

COMPARTIMOSS

Revista especializada en Tecnologías SharePoint



ENTREVISTA

MVC PARA SHAREPOINTEROS III

RECORDS MANAGEMENT EN SHAREPOINT 2013

SITIOS DE LA COMUNIDAD

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 por su propio contenido.

DIRECCIÓN GENERAL

- Gustavo Vélez
- Juan Carlos González
- Fabián Imaz
- Alberto Díaz

Contactate con nosotros

revista@compartimoss.com
gustavo@gavd.net
jcgonzalezmartin1978@hotmail.com
fabiani@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:
twitter.com/CompartiMOSScom

VISÍTENOS:

<http://www.compartimoss.com>

Contenidos

03

Editorial

Tiene antes sus ojos el número 17 de la revista

04. Documentos Relevantes, limitaciones por umbral

10. Introducción a las plantillas de elementos de contenido (Display templates)

16. Records Management en SharePoint 2013

Una de las características que nos ofrece SharePoint es la capacidad de implementar soluciones

23

Usar Framework JavaScript en nuestros desarrollos SharePoint

31. PowerShell Cmdlets: Crearlos o Morir

PowerShell es la herramienta por excelencia para administrar a SharePoint.

37. Entrevista

A Haarón González.

39. Novedades en las herramientas de desarrollo de Office para Visual Studio 2013

42. Creación de aplicaciones de Windows Phone

Con contenido de SharePoint parte 2

49. ASP.NET MVC para SharePointeros (III)

51

ASP.NET MVC para SharePointeros (III)

Cerraremos la serie de programación de aplicaciones de SharePoint con ASP.NET MVC

56. Sitios de la Comunidad

59. Configurando el acceso a usuarios externos

En nuestro sitio de colaboración de SharePoint Online

61. Como configurar el Content Type Hub de SharePoint 2010

65. Clonando SharePoint 2010

68. Cómo se hizo la Apps OTic-Visor de Lista de OnlineTic

71. Ejemplos de CAML en Client Object Model (JavaScript).

71. El autor destacado del Número 16

Tiene antes sus ojos el número 17 de la revista, otro gran número donde hemos tratado de superar al anterior y creemos que hemos logrado la meta que nos propusimos, aunque no hubiera sido posible sin la colaboración de los autores que, número a número, proveen artículos de primer nivel para ser publicados.

Como saben, cada 3 meses lanzamos la revista y muchos lectores la esperan para aprender y conocer acerca de SharePoint. Pero también desde el último año, desde la dirección, hemos intentado brindar nuevos beneficios a nuestros lectores, como la página web donde se accede a todo el contenido o las aplicaciones para leer la revista desde distintos dispositivos móviles.

Este número no será la excepción, y por este motivo queremos Compartir con ustedes una nueva iniciativa que está relacionada con los autores: a partir de este número y en cada uno de los siguientes, uno de ellos será reconocido como "autor destacado" y será candidato para participar en la elección del mejor artículo del año entre los 4 autores destacados del año.

El mecanismo es sencillo y transparente, el autor del artículo que sea más leído en una edición será el autor destacado en la siguiente edición de la revista, así que son ustedes, los lectores, quienes deciden quién es el autor destacado de cada número. Por supuesto, el reconocimiento al trabajo realizado por los autores no sería totalmente transparente si los directores, que comúnmente también suelen escribir artículos, participaran de esta iniciativa, por lo que dichos artículos quedarán excluidos de la votación, puesto que en la revista queremos premiar a todas aquellas personas que aportan su granito de arena escribiendo para cada número. Esperamos que disfruten este número, como nosotros disfrutamos haciéndolo.

ALBERTO DÍAZ
adiazcan@hotmail.com

JUAN CARLOS GONZÁLEZ
jcgonzalezmartin1978@hotmail.com

GUSTAVO VÉLEZ
gustavo@gavd.net

FABIÁN IMAZ
fabiani@siderys.com.uy





04

Documentos Relevantes, limitaciones por umbral

Resumen

Nos pasa a menudo escuchar hablar de threshold, boundaries y supported limits, pero la realidad es que como programadores o configuradores raras veces nos encontramos con situaciones límites, por un lado sólo por el hecho de trabajar con máquinas virtuales locales o hosteadas donde realizamos las personalizaciones, desplegamos, testeamos, por el otro lado raras veces nos toca trabajar con listas o bibliotecas con tantos elementos como para alcanzar estos valores establecidos.

Artículo

Entonces nos llega un requerimiento del Cliente donde el elemento web out-of-the-box Relevant Documents presenta el error siguiente:



Imagen 1.- Error elemento web out-of-the-box Relevant Documents.

Entonces hacemos memoria, buscamos información y empezamos a tomar conciencia de lo que está sucediendo. En este artículo vamos a hacer un breve repaso sobre qué son las limitaciones y por qué existen y cómo podemos sortear este problema desde el punto de vista de configuraciones o desde un elemento web custom.

Repasemos algunos conceptos.

- Límites máximos - “boundaries: límites absolutos que no se pueden exceder por diseño. Es importante entender estos límites para no hacer suposiciones incorrectas al diseñar una granja de servidores.
- Umbrales - “threshold”: Los umbrales son límites que tienen un valor predeterminado que no se puede exceder a menos que se modifique el valor. En ciertos casos, los umbrales se pueden exceder para dar cabida a desviaciones en el diseño de la granja de servidores, pero es importante entender que al hacerlo se puede ver afectado el rendimiento de la granja además del valor efectivo de otros límites.

- Límites admitidos - “Supported Limits”: límites configurables que se han establecido en un valor probado de manera predeterminada. Los límites máximos son límites absolutos que no se pueden exceder por diseño. Es importante entender estos límites para no hacer suposiciones incorrectas al diseñar una granja de servidores.

Cuando vemos estos conceptos nos surgen preguntas como: ¿qué pasa si mi lista supera la cantidad máxima de 5000 elementos? ¿Dejará de funcionar? ¿Por qué me han limitado? ¿Es posible cambiar el límite? ¿De dónde surge el valor 5000?

Intentaré responder a estas preguntas en los siguientes párrafos para lograr entender o, mejor dicho, comprender qué es lo que está sucediendo con mi SharePoint.

En SharePoint Server 2010, hay ciertos límites que son de diseño y que no se pueden exceder y otros límites que se establecen en valores predeterminados que el Administrador de la granja de servidores puede modificar.

El umbral y los límites admitidos se establecieron a través del testing y la observación del comportamiento de la granja mediante el aumento de la carga hasta el punto donde los servicios y las operaciones de dicha granja alcanzan un límite eficaz operacionalmente. Algunos servicios y componentes pueden tolerar una carga más alta que otros que en algunos casos se debe asignar un límite basado en el promedio de varios factores.

...Tener en cuenta que al crear y usar columnas indizadas, cada índice de columna adicional consume recursos extras de la base de datos...

Límite	Valor Máximo	Tipo de Límite	Notas
Elementos	30,000,000 por lista	Admitido	Se puede crear listas muy grandes usando vistas estándar, jerarquías de sitio y navegación de metadatos. Este valor puede variar según el número de columnas de la lista y el uso de la lista.
Umbral de la vista de lista	5,000	Umbral	Especifica la cantidad máxima de elementos de lista o biblioteca que puede procesar simultáneamente una operación de base de datos, como una consulta, fuera del intervalo diario de horas que establece el administrador y durante el cual las consultas no tienen restricciones.
Umbral de la vista de lista para auditores y administradores	20,000	Umbral	Especifica la cantidad máxima de elementos de lista o biblioteca que puede procesar simultáneamente una operación de base de datos, como una consulta, cuando un auditor o administrador con los permisos apropiados realiza la operación. Esta configuración funciona junto con Permitir invalidación de modelos de objetos.

Tabla 1.- Algunos Límites para Listas y elementos

Vamos a ver algunos de estos límites como ejemplo, en el sitio de Microsoft se puede obtener el listado completo. Resumiendo el cuadro anterior, las listas y bibliotecas de SharePoint 2010 pueden contener un número máximo de 30.000.000 de elementos. En un sitio SharePoint el umbral de vista de la lista es de 5000 elementos para los usuarios y de 20.000 para auditores y administradores.

SharePoint 2010 introdujo la funcionalidad de Límites de Lista (List Throttling) a raíz del mecanismo de bloqueo de SQL Server.

Cuando se consulta una lista que tiene más de 5.000 elementos el SQL Server aumenta el nivel de bloqueo de nivel de fila a nivel de bloqueo de tabla. Esto significa que cualquier consulta posterior será bloqueada por lo que la concurrencia de consultas no es posible. Lo que nos lleva a la conclusión que si las consultas (de elementos en listas o bibliotecas) concurrentes quedaron bloqueadas entonces no podrá existir la “colaboración”.

Veamos qué es lo que pasa en segundo plano:

1. Los datos de lista o biblioteca de una colección de sitios se almacenan en una tabla de base de datos de SQL Server, que usa consultas, índices y bloqueos para mantener globalmente el rendimiento, el uso compartido y la precisión.
2. Las vistas filtradas con índices de columna (y otras operaciones) crean consultas de base de datos identifican un subconjunto de columnas y filas y devuelven este subconjunto al equipo.
3. Los umbrales y los límites ayudan a limitar las operaciones y equilibrar los recursos para varios usuarios simultáneos.
4. Desarrolladores con privilegios pueden utilizar el modelo de objetos y modificar aumentando temporalmente los umbrales y límites en una aplicación custom.
5. Los administradores del conjunto de servidores pueden especificar ventanas de tiempo dedicadas para que todos los usuarios puedan realizar operaciones ilimitadas durante las horas de menor actividad.
6. Los profesionales de la información pueden usar vistas, estilos y límites de página adecuados para mostrar los datos en la página más rápidamente.

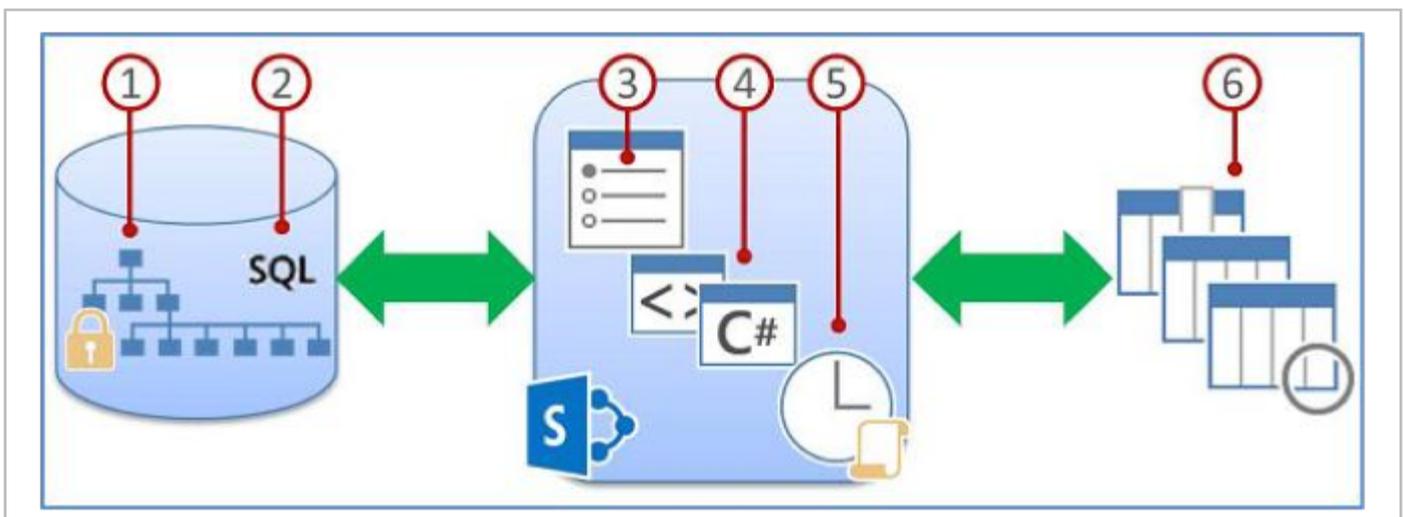


Imagen 2.- Proceso que tiene lugar en Segundo Plano

Para ayudar a mejorar el rendimiento de una biblioteca o lista grande, pueden indizar hasta 20 columnas. En general, el índice de una columna permite encontrar más rápido las filas que se desean en función de los valores de esa columna, incluso cuando se está trabajando con millones de elementos. Cuando se combinan índices con vistas filtradas, se puede obtener más rápido los elementos que se necesiten.

Es importante tener en cuenta que al crear y usar columnas indizadas, cada índice de columna adicional consume recursos extras de la base de datos y agrega alguna sobrecarga a cada operación para mantener el índice. Por lo tanto, debemos agregar índices únicamente a las columnas que se usarán activamente para filtrar en vistas de la lista o biblioteca.

Resumiendo, esta funcionalidad (List Throttling) protege al servidor de sobrecargas no intencionales limitando el número de elementos devueltos en la consulta. Si la vista es mayor al límite del umbral, entonces los usuarios obtienen el mensaje que los datos no pueden ser devueltos como se mostraba en la imagen del elemento web Relevant Documents.

Hasta aquí se han respondido la mayoría de las dudas que generalmente tenemos, ¿qué pasa si mi lista supera la cantidad máxima de 5000 elementos? Nada, deberemos tener en cuenta tener las distintas vistas de la lista con filtros apropiados para no devolver “todos los elementos” ¿Dejará de funcionar? No, si excede la cantidad de elementos devolverá un mensaje al usuario ¿Por qué me han limitado? Para evitar el bloqueo del SQL Server a nivel de tabla. ¿Es posible cambiar el límite? Sí, porque es del tipo umbral. ¿De dónde surge el valor 5000?, de testeos y pruebas hasta llegar a un valor ideal.

Ahora que tengo todas mis respuestas... ¿cómo hago para resolver el problema del elemento web Relevant Documents cuando en mi sitio existen bibliotecas que superan ampliamente el valor del umbral?

Solución 1: Modificar los valores de estos umbrales a nivel de Web Application

Cuando dentro de la empresa no hay una política establecida sobre las modificaciones a estos valores, o bien se encuentren en un ambiente más permisivo en cuanto a la modificación de estos parámetros, podemos modificar estos límites a nivel de Web Application.

La funcionalidad de Limitación de Recursos que incorpora SharePoint 2010 se configura a nivel de Web Application. Esta configuración se aplica a todos los Site Collections y Sitios bajo dicha Web Application.

De forma predeterminada, la comprobación de recursos está activa, por lo que un timer job se ejecutará cada 5 segundos. Este job revisa el estado de los recursos del servidor contra los indicadores de performance. Si la revisión devuelve un fallo 3 veces seguidas, se activa un estado de throttling. El servidor permanecerá en ese estado hasta que el chequeo devuelva un estado exitoso.

Los usuarios podrán ver el error 503 Servidor ocupado y deberán refrescar la página hasta ver si la solicitud se ha completado. Los recursos que este timer job revisa predeterminadamente son CPU del Servidor, memoria, Solicitudes en Cola y Tiempo de Espera de Solicitudes.

A través del SharePoint Central Administration:

1. Abrir el SharePoint 2010 Central Administration.
2. Hacer clic en Application Management.
3. En la sección Application Management , hacer clic en Manage Web Applications.
4. Seleccione una aplicación web (se activará el menú ribbon).
5. Hacer clic en el menú desplegable de Configuración General.
6. Hacer clic en la opción de Limitación de Recursos

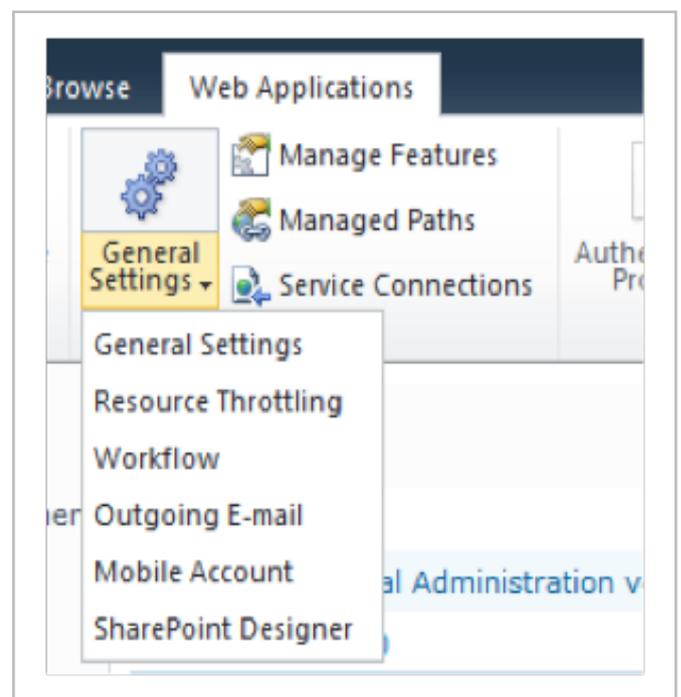


Imagen 3.- Acceso al Resource Throttling.

(Resource Throttling).

Entre las configuraciones de esta opción se podrán modificar por ejemplo, los siguientes valores de umbrales:

- Umbral de la vista de lista: por defecto es 5000.
- Umbral de vista lista para administradores y auditores: por defecto a 20000.
- Ventana de Tiempo diaria para consultas grandes.

