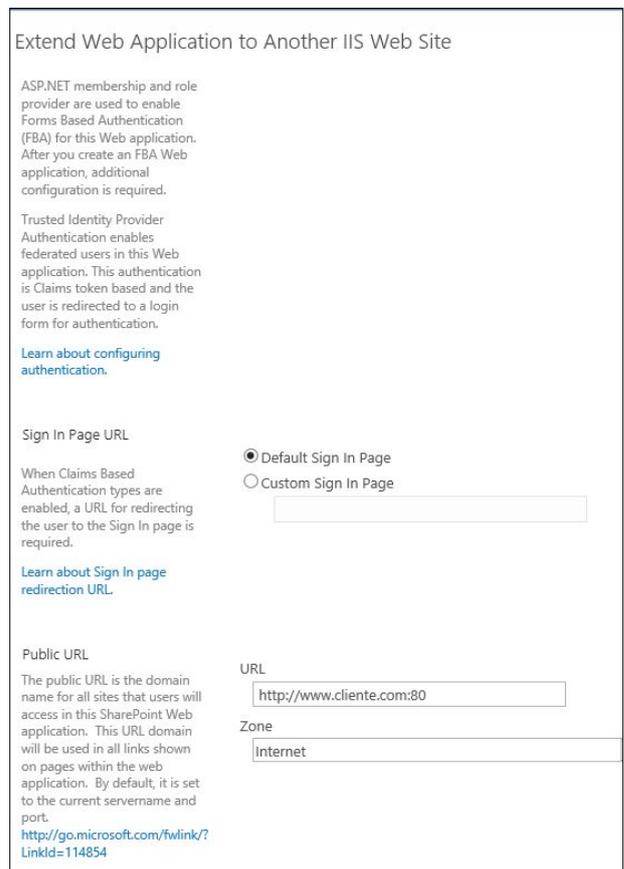


Imagen 5.- Resto de configuraciones necesarias en la extensión de la aplicación web.

Una vez SharePoint termina el proceso podremos comprobar como nuestra web application ahora dispone de 2 zonas definidas en Authentication Providers



Imagen 6.- Configuración de Authentication providers.

Si luego deseamos modificar algo podremos venir aquí y seleccionar la zona que deseamos modificar.

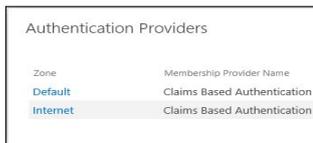


Imagen 7.- Authentication providers.

Otro cambio que podremos observar es que ahora en nuestro IIS encontraremos un nuevo site definido y que se ejecutara bajo el mismo Web Application Pool que nuestro primer sitio.

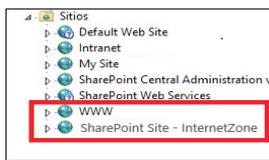


Imagen 8.- Sitios en IIS.

Esto que hemos realizado lo podemos aplicar también para nuestra Intranet, pero en este caso por ejemplo podríamos habilitar la validación por formulario para la zona de Ex-

tranet que será la que publicaremos al exterior.

Ahora nuestro cliente accede a su sitio público para cargar contenido de forma natural, <http://www.cliente.local> ve y tiene todas las características para editar su sitio, pero desde internet solo ven el contenidos publicado al entrar en <http://www.cliente.com> y no hay forma de que se pueda entrar desde fuera al `/_layouts/settings.aspx` porque directamente devuelve un 401.

Configuraciones en el Firewall

Los firewalls están cambiando y nos estamos encontrando con dispositivos que ya no solo abren y cierran puertos, ahora inspeccionan tráfico, IPS, etc. y algunos como este Cyberoam nos hacen de proxy inverso, 1 IP publica y múltiples sites publicados en el puerto 80 así nuestros visitantes no tienen que recordar puertos en nuestras urls o el cliente pagar por servicios de ISP para empresa con alto coste.

Esto es bastante sencillo si tenemos la característica de WAF activada en nuestro firewall Cyberoam, solo es cuestión de definir la interfaz que tiene nuestra IP publica el host que tiene la IP privada (Nuestro SharePoint), protocolo (HTTP/HTTPS), si tenemos traducción de puerto (Eje. Publicamos en internet el 80, pero en nuestro servidor en el 8080) y especificamos el header o nombre de dominio con el que publicamos nuestro site, esto es lo que nos permitirá crear tantas publicaciones de sitios necesitemos externamente sobre el puerto 80 en nuestro firewall (Proxy Inverso).

Nos me extenderé sobre la configuración del Firewall principalmente porque cada fabricante tiene un proceso distinto para alcanzar esto, pero el concepto es igual, solo cambia la forma de conseguirlo.

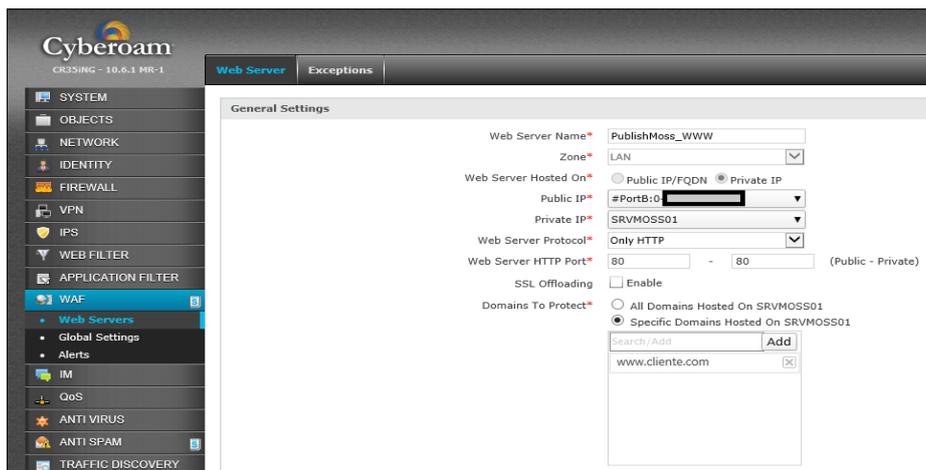


Imagen 9.- Configuración del firewall.

Conclusión

Como vemos, es posible con un despliegue mínimo de infraestructura crear un entorno bastante robusto y seguro para hospedar nuestro sitio de SharePoint On-Premise y nuestra intranet ambos expuestos a accesos externos. De esta forma podemos extender aún más las posibilidades de

una organización para explotar al máximo las cualidades y capacidades de SharePoint.

JORGE LUIS GONZÁLEZ YÁNEZ
 Responsable de Ingeniería y Sistemas
jorgelgy@hotmail.com



Introducción a las App Forms con Access

Resumen

En el artículo veremos un ejemplo paso a paso sobre cómo se puede construir un formulario de aplicación en SharePoint 2013 con Access Services (App Forms), así como las ventajas e inconvenientes en utilizarlo como herramienta para nuestras soluciones.

Microsoft ya anunció la “muerte” de InfoPath, el inicio del “periodo de gestación” de los Forms over SharePoint Lists

Artículo

En la SPC14 Microsoft ya anunció la “muerte” de InfoPath, el inicio del “periodo de gestación” de los Forms over SharePoint Lists (FoSL) y las alternativas actuales y futuras a aplicaciones o servicios para la creación de formularios en SharePoint.

Una de las alternativas que presentó Microsoft como actualmente ya disponibles son las que llamó “Apps Forms” y que en realidad todos conocemos como Access Services. Bueno, de hecho no todos lo conocemos, ya que la simple palabra “Access” nos suele generar cierto rechazo por sonar anacrónica y fuera de lugar, aunque deberíamos darle una oportunidad a Access 2013, ya que (si no lo conocéis) os sorprenderá cómo puede crear formularios de datos relacionales desde SharePoint en una interface que se encuentra a caballo entre la interface web más moderna y los detalles Access más clásicos.

Este artículo pretende mostrar un sencillo ejemplo de cómo crear una aplicación de formularios basada en Access Services 2013, mostrando sus principales características, ventajas y limitaciones.

Ante todo comentar que las App Forms generadas con Access Services, aunque se pueden alimentar de fuentes externas de datos (importaciones de datos), se almacenan completamente en el propio interior del fichero Access, el front-end de formularios que genera también es interno a Access y no puede ser utilizado para front-end externo de otros entornos, y los datos se almacena en el SQL Server

configurado en el servicio. Por lo tanto, al ser un entorno aislado, no puede utilizarse para interactuar con Workflows de SharePoint ni nada parecido. La única integración que tendrá con SharePoint es que desde el Sitio podremos crear y visualizar las Apps renderizadas como en formato web.

Como buen amante de la literatura, voy a utilizar como ejemplo la construcción de una aplicación para la gestión del inventario de una biblioteca, que además nos brindará un buen entorno para aplicar diversos niveles de vínculos relacionales.

Para ello vamos a crear 4 tablas distintas. 3 secundarias (Editoriales, Autores y Géneros) y una principal (Libros) que se alimentará mediante lookups de dichas tablas secundarias.

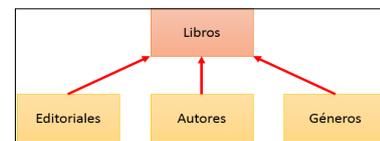


Imagen 1.- Esquema de tablas a utilizar en el ejemplo propuesto.

Antes de crear un nuevo App Form, es prerequisite tener instalado en nuestra máquina el Access 2013, que será nuestra herramienta de trabajo y con la que publicaremos el front-end diseñado en SharePoint. Para crear la App podemos hacerlo directamente desde SharePoint o desde el propio Access, ambos caminos nos conducirán finalmente a la edición desde el cliente Access.

Para iniciar la creación del formulario desde SharePoint, iremos a los Site Settings y pulsaremos en “Add an app”

Aquí buscaremos la Access App y crearemos una nueva pulsando sobre el icono correspondiente, tecleando el nombre de la App y pulsando en el botón de “Create”. Transcurridos unos segundos, esto nos llevará a una página donde nos informará que la App de Access se ha creado correctamente y que deberíamos empezar a trabajar en ella (Añadir tablas) desde el cliente de Access.

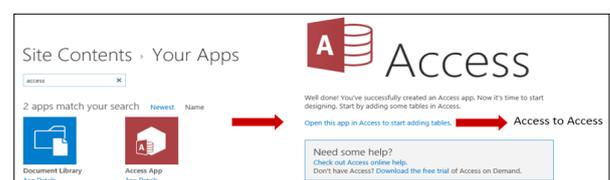


Imagen 2.- Creación de una Access App desde SharePoint.

Para iniciar la creación del formulario desde Access, deberemos ir al menú “Nuevo” desde el cliente de Access y seleccionar la plantilla “Custom Web App” para iniciar un App Form en blanco. Si pulsáramos en “Blank desktop database” estaríamos creando un fichero local de Access de los de toda la vida. El resto de opciones son plantillas ya pre-fabricadas para empezar una App Form con algo de contenido inicial (puede venirnos bien si lo que queremos construir tiene como base alguno de los contenidos de esas plantillas).

Una vez hayamos pulsado en “Custom web app” debemos introducir el nombre de nuestra aplicación y la localización de la misma (URL del site donde la vamos a publicar). Al pulsar en “Create” generará la App en el site indicado de SharePoint y podremos iniciar la edición de nuestro formulario.

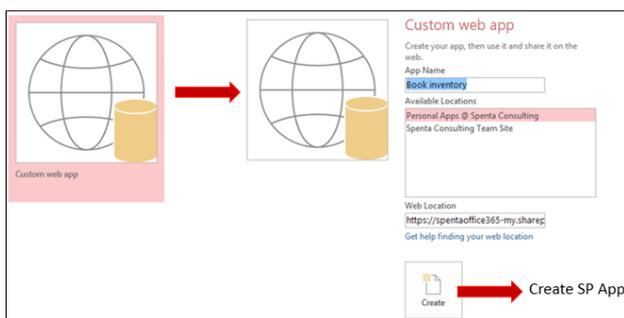


Imagen 3.- Creación de una Access App desde Access.

Tanto si hemos creado la Access App por SharePoint o por Access, accederemos a una pantalla dentro del nuestro Access Cliente, donde nos indicará que debemos empezar a crear tablas para la base de datos Access. Estas tablas se pueden crear a partir de fuentes externas ya existentes (Un propio documento de Access, Un documento Excel, Conexión ODBC a BD, un documento .csv o una SharePoint List), pero en nuestro ejemplo vamos a centrarnos en crear las tablas manualmente desde cero, y así podremos trabajar desde la base.

Por lo tanto, pulsaremos en el enlace (un tanto escondido a simple vista) de la columna de texto derecha, donde indica “add a new blank table”.