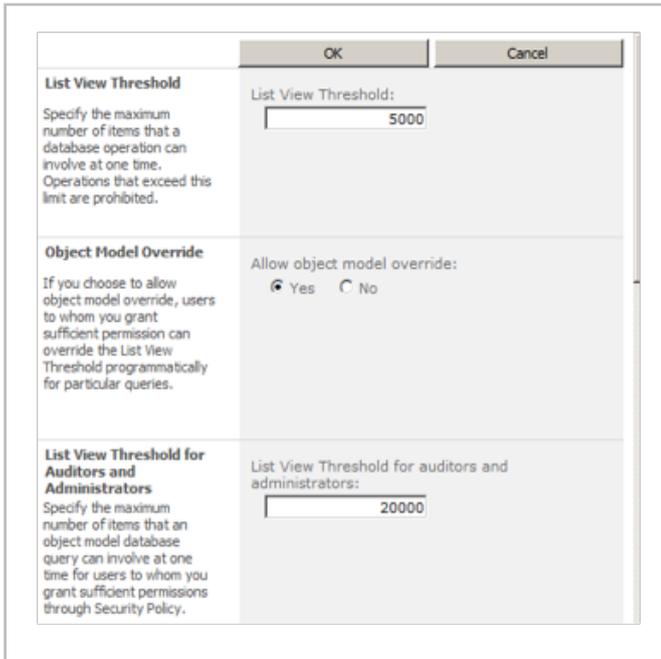


Imagen 4.- Valores predeterminados para los umbrales mencionados arriba.

Ventana de Tiempo diaria para consultas grandes. Esta opción permite crear una ventana de tiempo durante el día, el ideal cuando el uso es bajo, donde las consultas podrán ejecutarse sin estas limitaciones.

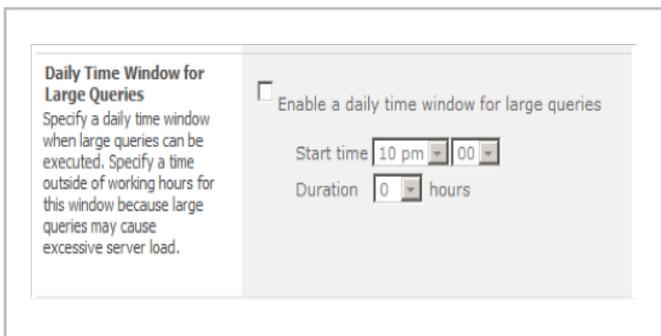


Imagen 5.- Habilitar ventana de tiempo para grandes consultas

A través de PowerShell

```
$site = SPSite http://[url del site collection]
```

Umbral de la vista de lista: especificar el número máximo de elementos que una operación de base de datos puede devolver por vez. Las operaciones que excedan este límite están prohibidas.

```
$site.WebApplication.MaxItemsPerThrottledOperation
```

Umbral de la vista de lista para Auditores y Administradores: especificar el número máximo de elementos que una consulta sobre la base de datos puede devolver a la vez para usuarios a quienes se les otorgó suficientes permisos según una Política de Seguridad.

```
$site.WebApplication.  
MaxItemsPerThrottledOperationOverride
```

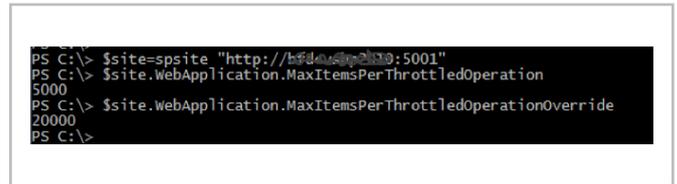


Imagen 6.- Obtener los valores de los umbrales de vista de lista a través de PS.



Imagen 7.- Establecer la ventana de tiempo a través de PS.

Solución 2: a través de un elemento web custom

Podemos aplicarla cuando por política de la empresa nos encontramos que la solución anterior no es viable y sin embargo es necesario ejecutar una consulta a una lista con una vista que devuelve más de 5000 elementos.

Para este caso en especial lo que se hizo fue programar un elemento web similar al out-of-the-box Relevant Documents.

Más allá de la presentación gráfica, lo interesante aquí es cómo ejecutar una consulta en todas las bibliotecas de documentos del Site Collection para traer aquellos documentos que tienen como autor al usuario logado, o bien el usuario logado es quien los ha modificado o es quien los ha desprotegido para realizar algún cambio, dentro de un período dado de tiempo.

Lo primero que definimos son las columnas que vamos a utilizar en la consulta CAML



Imagen 9.- Código - Definir las columnas a utilizar en el query.

Luego, tenemos que definir la consulta CAML, en el caso de documentos relevantes, debemos armar la cláusula Where de acuerdo a la lógica que ya se mencionó.

```
//Criterio de selección
string whereClause = "<Where><And><And><Or><Or><Eq><FieldRef ID='" + createdByID + "' />"
+ "<Value Type='Integer'><UserID/></Value>"
+ "</Eq><Eq><FieldRef ID='" + modifyByID + "' /><Value Type='Integer'><UserID/></Value></Eq></Or>"
+ "<Eq><FieldRef ID='" + checkOutUserID + "' /><Value Type='Integer'><UserID/></Value></Eq>"
+ "</Or><Geq><FieldRef ID='" + lastModifiedID + "' />"
+ "<Value Type='DateTime' IncludeTimeValue='True'> + fromDate + "</Value>"
+ "</Geq></And><Neq><FieldRef ID='" + relativeURL + "' />"
+ "<Value Type='Text'>Folder</Value></Neq></And></Where>";

//Orden
string orderByClause = "<OrderBy><FieldRef ID='" + lastModifiedID + "' Ascending='False' /></OrderBy>";
```

Imagen 9.- Código - Definir el criterio de selección

Si ejecutamos el query sin definir cómo se comportará con respecto a los límites establecidos, lo más probable es que nos encontremos con una SPQueryThrottledException con un mensaje avisándonos que se ha superado el valor máximo de elementos configurado por el administrador.

Para ello debemos cambiar la propiedad QueryThrottleMode del SPSiteDataQuery.

La enumeración SPQueryThrottleOption tiene tres valores: Default, Override y Strict.

```
//Configurar la consulta CAML
query.Query = whereClause + orderByClause;
query.QueryThrottleMode = SPQueryThrottleOption.Override;
query.RowLimit = uint.Parse(this.PageSize);

//Ejecutar la consulta para todas las bibliotecas de Documentos
query.Lists = "<Lists ServerTemplate='101' />";
```

Imagen 10.- Código - Definir el modo de consulta.

Si se utiliza el valor predeterminado, el umbral de la vista de lista estándar se aplica a todos los usuarios excepto los administradores del servidor local, que no están obligados a ese umbral. Si establece el modo de Override de consulta, los usuarios que tengan los permisos necesarios en la política de usuario de aplicación web pueden realizar la consulta en el umbral superior “administradores y auditores”. Finalmente, si se establece el modo de consulta a Strict, esto hace que el umbral de la vista de lista estándar se aplica a todos los usuarios.

La siguiente tabla muestra cuales umbrales se aplican de acuerdo al usuario para cada valor de la enumeración SPQueryThrottleOption:

Tipo de Usuario	Default	Override	Strict
Miembro del Sitio	Standard	Standard	Standard
Propietario del Sitio	Standard	Standard	Standard
Administrador del Site Collection	Standard	Standard	Standard
Web app policy: site collection admin	Standard	Más alto	Standard
Web app policy: site collection auditor	Standard	Más alto	Standard
Administrador de la Farm	Standard	Standard	Standard
Local server admin	Ilimitado	Ilimitado	Standard

Tabla 2.- Usuarios y Umbrales-

Si vemos la tabla anterior observamos que el Administrador del Site Collection no accede al umbral más alto, pero sí en caso de estar definido dentro de la política de la Web Application. Esto se define a nivel de Web Application dentro del Central Administration. Para terminar el armado

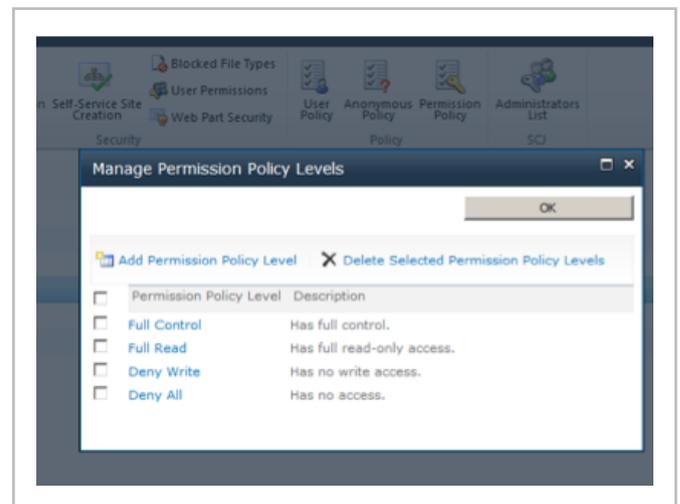


Imagen 11.- Política de Niveles de Permisos.

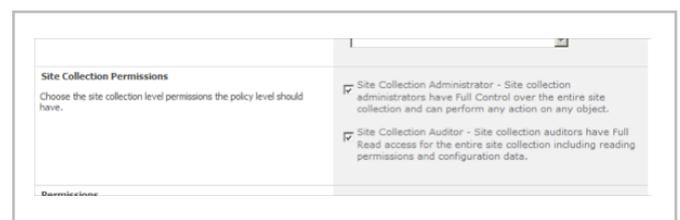


Imagen 12.- Política de nivel de permisos del Site Collection.

del query, nos resta definir las columnas que devolverá el mismo, el scope del SPSiteDataQuery y ejecutar la consulta.

Una vez que se obtienen los datos podrán visualizarse de acuerdo a los requerimientos de diseño elegidos.

SharePoint 2013

...tenemos que definir la consulta CAML, en el caso de documentos relevantes, debemos armar la cláusula Where de acuerdo a la lógica...

```

//Definir los campos que devolverá la consulta
string viewFields = "<FieldRef ID='" + titleID + "' />"
+ "<FieldRef ID='" + createdByID + "' Nullable='TRUE' />"
+ "<FieldRef ID='" + modifyByID + "' Nullable='TRUE' />"
+ "<FieldRef ID='" + lastModifiedID + "' Nullable='TRUE' />"
+ "<FieldRef ID='" + docIconID + "' Nullable='TRUE' />"
+ "<FieldRef ID='" + fileDirRef + "' Nullable='TRUE' />"
+ "<FieldRef ID='" + linkFileName + "' Nullable='TRUE' />"
+ "<FieldRef ID='" + relativeURL + "' Nullable='TRUE' />"
+ "<FieldRef ID='" + checkOutUserID + "' Nullable='TRUE' />";

query.ViewFields = viewFields;

//Query all the SPWebs in this SPSite
query.Webs = "<Webs Scope='SiteCollection'>";

SPSite site = SPContext.Current.Site;
{
    using (SPWeb web = site.OpenWeb())
    {

        //Run the query
        DataTable resultsTable = web.GetSiteData(query);
    }
}

```

Imagen 13.- Ejecución de la consulta SPSiteDataQuery

Para cerrar, en SharePoint 2013 mantiene este tipo de funcionalidad de límites y para este tipo de umbrales los mismos valores que en 2010 de forma predeterminada. Desde la administración central podrán cambiarse a nivel de Web Application, o bien a través del modelo de objetos programáticamente.

Bibliografía

1. SharePoint Server 2010 capacity management: Software boundaries and limits. Technet. [Online] [http://technet.microsoft.com/en-us/library/cc262787\(v=office.14\).aspx](http://technet.microsoft.com/en-us/library/cc262787(v=office.14).aspx).
2. Designing large lists and maximizing list performance (SharePoint Server 2010). Technet. [Online] [http://technet.microsoft.com/en-us/library/cc262813\(v=office.14\).aspx](http://technet.microsoft.com/en-us/library/cc262813(v=office.14).aspx).
3. Software boundaries and limits for SharePoint 2013. Technet. [Online] <http://technet.microsoft.com/en-us/library/cc262787.aspx>.
4. Query Throttling in SharePoint 2013. Andy Burns' SharePoint Blog. [Online] <http://www.novolocus.com/2013/04/26/query-throttling-in-sharepoint-2013/>.
5. Manage lists and libraries with many items. Office. [Online] <http://office.microsoft.com/en-us/sharepoint-server-help/manage-lists-and-libraries-with-many-items-HA102771361.aspx?CTT=1>.

SILVINA ANDREA PIZZARULLI

SharePoint Technical Leader en Baufest

<http://silvinapizzarulli.blogspot.com>



Introducción a las plantillas de elementos de contenido (Display templates)

Resumen

En anteriores versiones de SharePoint, para modificar la vista de los elementos de los WebParts, tanto diseñadores como desarrolladores debían tener conocimientos de XSL y XSLT para poder modificar sus “plantillas” y así poder conseguir el resultado visual que buscaban. SharePoint 2013, dentro de su cambio de orientación hacia las nuevas tendencias web, HTML5, introduce un nuevo modelo de plantillas de elementos que han llamado “Display templates” y que vamos a analizar en este artículo.

Artículo

SharePoint 2013 viene con muchos cambios orientados para una mejor integración con las nuevas tendencias web y, uno de estos cambios, muy evidente para los diseñadores, es la introducción de las plantillas de elementos de contenido que hacen uso de HTML y JavaScript y que vienen a sustituir a las antiguas plantillas XSL que obligaban a diseñadores y/o desarrolladores a tener que saber manejar XSLT para generar plantillas.

De esta forma, los nuevos WebParts que entran en juego con SharePoint 2013 hacen uso de estas nuevas plantillas, como por ejemplo el WebPart de Búsqueda de contenido que se hará indispensable para cualquier sitio de SharePoint 2013, dada su potencia y que complementa al antiguo Content Query WebPart en este nuevo paradigma. En la Imagen 1 vemos una ilustración del uso de plantillas para elementos de contenido en un WebPart de Búsqueda de Contenido (Content Search WebPart).



Imagen 1.- Ejemplo del uso de plantillas en un Web Part de Búsqueda de Contenido.

Ventajas de las plantillas de elementos de contenido

Las plantillas de elementos de contenido tienen muchísimas ventajas pero me centraré en las que considero más importantes:

- **Uso de HTML y JavaScript:** Es la primera gran ventaja que podemos observar y con la que nos quitaremos de encima las complicadas y tediosas plantillas XSL además de permitirnos generar plantillas potentes para mejorar la experiencia de los usuarios.
- **Plantillas a nivel de Sitios o Colección de Sitios** que nos permitirá disponer de estilos reutilizables para los WebParts de contenido y búsqueda y así mantener un aspecto común entre los elementos listados en nuestros sitios de SharePoint 2013 sin necesidad de un gran esfuerzo.
- **Posibilidad de generar plantillas para todo tipo de contenidos**, ya sean refinadores, resultados de búsqueda, etc.
- **Posibilidad de añadir propiedades administradas en las plantillas.**
- **Posibilidad usar la función “OnPostRender()”** para realizar acciones JavaScript tras la carga de los elementos.
- ...

Dónde se encuentran las plantillas de elementos de contenido

Disponemos de dos formas de acceder a ellas dependiendo de lo que deseemos hacer:

- **Para consultas y/o edición de propiedades.** En este caso tendremos que acceder al “Administrador de diseños” y posteriormente seleccionar la opción “5”.

Editar plantillas para mostrar” tal y como se muestra en la Imagen 2, lo que nos llevará a un listado de plantillas como el de la Imagen 3 donde podremos ver todas las plantillas que hay disponibles o incluso editar algunas propiedades importantes como podemos observar en la Imagen 4.

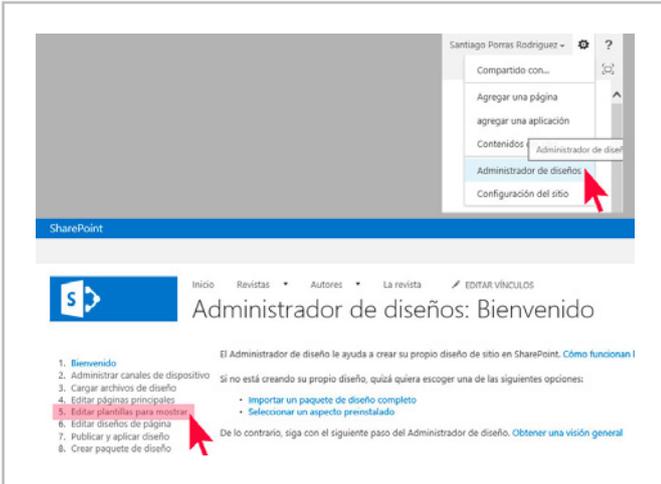


Imagen 2.- Acceso al listado de plantillas de elementos de contenido.

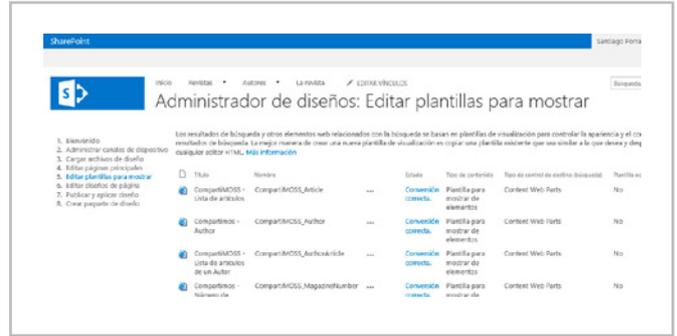


Imagen 3.- Listado de plantillas de elementos de contenido.

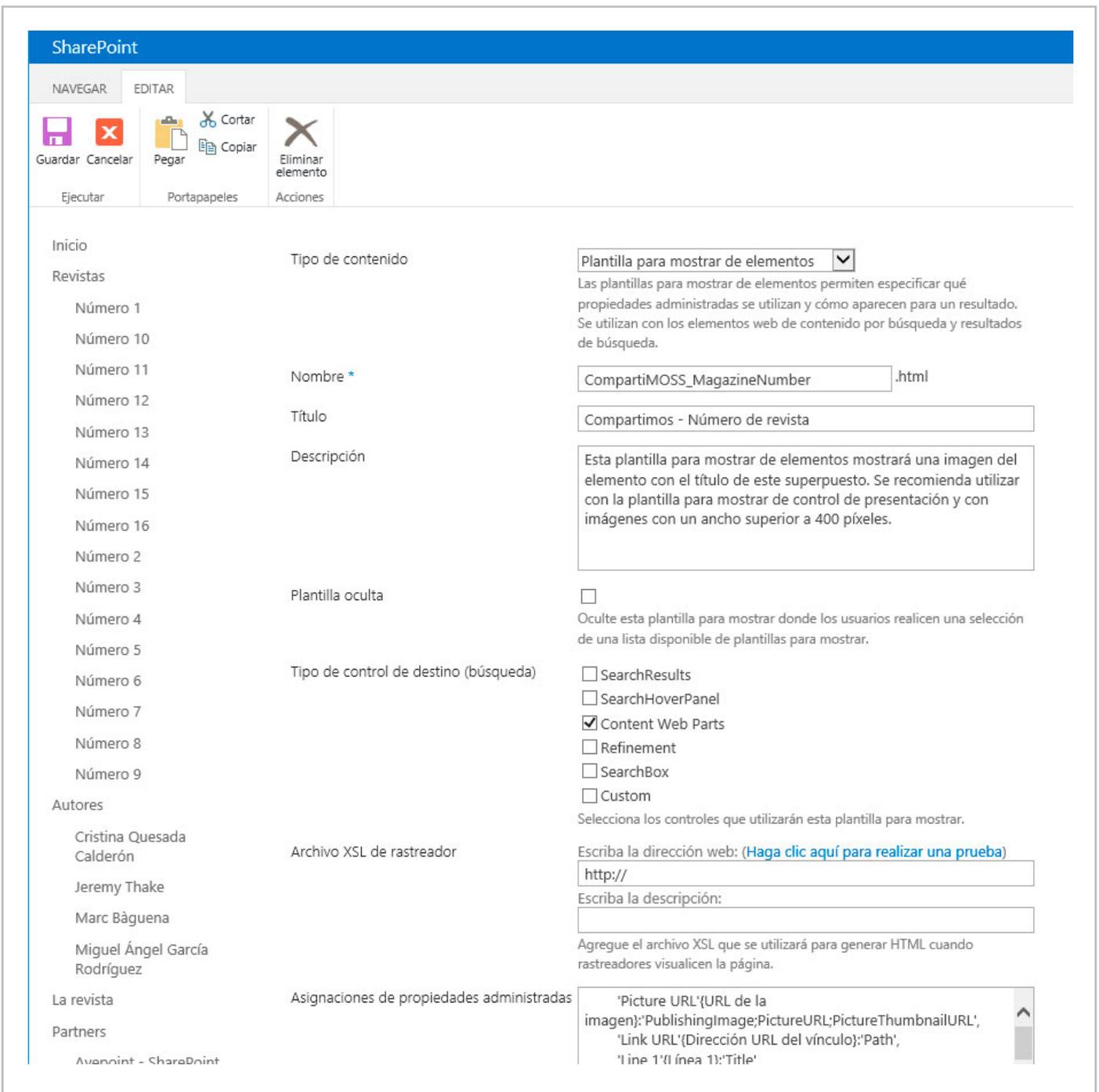


Imagen 4.- Edición de propiedades de una plantilla.

- Para creación y/o edición de código. Con cualquier programa de edición, WebMatrix, Dreamweaver o SharePoint Designer, por ejemplo, debemos acceder a la ruta que nos permite acceder a los archivos de diseño y que se muestra en la Imagen 5.

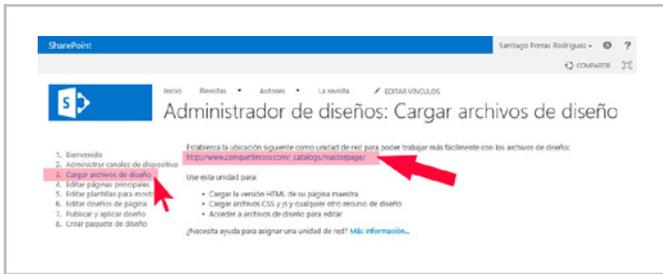


Imagen 5.- Acceder a los archivos de diseño del sitio.

Una vez hayamos accedido a esa ruta con nuestro programa de diseño – para el ejemplo he hecho uso de WebMatrix – podremos ir a la carpeta Display Templates donde encontraremos, separados en diferentes carpetas, los archivos que se corresponden a las plantillas de elementos de contenido, tal y como se puede ver en la Imagen 6.

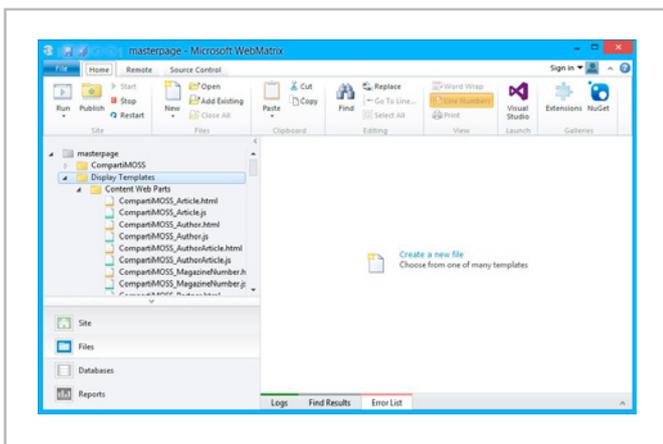


Imagen 6.- Acceder a las plantillas de elementos de contenido desde WebMatrix.

¿De qué se componen las plantillas de elementos de contenido?

Tal y como podremos comprobar desde nuestro programa de edición, para cada plantilla existen dos archivos, uno con la extensión .html y otro con la extensión .js pero, ¿hemos de tocar los dos? La respuesta es no. Veamos para qué es cada archivo.

- Archivo .html. Es el archivo que generamos nosotros para la plantilla y con el que debemos tratar en todo momento. En él iremos codificando y diseñando cómo queremos que se vean los elementos usando HTML y JavaScript mediante una forma de escribir el código (o pseudo-código) que permite a SharePoint realizar sustituciones dentro del mismo identificando una serie de etiquetas que debemos usar.

- Archivo .js. Este archivo lo genera SharePoint automáticamente a partir de nuestro archivo .html

sustituyendo las etiquetas especiales que hayamos usado al crear la plantilla, por su propio código. Esta generación se produce cada vez que realicemos una modificación en el archivo .html y guardemos los cambios, con lo que siempre dispondremos de la última versión. De igual manera, al proteger/desproteger los cambios del archivo .html, SharePoint automáticamente realizará la misma acción sobre este archivo .js.

NOTA: Nunca debemos tocar el archivo .js directamente porque podemos romper la plantilla o, simplemente porque si en algún momento editamos el .html, perderemos los cambios que hubiéramos realizado en el archivo .js.

Crear una nueva plantilla de elementos de contenido

El proceso, es bastante sencillo y sólo tenemos que tener en cuenta una serie de “normas” para su correcto funcionamiento.

La primera opción es cargar un archivo .html que hayamos creado con código de ejemplo para la plantilla y, para ello, accederemos a “Configuración de sitio → Páginas maestras y diseños de página → Display Templates” y accedemos a la carpeta correspondiente, en este caso, “Content Web Parts” tal y como se muestra en la Imagen 7.

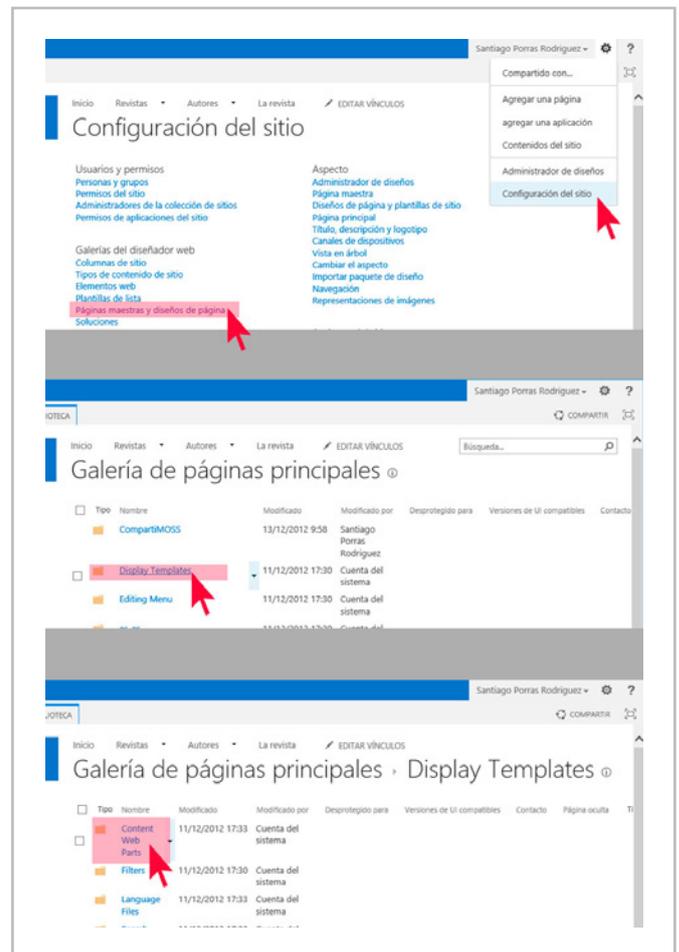


Imagen 7.- Acceder a los archivos de diseño del sitio para crear una nueva plantilla.

Ahora procederemos a crear la plantilla que será del tipo “Plantilla para mostrar de elementos” será de tipo “WebParts de contenido” y que se llama CompartiMOSS_MagazineNumber.html como vemos en la Imagen 8.

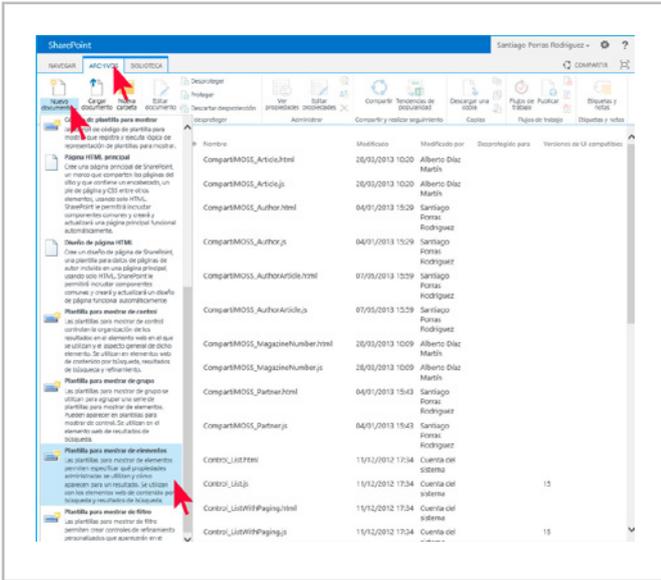


Imagen 8.- Crear una plantilla para elementos de contenido

Esto nos permitirá cargar un archivo .html que hayamos creado (aunque esté vacío) con nuestro editor de HTML tal y como se puede ver en la siguiente imagen

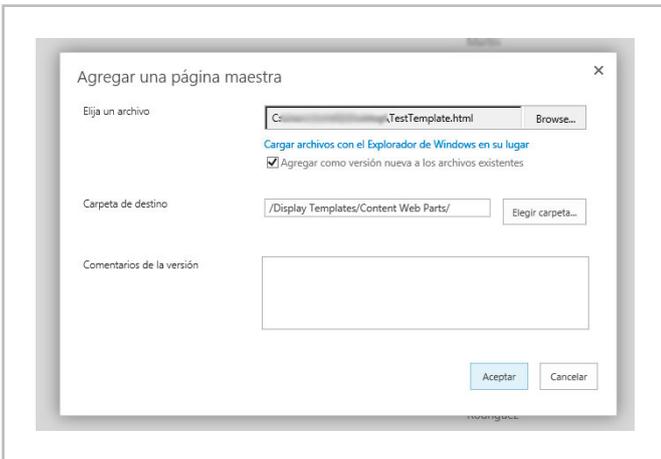


Imagen 9.- Cargar un archivo con la plantilla.

Una vez cargado el archivo, aparecerá la ventana de propiedades en la que tendremos que establecer, como mínimo, el tipo de contenido (si no lo establecimos en el paso de la Imagen 9), el nombre de la plantilla, su título, el tipo de control con el valor “Content WebParts” y marcar que tiene un archivo asociado, tal y como se muestra en la Imagen 10.

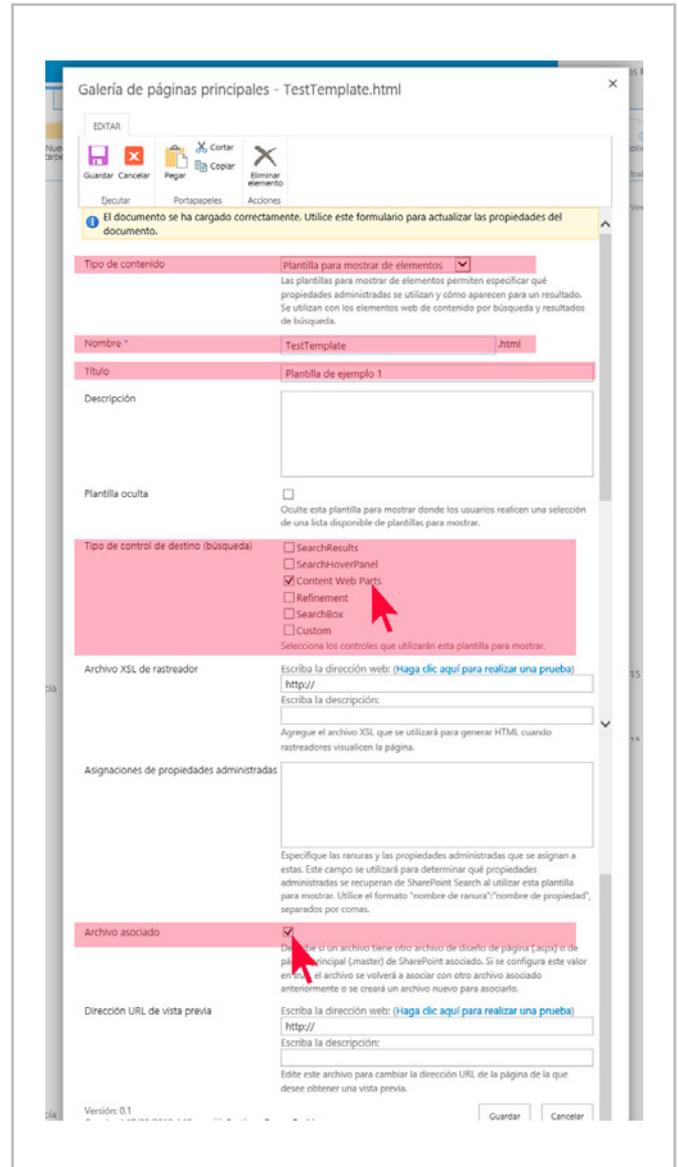


Imagen 10.- Editar las propiedades de la plantilla.

En el momento en que guardemos esta plantilla, podremos ver cómo de forma automática, SharePoint 2013 creará un archivo .js asociado con el mismo nombre que le hayamos puesto a la plantilla, que contendrá el código que realmente se usará para mostrar la plantilla y que es resultado de la sustitución de elementos que realiza el sistema.

NOTA: Para esta tarea, podemos aprovecharnos de la gran herramienta usada por los developers, Copy&Paste, y usarla para copiar desde nuestro editor de código o incluso desde el explorador de archivos, alguna plantilla (archivo .html) ya generada y cambiarle el nombre pero prestando mucha atención al siguiente punto.

Cómo editar las plantillas de elementos de contenido

Hemos llegado al punto crítico ya que entraremos en materia de código, así que para ilustrarlo mejor, veamos el uso para generar la plantilla de los elementos mostrados en la Imagen 1.