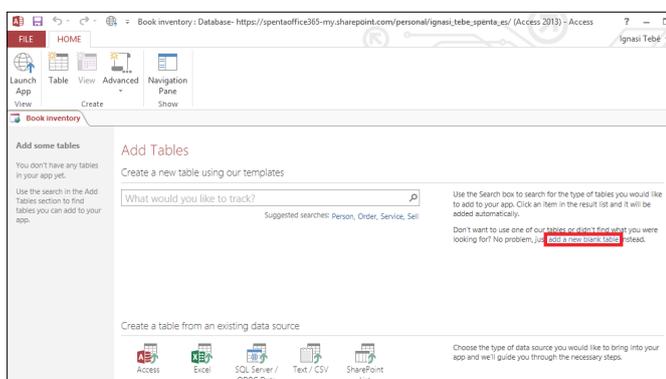


Imagen 4.- Pantalla inicial de la Access App.

Empezaremos creando una de las tablas secundarias. Por ejemplo, Editoriales. Una vez dentro del editor de campos de una tabla, veremos que el campo ID nos viene ya creado como “key”. A partir de aquí podemos insertar nombres de

nuevos campos en las filas posteriores, y seleccionar entre los diversos tipos de campos disponibles (que se aprecian en la siguiente imagen).

Field Name	Data Type
ID	AutoNumber
Nombre	Short Text
	Short Text
	Long Text
	Number
	Date/Time
	Currency
	Yes/No
	Hyperlink
	Image
	Calculated
	Lookup

Imagen 5.- Tabla de Editoriales y opciones de tipos de campo.

En este ejemplo, para la tabla de Editoriales, tan solo requiere crear un campo (Nombre) de tipo Short Text.

Veremos que en la zona inferior de la pantalla, nos aparece un cuadro de propiedades relativo al campo actualmente seleccionado. En ella podremos configurar diversos aspectos (según el tipo de campo), como la longitud del mismo (en caso de tipo string), el formato de la fecha (en caso de tipo data), el valor por defecto del campo, insertar reglas de validación y el mensaje de alerta en caso de no cumplir dicha validación, si el campo es obligatorio (requerido) o no, y si el campo requiere indexación o no. Este panel de configuración es esencial para conseguir que nuestros campos y formularios se muestren y comporten de la forma deseada.

General	
Limit Length	Yes
Character Limit	220
Label Text	Nombre de la editorial que publica el libro
Default Value	"Planeta"
Validation Rule	
Validation Text	
Required	Yes
Indexed	No

Imagen 6.- Tabla de configuración del campo para un tipo Short Text.

Una vez tengamos el campo “Nombre” creado y configurado correctamente, pulsaremos en el botón de Guardar, momento en el cual nos aparecerá una nueva ventana para solicitar el nombre de la tabla. Lo introducimos y pulsamos OK.

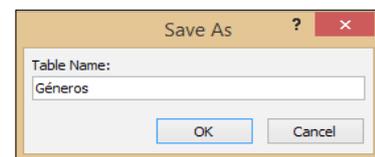


Imagen 7.- Introducción del título de la nueva tabla.

A continuación crearemos la segunda tabla secundaria, Géneros, que será igual de sencilla que la de editoriales, con un único campo con el nombre del género del libro (Short Text)

Field Name	Data Type
ID	AutoNumber
Género	Short Text

Imagen 8.- Esquema de la tabla de Géneros.

La tercera tabla secundaria, la de Autores, tendrá algo más de entidad, pues aparte del nombre del autor (Short Text), insertaremos también su Biografía (Long Text), su Foto (Image) y su Nacionalidad (Short Text).

Field Name	Data Type
ID	AutoNumber
Nombre	Short Text
Biografía	Long Text
Foto	Image
Nacionalidad	Short Text

Imagen 9.- Esquema de la tabla de Autores.

Finalmente crearemos la tabla principal de Libros, que contendrá los campos de Título (Short Text), Año Publicación (Short Text), Portada (Image) y las referencias a las tablas secundarias de Autor, Editorial y Género (Lookups)

Field Name	Data Type
ID	AutoNumber
Título	Short Text
Autor	Lookup
Editorial	Lookup
Año Publicación	Short Text
Portada	Image
Género	Lookup

Imagen 10.- Esquema de la tabla de libros.

Hay que tener en cuenta que al definir un campo de Lookup, nos aparecerá un Wizard donde podremos indicarle fácilmente si queremos crear valores propios o basarnos en una tabla ya existente, y en este último caso, seleccionar la tabla, el campo de la misma a vincular y el tipo de relación que tendrán los elementos vinculados, a elegir entre:

- No dejar eliminar un elemento de la tabla de destino, si tiene elementos creados en la tabla origen con el valor que se quería eliminar (Si tengo un libro de "Santiago Posteguillo" no me dejaría borrar dicho autor).
- Borrar los elementos de la tabla origen vinculados al elemento borrado de la tabla destino (si elimino el autor "Orson Scott Card" me borraría también todos sus libros).

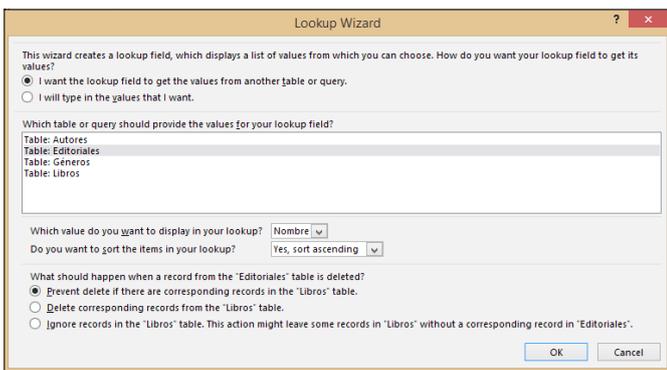


Imagen 11.- Wizard de configuración de Lookups.

También debo advertiros de que en el caso de que hayáis creado previamente la tabla principal (Libros) con campos

tipo "Short Text" y posteriormente las tablas secundarias (Autores, Editoriales, Géneros...), si posteriormente cambias esos campos por "Lookup" a dichas tablas, puede que observes un comportamiento extraño en la entrada de datos del formulario de lista de la tabla principal. A mí me ocurrió que me pedía los números enteros (ID) de los Autores y Editoriales, en lugar de hacer un Intellisense del texto en sí. Para solucionar este "bug" tendríamos que editar la vista en cuestión en el Access, eliminar los campos y volverlos a introducir desde el listado lateral de campos existentes. Al hacer esto, veréis que os vuelve a solicitar los nombres (y no los números) de los campos de tipo lookup.

Llegados a este punto, ya tenemos construidas todas las tablas de nuestra aplicación. Veremos que si navegamos por ellas, se nos han creado 2 vistas por defecto para cada una de ellas.

La vista "List" nos permitirá crear nuevos elementos mediante un formulario de introducción de datos, visualizar un listado de todos los elementos creados, visualizar el detalle del elemento seleccionado en el listado de elementos (con los metadatos que elijamos en el editor de vistas) e incluso visualizar sus elementos relacionados y crear directamente elementos relacionados desde esta vista (por ejemplo visualizar/crear los libros de una editorial concreta).



Imagen 12.- Vista "List" de la tabla Editoriales.

La vista Datasheet nos permitirá editar y modificar rápidamente los registros de una tabla, como si de un Excel se tratara.

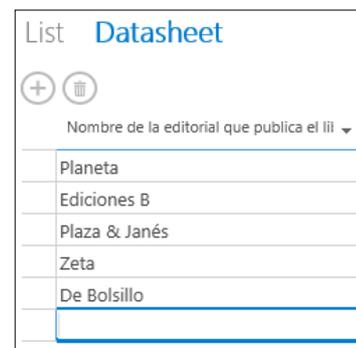


Imagen 13.- Vista "Datasheet" de la tabla Editoriales.

Access nos permite modificar estas 2 vistas existentes o crear nuevas vistas para nuestras tablas. Teniendo en cuenta que dichas vistas forman una parte importante del resultado final de la aplicación, es altamente aconsejable retocarlas (o crear nuevas si así se requiere) para facilitar la lectura y navegación de los datos que almacena nuestra aplicación.

Para modificar una vista de una tabla existente, tendremos que navegar a la lista deseada en nuestra pestaña base del cliente Excel, y seleccionar "Edit" en el menú desplegable del icono de settings de la vista ya existente. Si en lugar de eso pulsamos en el icono del "+" a la derecha de las vistas existentes, crearemos una vista nueva.

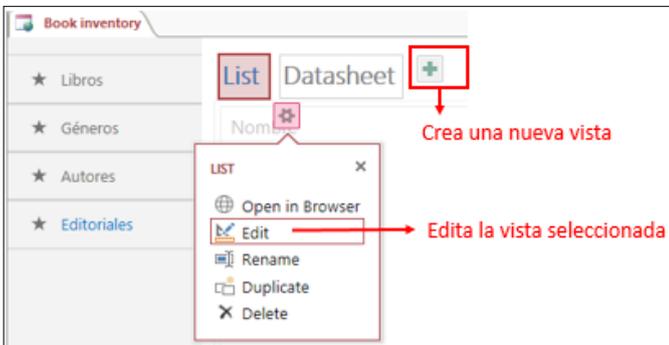


Imagen 14.-Acceso a la edición / creación de vistas de una tabla existente.

Una vez estemos en modo edición/creación de vista, veremos que podemos arrastrar con el ratón cualquier campo disponible en la tabla actual o tablas relacionadas.



Imagen 15.-Listados de campos de las diversas tablas con los que podemos construir las vistas.

El editor de vistas es realmente fluido y a parte del drag & drop de campos, permite también redimensionarlos, eliminarlos y modificarlos de la forma más intuitiva posible. Por ejemplo, para redimensionar el campo "Foto" de la vista "List" de la tabla de Autores, tan solo tendremos que pulsar con el ratón encima del campo y "coger" el cuadradito de la esquina para arrastrarlo y redimensionar el campo a nuestro antojo. Veremos como el listado se va reestructurando a medida que añadimos, movemos o redimensionamos campos de forma totalmente automática. Este es uno de los componentes más "cools" de las Access Apps, y el que dice que también tendrán los FoSL.

También hay que considerar que si vinculamos otras tablas a la tabla actualmente seleccionada, se nos creará un listado con elementos vinculados en la zona inferior de nuestra vista de la tabla. Su función será mostrar los elementos relacionados con el elemento que se esté visualizando en ese momento (por ejemplo, si estamos en Editoriales, y seleccionamos una en concreto, se nos mostrarían aquí todos los libros de dicha editorial). Esta visualización de datos de otra tabla se puede configurar pulsando con el botón derecho encima del nombre de su pestaña, pudiendo seleccionar parámetros tales como la tabla origen, el campo que la vincula, una selección de un total de 4 campos a mostrar de la tabla vinculada, la vista a la que vincula, y el campo y tipo de ordenación por el que queremos que nos muestre el listado de elementos relacionados.

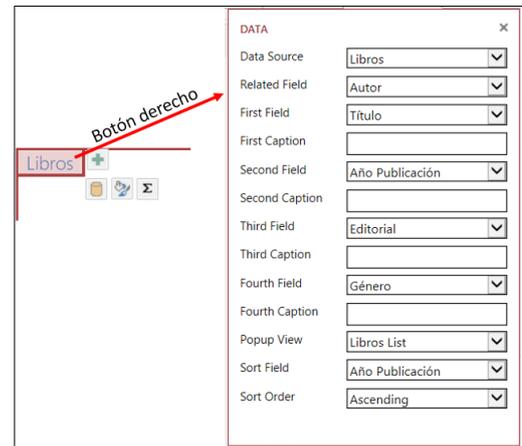


Imagen 16.-Menú de propiedades de las Pestañas a tablas vinculadas.

Cada vez que guardamos los cambios, podremos observar los resultados en la App publicada vinculada automáticamente entre Access y SharePoint.

A continuación os muestro cómo modifiqué la vista "List" de la tabla de "Libros", y el resultado visual final de la misma. Os invito a jugar de la misma forma y comprobar por vosotros mismos los resultados.

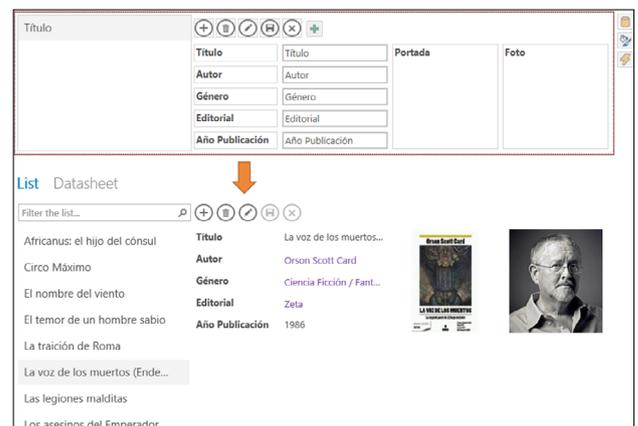


Imagen 17.- Relación entre edición de una vista y su resultado final visual.