...los nuevos WebParts que entran en juego con SharePoint 2013 hacen uso de estas nuevas plantillas...



```

<htmlxmlns:mso="urn:schemas-microsoft-com:office:office"xmlns:msdt="uuid:C2F41010-65B3-11d1-A29F-00AA00C14882">
<head>
<title>Compartimos - Número de revista</title>

<!--[if gtemso 9]><xml>
<mso:CustomDocumentProperties>
<mso:TemplateHiddenmsdt:dt="string">0</mso:TemplateHidden>
<mso:ManagedPropertyMappingsdt:dt="string">
&#39;Picture URL&#39;{URL de la imagen};&#39;PublishingImage;PictureURL;PictureThumbnailURL&#39;;
&#39;Link URL&#39;{Dirección URL del vínculo};&#39;Path&#39;;
&#39;Line 1&#39;{Línea 1};&#39;Title&#39;;
&#39;Line 2&#39;{Línea 2};&#39;PublishingPageContentOWSHTML&#39;;
&#39;SecondaryFileExtension&#39;;
&#39;ContentTypeId&#39;;
</mso:ManagedPropertyMapping><mso:MasterPageDescriptionmsdt:dt="string">Esta plantilla para mostrar de elementos mostrará una imagen del elemento con el título de este superpuesto. Se recomienda utilizar con la plantilla para mostrar de control de presentación y con imágenes con un ancho superior a 400 píxeles.</mso:MasterPageDescription><mso:ContentTypeId msdt:dt="string">0x0101002039C03B61C64EC4A04F5361F385106603</mso:ContentTypeId><mso:TargetControlType msdt:dt="string">#Content Web Parts;#</mso:TargetControlType><mso:HtmlDesignAssociated msdt:dt="string">1</mso:HtmlDesignAssociated><mso:HtmlDesignConversionSucceeded msdt:dt="string">True</mso:HtmlDesignConversionSucceeded><mso:HtmlDesignStatusAndPreview msdt:dt="string">http://www.compartimoss.com/_catalogs/masterpage/Display%20Templates/Content%20Web%20Parts/CompartimOSS_MagazineNumber.html, Conversión correcta.</mso:HtmlDesignStatusAndPreview></mso:CustomDocumentProperties></xml>![endif]-->
</head>

<body>

<!--
Warning: Do not try to add HTML to this section. Only the contents of the first <div> inside the <body> tag will be used while executing Display Template code. Any HTML that you add to this section will NOT become part of your Display Template.
-->
<script>

$includeLanguageScript(this.url,"~sitecollection/_catalogs/masterpage/Display Templates/Language Files/{Locale}/CustomStrings.js");
</script>

<!--
Use the div below to author your Display Template. Here are some things to keep in mind:
* Surround any JavaScript logic as shown below using a "pound underscore" (#_ ... _) token inside a comment.

* Use the values assigned to your variables using an "underscore pound equals" (_#= ... =#_) token.
-->
<divid="Item_LargePicture">

<!--#_
varencodedId =
$htmlEncode(ctx.ClientControl.get_nextUniqueId() + "_largePicture_");

varlinkURL = $getItemValue(ctx, "Link URL");
linkURL.overrideValueRenderer($urlHtmlEncode);

var line1 = $getItemValue(ctx, "Line 1");
//Override site name on Title
var title = line1.value.replace("CompartimOSS\r\n\r\n", "\r\n\r\n", "");

```

```

var line2 = $getItemValue(ctx, "Line 2");

varpictureURL = $getItemValue(ctx, "Picture URL");
varpictureId = encodedId + "picture";
varpictureMarkup =
Srch.ContentBySearch.getPictureMarkup(pictureURL, 228, 300, ctx.CurrentItem, "cbs-largePictureImg", line1, pictureId);

varcontainerId = encodedId + "container";
varpictureLinkId = encodedId + "pictureLink";
varpictureContainerId = encodedId + "pictureContainer";
vardataContainerId = encodedId + "dataContainer";
vardataContainerOverlayId = encodedId + "dataContainerOverlay";
var line1LinkId = encodedId + "line1Link";
var line1Id = encodedId + "line1";
var line2Id = encodedId + "line2";

AddPostRenderCallback(ctx, function()
{
$("#"+containerId).hover(
function(){
$('.divOverlay', this).css('top', '0');
},
function(){
$('.divOverlay', this).css('top', '300px');
});
});
_#-->

<divclass="divCompartiMOSSMagazineNumber" id=" _#= containerId =#_">
<divclass="divPicture" id=" _#= pictureContainerId =#_">
<a href=" _#= linkURL =#_" title=" _#= $htmlEncode(title) =#_" id=" _#= pictureLinkId =#_" _#= pictureMarkup =#_">
</a>
</div>
<divclass="divOverlay" id=" _#= dataContainerId =#_">
<aclass="linkOverlay" href=" _#= linkURL =#_" title=" _#= $htmlEncode(title) =#_" id=" _#= line1LinkId =#_">
<h2class="ms-nowrap" id=" _#= line1Id =#_" _#= title =#_</h2>
<divtitle=" _#= $htmlEncode(line2) =#_" id=" _#= line2Id =#_" _#= line2 =#_</div>
</a>
</div>
</div>

</div>
</body>
</html>

```

Analizamos el código

- La cabecera contiene un bloque con las propiedades que mapea la plantilla correspondiéndose con el tipo de contenido seleccionado en las propiedades.

- Tal y como se indica en el primer comentario dentro del <body>, sólo se mostrará el contenido que se encuentre dentro del primer <div> dentro del <body>.

- Para obtener los valores de las propiedades mapeadas tendremos que hacer uso de la sentencia de javascript que sigue:

```
var line1 = $getItemValue(ctx, "Line 1");
```

- Para usar los valores de las variables de JavaScript dentro del HTML que escribamos, debemos hacer uso de las siguientes marcas que SharePoint sustituirá por su valor correspondiente.

```
_#= Nombre_variable_JavaScript =#_
```

Con estas normas, podremos empezar a diseñar nuestras plantillas de elementos de contenido y así poder enriquecer nuestros sitios de SharePoint pero, aún nos queda asignársela a algún WebPart, así que, en el siguiente punto veremos cómo usarlas en un WebPart de Búsqueda de Contenido.

Usar las plantillas de elementos de contenido

Para usar esta nueva plantilla, tan sólo tendremos que ir al WebPart deseado, en este caso un WebPart de Búsqueda de Contenido, editarlo mediante el menú contextual y seleccionando “Editar elemento web” tal y como se muestra en la siguiente imagen.

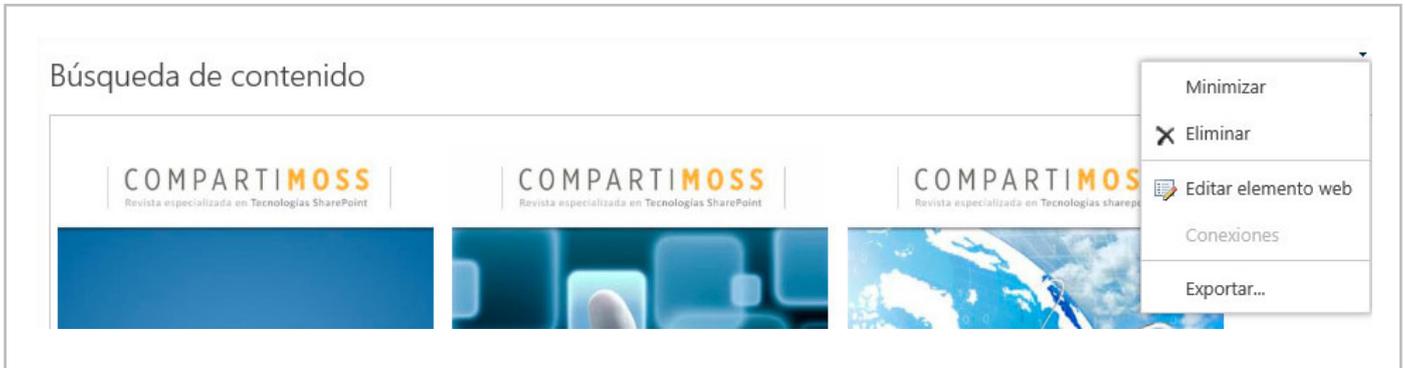


Imagen 12.- Editar Web Part de Búsqueda de Contenido.

Cuando aparezcan las propiedades de edición del Web Part, en el desplegable “Elemento”, podremos seleccionar nuestras nuevas plantillas, en este caso, CompartiMOSS – Número de revista tal y como se puede ver en la Imagen 13.

Aceptamos, guardamos la página y... ¡todo hecho! Acabamos de crear, editar y usar nuestra primera plantilla de elementos de contenido. Espero que os haya servido para descartar el uso de las antiguas plantillas XSLT y comenzar a aprovechar las bondades del uso de HTML + JavaScript para estas tareas.

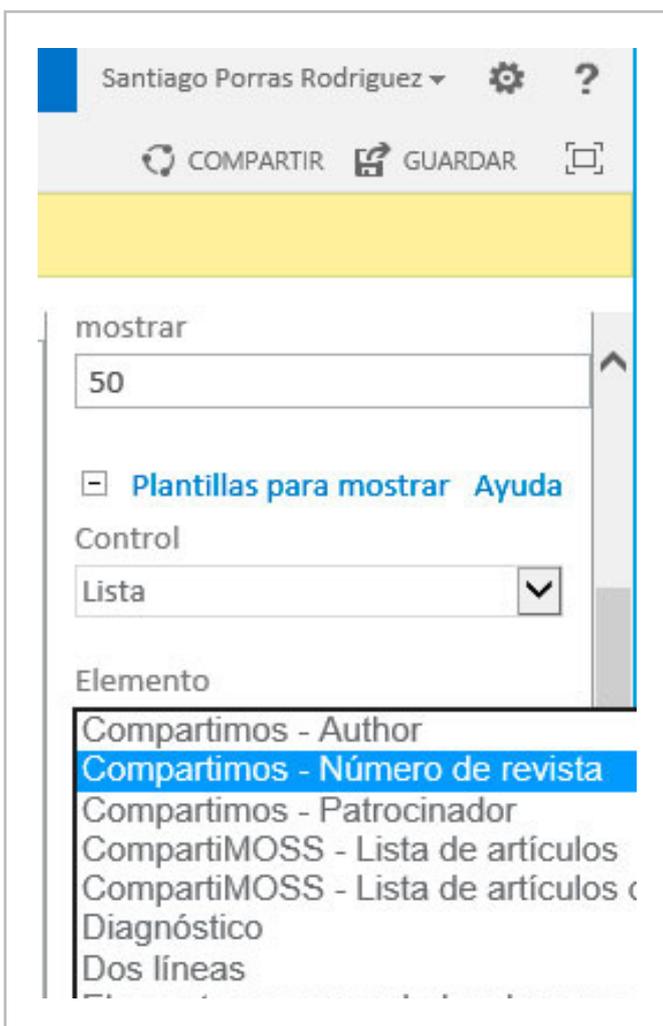


Imagen 13.- Selección de plantilla.



Imagen 14.- Resultado final.

SANTIAGO J. PORRAS RODRÍGUEZ
 Nokia Developer Champion
 UX Developer
<http://geeks.ms/blogs/santyptr>
[@saintwukong](https://twitter.com/saintwukong)





16

Records Management en SharePoint 2013

Resumen

Una de las características que nos ofrece SharePoint es la capacidad de implementar soluciones ECM (Enterprise Content Management), y no sólo desde un punto de vista de gestión documental natural de control de versiones, tipos de contenido y plantilla, sino también desde un punto de vista de gestión de las políticas de negocio o legales que obligan a las empresas a mantener seguros sus documentos. En este artículo nos adentraremos en la Gestión de Registros que SharePoint nos ofrece como parte del ciclo de vida de un ECM.

Artículo

Un Record o Registro es una entidad física o digital, con importancia organizacional y que requiere de unas políticas de retención. Por motivos legales o de negocio, las empresas necesitan aplicar estas políticas de retención, bloquear registros para que puedan ser editados, disponer de ellos cuando sea necesario, manteniendo unas políticas de administración y gestión que permita mantener la información de la empresa cumpliendo con esas leyes o con el propio negocio.

¿Qué es Records Management?

Records Management o Gestión de Registros (no tengo claro que la traducción exista en Español) es la práctica de identificar, clasificar, archivar, preservar y destruir los registros de acuerdo a una serie de reglas o estándares predefinidos.

Bajo la definición de un Plan de Archivo, debemos de asegurar la información afectada y cumplir con estas políticas definidas, y, para los registros digitales, SharePoint tiene una solución. Entre todas las funcionalidades de ECM de SharePoint, éstas son las que se aplican a la gestión de registros:

- **Identificador de Documentos.** Todo registro debe de tener un identificador único en la empresa, y para eso desde SharePoint 2010 tenemos el servicio de Id de Documentos.
- **Metadatos empresariales.** El servicio de metadatos mejora el uso de la terminología en toda la empresa y

permite clasificar el contenido a nivel empresarial. Este servicio permite tener los conjuntos de términos para todos nuestros sitios de SharePoint, mejorando la búsqueda y el descubrimiento de los contenidos. Desde el punto de vista de un registro, nos permite tener metadatos consistentes en toda la organización y mantener reglas que categoricen o almacenen los registros.

- **Records center.** El Centro registros es un sitio que actúa como repositorio central donde almacenar y gestionar los registros. SharePoint incluye una plantilla de sitios que permite gestionar los registros de la empresa, con todas las funcionalidades necesarias.

- **In-Place Records Management.** Junto con las funcionalidades de Records Management, desde SharePoint 2010 tenemos la posibilidad de llevar esta gestión desde el sitio, en modo In-Place. En vez de mover los documentos al Centro de Registros para declararlo como registro, se puede declarar y administrar el registro desde el sitio donde fue creado. Las políticas son aplicadas a nivel de tipos de contenido o directamente en las bibliotecas de documentos, y los documentos pueden ser declarados como registros, manual o automáticamente.

- **Auditoría de Colección de Sitios.** La gestión de registros necesita de una intensiva auditoría para conocer lo que está pasando, por ejemplo, cuando declaramos un registro y por qué, cuando fue eliminado o cambiado, etc. Para eso, utilizaremos la configuración de auditoría que SharePoint ofrece a nivel de colección de sitios.

- **Organizador de contenido.** El organizador de contenido de SharePoint 2010 ayuda a las empresas a clasificar el contenido, moviéndolo a su lugar correspondiente, basándose en los metadatos del documento.

- **Detalles de cumplimiento.** SharePoint 2010 incluye una opción en las bibliotecas que permiten gestionar la información de negocio de los documentos, por ejemplo, políticas de retención, documentos retenidos, documentos declarados como registro, etc.

- **Retención y eDiscovery.** En SharePoint 2010 se incluye esta característica que permite encontrar los documentos que cumplen con una determinada búsqueda, tipo de contenido o política, y ponerlos en retención. Esta retención suele ser necesaria, en la gestión de registros, para auditorías o investigación de negocios. En la última versión, SharePoint 2013 permite también encontrar documentos de Exchange y Lync.

- **Content Type Hub.** El concentrador de tipos de contenido fue una característica nueva de SharePoint 2010, solicitada por los usuarios, que permite tener un sitio centralizado donde administrar los tipos de contenido y publicarlos a las distintas colección de sitios suscritas, ya que la definición de los tipos de contenido es únicamente a nivel de colección de sitios. Para la gestión de registros es importante tener declarados globalmente los tipos de documentos (tipos de contenidos) y poder saltarse la limitación del producto de la definición a sólo una colección de sitios, porque, probablemente, tengamos más de una colección de sitios y, más probable, el centro de registros esté en una aplicación web independiente.

- **Multi-Level Retention.** SharePoint 2010 incluye el concepto de poder configurar diferentes fases de retención, para permitir la posibilidad de definir diferentes estados del registro. Lo normal es que los registros pasen por un periodo de retención y que pasado ese periodo se establezca otro periodo de salvaguarda con la destrucción del registro, y con ello, el documento.

- **Send to connections.** Las capacidades del menú de "Enviar a" en SharePoint 2010 se ampliaron, con respecto a SharePoint 2007. En SharePoint 2010 no sólo se puede hacer una copia del documento, sino que es posible moverlo o, incluso, moverlo y dejar un link que enlace con el destino final del documento. Esta característica es necesaria para mover documento a Centro de registros y que estos sean enrutados a su lugar correspondiente, una vez que lleguen a él.

- **Sitios de buzones.** SharePoint 2013 permite integrarnos con Exchange 2013 con una nueva funcionalidad llamada Site Mailbox. Pensemos que un registro también incluye a los correos electrónicos, y con esta funcionalidad, podemos emplear las mismas políticas y el centro de registro para estos documentos.

- **Políticas de sitio.** Una extensión que permite aplicar políticas a nivel de sitio, incluyendo cumplimientos.

Teniendo en cuenta las funcionalidades que SharePoint 2013 nos ofrece, podemos planificar la gestión de registros con las siguientes fases:

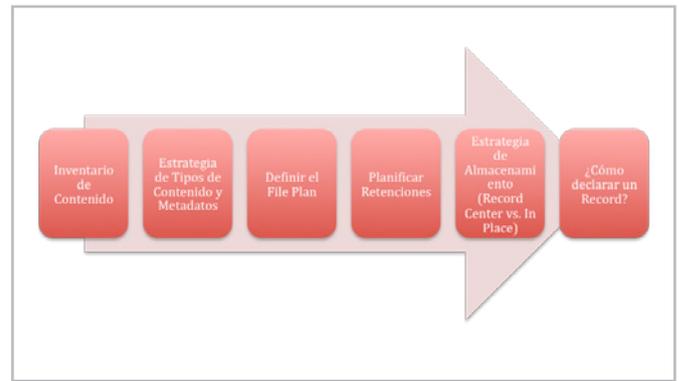


Imagen 1.- Fases de la planificación de registros.

Hacer un inventario del contenido electrónico de la empresa, definir una estrategia, alineada con el negocio, de tipos de contenido de Metadatos empresariales, definir un plan de archivo para esos tipos de contenido, planificando las retenciones necesarias en cada caso y las distintas fases por las que un registro debe de pasar, todo esto teniendo claro cómo vamos a almacenar toda esta información en SharePoint, en las bases de datos de contenido o en Blob Storage, y documentar como y cuando se debe de declarar un registro.

Centro de Registros

El Centro de Registros es una plantilla de sitio en SharePoint que combina características estándar de gestión de proyectos empresariales con funcionalidades especializadas de gestión de registros. Por definición, los registros no serán nunca modificados por el Sistema y los administradores del sistema pueden configurarlo para prevenir modificaciones.

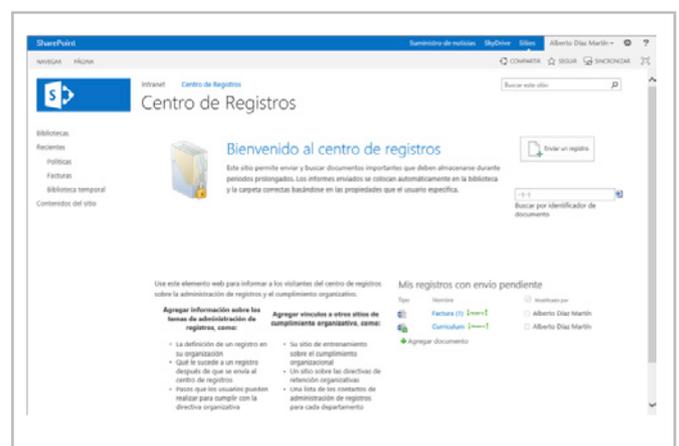


Imagen 2.- Centro de registros de SharePoint 2013.

Mediante la definición de reglas del organizador de contenido, clasificamos los documentos que llegan al Centro de Registro. Esta clasificación se basa en el tipo de contenido del documento y los metadatos de los mismos, para enrutar el contenido.



Tipo de contenido de envío *
 Al seleccionar un tipo de contenido, se determinan las propiedades que se pueden usar en las condiciones de esta regla. Asimismo, los envíos que cumplan esta regla recibirán el tipo de contenido seleccionado aquí cuando se coloquen en una ubicación de destino.

Tipo de contenido:
 Grupo:
 Tipo:

Nombres alternativos:
 Este tipo de contenido tiene nombres alternativos en otros sitios:

Agregar nombre alternativo:

Nota: al agregar el tipo "", los documentos de tipo de contenido desconocido se clasifican según esta regla.

Lista de nombres alternativos:

Condiciones
 Para que cumpla esta regla, las propiedades de un envío deben cumplir todas las condiciones de propiedades especificadas (por ejemplo, "Si Fecha de creación es anterior a 1/1/2000").

Condiciones basadas en propiedades:
 Propiedad:
 Operador:
 Valor:

[\(Agregar otra condición\)](#)

Ubicación de destino *
 Especifique dónde desea colocar el contenido que coincida con esta regla.

Al enviar a otro sitio, los sitios disponibles se toman de la lista de otros sitios con organizadores de contenido, de acuerdo a cómo lo haya definido el administrador del sistema.

Marque la casilla "Crear automáticamente una carpeta

Destino:

Ejemplo: /sitios/CentroDeDocumentación/Documentos/
 Crear automáticamente una carpeta para cada valor único de una propiedad:
 Seleccionar una propiedad (debe ser una propiedad requerida y de valor único):
 Especificar el formato para el nombre de carpeta:

Imagen 3.- Clasificación de contenido.

Declaración de Registros

Para declarar un registro existen cuatro opciones: realizarlo usando las fases de la directiva de información del tipo de contenido, mediante un flujo de trabajo, que se cree automáticamente para todos los documentos de una biblioteca o que el usuario lo pueda realizar de forma manual, esto es, una vez que el documento se encuentra en el Centro de Registro, o en principio, en cualquier ubicación, existe una opción para realizar la declaración del documento como registro. Esto se tiene que configurar en la configuración de la colección de sitio, en la página de Configuración de declaración como registro o a nivel de biblioteca de documentos

Configuración de declaración como registro

Restricciones de registro
 Permite especificar restricciones para aplicar a un documento o elemento cuando se le haya declarado como registro. El cambio de esta configuración no afectará a los elementos que ya hayan sido declarados como registros. Nota: la configuración de la directiva de administración de información también puede especificar diferentes directivas para los registros y para los que no lo son.

Sin restricciones adicionales
 Los registros no tienen más restricciones que los elementos que no son registros.
 Bloquear eliminación
 Los registros pueden editarse, pero no eliminarse.
 Bloquear edición y eliminación
 Los registros no pueden editarse ni eliminarse. Todo cambio requerirá que la declaración como registro se revoque.

Disponibilidad de declaración como registro
 Permite especificar si todas las bibliotecas y listas de este sitio deben hacer que la declaración manual como registro esté disponible de manera predeterminada. Cuando la declaración manual como registro no esté disponible, los registros solo pueden declararse como tales mediante una directiva o un flujo de trabajo.

La declaración manual como registro en listas y bibliotecas debe:
 Estar disponible en todas las ubicaciones de manera predeterminada
 No estar disponible en todas las ubicaciones de manera predeterminada

Roles de declaración
 Permite especificar qué roles de usuario pueden declarar o revocar la declaración como registro.

La declaración como registro puede ser realizada por:
 Todos los administradores y colaboradores de la lista
 Solo los administradores de la lista
 Solo acciones de directiva

La revocación de declaración como registro solo puede ser realizada por:
 Todos los administradores y colaboradores de la lista
 Solo los administradores de la lista
 Solo acciones de directiva

Imagen 4.- Declaración de un registro.

La declaración de un registro implica una configuración de restricciones de registro, que bloquean o no el documento para eliminación o edición, y la configuración de los roles que pueden declarar o revocar la declaración de un registro.

Plan de Archivo

Dentro de la gestión de registros se define el File Plan o Plan de Archivo. Este plan define las directivas de retención, teniendo en cuenta las distintas fases por las que un registro pasa y los niveles de auditorías necesarios para cada uno de los tipos de archivo o contenido. SharePoint permite definir estas políticas a nivel de sitio, como plantillas, a nivel de tipo de contenido o a nivel de biblioteca de documentos, pudiendo sobrescribir cualquier política aplicada desde un nivel superior.

...El Centro de Registros es una plantilla de sitio en SharePoint que combina características estándar de gestión de proyectos...



Retención
Para programar la administración y disposición del contenido, especifique una secuencia de fases de retención. Si especifica varias fases, tendrán lugar sucesivamente en el orden en que aparecen en esta página.

Nota: Si la característica Retención basada en biblioteca y carpeta está activa, los administradores de la lista pueden invalidar las directivas de tipos de contenido con sus propias programaciones de retención. Para impedirlo, desactive la característica en la colección de sitios.

Habilitar Retención

Elementos que no son registros
Especifique cómo administrar la retención en los elementos que no han sido declarados como registros:

Evento	Acción	Periodicidad
Creado + 1 días	Enviar a la ubicación Centro de Registro	No
Creado + 15 días	Declarar como registro	No
Declarado como registro + 5 años	Eliminar todas las versiones anteriores	No

[Agregar una fase de retención...](#)

Nota: puede especificar una directiva diferente que se aplique una vez que un elemento haya sido declarado como registro.

Registros
Especifique cómo administrar la retención en los registros:

Usar la misma directiva de retención que para los elementos que no son registros
 Defina las diferentes fases de retención para los registros:

Imagen 5.- Configuración del Plan de Archivo.

La configuración de retención anterior, define las siguientes fases:

- Creado + 1 días. La primera fase se activa 1 días después de la fecha de creación del documento y lo transfiere al Centro de Registro, moviéndolo y dejando un vínculo de enlace al mismo.

Evento

Permite especificar qué causa que la fase se active:

Período de tiempo basado en las propiedades del elemento:
Período de tiempo: Creado + 1 días

Establecer mediante una fórmula de retención personalizada instalada en este servidor:

Acción

Cuando se desencadene esta fase, realice la siguiente acción:
Transferir a otra ubicación

Tipo de transferencia:
 Copiar
 Mover
 Mover y dejar un vínculo

Ubicación de destino: Centro de Registro

Razón para la transferencia: Transferido debido a una directiva organizativa

Periodicidad

Esta fase se ejecutará una vez de acuerdo con el evento definido arriba. Use la periodicidad para forzar que la fase repita la acción.

Repetir la acción de esta fase hasta que la siguiente fase esté activa
Después de que la fase se desencadene por primera vez, la acción de la fase se repetirá indefinidamente hasta que se desencadene la siguiente fase.
Periodicidad: años

Aceptar Cancelar

Imagen 6.- Configuración primera fase

- Creado + 15 días. Una vez han pasado 15 días de la creación del documento, este es declarado como registro de forma automática.

...Cuando el documento es declarado como registro, manual o automáticamente, se eliminan todas las versiones anteriores...

Evento

Permite especificar qué causa que la fase se active:

Período de tiempo basado en las propiedades del elemento:
Período de tiempo: Creado + 15 días

Establecer mediante una fórmula de retención personalizada instalada en este servidor:

Acción

Cuando se desencadene esta fase, realice la siguiente acción:
Declarar como registro

Esta acción declarará al elemento como registro. Si hay una directiva de retención diferente definida para los registros, el elemento comenzará a ejecutar la directiva del registro.

Periodicidad

Esta fase se ejecutará una vez de acuerdo con el evento definido arriba. Use la periodicidad para forzar que la fase repita la acción.

Repetir la acción de esta fase hasta que la siguiente fase esté activa
Después de que la fase se desencadene por primera vez, la acción de la fase se repetirá indefinidamente hasta que se desencadene la siguiente fase.
Periodicidad: años

Aceptar Cancelar

Imagen 6.- Configuración declaración de registro transcurridos 15 días

- Declarado como registro + 5 años. Cuando el documento es declarado como registro, manual o automáticamente, se eliminan todas las versiones anteriores.

Evento

Permite especificar qué causa que la fase se active:

Período de tiempo basado en las propiedades del elemento:
Período de tiempo: Declarado como registro + 5 años

Establecer mediante una fórmula de retención personalizada instalada en este servidor:

Acción

Cuando se desencadene esta fase, realice la siguiente acción:
Eliminar todas las versiones anteriores

Esta acción eliminará todas las versiones anteriores de este documento.

Periodicidad

Esta fase se ejecutará una vez de acuerdo con el evento definido arriba. Use la periodicidad para forzar que la fase repita la acción.

Repetir la acción de esta fase hasta que la siguiente fase esté activa
Después de que la fase se desencadene por primera vez, la acción de la fase se repetirá indefinidamente hasta que se desencadene la siguiente fase.
Periodicidad: años

Aceptar Cancelar

Imagen 7.- Configuración declaración de registro transcurridos 5 años.

Por definición, los periodos puedes ser al Crear un documento, al Modificarlo o al Declararlo como registro, mientras que las acciones son:

- Mover a la papelera de reciclaje.
- Eliminar permanentemente.
- Transferir a otra ubicación.
- Iniciar un flujo de trabajo.
- Saltar a la siguiente fase.
- Declarar como registro.
- Eliminar borradores anteriores.
- Eliminar todas las versiones anteriores.

Además de poder habilitar los eventos que se van a auditar para este tipo de contenido:

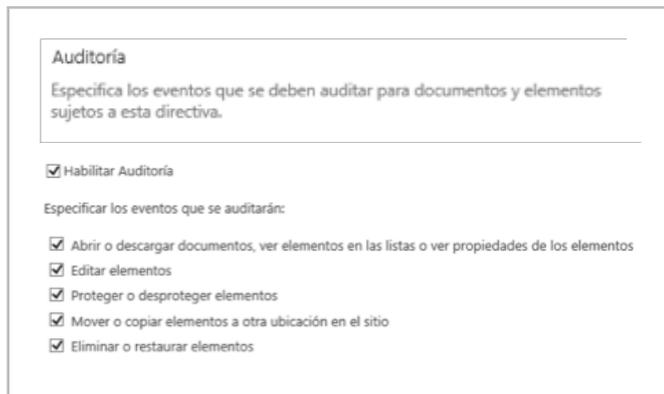


Imagen 7.- Opciones de configuración de auditoría.

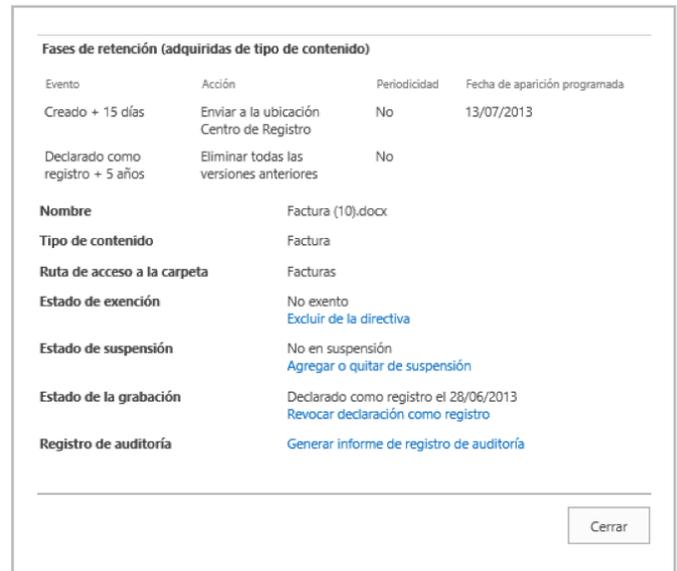


Imagen 8.- Informe de cumplimiento.

Este informe nos ofrece toda la información, del elemento actual, relativo a las fases y a las políticas que tiene que cumplir. Además, nos ofrece el estado de registro, el estado de suspensión y de exención, junto con un informe de registro de auditoría, si esta estuviera activada.

Desde la Administración del Centro de Registros podemos acceder a toda una serie de informes sobre actividades de contenido, directiva de administración de información y seguridad y configuración del sitio.



Imagen 9.- Informes de auditoría.

En cualquier momento, tenemos que poder extraer información sobre los registros y sus documentos asociados, por lo tanto, es importante que configuremos correctamente la auditoría de los contenidos.

Suspensión

Por motivos del negocio, cuando un registro está bajo investigación o auditoría, es posible ponerlo en estado de suspensión. Cuando un registro se encuentra en suspensión, está suspendido para cualquier caducidad de alguna política que se le aplique, y está protegido ante la destrucción. De nuevo en la Administración del Centro de Registros, podemos buscar estos documentos y suspenderlo, además de poder crear motivos de suspensión, necesarios para suspender un documento.

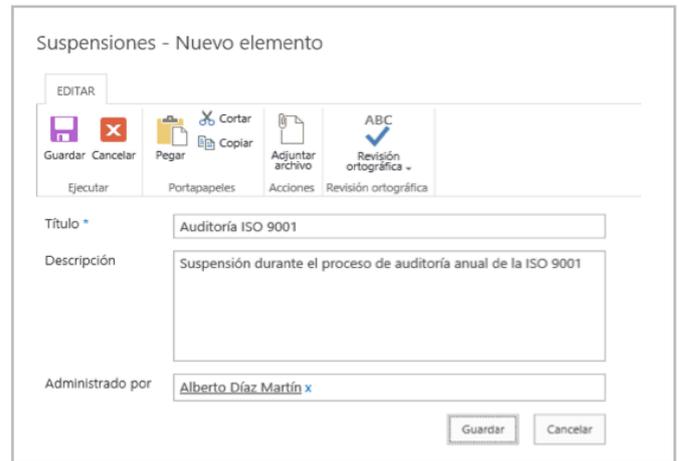


Imagen 10.- Nuevo elemento de Suspensión.

Que pueden ser aplicadas a nivel del centro de registro o para un registro en concreto.

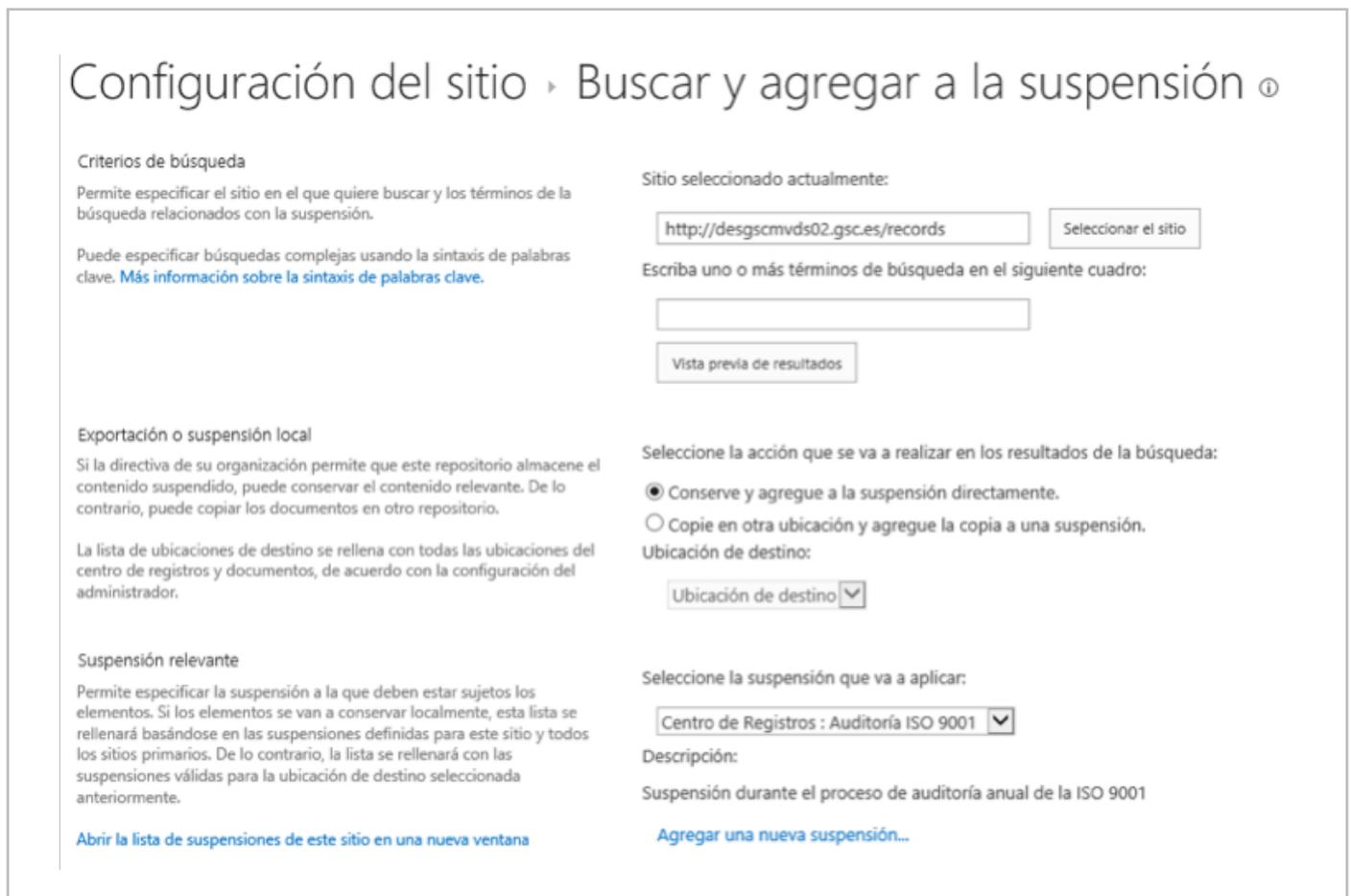


Imagen 11.- Aplicación de Suspensión.