De esta forma rápida y sencilla, hemos podido construir una simple, pero llamativa base de datos de libros por autor, editorial y género. Lo cierto es que el resultado visual de las mismas, teniendo el juego que da el editor de vistas, está bastante conseguido y permitirá cumplir a la perfección con la finalidad de nuestras aplicaciones de bases de datos relacionales.

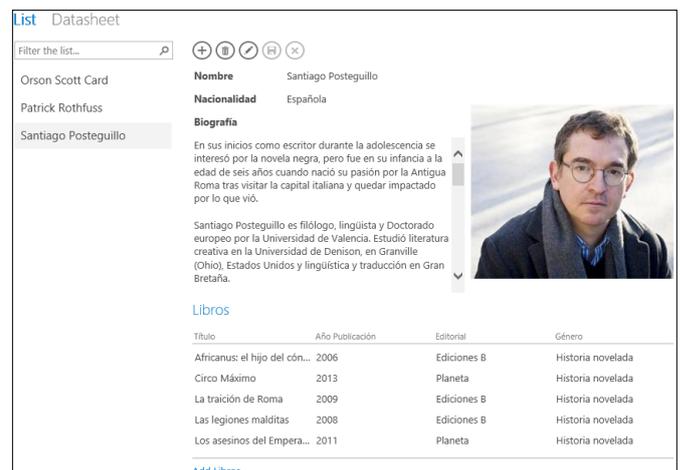


Imagen 18.- Ejemplo de la vista retocada para la tabla de autores.

Se nos acaba el espacio para explicaros más detalles de la construcción de formularios con Access Services, aunque antes de finalizar quisiera comentar que el editor permite crear también fácilmente desplegables en cascada (Cascading dropdowns), aplicar formato condicional a los campos (como ocultarlos en determinadas condiciones y cambiar su color de fondo) y aplicar lógica de campos al estilo InfoPath mediante la inclusión de macros con flujo de condiciones y acciones al más puro estilo workflows de SharePoint Designer. Para crear una macro, tendremos que ir a una vista de una tabla y pulsar el botón izquierdo del ratón. Se nos abrirá una ventana con las opciones de configurar los datos del campo, configurar su formato (visible, invisible...), y el icono de insertar una macro.



Imagen 19.- Acceso al editor de macros

Una vez dentro del Editor de macros, podremos elaborar nuestra lógica para cada campo al construir un flujo de ejecución de determinadas acciones en base a una serie de condiciones.

Como punto final, os dejo una pequeña tabla con las principales ventajas y desventajas del uso de Access Services. Lo podríamos resumir como que es altamente recomendable para aplicaciones de gestión de datos relacionales au-

to-contenidas (que no interactúen con Workflows u otros circuitos/interfaces corporativos).

PROS	Contras
<ul style="list-style-type: none"> • Access 2013 se integra perfectamente con SharePoint (crea una App en el site). • Ideal para gestionar tablas de datos relacionales. • Interface amigable. No requiere código (wizard para macros). • Plantillas de tablas disponibles para acelerar la creación de nuevos forms. • Se pueden paquetizar las aplicaciones y subirlas a la Microsoft App Marketplace • Admite una gran variedad de lógica, formato condicional (Macros) y cascading dropdowns. • Permite obtener datos de fuentes externas. 	<ul style="list-style-type: none"> • Un Workflow no puede acceder a los campos del formulario. • No incorpora seguridad interna (vistas, campos...). • La App resultante pertenece a la BD del propio fichero Access y no se puede utilizar como frontend de una fuente de datos externa a ella (SharePoint List, por ejemplo). • Seguramente encontremos reticencias por parte de los clientes a utilizar Access por resultar un concepto un tanto anacrónico.

Hasta aquí la introducción básica sobre las App Forms. Quedaría mucho por detallar todavía, pero como diría aquél gran autor (Michael Ende), eso sería otra historia que deberíamos contar en otra ocasión.

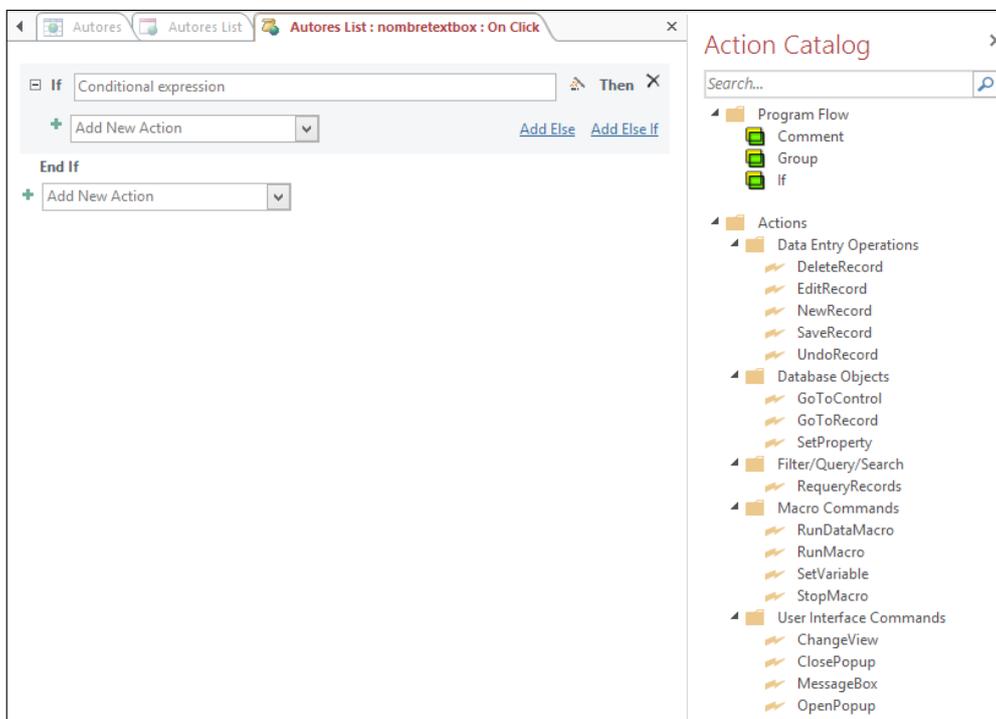


Imagen 20.- Editor de macros, con el flujo de condiciones/acciones en la zona central y el menú de acciones y lógica a la derecha.

Administración de SharePoint Online mediante PowerShell

Resumen

Al igual que sucede con SharePoint OnPremise, es posible realizar ciertas funciones de administración de SharePoint Online por medio de PowerShell, aunque las posibilidades son mucho más limitadas en comparación con todas las funcionalidades de administración disponibles en una instalación OnPremise de SharePoint.

Artículo

En este artículo veremos algunas de las posibilidades de administración de SharePoint Online a través de la interfaz de línea de comandos y el uso de PowerShell. Como se verá en el artículo, a pesar de las “aparentes” limitaciones que el reducido número de comandos PowerShell disponibles para SharePoint Online supone, estas se pueden superar por medio del uso del modelo de objetos en cliente (CSOM) desde PowerShell.

Preparando el entorno para administrar SharePoint Online con PowerShell

Para poder administrar SharePoint Online por medio de PowerShell [necesitamos cumplir una serie de pre-requisitos](#) que pasan por:

- Descargar e instalar [Windows Management Framework 3.0](#).
- Descargar e instalar [SharePoint Online Management Shell](#).

Una vez que tenemos listo el entorno de Administración de SharePoint Online con PowerShell, el siguiente paso consiste en elegir la herramienta de Administración que se va a utilizar. Las posibilidades son dos:

- El propio SharePoint Online Management Shell, disponible una vez se cumplen los pre-requisitos necesarios para la administración de SharePoint Online.
- La aplicación ISE (Integrated Shell Script Environment) disponible no solo en las versiones de servidor de Windows, sino también en las versiones cliente. Esta aplicación proporciona capacidades avanzadas para crear (incluyendo soporte de intellisense, depuración de scripts, etc) y ejecutar scripts PowerShell.

A título particular, os recomiendo utilizar PowerShell ISE. En cualquier caso, elijáis la herramienta que elijáis para poder administrar SharePoint Online tendréis que escribir y ejecutar una secuencia mínima de comandos PowerShell que os permitirán conectaros a un tenant particular de SharePoint Online:

```
$sUserName="<YourOffice365Account>"
$message="Introduce your SPO Credentials"
$sPOAdminCenterUrl="https://<YourDomain>-admin.sharepoint.com/"
$msolcred = get-credential -UserName $sUserName -Message $message
Connect-SPOService -Url $sPOAdminCenterUrl -Credential $msolcred
```

La ejecución de la secuencia anterior produce como resultado la petición de credenciales de acceso al tenant de SharePoint Online en Office 365:

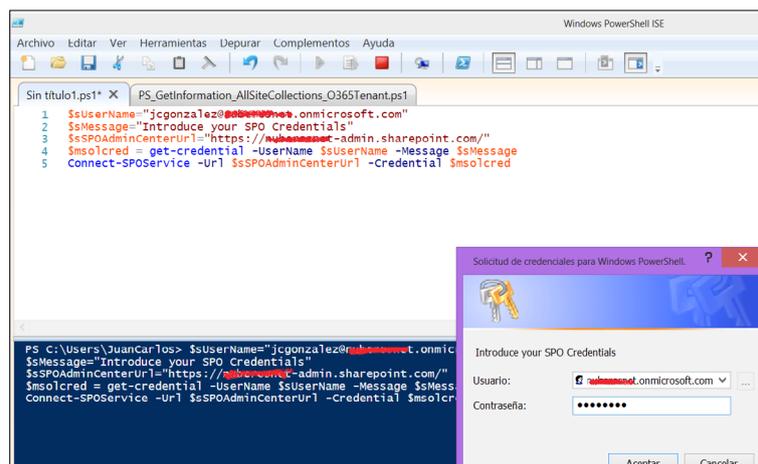


Imagen 1.- Conexión al tenant de SharePoint Online en Office 365 desde PowerShell.

A partir de aquí, ya estamos listos para poder hacer uso de los comandos PowerShell disponibles para SharePoint Online. Pero antes, os recomiendo obtener un listado rápido de los mismos mediante la siguiente secuencia PowerShell:

```
$spoCmdlets=Get-Command | where {$_.ModuleName -eq "Microsoft.Online.SharePoint.PowerShell"}
Write-Host "There are "$spoCmdlets.Count "Cmdlets in SharePoint Online"
$spoCmdlets
```

Si ejecutáis el script anterior, obtendréis que se dispone de un total de 30 comandos PowerShell para SharePoint Online.

Notas:

- Os recomiendo revisar este artículo sobre el [uso de Windows PowerShell ISE en Windows 8](#).
- El número de comandos PowerShell disponible para SharePoint Online puede variar en el tiempo, ya que Microsoft puede añadir nuevos comandos.