Conclusiones

Todas estas capacidades ponen a SharePoint 2013 en un lugar privilegiado dentro de los gestores de registros, aunque os recomiendo planificar cuidadosamente todo el proceso, porque son muchos componentes por separado que poner a jugar para el mismo rol. Por su puesto, tenemos ciertas limitaciones que SharePoint no cubre, como la gestión de registros físicos, la seguridad por metadatos, el reemplazo de políticas o elementos obsoletos, que se suelen necesitar en este tipo de gestiones, pero para esto, tenemos soluciones de terceros que amplían y mejoran las capacidades de SharePoint, por ejemplo, Collabware CLM.

ALBERTO DIAZ MARTIN

MVP SharePoint

adiazcan@Hotmail.com

@adiazcan

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





conferencia

workshop desarrollo



workshop IT



workshop negocio



Madrid → 10th October

www.iberiansharepointconference.com

Usar Framework JavaScript en nuestros desarrollos SharePoint Parte II

Resumen

En el artículo de este número vamos a continuar introduciendo Frameworks de JavaScript. En esta ocasión es el turno de ExtJS de **Sencha**. ExtJS es uno de los Frameworks más populares y más utilizados para el desarrollo web. Proporciona un rendimiento muy rápido y veloz en las aplicaciones que lo utilizan debido a que el lenguaje generado es perfectamente interpretado por los navegadores modernos. Otra de las ventajas que tiene ExtJS es que tiene un IDE propio (Sencha Architect), que aunque no se trata de Visual Studio, proporciona una interfaz de desarrollo relativamente simple y ahorra bastante trabajo.

Artículo

ExtJS es uno de los Frameworks JavaScript más famosos del mercado cuya última versión es la 4.2. Se caracteriza por simplificar la creación de Apps.

Sus principales ventajas:

- Prototipos y APPS de una forma muy sencilla y rápida.
- Resultado obtenido es HTML5 con lo cual obtienes una compatibilidad con todos los navegadores y dispositivos.
- Posibilidad de crear aplicaciones móviles con un código muy semejante (ya lo veremos en otros artículos)
- Arquitectura MVC

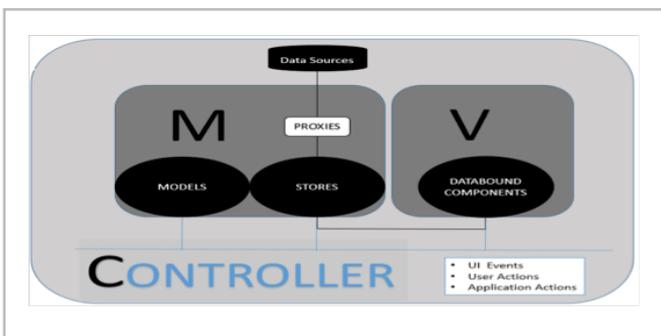


Imagen 1.- Arquitectura Proyecto ExtJs.

El principal inconveniente que tiene es que su curva de aprendizaje no es tan rápida, y para hacer cosas relativamente interesantes con él, es necesario conocer bastante bien JavaScript, lo que en los primeros proyectos nos puede hacer tirar la toalla y ocasionar un poco de frustración.

Al igual que en el anterior número, en este artículo vamos a explicar cómo desarrollar un APP en la que vamos a mostrar información de los Autores de CompartiMOSS. Vamos a poder realizar operaciones Create Read Update Delete contra una lista de SharePoint y visualizaremos la información obtenida en un GRID haciendo uso de la API REST para leer la información y también utilizaremos el modelo de objetos cliente de JavaScript para ver cómo podemos integrar otras librerías con este Framework.

Requisitos Previos

Antes de ponernos a trabajar hay que descargarse el Framework Sencha de la siguiente página: <http://www.sencha.com/products/extjs/download/>. Su uso, siempre que no se implemente ninguna aplicación “comercial”, es gratuito.

Creación de la APP SharePoint-Hosted mediante Visual Studio

Abrimos Visual Studio 2012 y seleccionamos un proyecto de este tipo:

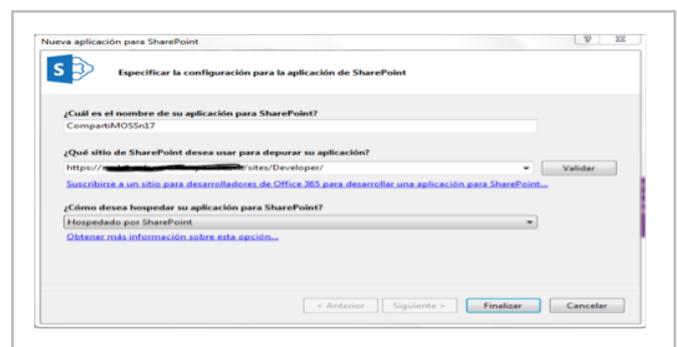


Imagen 2.- Creación de la Aplicación SharePoint-Hosted.

En primer lugar, agregamos al proyecto una lista llamada Autores en la que añadiremos los datos que queremos mostrar en el Grid. Esta lista contendrá los campos mencionados anteriormente: (Autor, PuestoTrabajo, Twitter y Blog). A esta lista le añadiremos datos de ejemplo para comprobar el correcto funcionamiento, para ello basta con editar el archivo “elements.xml” que se encuentra ubicado dentro de la Instancia de la lista y añadir los datos que creamos conveniente.

El siguiente paso es añadir los ficheros JavaScript y CSS que vienen en el Framework de Sencha en nuestro proyecto. Los JavaScript los ubicaremos dentro de la carpeta de Scripts y los CSS dentro de la carpeta de Content. Estos ficheros son los siguientes: ext-all.js, include-ext.js, options-toolbar.js, ext-theme-classic-all.css y example.css. En la página default.aspx añadimos referencias a las librerías JavaScript y archivos .CSS de la siguiente forma:

```
<script type="text/javascript"
src="../../Scripts/App.js"></script>
  <script
type="text/javascript"src="../../Scripts/ext-
all.js"></script>
  <script type="text/javascript"
src="../../Scripts/include-ext.js"></script>
  <script type="text/javascript"
src="../../Scripts/options-
toolbar.js"></script>

  <link rel="stylesheet" type="text/css"
href="../../Content/App.css" />
  <link rel="stylesheet" type="text/css"
href="../../Content/ext-theme-classic-all.css"
/>
  <link rel="stylesheet" type="text/css"
href="../../Content/example.css" />
```

En la misma página default.aspx también vamos a incluir el siguiente div que es el lugar donde vamos a desplegar el desarrollo realizado.

```
<div id="editor-grid"></div>
```

Finalmente creamos un fichero JavaScript en blanco que le vamos a llamar "Row-editing.js" que será donde realizaremos la implementación de la solución. Con esta parte ya nos podemos poner manos a la obra con el desarrollo JavaScript.

Introducción al desarrollo con Sencha ExtJS

Para introducirse en el desarrollo con ExtJS mi recomendación es leer los tutoriales y ejemplos que hay en la Web del producto y una vez entendidos los casos bases, empezar a complicarlos. En los mismos ejemplos obtendremos y seremos conscientes de toda la potencia que tiene ExtJS. El consultar la ayuda nos indicará todos los eventos y valores que tienen los diversos artefactos que vamos a utilizar.

En este ejemplo lo que vamos a realizar es añadir un GRID en el que podemos realizar operaciones CRUD contra una lista de SharePoint, en nuestro ejemplo queremos consultar los Autores de la revista CompartiMOSS. Para ello el primer paso es crearnos un modelo en el que vamos a definir que

valores va a tener, en nuestro caso: (Title, PuestoTrabajo, Blog, Twitter) para ello añadimos este código: Dado que aún

```
Ext.define('Autor', {
    extend: 'Ext.data.Model',
    fields: [
        'Title',
        'Trabajo',
        'Blog',
        'Twitter',
    ]
});
```

no vamos a realizar la llamada a la API REST de SharePoint, vamos a introducir una serie de valores aleatorios con la finalidad de comprobar el funcionamiento del grid:

```
// Generate mock employee data

var data = (function() {

    var lasts = ['Jones', 'Smith',
'Lee', 'Wilson', 'Black', 'Williams',
'Lewis', 'Johnson', 'Foot', 'Little', 'Vee',
'Train', 'Hot', 'Mutt'],

        firsts = ['Fred', 'Julie',
'Bill', 'Ted', 'Jack', 'John', 'Mark',
'Mike', 'Chris', 'Bob', 'Travis', 'Kelly',
'Sara'],

        lastLen = lasts.length,
        firstLen = firsts.length,
        usedNames = {},
        data = [],
        s = new Date(2007, 0, 1),
        eDate = Ext.Date,
        now = new Date(),

        getRandomInt =
Ext.Number.randomInt,

        generateName = function() {
```

```

        var name =
firsts[getRandomInt(0, firstLen - 1)] + ' '
+ lasts[getRandomInt(0, lastLen - 1)];

        if (usedNames[name]) {
            return generateName();
        }

        usedNames[name] = true;

        return name;
    };

    while (s.getTime() < now.getTime())
    {
        var ecoun = getRandomInt(0, 3);
        for (var i = 0; i < ecoun; i++)
        {
            var name = generateName();

            data.push({
                name : name,
                trabajo:
name.toLowerCase().replace(' ', '.') +
 '@sencha-test.com',

                blog:
'http://compartimos.com/blog/' + name ,
                twitter: '@' + name
            });
        }

        s = eDate.add(s, eDate.MONTH,
1);
    }

    return data;
})();

```

El siguiente paso es añadir un Store a nuestra APP. ¿Qué es un Store? Un Store es un objeto que encapsula el modelo de datos en la parte del cliente, de tal forma que en él sirva de “cache de datos” y sea el encargado de solicitar los datos a la API REST de SharePoint, se encarga de mapear la información recibida con nuestro modelo de objetos

que nos hemos creado anteriormente. Además este objeto tendremos que configurarlo en los Grids, Combos y demás Widgets que queremos introducir en nuestros desarrollos.

```

// create the Data Store

var store = Ext.create('Ext.data.Store',
{
    // destroy the store if the grid is
destroyed

    autoDestroy: true,

    model: 'Autor',

    proxy: {
        type: 'memory'
    },

    data: data,

    sorters: [{
        property: 'name',
        direction: 'ASC'
    }]
}); return data;
})();

```

Los datos que le hemos pasado al Store es que se base en el modelo que hemos creado anteriormente, y que tenemos la información fija al seleccionar el tipo del proxy “memory” para ello lo que hacemos es indicarle que los datos corresponden con la variable “data” creado anteriormente.

El siguiente paso es crearnos el Grid donde se van a mostrar todos estos datos para ello tenemos que añadir el siguiente código:

```

var rowEditing =
Ext.create('Ext.grid.plugin.RowEditing', {
    clicksToMoveEditor: 1,

    autoCancel: false

});

// create the grid and specify what
field you want

```

```

// to use for the editor at each column.
var grid = Ext.create('Ext.grid.Panel',
{
    store: store,
    columns: [{
        header: 'Name',
        dataIndex: 'name',
        flex: 1,
        editor: {
            // defaults to textfield if
            no xtype is supplied
            allowBlank: true
        }
    }, {
        header: 'Trabajo',
        dataIndex: 'trabajo',
        width: 160,
        editor: {
            // defaults to textfield if no
            xtype is supplied
            allowBlank: true
        }
    }, {
        header: 'Blog',
        dataIndex: 'blog',
        width: 160,
        editor: {
            // defaults to textfield if
            no xtype is supplied
            allowBlank: true
        }
    }
    ],
    renderTo: 'editor-grid',

```

```

width: 600,
height: 400,
title: 'Autor CompartiMOSS',
frame: true,
tbar: [{
    text: 'Add Autor',
    iconCls: 'employee-add',
    handler: function () {
        rowEditing.cancelEdit();

        // Create a model instance
        var r = Ext.create('Autor',
        {
            });

        store.insert(0, r);
        rowEditing.startEdit(0, 0);
    }
}, {
    itemId: 'removeEmployee',
    text: 'Remove Autor',
    iconCls: 'employee-remove',
    handler: function () {
        var sm =
        grid.getSelectionModel();
        rowEditing.cancelEdit();

        store.remove(sm.getSelection());
        if (store.getCount() > 0) {
            sm.select(0);
        }
    }
}

```

```

    },
    disabled: true
  }],
  plugins: [rowEditing],
  listeners: {
    'selectionchange': function
(view, records) {
  grid.down('#removeEmployee').setDisabled(!re
cords.length);
}
}
});

```

Este código es muy semejante al que utilizamos en cualquier desarrollo en .NET es decir, indicamos el origen de datos (Store), indicamos las Columnas que va a tener el Grid y su nombre en el Store, en cada columna podemos indicar el tipo de campo que es (fecha, número) del mismo modo que podemos indicar validaciones en los mismos.

También resulta bastante llamativo que hemos añadido dos funciones para poder añadir y eliminar registros sobre el propio Grid y que se queden reflejados en el Store. Finalmente añadimos un plugin que permite poder editar registros directamente en el Grid. Si realizamos estas modificaciones y ejecutamos la APP el resultado sería el siguiente:

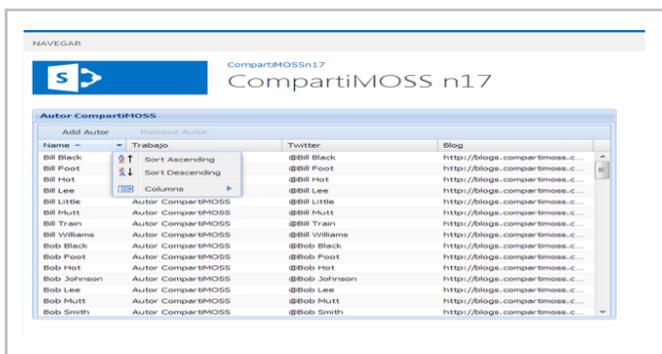


Imagen 3.- Aplicación Demo desplegada.

Como podemos observar ya tenemos los datos en el Grid, y también se aprecian alguna de las características de los mismos como son el orden de las columnas, seleccionar las columnas que queremos visualizar, además de la posibilidad de añadir, editar y eliminar datos desde el propio Grid.

Consumir datos de SharePoint

Una vez ya tenemos la aplicación funcionando con los datos de demostración, viene la parte más divertida y es como hacer que la App obtenga los datos de SharePoint. Para ello tenemos que realizar modificaciones en el Store definido previamente, en lugar de obtener los datos fijos, va a realizar una llamada a la API REST para obtener los datos y mostrarlos en el GRID, para ello tenemos que hacer las siguientes modificaciones en el Store:

```

var store = Ext.create('Ext.data.Store', {
    // destroy the store if the grid is
    // destroyed

    autoLoad:true,

    autoDestroy: true,

    model: 'Autor',

    proxy: {
        type: 'ajax',

        url: '
/_api/web/Lists/getbytitle(\'Autores\')/item
s()?select=Title,Trabajo,Twitter,Blog',

        headers: { "Accept":
"application/json; odata=verbose" },

        reader: {
            type: 'json',
            root: 'd.results'
        }
    },

    sorters: [{
        property: 'name',
        direction: 'ASC'
    }]
});

```

La principal modificación es que hemos cambiado el Proxy por una consulta Ajax que lo que hace es realizar una petición a la API y obtiene los resultados en formato JSON. En la variable root, indicamos el lugar desde el que queremos obtener los datos, dependiendo de la llamada API REST que realicemos los datos se encuentran en una ubicación u otra.

Ahora observamos que se muestran los datos de nuestra lista de SharePoint.

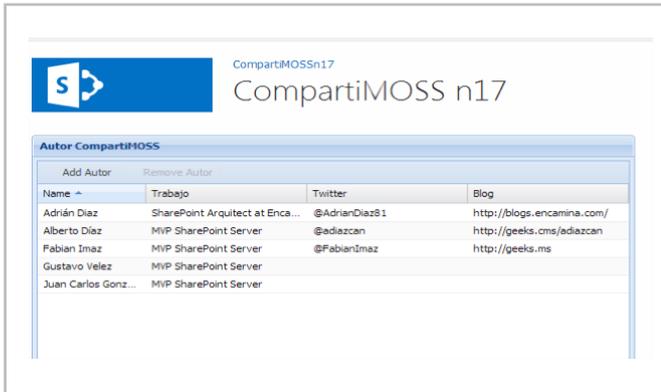


Imagen 4.- Aplicación con Datos de SharePoint

Aún no hemos terminado con el desarrollo, si intentamos hacer cualquier operación CRUD en el Grid, aunque no da error, el resultado no se refleja en nuestra lista. Esto es por dos motivos: el primero es el funcionamiento del Store, el Store es el “encargado” de almacenar los datos en local y posteriormente trasladar estos cambios a la lista. La primera parte lo hace correctamente, pero en el momento que se intenta sincronizar con la lista se produce este error:

✖ Failed to load resource: the server responded with a status of 403 (FORBIDDEN)

¿Por qué se produce este error?, porque en el Store no hemos creado la estructura adecuada para la comunicación por REST. Para solucionarlo tenemos dos opciones: modificar estas llamadas a la API REST de SharePoint o bien añadir una función dentro del grid que cada vez que se realice una operación realice esta operación en la lista de SharePoint.

Para este ejemplo he decidido añadir una función que se encargue de hacer las operaciones en SharePoint. El motivo es demostrar que dentro de Sencha también podemos hacer uso de otras librerías, en este ejemplo voy a llamar al modelo de objetos JavaScript.