Uso de los comandos PowerShell por defecto para SharePoint Online

Como se puede deducir del listado de comandos PowerShell disponibles por defecto para SharePoint Online, con los mismos se pueden hacer actividades como por ejemplo:

- Listar todas las colecciones de sitios disponibles en un tenant de SharePoint Online y su información detallada:

```
$spoSites=Get-SPOSite | Select *
foreach($spoSite in $spoSites)
{
    $spoSite
}
```

La salida correspondiente por pantalla releva el tipo de información que nos proporciona el comando Get-SPOSite para cada colección de sitios:

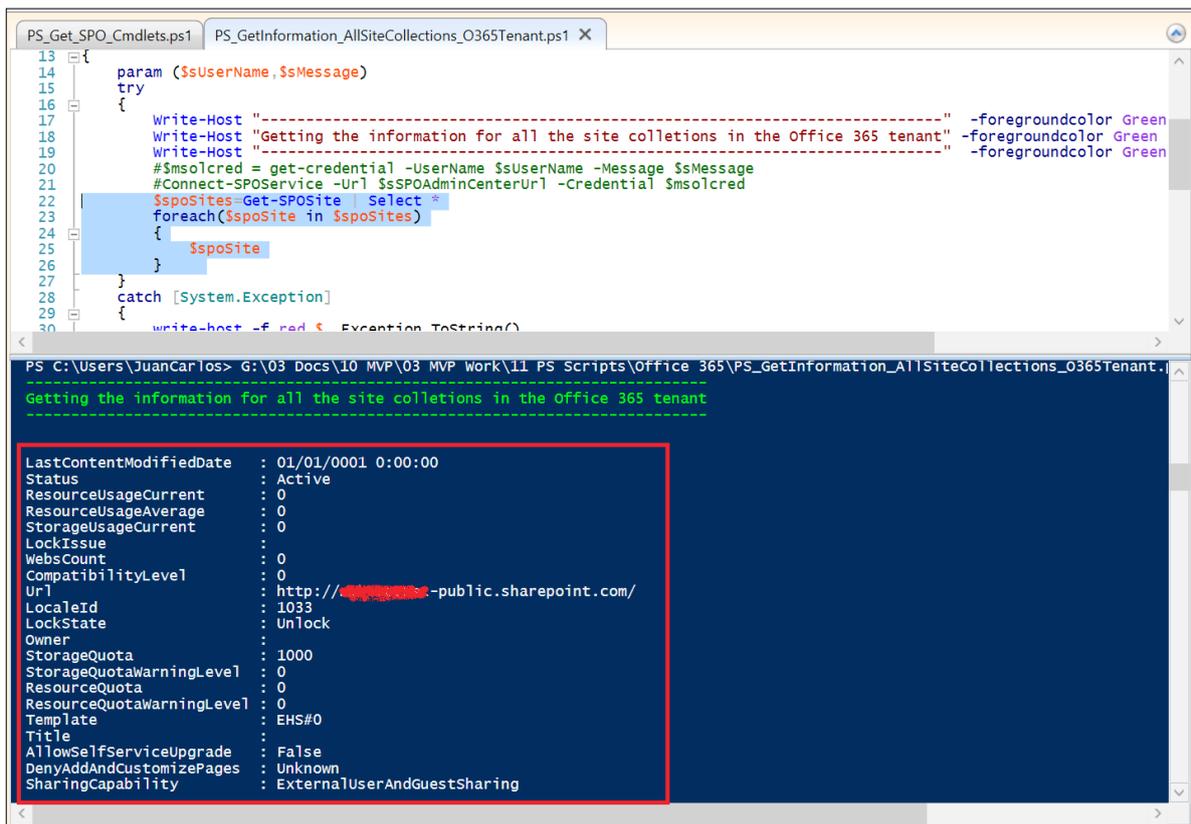


Imagen 2.- Resultado de la ejecución de Get-SPOSite.

- Listar todos los usuarios de todas las colecciones de sitios de un tenant de SharePoint Online.

```
$spoSites=Get-SPOSite | Select *
foreach($spoSite in $spoSites)
{
    Write-Host "Users for "$spoSite.Url -ForegroundColor Blue
    Get-SPOUser -Site $spoSite.Url
    Write-Host
}
}
```

En este caso, para cada sitio disponible en el tenant de

SharePoint Online se visualiza el nombre para mostrar del usuario, el login de usuario y el grupo de SharePoint al que pertenece:

Para cada sitio disponible en el tenant de SharePoint Online se visualiza el nombre para mostrar del usuario, el login de usuario y el grupo de SharePoint



```
PS C:\Users\JuanCarlos> G:\03 Docs\10 MVP\03 MVP Work\11 PS Scripts\Office 365\PS_Get_SPOUsers_AllSiteCollections.ps1
-----
Getting the information for all the site collections in the Office 365 tenant
-----
Users for http://[redacted]-public.sharepoint.com/

Display Name                Login Name                Groups
-----
Company Administrator      s-1-5-21-647558889-2455673044-22145243... {}
Everyone                   true                     {Style Resource Readers}
Everyone except external users spo-grid-all-users/a5bfce6b-582a-4753-... {Visitors}
Juan Carlos González       live.com#balls_best@msn.com {}
Juan Carlos Gonzalez Martin jcgonzalez@[redacted].onmicrosoft.com {}
NT AUTHORITY\authenticated users windows                  {Style Resource Readers}
SharePoint App             app@sharepoint          {}
_SPOCacheFull              ylo001\_spocachefull   {}
_SPOCacheRead              ylo001\_spocacheread   {}
System Account             SHAREPOINT\system       {Owners}
YLO001\_spocrawler_21_3603 ylo001\_spocrawler_21_3603 {}

Users for https://[redacted].sharepoint.com/
Alberto Diaz Martin        live.com#adiazcan@hotmail.com {}
Alberto Morales           amorales@[redacted].onmicrosoft.com {}
Aplicación de SharePoint  app@sharepoint          {}
```

Imagen 3.- Listado de usuarios por colección de sitios en un tenant de SharePoint Online.

- **O bien crear una nueva colección de sitios en un tenant de SharePoint Online:**

```

$$SiteColTitle="SPO PowerShell Site Col"

$$SiteColUrl="https://<Office365Domain>.sharepoint.com/sites/SPOPowerShellSiteC"

$$Owner="<Office365UserLogin>@<Office365Domain>.onmicrosoft.com"

$$LocaleID=3082

$iTemplateID="STS#0"

$iStorageQuota=1024

New-SPOSite -Title $$SiteColTitle -Url $$SiteColUrl -Owner $$Owner -LocaleId $$LocaleID -Template $iTemplateID -StorageQuota $iStorageQuota

```

secuencia de comandos que hacen uso de Add-Type para agregar un ensamblado a partir de la ruta del mismo:

```

Add-Type -Path "<CSOM Path>\Microsoft.SharePoint.Client.dll"

Add-Type -Path "<CSOM_Path>\Microsoft.SharePoint.Client.Runtime.dll"

```

- A partir de aquí, sólo hay que seguir las reglas conocidas en el uso del Modelo de Objetos en Cliente:
 - Definir en primer lugar un objeto de tipo ClientContext a partir de la Url de una Colección de Sitios de SharePoint Online.
 - Indicar unas credenciales de conexión validas a la Colección en cuestión.
 - Comenzar a realizar operaciones en la Colección teniendo en cuenta que en primer lugar hay que inicializar las operaciones mediante el método Load() de ClientContext para a continuación proceder a ejecutar dicha operación con el método ExecuteQuery().
- Por ejemplo, el siguiente script PowerShell muestra como listar todos los sitios bajo una Colección de Sitios en SharePoint Online:

El uso de PowerShell para administrar y trabajar con SharePoint Online es posible no sólo a través de los comandos PowerShell proporcionados por defecto

La salida por pantalla correspondiente es la que se muestra a continuación:

Uso del modelo de objetos en cliente desde PowerShell

Más allá de lo que podemos hacer con los comandos PowerShell disponibles por defecto para SharePoint Online, podemos hacer uso del Modelo de Objetos en Cliente lo que incrementa notablemente las posibilidades de administrar y operar SharePoint Online desde la interfaz de línea de comandos. Para poder hacer uso del Modelo de Objetos en Cliente en PowerShell tenemos que seguir las siguientes pautas:

- En primer lugar, necesitamos disponer de los ensamblados del modelo de objetos en cliente que podemos obtener o bien de una instalación de SharePoint 2013 o bien a partir del [SharePoint Online Client Components SDK](#) y del [SharePoint Server 2013 Client Components SDK](#).
- Una vez que tenemos disponibles los ensamblados, tenemos que cargar los mismos en el entorno de ejecución de PowerShell utilizando la siguiente

```

$host.Runspace.ThreadOptions = "ReuseThread"

#Definition of the function that gets all the site collections information in a SharePoint Online tenant
function Get-SPOSitesInSC
{
    param ($sSiteColUrl,$sUsername,$sPassword)
    try
    {
        Write-Host "-----" -foregroundcolor Green
        Write-Host "Getting all the sites in a SharePoint Online Site Collection" -foregroundcolor Green
        Write-Host "-----" -foregroundcolor Green

        #Adding the Client OM Assemblies
        Add-Type -Path "<CSOM Path>\Microsoft.SharePoint.Client.dll"
        Add-Type -Path "<CSOM_Path>\Microsoft.SharePoint.Client.Runtime.dll"

        #SPO Client Object Model Context
        $spoCtx = New-Object Microsoft.SharePoint.Client.ClientContext($sSiteColUrl)
        $spoCredentials = New-Object Microsoft.SharePoint.Client.SharePointOnlineCredentials($sUsername, $sPassword)
        $spoCtx.Credentials = $spoCredentials

        #Root Web Site
        $spoRootWebSite = $spoCtx.Web

        #Collection of Sites under the Root Web Site
        $spoSites = $spoRootWebSite.Webs

        #Loading operations
        $spoCtx.Load($spoRootWebSite)
        $spoCtx.Load($spoSites)
        $spoCtx.ExecuteQuery()

        #We need to iterate through the $spoSites Object in order to get individual sites information
        foreach($spoSite in $spoSites){
            $spoCtx.Load($spoSite)
            $spoCtx.ExecuteQuery()
            Write-Host $spoSite.Title " - " $spoSite.Url -ForegroundColor Blue
        }
        $spoCtx.Dispose()
    }
    catch [System.Exception]
    {
        write-host -f red $_.Exception.ToString()
    }
}

#Required Parameters
$sSiteColUrl = "https://<YourSharePointSiteCollectionUrl>"
$sUsername = "<YourSPOUser>"

#sPassword = Read-Host -Prompt "Enter your password: " -AsSecureString
$sPassword=convertto-securestring "<SPOUserPasswrod>" -asplaintext -force
Get-SPOSitesInSC -sSiteColUrl $sSiteColUrl -sUsername $sUsername -sPassword $sPassword

```

```

PS C:\Users\JuanCarlos> G:\03 Docs\10 MVP\03 MVP Work\11 PS Scripts\Office 365\PS_Get_SPO_SiteCollectionSites.ps1
-----
Getting all the sites in a SharePoint Online Site Collection
-----
C. Búsquedas - https://[redacted].sharepoint.com/sites/SPSaturdayCo1/CBusquedas
C. Decisional - https://[redacted].sharepoint.com/sites/SPSaturdayCo1/BICenter
C. Documentación - https://[redacted].sharepoint.com/sites/SPSaturdayCo1/CDocumentacion
C. Registros - https://[redacted].sharepoint.com/sites/SPSaturdayCo1/CRegistros
GScan Online Lite - https://[redacted]-8e843979cb8113.sharepoint.com/sites/SPSaturdayCo1/GradientGloadScanSharepointSPA
Project Management - https://[redacted]-8e843979cb8111.sharepoint.com/sites/SPSaturdayCo1/Project Management
Proyectos - https://[redacted].sharepoint.com/sites/SPSaturdayCo1/Proyectos
Shreyan Sticky Notes - https://[redacted]-8e843979cb8110.sharepoint.com/sites/SPSaturdayCo1/ShreyanStickyNotes
Survey Management Lite - https://[redacted]-8e843979cb810F.sharepoint.com/sites/SPSaturdayCo1/SurveyManagement

```

Imagen 4.- Listado de sitios de una Colección de Sitios de SharePoint Online.

Conclusiones

El uso de PowerShell para administrar y trabajar con SharePoint Online es posible no sólo a través de los comandos PowerShell proporcionados por defecto, sino también a partir de las posibilidades del uso del Modelo de Objetos en Cliente desde scripts y funciones PowerShell. De esta forma se habilitan escenarios interesantes en el trabajo con SharePoint Online ya que se pueden automatizar tareas como la creación de contenedores de información (Colecciones de Sitios, Sitios, Listas, Bibliotecas de Docu-

mentos), carga masiva de documentación, migración desde SharePoint On-Premise, y finalmente el despliegue de soluciones personalizadas.

JUAN CARLOS GONZÁLEZ MARTÍN

MVP SharePoint Server | Arquitecto de Soluciones en LKS

jcgonzalezmartin1978@hotmail.com

jc.gonzalez@lks.es

[@jcg1978](https://www.linkedin.com/in/jcgm1978)

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



46

Automatizaciones para editar el IIS en SharePoint 2010/2013

Resumen

A menudo caemos en la tentación de editar configuraciones en el IIS a mano. Acciones tan típicas para un administrador como cambiar un encabezado de host, asignar un puerto, editar el web.config o cambiar el usuario de un pool de aplicación pueden traernos de cabeza en un entorno de SharePoint si las hacemos manualmente. Estos típicos cambios no quedan reflejados en el entorno y no se replicarán cuando decidamos añadir nuevos servidores a nuestra granja. En este artículo repasaremos las diferentes técnicas y mecanismos para poder configurar las aplicaciones web de SharePoint de forma correcta.

Introducción

SharePoint es un producto fantástico en cuanto a configuración y escalabilidad se refiere. Todas las configuraciones y todas las acciones que realizamos en el entorno mediante la Administración Central se almacenan en sus bases de datos para ser aplicadas en todas las máquinas que forman parte del conjunto o granja de servidores. Por ejemplo, si configuramos el correo, una plantilla de formulario, una cuenta administrada, la ejecución de un job... o un sinfín de opciones más, éstas quedan registradas en las bases de datos de SharePoint y se propagan a todos los servidores de la granja necesarios (los que tengan activados los roles o servicios correspondientes).

Y en el caso de las aplicaciones web, no es necesario crearlas una vez por servidor, ya que se crean de forma centralizada y los cambios se distribuyen a todos los servidores que tienen el servicio de aplicación web activo. Imaginaros un entorno con 10 frontales web... ¿sería una locura tener que crear una aplicación 10 veces!

A la hora de escalar nuestra granja para añadir más servidores, éstos cogen todos estos parámetros de configuración automáticamente según el rol que le hayamos dado al servidor o los servicios que hayamos activado. Y si el servidor es un frontal web, replicará todas las aplicaciones con toda la configuración que le hayamos dado. De esta forma, es rápido y sencillo escalar nuestros entornos.

Para mantener esta homogeneidad y que todo funcione correctamente, Microsoft recomienda que todos nuestros despliegues de personalizaciones se automaticen en pa-

quetes WSP. Estos paquetes se despliegan sobre todos los servidores de la granja necesarios de manera automática. Si añadimos un nuevo servidor a la granja, SharePoint le desplegará también los WSP que necesite sin que sea necesaria nuestra intervención.

Los desarrolladores están cada vez más concienciados en hacer sus despliegues mediante paquetes de solución WSP,

ya que son la forma correcta de hacer que todas las personalizaciones, artefactos y código se distribuyan correctamente por todos los servidores. Pero los administradores de IT no lo tienen tan fácil. Hay numerosas operaciones de configuración que puede ser necesario repetir en todas las máquinas de la granja, como editar un web.config, añadir una entrada en el registro de Windows, cambiar algún archivo, etc. Es posible realizar cualquiera de estas acciones de forma manual, pero lo recomendable es intentar automatizarlas para que se desplieguen por toda la granja. De esta forma ahorraremos tiempo evitando repetir una misma acción en varias máquinas y reduciremos el error humano. Además de este despliegue automático, es necesario buscar los mecanismos que hagan que estos cambios se repliquen solos cuando añadamos nuevos servidores a la granja.

En este artículo nos centraremos en diversos mecanismos que nos permitirán configurar el IIS de manera automática, haciendo que los cambios se desplieguen en todas las máquinas necesarias y que se repliquen cuando se añadan nuevos servidores. Modificar parámetros a mano en el IIS no es una buena práctica, existen mecanismos en SharePoint para automatizarlos y almacenarlos en las bases de datos de configuración. Los cambios manuales están sujetos al error humano, ya que podemos cometer errores si tenemos que replicarlos en todas las máquinas. Además, no quedan registrados en ningún lugar. Si no los documentamos bien, podemos olvidar que los hicimos y no se replicarán cuando decidamos añadir nuevos servidores a nuestra granja. A continuación veremos cómo abordar las configuraciones más importantes en el IIS:

Edición del web.config

Los cambios en el web.config pueden realizarse usando el objeto SPWebConfigModification, que contiene las propiedades y métodos necesarios para añadir nuevos nodos y editar los existentes.

Cuando modificamos el web.config de una aplicación usando este objeto, los cambios se almacenan en un listado en la base de datos de configuración y se propagan en todos los servidores en los que existe la aplicación. Además, si añadimos servidores nuevos en un futuro, SharePoint les hará estos cambios automáticamente, ya que los tiene almacenados. Otra gran ventaja es que disponemos de operaciones para eliminar los cambios de este listado, lo que propiciará que se eliminen estas modificaciones del web.config en todos los servidores.

Este objeto se encuentra en el namespace Microsoft.SharePoint.Administration y puede usarse desde el modelo de objetos o desde PowerShell. Como el artículo está destinado principalmente a administradores de IT, veremos los ejemplos con PowerShell.

Para ilustrar el uso de este comando haremos un ejemplo práctico. Vamos a añadir un atributo al nodo "httpRuntime" llamado "executionTimeout" con un valor de 300 (Imagen 1).

```

Antes:
<system.web>
<httpHandlers />
<customErrors mode="On" />
<httpRuntime maxRequestLength="51200" requestValidationMode="2.0" />
<authentication mode="Forms">
  <forms loginUrl="/_login/default.aspx" />
</authentication>
<identity impersonate="true" />
Después:
<system.web>
<httpHandlers />
<customErrors mode="On" />
<httpRuntime maxRequestLength="51200" requestValidationMode="2.0" executionTimeout="300"/>
<authentication mode="Forms">
  <forms loginUrl="/_login/default.aspx" />
</authentication>
<identity impersonate="true" />
    
```

Imagen 1.- Archivo web.config antes y después de modificar.

Primero cargamos el Snap-in con los cmdlet de SharePoint y creamos un objeto SPWebConfigModification. No existe un cmdlet de PowerShell para esta operación, así que hay que crear un nuevo objeto usando el modelo de objetos con el cmdlet New-Object:

```

Add-PSSnapin Microsoft.SharePoint.PowerShell
$webConfigMod = New-Object Microsoft.SharePoint.Administration.SPWebConfigModification
    
```

Una vez tenemos creado el objeto, asignaremos las propiedades:

```

$webConfigMod.Path = "configuration/system.web/httpRuntime"
$webConfigMod.Type = 1
$webConfigMod.Name = "executionTimeout"
$webConfigMod.Value = 300
$webConfigMod.Sequence = 0
$webConfigMod.Owner = "mita_fix_1"
    
```

Estas propiedades contienen el cambio que necesitamos.

Son las siguientes:

Path: Es la ruta XPath al elemento del web.config que queremos modificar. En el ejemplo vamos a modificar un nodo llamado httpRuntime, que está dentro de system.web, que está dentro de configuration. En la Imagen 1 puede visualizarse la ubicación.

Type: es el tipo de modificación a realizar. Esta propiedad es del tipo SPWebConfigModificationType y puede tener 3 posibles valores:

- **EnsureChildNode:** Especifica que la modificación del web.config creará un nodo hijo en el nodo especificado en el valor Path. Valor = 0.
- **EnsureAttribute:** Especifica que la modificación del web.config creará un atributo en el nodo especificado en el valor Path. Valor = 1.
- **EnsureSection:** Especifica que la modificación del web.config creará un nodo único. Valor = 2

En este caso, se ha especificado el Type = 1 para crear un nuevo atributo dentro de httpRuntime, tal y como puede verse en la Imagen 1.

Name: el nombre de la sección o atributo que queremos crear/modificar. En este caso, queremos crear e atributo de nombre executionTimeout.

Value: el valor a darle al atributo que hemos creado. En este caso, queremos darle 300.

Sequence: si creamos múltiples modificaciones con SPWebConfigModification sobre un mismo atributo, la propiedad Sequence indica la prioridad, siendo 0 el más prioritario. En este caso sólo hay una modificación, así que hemos dejado el valor 0, que dará preferencia sobre próximas modificaciones.

Owner: sirve para identificar quién realizó esta modificación. El valor puede ser arbitrario. Se recomienda dar distintos valores en esta propiedad porque puede ser útil a posteriori para encontrar una modificación para eliminarla. Yo suelo asignar un valor distinto en cada modificación para tenerlas a todas identificadas.

Tras asignar todas las propiedades, debemos obtener la aplicación web con el cmdlet Get-SPWebApplication y añadirle la modificación:

```

$webApp = Get-SPWebApplication "http://b13-sp:4242/"
$webApp.WebConfigModifications.Add($webConfigMod)
    
```

A continuación, haremos un Update de la aplicación para que guarde estos cambios:

```

$webApp.Update()
    
```

Y, por último, debemos decirle al objeto SPWebService (el

servicio padre de todos los SPWebApplication) que aplique estos cambios en toda la granja:

```
$webApp.parent.ApplyWebConfigModifications()
```

De esta forma, veremos cómo los web.config de esta aplicación web se actualizan en todos los servidores de la granja añadiendo este cambio. El script completo sería:

A la hora de escalar nuestra granja para añadir más servidores, éstos cogen todos estos parámetros de configuración automáticamente

```
Add-PSSnapin Microsoft.SharePoint.PowerShell

$webConfigMod = New-Object Microsoft.SharePoint.Administration.SPWebConfigModification
$webConfigMod.Path = "configuration/system.web/httpRuntime"
$webConfigMod.Name = "executionTimeout"
$webConfigMod.Value = 300
$webConfigMod.Sequence = 0
$webConfigMod.Type = 1
$webConfigMod.Owner = "mita_fix_1"
$webApp = Get-SPWebApplication "http://b13-sp:4242/"
$webApp.WebConfigModifications.Add($webConfigMod)
$webApp.Update()
$webApp.parent.ApplyWebConfigModifications()
```

Como hemos dicho, estos cambios se registran en un listado en la base de datos de configuración para poder replicarse en todos los servidores existentes y futuros de la granja. Podemos consultar el listado llamando a la propiedad WebConfigModifications de la aplicación web:

```
$webApp.WebConfigModifications
```

Lo que listará las modificaciones guardadas (Imagen 2).

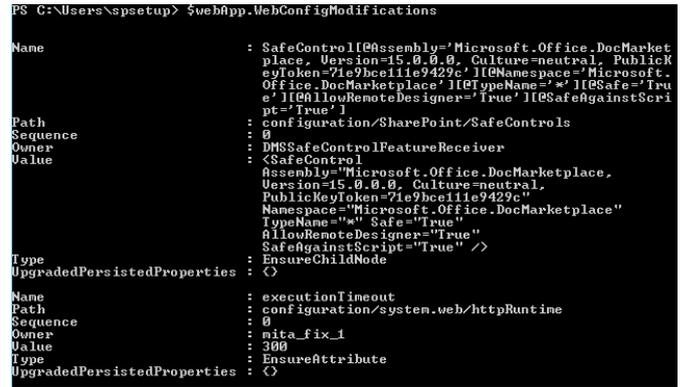


Imagen 2.- Listado de las modificaciones del web.config guardadas en base de datos.

Si quisiéramos borrar una modificación, debemos buscarla en este listado y eliminarla. Para ello es útil el valor que hemos dado en la propiedad Owner, que permitirá encontrar la modificación que queramos rápidamente. Para eliminar la modificación anterior, buscamos el owner "mita_fix_1", lo eliminamos del listado y actualizamos el objeto SPWebService:

```
$modifications = $webApp.WebConfigModifications | ? {$_.Owner -eq "mita_fix_1"}
foreach ($mod in $modifications) { $webApp.WebConfigModifications.Remove($mod) }
$webApp.Update()
$webApp.parent.ApplyWebConfigModifications()
```

Tras ejecutar esto, la modificación desaparecerá de los web.config de todos los servidores.

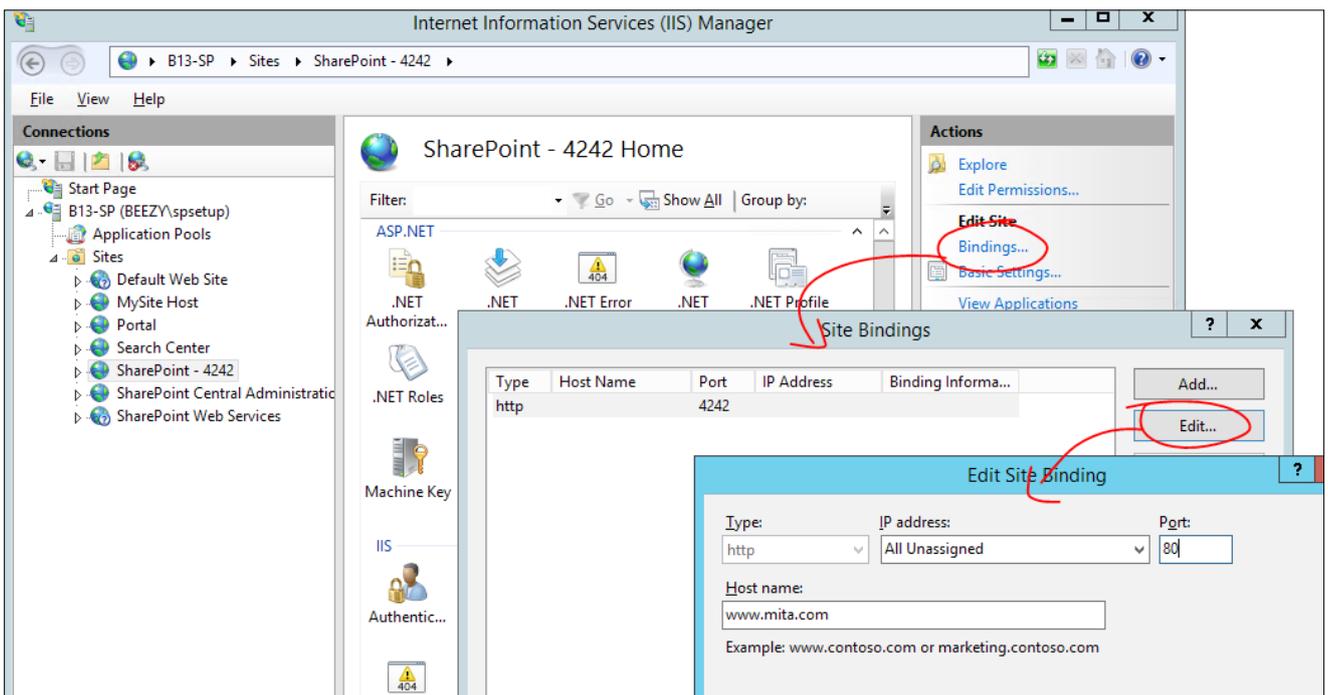


Imagen 3.- Cambio manual del host header y puerto. ¡No se debe hacer!