Para ello en primer lugar tenemos que añadir un Evento al listener al Store, de tal forma que cada vez que se añada/modifique o elimine un elemento al Store este sea notificado de tal forma que quedaría de la siguiente forma:

```
listeners: {
    'update': function (view,
records) {
        if (records.data.Id == '') {
            AddRecord(records.data);
        }
    }
}
```

```
else
{
UpdateRecord(records.data);
}
debugger;
store.reload();
},
'remove': function (view,
records) {
    RemoveRecord(records.data);
    store.reload();
}
```

Una vez realizada esta modificación tan solo nos queda definirnos las funciones correspondientes para hacer las operaciones contra SharePoint. Estas funciones quedarían de la siguiente forma:

```
var context =
SP.ClientContext.get_current();

function AddRecord(data)
{
    var oList =
context.get_web().get_lists().getByTitle('Autores');

    var itemCreateInfo = new
SP.ListItemCreationInformation();

    this.oListItem =
oList.addItem(itemCreateInfo);

    oListItem.set_item('Title', data.name);

    oListItem.set_item('Trabajo',
data.Trabajo);

    oListItem.set_item('Twitter',
data.Twitter);

    oListItem.set_item('Blog', data.Blog);
}
```

```

oListItem.update();

context.load(oListItem);

context.executeQueryAsync(
    Function.createDelegate(this,
this.onQuerySucceeded),
    Function.createDelegate(this,
this.onQueryFailed)
);
}

function onQuerySucceeded() {
alert('Acción Realizada');
}

function onQueryFailed(sender, args) {
alert('Error');
}

function UpdateRecord(data)
{
    var oList =
context.get_web().get_lists().getTitle('Autores');

    this.oListItem =
oList.getItemById(data.Id);

    oListItem.set_item('Title', data.name);

    oListItem.set_item('Trabajo',
data.Trabajo);

    oListItem.set_item('Twitter',
data.Twitter);

    oListItem.set_item('Blog', data.Blog);

    oListItem.update();

```

```

context.executeQueryAsync(
    Function.createDelegate(this,
this.onQuerySucceeded),
    Function.createDelegate(this,
this.onQueryFailed)
);
}

function RemoveRecord(data)
{
    var oList =
context.get_web().get_lists().getTitle('Autores');

    this.oListItem =
oList.getItemById(data.Id);

    oListItem.deleteObject();

    context.executeQueryAsync(
        Function.createDelegate(this,
this.onQuerySucceeded),
        Function.createDelegate(this,
this.onQueryFailed)
    );
}

```

Con estas modificaciones ya tenemos listo el GRID.

Reseña de Sencha Architect

Sencha tiene un IDE propio para desarrollar sus aplicaciones JavaScript utilizando su propio IDE, se puede obtener una versión de evaluación desde el siguiente link <http://www.sencha.com/products/architect/>. Este IDE nos ahorra bastante trabajo ya que nos evita introducir gran cantidad de código. También tiene inconvenientes ya que es una interfaz muy poco intuitiva para introducir código, está más pensado para hacer bocetos de Apps y una vez el boceto está validado empezamos a introducirle funcionalidad desde otro entorno.

Mi experiencia personal es que el IDE es bueno para el tema del diseño, y la arquitectura de la aplicación pero cuando hay que empezar a añadir funcionalidad a la aplicación aún le falta mucho.

Este es el entorno de trabajo de Sencha Architect



Imagen 5.- Entorno de Sencha Architect

La pantalla está dividida en tres partes:

- A la izquierda las artefactos que podemos incluir en nuestra aplicación.
- En el centro la visualización del desarrollo que estamos implementando (o el código generado)
- A la derecha visualizamos la arquitectura del proyecto generado, y en caso de tener algún elemento seleccionado las propiedades, eventos y funciones que llevan asociados.

La gran ventaja de utilizarlo es que tenemos el código mucho mejor estructurado, cada artefacto introducido en nuestra App lo tenemos en un fichero separado. ¿Qué ventajas tiene? Pues de esta forma tenemos nuestro código mucho más escalable y mantenible. ¿y qué desventajas tenemos? Que tenemos una aplicación más pesada, ya que es necesario descargarse más ficheros para que empiece a funcionar.

Conclusiones

En estos o dos artículos hemos visto dos Frameworks JavaScript totalmente distintos, uno más estricto en cuanto a la forma de desarrollar como es ExtJS y otro un tanto más dinámico como es el de Kendo. Pero lo bueno es que ambos Frameworks los podemos utilizar dentro de SharePoint,

ya que nuestro servidor favorito es una plataforma en la que podemos desarrollar haciendo uso de casi cualquier recurso existente, lo que hace cualquier desarrollador web pueda ser un desarrollador de SharePoint.

Ahora bien si me preguntáis cuál de los dos Frameworks expuestos me gusta más, mi respuesta es que depende de lo que queramos hacer y depende del proyecto.

Las ventajas de Kendo son:

- Exige una curva de aprendizaje mucho menor por parte del desarrollador.
- Más sencillo de maquetar.
- Mucho más soporte en foros al basarse en JQuery.

Mientras que las ventajas de uso de ExtJS son:

- Aplicaciones mucho mejor estructuradas.
- Mejor funcionamiento en todos los navegadores.
- Posibilidad de crear aplicaciones móviles.
- Mucha más documentación.

ADRIÁN DIAZ CERVERA

SharePoint Architect en Encamina

MCPD SharePoint 2010, MAP y MCC 2012

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

adiaz@encamina.com @AdrianDiaz81

...resulta bastante llamativo que hemos añadido dos funciones para poder añadir y eliminar registros sobre el propio Grid...



31

PowerShell Cmdlets: Crearlos o Morir

Resumen

PowerShell es la herramienta por excelencia para administrar a SharePoint. Aunque muchas de las operaciones de administración de SharePoint se pueden realizar con la Interfaz de Usuario, las posibilidades, rapidez y facilidad de uso que ofrece PowerShell lo convierte en el instrumento favorito de los administradores del sistema. A pesar de que SharePoint ofrece por defecto cientos de comandos (llamados Cmdlets), es posible programar nuevos cuando se considere necesario.

Artículo

PowerShell es el Framework para automatización de tareas de Microsoft, consistente de un programa para ejecutar comandos y un lenguaje para ejecutar scripts basado en .NET. PowerShell ofrece acceso a todos los objetos de Windows y prácticamente todos sus servidores, incluyendo a SharePoint.

Aunque desde la versión 2007 de SharePoint fue posible utilizar PowerShell como herramienta de administración, apenas en la versión 2010 obtuvo un conjunto de comandos suficientemente sólido para poder trabajar cómodamente con el Portal. SharePoint 2013 sigue por el camino trazado, aumentando el soporte con nuevos comandos y erradicando totalmente la herramienta original de administración, stsadm.

SharePoint instala por defecto cientos de comandos que permiten realizar tareas administrativas, yendo desde la creación básica de Listas y Sitios, hasta la configuración de subsistemas completos como el Motor de Búsqueda o todos sus Servicios. A pesar de que la Interfaz de Usuario de Administración (la "Administración Central", junto con las páginas de Administración de Colecciones de Sitios, Sitios y Listas) de SharePoint ofrece una forma fácil para configurar los aspectos más comunes y triviales del servidor, hay muchas regiones que no están disponibles de esta forma, y que solo se pueden manejar utilizando a PowerShell. El ejemplo más claro es el Motor de Búsqueda, el que debe ser configurado utilizando los comandos disponibles de PowerShell y no dispone de Interfaz de Usuario para hacerlo.

Otra faceta importante de PowerShell es la rapidez con la

que se puede crear funcionalidad específica, orientada a la Administración, sin necesidad de crear programas "reales" con Visual Studio. Por poseer PowerShell su propio lenguaje de programación que no necesita ser compilado, es posible "ensamblar" comandos de tal forma que formen un conjunto funcional, y hacerlos ejecutar con un solo clic.

Pero, aunque SharePoint ofrece multitud de comandos para PowerShell, ellos son de uso general. Cuando se necesita crear comandos especializados para facilitar alguna tarea en SharePoint, es posible programarlos utilizando Visual Studio, la herramienta de programación por defecto de Microsoft. Esto abre la posibilidad de automatizar el Portal de forma prácticamente ilimitada, facilitando y haciendo más eficiente la labor de los Administradores.

Una pequeña introducción al uso de PowerShell

Cuando se instala SharePoint, se instala al mismo tiempo una consola de PowerShell especializada para trabajar con los comandos del Servidor. Esto facilita el uso de la herramienta, pero también es posible utilizar la consola de Windows PowerShell mismo registrando el conjunto de comandos (llamados "Cmdlets") de SharePoint por medio del comando "Add-PSSnapin Microsoft.SharePoint.PowerShell".

El uso de PowerShell puede resultar extraño en el principio para administradores que lo utilizan por primera vez, pero Microsoft ha introducido suficientes posibilidades dentro de la herramienta misma para hacer su aprendizaje relativamente fácil. Por ejemplo, dispone de una función de "Ayuda" que describe la sintaxis y uso de cada uno de los comandos existentes. Para ver su funcionamiento, inicie una consola de PowerShell de SharePoint y ejecute el comando "Get-Help". PowerShell dispone de un mecanismo para descargar los archivos más recientes de ayuda, así que inicialmente PowerShell pregunta si se desea hacer un update de la información, lo que descarga la última información disponible desde los servidores de Microsoft.

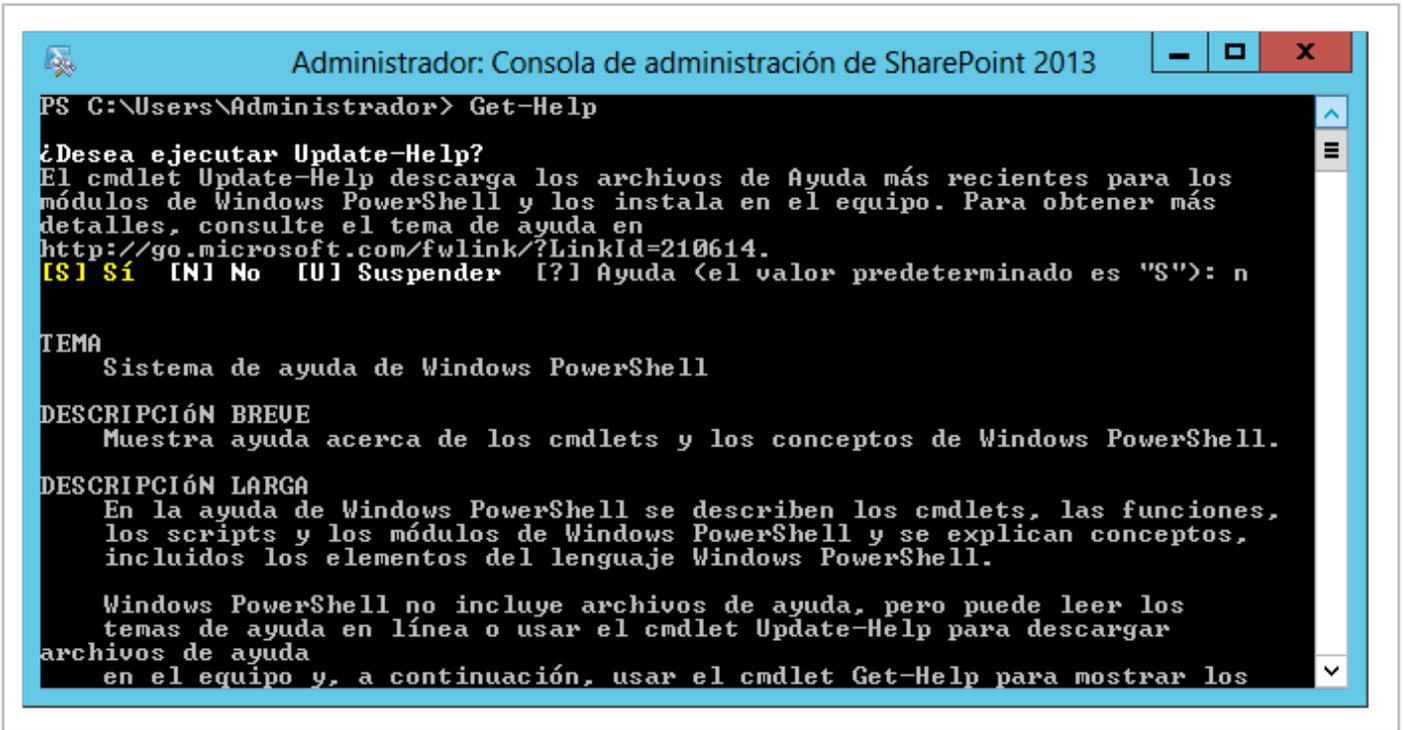


Imagen 1.- La función de Ayuda de PowerShell.

El comando "Get-Help" solo muestra información sobre el uso del comando mismo. Cuando se combina con otro Cmdlet, muestra la sintaxis del comando a investigar. Por ejemplo, "Get-Help Get-SPSite" da toda la información necesaria para la utilización del Cmdlet "Get-SPSite".

El problema con el anterior script es que es necesario conocer el nombre del Cmdlet ("Get-SPSite" en el ejemplo) antes de poder buscar su información de ayuda. Afortunadamente PowerShell puede utilizar información parcial para encontrar el Cmdlet apropiado. Por ejemplo, se sabe que es necesario utilizar el Cmdlet que obtiene el objeto relativo a una Colección de Sitios, pero solo se sabe que el comando tiene algo con "site" en su nombre. Si se utiliza "Get-Help site", PowerShell muestra todos los Cmdlets con "site" en su nombre:

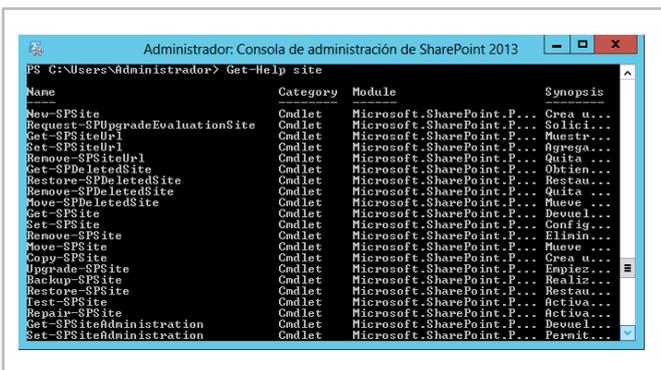


Imagen 2.- Encontrando todos los Cmdlets con "site" en su nombre.

Esta manera de filtrar también incluye caracteres comodines. Por ejemplo, para encontrar todos los Cmdlets que terminan en "site", como "Get-SPSite" y "Repair-SPSite", pero no "Get-SPSiteUrl", se puede utilizar "Get-Help *site".

Una segunda forma de encontrar información sobre el uso de un Cmdlet es utilizar el parámetro "-Examples". Por ejemplo, para ver ejemplos de utilización de "Get-SPSite" se puede usar "Get-Help Get-SPSite -Examples":

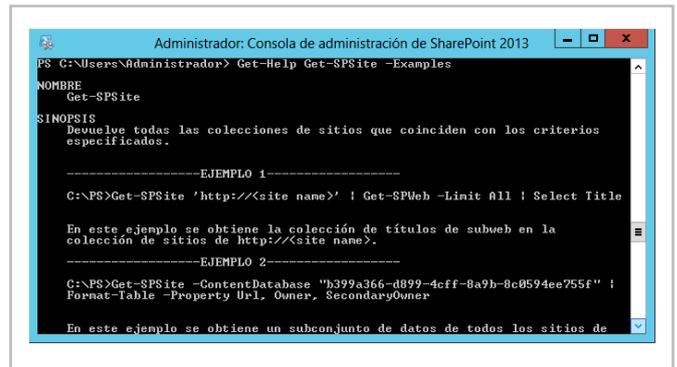


Imagen 3.- Ejemplos de utilización de un Cmdlet.

Los Cmdlets consisten de dos partes: un "verbo" ("Verb") y un "sustantivo" ("Noun") separados por un guion. Una forma alternativa para encontrar el mismo resultado mostrado anteriormente es la utilización del Cmdlet "Get-Command" en lugar de "Get-help". En este caso el Cmdlet está formado por el verbo "Dar" ("Get") y el sustantivo "Comando" ("Command"). Este último Cmdlet también permite filtrar por el verbo o por el sustantivo. Así, si se necesita encontrar todos los Cmdlets para "remover", se puede utilizar "Get-Command -Verb Remove", y para encontrar todos los Cmdlets relacionados con SharePoint se puede usar "Get-Command -Noun sp*"

Finalmente, PowerShell permite encadenar comandos por medio de un "tubo" ("pipe" en inglés, utilizando el carácter "|") y redirigirlos (usando el carácter ">") a un tipo de salida de pantalla diferente. Por ejemplo, para



mostrar todos los Cmdlets de SharePoint en una ventana que permite trabajar más cómodamente con ellos se puede encadenar el comando hacia una ventana con un grid mediante "Get-Command -Noun sp* | Out-GridView", como muestra la Imagen 4. Y para hacer que la salida no sea por pantalla sino hacia un archivo local se puede utilizar "Get-Command -Noun sp* > c:\com.txt", lo que crea un archivo "c:\com.txt" con la salida que de otra forma se vería en la pantalla de PowerShell.

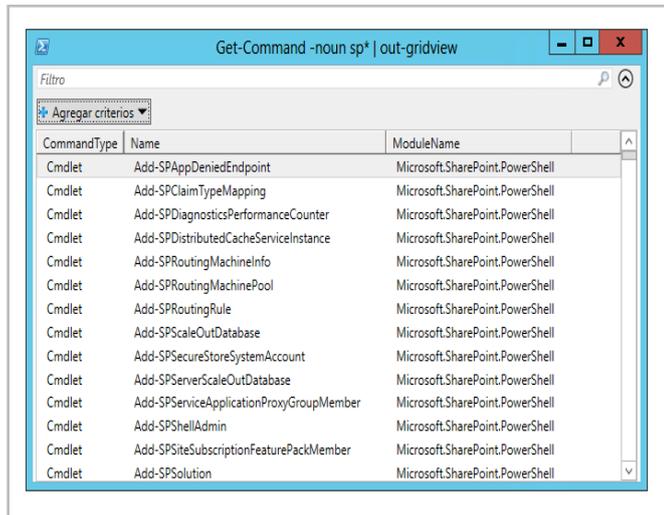


Imagen 4.- Utilización de otras formas de salida de la información.