Cambiar/asignar encabezado de host y puerto

El encabezado de host (host header) de una aplicación y su puerto pueden cambiarse fácilmente desde la consola del IIS. Por ejemplo, tengo una aplicación sin encabezado de host y con puerto 4242 y la quiero cambiar por el encabezado de host www.mita.com y puerto 80 (ver Imagen 3):

Aunque sea un paso sencillo y tentador, no se debe hacer así por tres razones: SharePoint tiene almacenado internamente la configuración de la aplicación web cuando ésta se creó. Si se cambia manualmente estos parámetros, no se replicarán al resto de servidores que tengan esta aplicación web. Además, en determinados escenarios, SharePoint podría sobrescribir estos valores automáticamente. Y por último, cuando se añadan nuevos servidores a la granja, SharePoint creará las aplicaciones con la configuración original que tenía registrada, lo que hará que haya diferencias entre frontales web que debían ser exactamente iguales.

Existe un truco para obtener la configuración de encabezado de host y puerto que SharePoint tiene registrado para una aplicación web, es ejecutar las siguientes líneas de PowerShell:

```
$webApp = Get-SPWebApplication "http://b13-sp:4242/"
$iisSettings = $webApp.GetIisSettingsWithFallback("Default")
$iisSettings.ServerBindings
```

Si seguimos el ejemplo anterior, aunque haya cambiado el encabezado de host y Puerto manualmente, SharePoint sigue teniendo almacenada internamente la configuración original (Imagen 4):

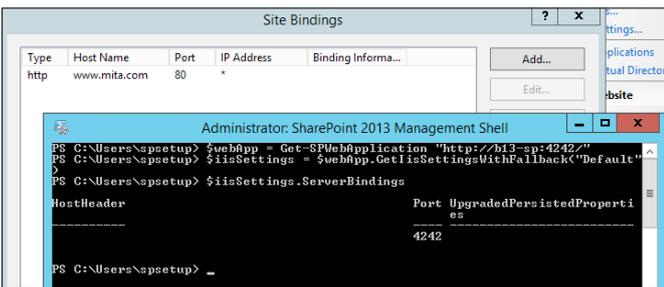


Imagen 4.- Por mucho que cambie el host header y puerto a mano, SharePoint no almacena ese cambio internamente.

Como hemos dicho, este cambio no se ha realizado correctamente y podemos tener problemas en el futuro, ya que SharePoint tiene almacenada una configuración distinta a la que hay realmente en el IIS.

Para cambiar estos parámetros y dejar los cambios registrados en SharePoint, debemos des-extender y re-extender la aplicación web. Al des-extender se eliminará la conexión con el sitio web de IIS. Cuando volvamos a extender la aplicación, se creará un nuevo sitio en el IIS con los nuevos parámetros de encabezado de host, puerto y seguridad que especifiquemos. Estos cambios quedarán correcta-

mente registrados en SharePoint.

cuando se añadan nuevos servidores a la granja, SharePoint creará las aplicaciones con la configuración original que tenía registrada

Si ya tenemos la aplicación extendida para varias zonas, bastará con des-extender de la zona para la que queramos cambiar el encabezado de host y puerto.

Debemos seleccionar la aplicación en la Administración Central, pulsar el desplegable de "Delete" en la cinta y hacer clic en "Remove SharePoint from IIS Web Site". En el cuadro de diálogo, seleccionaremos la zona para la que queramos cambiar el encabezado de host y puerto (en mi caso la zona "Default", la única que tengo) y eliminaremos. Si sólo teníamos contenido de SharePoint, podemos marcar la opción "Yes" para que elimine el sitio físico del IIS, ya que es más limpio crearlo de nuevo al re-extender. Si tenemos alguna personalización (no es recomendable) podemos marcar "No" y luego usaremos ese sitio para crear la aplicación (ver Imagen 5):

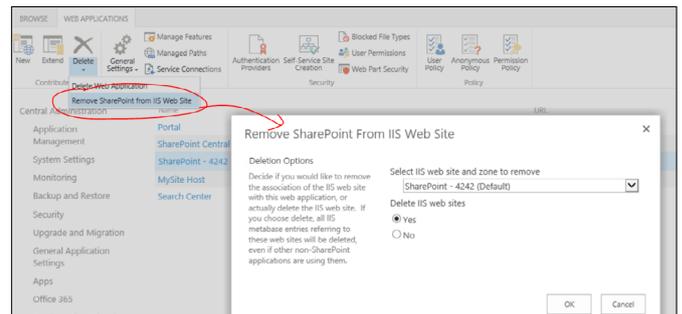


Imagen 5.- Des-extendemos la aplicación para la zona Default.

A continuación, pulsamos el botón "Extend" y rellenamos los parámetros que queremos de encabezado de host, puerto, seguridad y conexiones para crear la aplicación en la zona para la que habíamos eliminado (en mi caso la Default) (Imagen 6):

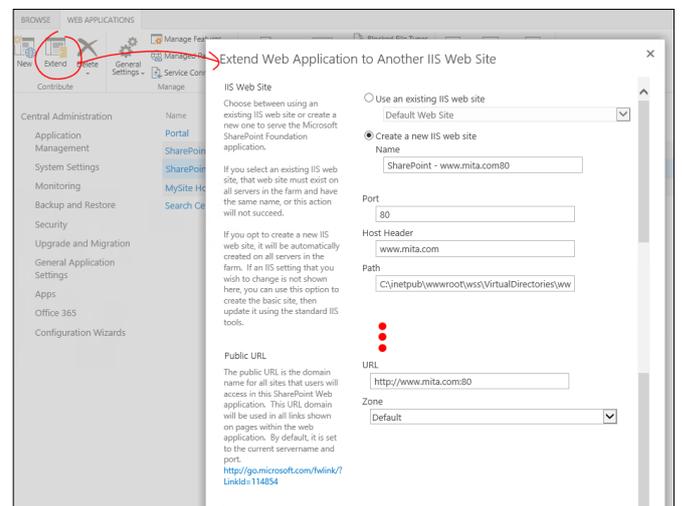


Imagen 6.- Re-extendemos la aplicación para la zona Default.

Si no habíamos borrado el anterior sitio web del IIS, podemos asignarlo marcando la opción "Use an existing IIS web site". Eso sí, antes hay que cambiarle manualmente el encabezado de host y puerto al sitio desde el panel del IIS.

Una vez hayamos hecho esto, hay que comprobar que se haya actualizado la nueva URL en las asignaciones de acceso alternativas (Administración Central > Administración de Aplicaciones > Configurar Asignaciones de Acceso Alternativas) y si no es el caso, cambiarla por la nueva para la zona que hemos re-extendido.

Tras este cambio, si volvemos a ejecutar las líneas de PowerShell, veremos que el nuevo encabezado de host y el nuevo puerto han quedado registrados en SharePoint (ver Imagen 7). Estos cambios se replicarán en todos los servidores frontal web y también se asignarán en futuros servidores que se añadan a la granja.

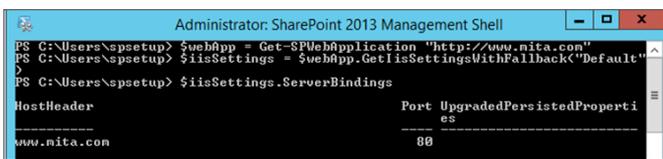


Imagen 7.- SharePoint ha almacenado nuestros cambios.

Podéis consultar el proceso con más detalle en TechNet:

[http://technet.microsoft.com/en-us/library/cc262366\(v=office.15\).aspx](http://technet.microsoft.com/en-us/library/cc262366(v=office.15).aspx)

Cambiar seguridad en la aplicación

Existen vías alternativas en la Administración Central para cambiar la seguridad de nuestras aplicaciones sin que sea necesario hacer un cambio a mano en el panel de IIS o en el web.config. Las más importantes son:

Proveedor de autenticación: podemos cambiar el proveedor de autenticación desde la sección de "Administración de Aplicaciones" de la Administración Central. Marcamos la aplicación deseada y pulsamos el botón "Proveedores de autenticación" en la cinta. Esto despejará los proveedores de autenticación que tengamos en cada zona (cada zona es un sitio de IIS con su web.config). Haciendo clic en uno de ellos, accederemos a la página en la que podremos editar la autenticación y cambiar entre Windows (con autenticación NTLM, Kerberos o básica), FBA u otro proveedor (ver Imagen 8):

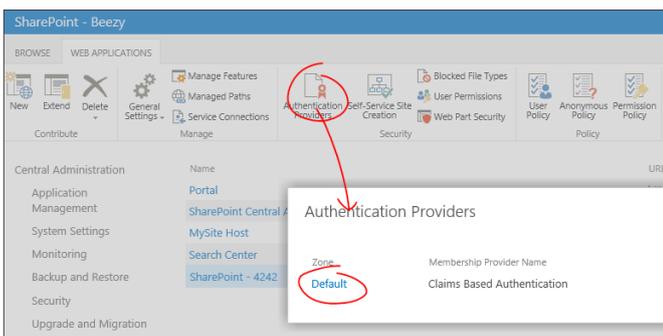


Imagen 8.- Debemos cambiar las opciones de autenticación en la Administración Central y nunca en el IIS.

Hay que tener en cuenta que la configuración de FBA u

otro proveedor distinto a Windows necesitan varias configuraciones adicionales en el web.config de la aplicación. Recordad no implementarlas a mano y usar las técnicas vistas en la anterior sección.

Acceso anónimo: podemos habilitar el acceso anónimo desde la misma sección a la que hemos accedido para cambiar el proveedor de autenticación. Junto a los distintos tipos de autenticación, veremos una sección para habilitar el acceso anónimo. No hay que configurarlo nunca a mano en el IIS ya que, además de que no se desplegará al resto de servidores, no funcionará correctamente el acceso anónimo.

Cambio de la cuenta del pool de aplicación: no se debe cambiar la cuenta de un pool de aplicación (application pool) del IIS de una aplicación SharePoint de forma manual en el panel de administración del IIS. La forma correcta de hacer este cambio es en la Administración Central -> Seguridad -> Configurar Cuentas de Servicio. Aquí podremos seleccionar el pool de la aplicación web o aplicación de servicio que deseemos y cambiar la cuenta que lo ejecuta.

Conclusiones

Hemos visto las técnicas más comunes para automatizar cambios en el IIS y evitar la edición manual. En la medida de lo posible, hay que evitar estos cambios manuales, ya que pueden causarnos muchos problemas si se nos olvida implementarlos en algún servidor o si añadimos nuevos servidores a la granja.

Aunque estas sean las recomendaciones, pueden existir cambios que sean más complicados de automatizar. Si se llega a este punto, me gustaría animar al lector a que intente buscar soluciones alternativas a la edición manual jugando con todas las herramientas que ofrece el entorno. Cuantas menos configuraciones manuales tengamos, mucho mejor funcionará nuestra granja. Quiero proponer rápidamente una serie de opciones que conviene investigar para estos casos y como ayuda general para automatizar cambios:

- Podemos forzar cambios en el web.config, en archivos del servidor e incluso en el registro de Windows desde una solución WSP. Incluso se puede hacer un WSP que añada opciones a la Administración Central que nos permitan hacer cambios comunes, llamar a servicios o cmdlets que hagan cambios, o cualquier cosa que se nos ocurra.
- Muchas de las configuraciones que se hacen en el panel de administración de IIS tienen traducción en el web.config, intentad buscar el cambio que se hace en el web.config para propagarlo con el objeto SPWebConfigModifications (como se ha explicado en este artículo).
- Si hay que hacer cambios sobre alguna propiedad del pool de aplicación, quizás puedas usar el objeto SPApplicationPool, que permite cambiar alguna

de las propiedades.

- Cuando se desarrolla en ASP.NET, es muy típico almacenar en el web.config nodos del tipo “add key” dentro del nodo “appSettings” para guardar variables (por ejemplo datos de conexión, nombres de usuario, nombres de servidor, enlaces...). En el caso de SharePoint os propongo una solución alternativa más limpia, el uso de property bags (<http://social.technet.microsoft.com/wiki/contents/articles/11126.sharepoint-2010-introduction-to-property-bags.aspx>). Incluso hay proyectos en CodePlex para crearlas desde la Ad-

ministración Central (pbs2010.codeplex.com).

Espero que el artículo os sirva de ayuda si queréis abordar cambios similares. Usando estas técnicas conseguiremos granjas más limpias y escalables.

MIGUEL TABERA

MVP de SharePoint Server

miguel.tabera@outlook.com

@migueltabera

www.sinsharepointnohayparaiso.com



Creación de Soluciones personalizadas y automatizadas con Nintex

Resumen

Hoy en día el aumento de la productividad a través de procesos organizacionales optimizados es un inquilino clave de todas las organizaciones que buscan mantener su relevancia en la constante evolución del mundo de los negocios. Encontrar maneras de reducir la cantidad de tiempo y esfuerzo necesarios para completar el trabajo del proyecto se está volviendo aún más crucial, ya que las empresas empiezan a luchar por la optimización de los flujos de trabajo. Para abordar esta preocupación tenemos Nintex, que ofrece a los usuarios la libertad para crear soluciones personalizadas y automatizadas para sus necesidades en cuanto a los flujos de trabajo.

Artículo

Nintex es un conjunto de productos que cubre la gran mayoría de los flujos de trabajo, gestión de procesos de negocio y formas relacionadas con escenarios de SharePoint. Ha simplificado la creación de flujos de trabajo, facilitando la creación a cualquier usuario, con una sencilla interfaz gráfica totalmente integrada en SharePoint que permite crear flujos de trabajo en minutos.

Las principales características de Nintex son las siguientes:

- **Diseño rápido, fácil y sin código**

Nintex proporciona una interfaz muy amigable y fácil de usar, basta con arrastrar, soltar y configurar las diferentes cajas de acciones. No hay solución de código para los usuarios de negocio, analistas y programadores, lo que hace que Nintex sea totalmente transparente y muy visual. Gracias a esta presentación sencilla y sin código, se puede prescindir del departamento de IT para realizar los flujos de trabajo, con lo que se aumenta la cobertura de personal que puede llegar a realizar y gestionar los flujos.

- **Acciones definidas por el usuario**

Estas acciones conocidas como Nintex UDA (User Defined Actions) proporcionan un método escalable para implementar la lógica de negocio reutilizable a través de sus flujos de trabajo. Esto puede ser útil cuando se reutiliza un proceso de flujo de trabajo a través de múltiples sitios o cuando se crea un complejo conjunto de acciones de flujo de trabajo que los usuarios finales pueden configurar y reutilizar fácilmente.

- **Gestión del estado en tiempo real**

Permite gestionar en cualquier momento el estado del flujo. Se puede visualizar rápidamente en que parte se encuentra parado el flujo, ver las estadísticas de las diferentes acciones y detectar cualquier posible error.

- **Integración total con SharePoint**

El diseñador de Nintex Workflow trabaja directamente en SharePoint y no requiere de una instalación en cliente, esta es una gran victoria, ya que elimina el software de cliente, así como el control de IT y la gestión necesaria para implementar este programa.

- **Conexión con otros sistemas**

Permite crear cuentas de usuario en Active Directory, Exchange Server y Office Communications Server.

Lectura y escritura de datos de Microsoft Dynamics CRM.

Establece el calendario y elementos de tarea en Exchange Server.

Proporciona un fácil acceso a datos y procesos de la línea de negocio y sistemas externos a través de servicios web, SQL, LDAP, XML, BizTalk y middleware comunes y otras interfaces abiertas.

- **Conexión entre Nintex Workflow y Office 365**

Nintex Live extiende Nintex Workflow para conectar SharePoint con Office 365 y un número potencialmente infinito de servicios basados en la nube como Yammer, Twitter, Bing,...

- **Capacidades móviles**

El móvil se ha convertido en un negocio en marcha y Nintex no se ha quedado atrás en este sector, ya que ahora ofrece la aplicación de Nintex Mobile para sacar más partido de sus procesos comerciales ya que permiten que los usuarios interactúen con estos cuando están fuera de la oficina.

- **Diseño de formularios**

Nintex Forms es un diseñador basado en la web que permite crear formularios en SharePoint de forma rápida y sencilla. Después se pueden emplear dichos formularios en la mayor parte de los dispositivos

móviles a través de internet, estés donde estés y en cualquier momento. Nintex Forms se integra a la perfección con Nintex Workflow con el fin de automatizar los procesos comerciales y de proporcionar sofisticadas aplicaciones de SharePoint.

- **Rápida rentabilidad de la inversión**

Nintex es un producto que proporciona una rápida rentabilidad de la inversión, dado que nos permite agilizar la creación de procesos, detectar rápidamente los cuellos de botella y aumentar la eficiencia, esto puede reducir drásticamente los costes.

- **Constantes de Nintex**

Nintex nos proporciona la capacidad de crear constantes y poder gestionarlas a nivel de sitio, colección de sitios o granja. Pueden utilizarse en los flujos de trabajo de una manera muy similar a las variables, con la diferencia de que como su propio nombre indica “son constantes”, por lo que tienen un valor definido y se pueden utilizar tanto para guardar unas credenciales, una cadena de conexión, un texto,...

Las constantes nos pueden ser muy útiles a la hora de mover los flujos de trabajo entre entornos diferentes, ya que si las utilizamos no tendremos la necesidad de tener que realizar cambios en el flujo, simplemente cambiaremos el valor de las constantes según nos interese.

Un caso práctico: Solicitud de espacios

Para ver todo el escenario de montaje y despliegue de un flujo de trabajo con Nintex, se plantea a modo de ejemplo realizar un flujo de trabajo para solicitar un espacio de trabajo, ocio, proceso o subproceso en SharePoint. Estos espacios pueden ser plantillas de sitio propias de SharePoint o personalizadas. En esta solicitud tendremos en cuenta dos perfiles, uno será el usuario solicitante y otro el usuario administrador encargado de gestionar la aprobación del flujo de trabajo. El usuario solicitante deberá definir los permisos del espacio solicitado indicando los usuarios que tendrán el rol de propietarios y los que tendrán el rol de miembros.

Creación de la lista

En primer lugar hay que crear una lista a la que se le asociará el flujo de trabajo y donde quedarán guardados los diferentes elementos de las solicitudes. Los campos que hemos utilizado en este caso son los siguientes:

Columna (clic para editarla)	Tipo	Necesario
Título	Una línea de texto	✓
Descripción	Varias líneas de texto	✓
Tipo	Opción	✓
Propietarios	Persona o grupo	✓
Miembros	Persona o grupo	

Imagen 1.-Columnas de la tabla de solicitudes de espacios.

Creación del formulario

Una vez creada la lista podemos configurar un formulario de Nintex para la solicitud, clicando sobre el icono de Nin-

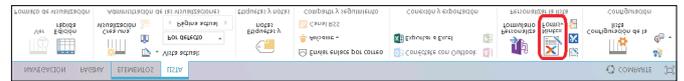


Imagen 2.-Ribbon de la lista.

Por defecto nos aparece el formulario con la plantilla de Nintex Forms, pero esta plantilla se puede cambiar accediendo a la administración central de SharePoint y en el apartado “Nintex Forms Management” seleccionamos “Manage device layouts”. En esta página tenemos la opción de importar la plantilla del formulario en formato XML. Configurar una plantilla nos aporta mucha agilidad a la hora de crear los diferentes formularios, ya que algunas características como el logotipo o los estilos del formulario se mantienen de forma estática en todos los formularios corporativos. Nintex Forms también nos ofrece la posibilidad de configurar el formulario para diferentes plataformas móviles como iPhone, Ipad, Android, Windows Phone,...

Para realizar la migración de formularios entre diferentes entornos, Nintex Forms nos ofrece una solución muy sencilla gracias a los botones de Importar y Exportar que se encuentran en la Ribbon.

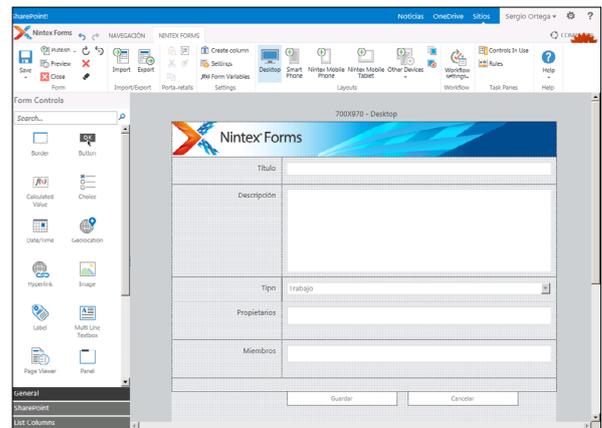


Imagen 3.-Aspecto del diseñador de formularios de Nintex.

Creación del flujo de trabajo

El siguiente paso será crear el flujo de trabajo asociado a la lista de solicitudes, esto se puede realizar desde la Ribbon de Nintex Form o desde la Ribbon de la lista, clicando en ambas sobre el botón de “Configuración del flujo de trabajo”. Ahora empezamos a construir el flujo de trabajo arrastrando, soltando y configurando las diferentes cajas de acciones que necesitamos.

Proporciona una interfaz muy amigable y fácil de usar, basta con arrastrar, soltar y configurar las diferentes cajas de acciones

Cuando tengamos el flujo construido podemos configurar desde el apartado “Workflow settings” la manera en que se iniciará (al crearse o modificarse una solicitud, o manualmente desde las opciones del elemento).

El último paso para que empiece a funcionar el flujo es

publicarlo, esto se hace desde el botón publicar disponible en la Ribbon.

Al igual que Nintex Forms, para migrar los flujos de trabajo entre entornos se hace fácilmente con los botones de Importar y Exportar.

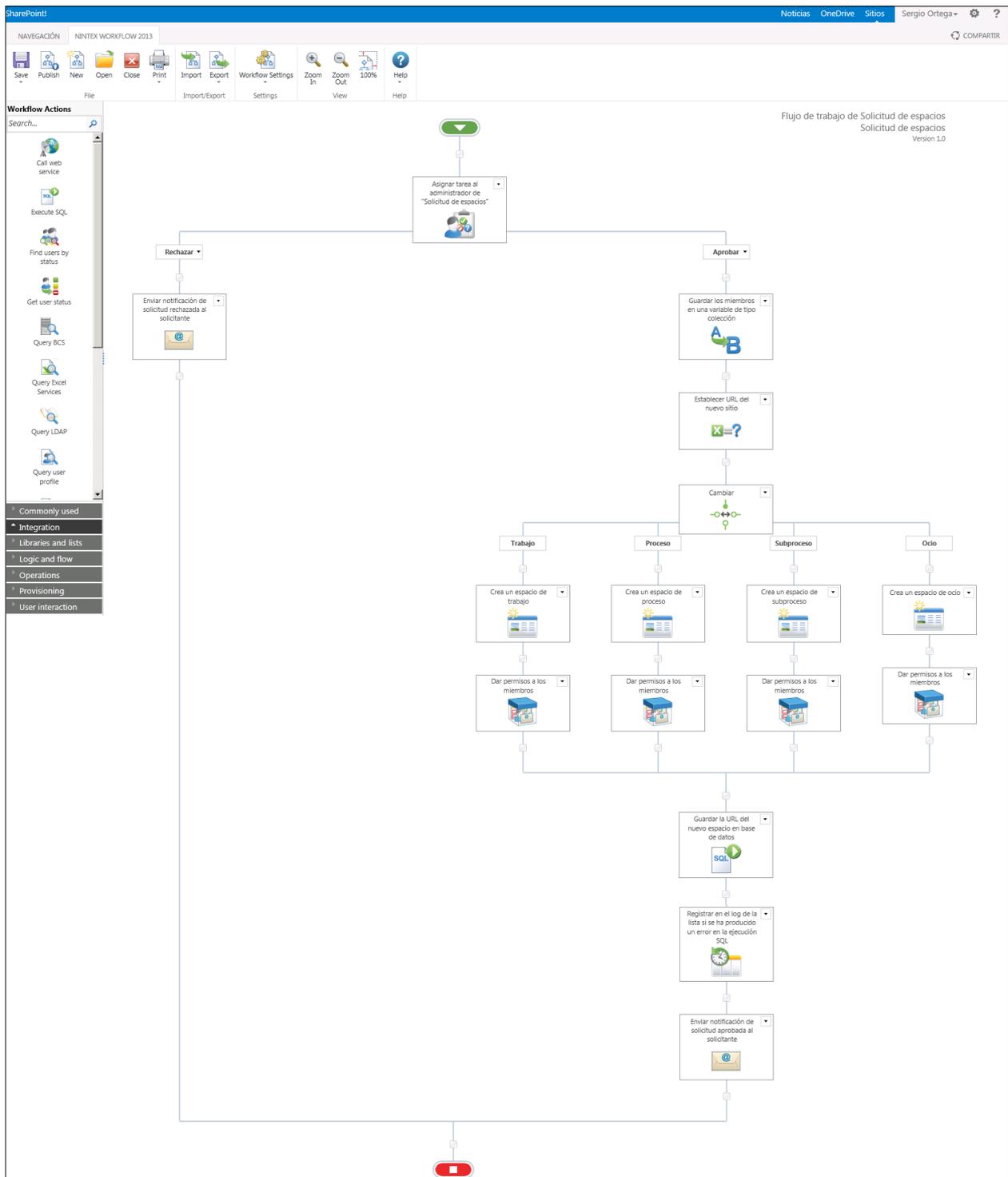


Imagen 4.-Flujo de trabajo para la solicitud de espacios.

Flujo de trabajo corriendo

Cuando se realiza una solicitud se puede ver en tiempo real el estado en el que se encuentra el flujo. Nintex nos ofrece una interfaz muy similar a la del diseñador de flujos, pero en este caso no se permite editar, muestra el conjunto de cajas del flujo y muestra de color verde aquellas acciones que ya se hayan completado y en color amarillo la acción

en la que se encuentra actualmente. Las cajas de color gris representan las acciones que aún no se han ejecutado. Es una manera muy visual de ver rápidamente en qué punto se encuentra el flujo y detectar cualquier tipo de error. Desde esta página el administrador puede terminar el flujo de trabajo, responder a la tarea y ver las estadísticas del flujo.



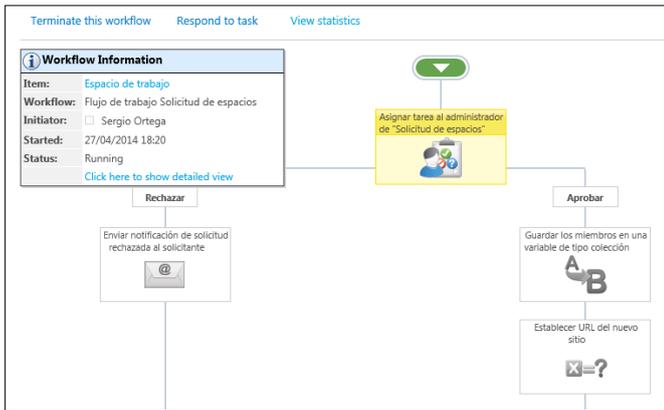


Imagen 5.-Visualización del flujo de trabajo en curso.

Webparts de Nintex Workflow

Nintex nos ofrece dos Web Parts muy útiles para el escenario de solicitudes que hemos puesto de ejemplo, una muestra las tareas que un usuario tiene asignadas y la otra muestra las solicitudes que ha realizado el usuario.

Flujos de trabajo de sitio

A parte de poder crear un flujo de trabajo asociado a una lista, Nintex nos permite crear un flujo de trabajo a nivel de sitio. Los flujos de sitio se pueden programar en el tiempo para que se ejecuten diariamente, semanalmente,... Esto puede ser muy útil por ejemplo si queremos recopilar las estadísticas de un sitio de SharePoint y guardarlas en base de datos, además permite configurar un formulario de inicio, y se podría utilizar para enviar una notificación por correo a los usuarios.

Despliegue automático de flujos de trabajo y constantes

Cuando estamos ante un proyecto en el cual se desarrollan una gran cantidad de procesos diferentes, a la hora de mover nuestros flujos de trabajo de entornos de desarrollo a entornos de producción nos encontramos con un coste significativo si dicha tarea la tenemos que realizar manualmente importando y exportando los flujos. Por esta razón es una buena práctica añadir el despliegue de los flujos a nuestra solución de Visual Studio.

Publicar un flujo de trabajo

Para realizar el despliegue del flujo de trabajo sobre una lista de SharePoint, lo que haremos será llamar al método que publica el flujo en la activación de la característica a nivel de sitio.

En primer lugar se tiene que añadir la referencia a la dll de Nintex Workflow y una referencia web al web service de Nintex con la dirección: “_vti_bin/nintexworkflow/workflow.asmx”, a esta referencia web la hemos llamado “WorkflowWS”.

Una vez añadidas las referencias este sería el código necesario para la publicación del flujo:

```
using Nintex.Workflow;

//RUTA DONDE SE ENCUENTRA EL FLUJO DE TRABAJO
private const string filewfname = "NWSolicitudEspacios\\Espacios.nwf";
//NOMBRE DE LA LISTA DONDE SE VA A PUBLICAR EL FLUJO DE TRABAJO
private const string listname = "Solicitud de espacios";
//NOMBRE DEL FLUJO DE TRABAJO
private const string wfname = "Flujo de trabajo Solicitud de espacios";
//DIRECCIÓN DEL WEB SERVICE DE NINTEX WORKFLOW
private const string WFServiceUrl = "_vti_bin/nintexworkflow/workflow.asmx";

public override void FeatureActivated(SPFeatureReceiverProperties properties)
{
    PublishNWF(properties, filewfname, listname, wfname);
}

public static void PublishNWF(SPFeatureReceiverProperties properties, string filewfname, string listname, string wfname)
{
    try
    {
        SPWeb web = (SPWeb)properties.Feature.Parent;

        byte[] nwfFile = GetWorkflowNWF(properties, filewfname);

        //CONSTRUCCIÓN DE LA URL DEL WEB SERVICE DE NINTEX A PARTIR DEL SITIO DONDE SE ACTIVE LA CARACTERÍSTICA
        string webServiceUrl = web.Url;
        if (!webServiceUrl.EndsWith("/"))
            webServiceUrl += "/";
        webServiceUrl += WFServiceUrl;
        SPSecurity.RunWithElevatedPrivileges(delegate()
        {
            //DECLARACIÓN DEL WEB SERVICE DE NINTEX
            WorkflowWS.NintexWorkflowWS service = new WorkflowWS.NintexWorkflowWS();
            service.Url = webServiceUrl;
            service.UseDefaultCredentials = true;
            //LLAMADA AL MÉTODO DE PUBLICAR WORKFLOW
            service.PublishFromNWF(nwfFile, listname, wfname, true);
        });
    }
    catch (Exception ex)
    {
        throw ex;
    }
}

private static byte[] GetWorkflowNWF(SPFeatureReceiverProperties properties, string filewfname)
{
    using (Stream nwfFile = properties.Definition.GetFile(filewfname))
    {
        byte[] buffer = new byte[1024];
        using (MemoryStream memoryStream = new MemoryStream())
        {
            int read;
            while ((read = nwfFile.Read(buffer, 0, buffer.Length)) > 0)
            {
                memoryStream.Write(buffer, 0, read);
            }

            return memoryStream.ToArray();
        }
    }
}
```

Además de publicar el flujo de trabajo, el SDK de Nintex nos permite realizar muchas otras acciones como eliminar un flujo de trabajo o publicar y programar un flujo de trabajo a nivel de sitio.

Publicar una constante

Al igual que con la publicación del flujo de trabajo, se tienen que añadir las referencias de Nintex Workflow a nuestro proyecto. En esta ocasión llamaremos al método de creación de constantes en la activación de la característica a nivel de Site Collection.

En el ejemplo se muestra como crear una constante de tipo String y otra de tipo Credencial:

```
private void CreateSiteCollectionWorkflowConstants(SPSite site)
{
    //DECLARACIÓN DE LAS CREDENCIALES
    var cred = new CredentialValue(Constants.WorkflowConstants.admuser, Constantes.WorkflowConstants.admpwd);
    if (cred == null) return;
    var serializer = new XmlSerializer(typeof(CredentialValue));
    var sb = new StringBuilder();
    using (var sw = new StringWriter(sb))
    {
        serializer.Serialize(sw, cred);
    }
    var valueCred = sb.ToString();

    //LLAMADA AL MÉTODO PARA CREAR LAS CREDENCIALES
    //WorkflowConstant(string title, string description, CredentialValue value, bool sensitive, Guid siteId, Guid webId, WorkflowConstant.WorkflowConstantType type, bool adminOnly)
    var credConstante = new WorkflowConstant("AdminCred", "Administrador de la colección de sitios de SharePoint", valueCred, true, site.ID, Guid.Empty, WorkflowConstant.WorkflowConstantType.Credential, false);

    //LLAMADA AL MÉTODO UPDATE PARA PUBLICAR LAS CREDENCIALES
    credConstante.Update();

    //LLAMADA AL MÉTODO PARA CREAR LA CONSTANTE DE TIPO STRING
    // WorkflowConstant(string title, string description, string value, bool sensitive, Guid siteId, Guid webId, WorkflowConstant.WorkflowConstantType type, bool adminOnly)
    var stringConstante = new WorkflowConstant("Variable String", "Variable de ejemplo de tipo String", Constantes.WorkflowConstants.variableString, false, site.ID, Guid.Empty, WorkflowConstant.WorkflowConstantType.String, false);

    //LLAMADA AL MÉTODO UPDATE PARA PUBLICAR LA CONSTANTE DE TIPO STRING
    stringConstante.Update();
}
```

Conclusión

NINTEX, como suite de soluciones de trabajo, resulta una herramienta extremadamente útil para nuestro entorno SharePoint, mejorando la productividad en diferentes aspectos.

Como último punto, nos planteamos la idoneidad del producto en las diferentes casuísticas que podemos encontrarlo. Dentro de todas las soluciones, NINTEX Workflows, podemos situarlo en la siguiente escala:

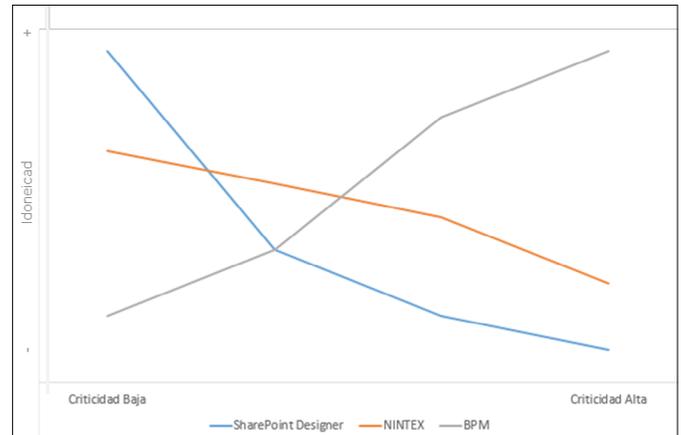


Imagen 6.-Escala de motores de workflows.

Resulta sorprendente como una solución de Workflows bajo SharePoint Designer prácticamente se descarta cuando hablamos del soporte a procesos críticos de negocio, tanto por las actividades disponibles como las capacidades de gobiernos del proceso. De la misma forma, para procesos simples y de baja criticidad, una suite BPM completa de mercado como AgilePoint, por ejemplo, resultaría bastante más pesada que en el caso de SharePoint Designer.

Finalmente, NINTEX, por ser un producto ligero y hospedado bajo SharePoint, nos ofrece una cobertura muy interesante para el proyecto de criticidad media, con todos los monitores y capacidades que el producto ofrece, descritas a lo largo del presente artículo.

SERGIO ORTEGA

SharePoint Consultant at The Birchman Group
sergio.ortegamartin@birchmangroup.com
<http://birchman4sharepoint.wordpress.com>

La vista Datasheet nos permitirá editar y modificar rápidamente los registros de una tabla, como si de un Excel se tratara

Nosotros



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

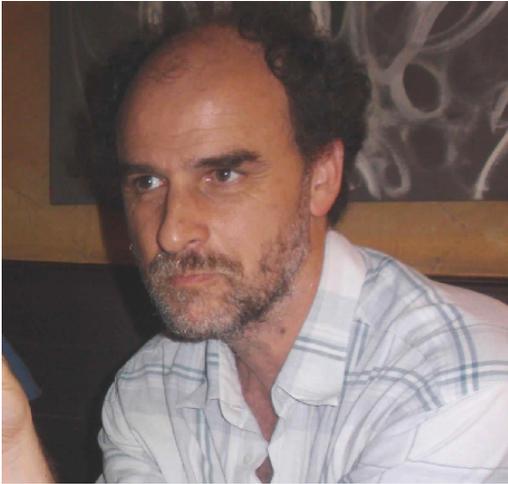
Fabián Imaz, MVP de SharePoint Server trabaja en el mundo del desarrollo de software desde hace más de 10 años, teniendo la suerte de trabajar en distintas arquitecturas y tecnologías Microsoft. Pertenece a la firma Siderys, <http://www.siderys.com> empresa de desarrollo de Software especializada en 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

| COMPARTIMOSS |