Creación de Cmdlets para SharePoint

El SDK de SharePoint ofrece extensa información sobre todos los Cmdlets disponibles para el sistema en la página <http://technet.microsoft.com/en-us/library/ee890108.aspx>. Como ya se ha indicado, aunque los Cmdlets por defecto cubren prácticamente todos los aspectos de administración de SharePoint, eventualmente es necesario crear comandos personalizados que hagan alguna tarea especializada, de tal forma que estén integrados entre los Cmdlets de SharePoint por defecto y se puedan utilizar desde la consola de PowerShell de SharePoint sin necesidad de registrarlos específicamente.

Para crear Cmdlets para SharePoint se puede utilizar Visual Studio y sus plantillas para trabajar con el Portal. El siguiente ejemplo muestra cómo crear un Cmdlet básico (para mostrar el título de un Sitio) para SharePoint 2013 utilizando Visual Studio 2012, pero los pasos son exactamente los mismos si se desea utilizar SharePoint 2010 y Visual Studio 2010.

Inicie Visual Studio y seleccione "Nuevo proyecto → SharePoint 2013 - Proyecto vacío" desde la plantilla de "VisualC# → Office/SharePoint → Soluciones de SharePoint". Defina un nombre para el proyecto ("SPTestCmdlet01" en el ejemplo) y su ubicación. En la ventana de selección de tipo de proyecto, elija "Implementar como solución de granja de servidores".

Agregue referencias en el proyecto a "Microsoft.SharePoint.PowerShell" (que se puede encontrar en "C:\Windows\Microsoft.NET\assembly\GAC_MSIL\Microsoft.SharePoint.PowerShell\v4.0.15.0.0.0__71e9bce111e9429c\Microsoft.SharePoint.PowerShell.dll") y a "System.Management.Automation" (desde "C:\Program Files(x86)\Reference Assemblies\Microsoft\WindowsPowerShell\3.0\System.Management.Automation.dll").

Cree un nuevo folder en el proyecto (llamado "SPCmdletGetWebTitle" en el ejemplo), y dentro del folder agregue una nueva clase (menú contextual del folder → Agregar → Clase) llamada "SPGetWebTitle.cs". Reemplace el código original por el siguiente:

```
using Microsoft.SharePoint;
using Microsoft.SharePoint.PowerShell;
using System;
using System.Collections.Generic;
using System.Management.Automation;

namespace SPTestCmdlet01.SPCmdletGetWebTitle
{
    [Cmdlet(VerbsCommon.Get, "SPWebTitle")]
    class SPGetWebTitle :
        SPGetCmdletBase<string>
    {
        [Parameter(Position = 0, Mandatory = true)]
        public string URL = string.Empty;

        protected override
        IEnumerable<string> RetrieveDataObjects()
        {
            try
            {
                using (SPSite mySite = new
                SPSite(URL))
                {
                    using (SPWeb myWeb =
                    mySite.OpenWeb())
                    {
                        return new string[]
                        { myWeb.Title };
                    }
                }
            }
            catch (Exception ex)
            {
                WriteError(ex,
                ErrorCategory.NotSpecified, null);
                return null;
            }
        }
    }
}
```

Inicialmente se necesitan referencias using a “Microsoft.SharePoint”, “Microsoft.SharePoint.PowerShell” y “System.Management.Automation”. El primer punto importante es definir qué tipo de verbo va a utilizar el Cmdlet en la decoración de la clase. Para el ejemplo, que es un Cmdlet del tipo “Dar”, se debe utilizar “VerbsCommon.Get”. PowerShell solamente acepta un número de verbos predefinidos en la enumeración “System.Management.Automation.VerbsCommon”, no es posible crear verbos nuevos. La decoración de la Clase requiere también la cadena que se va a utilizar como sustantivo, “SPWebTitle” para el ejemplo.

La Clase debe ser “public” y heredar de la clase base “SPGetCmdletBase” para el ejemplo pues se está utilizando el verbo “Get”. Hay cuatro herencias disponibles para trabajar con objetos persistentes, es decir, objetos que deben poder mantener la información contenida para pasarla a otra operación: SPRemoveCmdletBase, SPNewCmdletBase, SPSetCmdletBase y SPGetCmdletBase. Para todos los otros tipos de verbos que son utilizados en objetos no persistentes, se debe utilizar SPCmdlet como clase base para heredar. Los objetos no persistentes ejecutan tareas que no requieren mantener información, por ejemplo un Cmdlet para iniciar una configuración o para detener un Servicio.

Los cmdlets aceptan parámetros, que son definidos por una variable pública, como indican las dos líneas de código enseguida de la declaración de la Clase. En el caso del ejemplo se necesita conocer el URL del Sitio, por lo que se declara una variable de tipo cadena llamada “URL” que, en este caso, se inicializa sin contenido (“string.Empty”) aunque se le puede dar un valor por defecto. La propiedad se debe decorar con un atributo “Parameter” que define tanto su posición entre todos los parámetros a utilizar como si es obligatorio o no.

Finalmente se sobrescribe la rutina “RetrieveDataObjects()” que es la que realiza el trabajo dentro de SharePoint. En este caso solamente se instancian dos objetos conteniendo la Colección de Sitios (SPSite MySite) y el Sitio de nivel superior de la Colección (SPWeb myWeb) para devolver el título del Sitio. Es de crucial importancia manejar siempre una buena estrategia de captura de errores para evitar que si ocurre algún problema, PowerShell deje de trabajar sorpresivamente.

Para implementar la función de Ayuda del Cmdlet es necesario crear primero un folder en el proyecto para la carpeta asignada de SharePoint que contiene los archivos xml respectivos (“C:\Program Files\Common Files\microsoft shared\Web Server Extensions\15\CONFIG\PowerShell\Help”). Seleccione el proyecto en el Explorador de Soluciones de Visual Studio y utilizando el menú contextual seleccione “Agregar

→ Carpeta asignada de SharePoint”. Navegue a “CONFIG → PowerShell → Help → en-US” y acepte los cambios. Esto crea un folder “en-US” en el proyecto de Visual Studio. Dentro de la carpeta cree un archivo xml (llamado “SPTestCmdlet01-Help.xml” en el ejemplo).

```

1 <?xml version="1.0" encoding="utf-8" ?>
2
3 <helpItems schema="maml">
4
5 <!-- v 1.3.0.21 -->
6 <command:command xmlns:maml="http://schemas.microsoft.com/maml/2004/10"
7   xmlns:command="http://schemas.microsoft.com/maml/dev/command/2004/10"
8   xmlns:dev="http://schemas.microsoft.com/maml/dev/2004/10">
9   <command:details>
10    <command:name>
11     Get-SPWebTitle
12    </command:name>
13    <maml:description>...</maml:description>
14    <maml:copyright>...</maml:copyright>
15    <command:verb>Get</command:verb>
16    <command:noun>SPWebTitle</command:noun>
17    <dev:version></dev:version>
18  </command:details>
19  <maml:description>...</maml:description>
20  <!-- Cmdlet syntax section -->
21  <command:syntax>...</command:syntax>
22  <!-- Cmdlet parameter section -->
23  <command:parameters>...</command:parameters>
24  <!-- Input - Output section -->
25  <command:inputTypes>...</command:inputTypes>
26  <!-- Return - Output section -->
27  <command:returnValues>...</command:returnValues>
28  <!-- Error section -->
29  <command:terminatingErrors />
30  <command:nonTerminatingErrors />
31  <!-- Notes section -->
32  <maml:alertSet>...</maml:alertSet>
33  <!-- Example section -->
34  <command:examples>...</command:examples>
35  <!-- Link section -->
36  <maml:relatedLinks>...</maml:relatedLinks>
37  </command:command>
38 </helpItems>

```

Imagen 5.- Archivo de ayuda para el Cmdlet.

El archivo está dividido en varias secciones que permiten agregar información sobre la sintaxis, los parámetros, el tipo de entrada y salida de datos, ejemplos, etc. Este archivo puede ser bastante largo, por lo que probablemente la mejor idea es copiar uno de los ya existentes y modificar su contenido de acuerdo al Cmdlet que se esté programando. Microsoft proporciona también información detallada sobre las secciones que se pueden crear en el archivo en el sitio <http://blogs.msdn.com/b/powershell/archive/2006/09/14/draft-creating-cmdlet-help.aspx>.

No es obligatorio definir archivos de ayuda para Cmdlets, aunque si es recomendable hacerlo para auto-documentar el comando. Para el ejemplo se ha creado el archivo de ayuda localizado para ingles en la carpeta en-US. Es posible crear archivo localizados para cualquier otro idioma en su propia carpeta (“es-ES” por ejemplo para castellano), aunque si se definen archivos de ayuda, es necesario tener siempre un archivo en el directorio “en-US” que es el que PowerShell usa por defecto si no hay información de localización disponible.

Finalmente es necesario registrar el Cmdlet para que se integre con los otros comandos por defecto de SharePoint. Cree una nueva carpeta compartida de SharePoint que apunte a “CONFIG → PowerShell → Registration” de la misma forma a como se creó la carpeta para ayuda.

Agrégueme un archivo xml que puede tener cualquier nombre, aunque es recomendable utilizar uno que esté relacionado al Cmdlet que se está programando ("SPTestCmdlet01.xml" en el ejemplo). Reemplace el código original del archivo con el siguiente:

```
<?xml version="1.0" encoding="utf-8" ?>
<ps:Config
xmlns:ps="urn:Microsoft.SharePoint.PowerShell"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:Microsoft.SharePoint.PowerShell SPCmdletSchema.xsd">
  <ps:Assembly
Name="$SharePoint.Project.AssemblyFullName$"
  >
    <ps:Cmdlet>
      <ps:VerbName>Get-SPWebTitle</ps:VerbName>
      <ps:ClassName>SPTestCmdlet01.SPCmdletGetWebTitle.SPGetWebTitle</ps:ClassName>
      <ps:HelpFile>SPTestCmdlet01-Help.xml</ps:HelpFile>
    </ps:Cmdlet>
  </ps:Assembly>
</ps:Config>
```

El archivo define el ensamblado a usar por medio de un macro ("SharePoint.Project.AssemblyFullName\$"), lo mismo que el nombre del Cmdlet que se va a utilizar cuando se llame desde PowerShell ("Get-SPWebTitle"). Es también necesario especificar el nombre de la Clase que define el Cmdlet y el nombre de su archivo de ayuda. Si no se utiliza un archivo de ayuda, es obligatorio de todas formas poner algunos caracteres en la etiqueta "ps:HelpFile" aunque no tengan significado alguno.

La Imagen 6 muestra la estructura final del proyecto en Visual Studio.

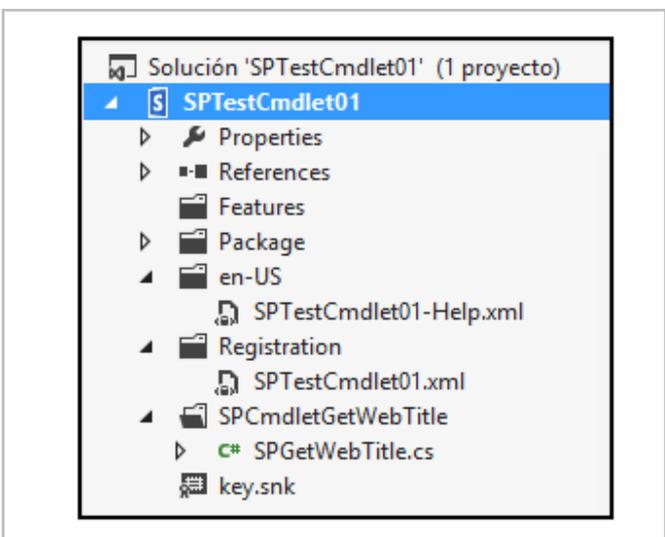


Imagen 6.- Estructura del Proyecto en Visual Studio 2012.

El nuevo Cmdlet está finalizado y listo para funcionar utilizando sus archivos de código, de ayuda y de configuración. Para poder depurar fácilmente desde Visual Studio es necesario acoplar el proyecto al ejecutable de PowerShell registrando los módulos de SharePoint. Para hacer esto, seleccione el proyecto en el Explorador de Soluciones de Visual Studio y utilizando su menú contextual seleccione "Propiedades". En la pantalla de propiedades, seleccione la pestaña de "Depurar" y en "Acción de inicio" utilice en la opción "Programa externo de inicio" el valor:

C:\Windows\System32\WindowsPowerShell\v1.0\PowerShell.exe

En la sección "Opciones de inicio", en "Argumentos de la línea de comandos" utilice:

-NoExit " & 'C:\Program Files\Common Files\microsoft shared\Web Server Extensions\15\CONFIG\PowerShell\Registration\SharePoint.ps1'"

Después de configurar a Visual Studio de esta forma, es posible colocar puntos de interrupción en cualquier línea de código, depurar (F5) y seguir el funcionamiento del Cmdlet de la misma forma que cualquier otro programa desarrollado con VS.

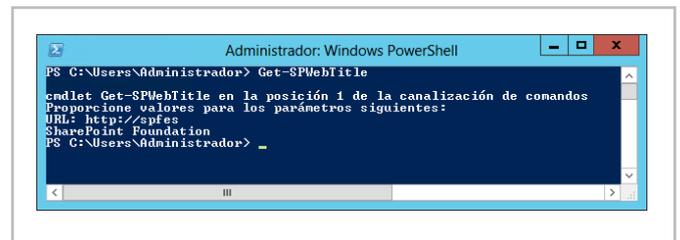


Imagen 6.- El Cmdlet funcionando.

Compruebe también que la Ayuda del Cmdlet funciona correctamente:

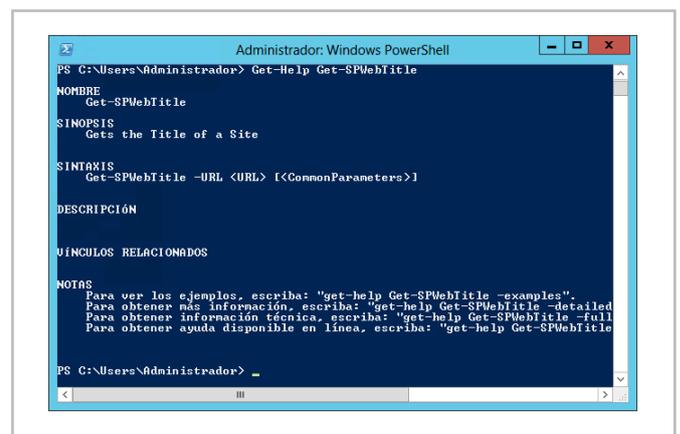


Imagen 7.- El archivo de Ayuda del Cmdlet.

...Los cmdlets aceptan parámetros, que son definidos por una variable pública...

Lo mismo que los ejemplos configurados en la Ayuda:

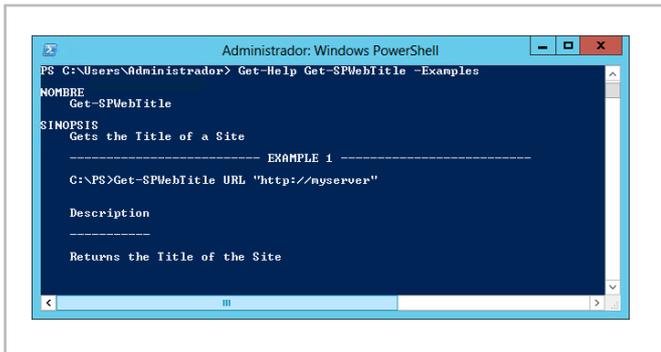


Imagen 8.- El archivo de ayuda mostrando los ejemplos de utilización del Cmdlet.

Finalmente, para distribuir e instalar el Cmdlet, empaque el proyecto de Visual Studio como una Solución de SharePoint (archivo .wsp): en Visual Studio seleccione el proyecto, usando su menú contextual seleccione "Publicar" y en la ventana de configuración utilice "Publicar en el sistema

de archivos" indicando la ubicación de destino. El archivo de Solución de SharePoint generado se puede instalar en cualquier servidor o granja de SharePoint de la misma forma que cualquier otra Solución con personalizaciones para el Portal.

Conclusión

PowerShell es la herramienta por excelencia para administrar a SharePoint. PowerShell no solamente dispone de cientos de comandos para realizar su trabajo con SharePoint, sino que debido a su extensibilidad es posible crear nuevos Cmdlets personalizados con Visual Studio que se integran totalmente con los comandos por defecto suministrados por SharePoint.

GUSTAVO VELEZ

MVP de SharePoint Server

gustavo@gavd.net

<http://www.gavd.net>

No es necesario esperar cuando se descargan archivos voluminosos

SharePoint Quick Previewer

- > Visor de archivos de alta velocidad
- > No necesita visores especiales en los equipos cliente
- > Soporte para archivos MUY grandes

www.kwizcom.com



Haciendo a SharePoint aún mejor!



i

37

Entrevista a Haaron González

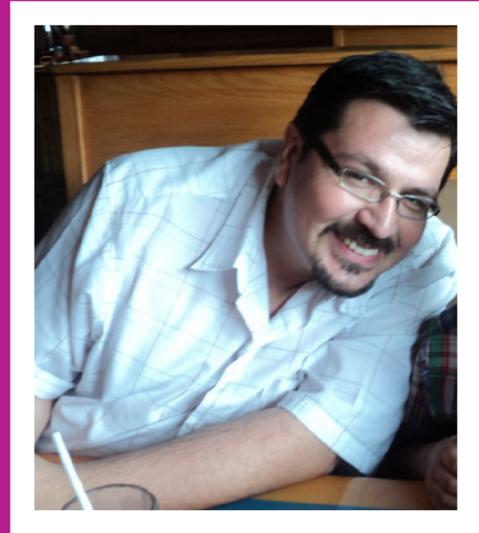
Mi nombre es Haaron Ignazio González Hernández y no pude protestar cuando en el registro civil de mi país las letras “H” y “Z” en mi nombre fueron negligentemente otorgadas por el resto de mi vida.

Desde 1997 tomé la decisión de ser un profesional de la información dedicado a tecnología Microsoft y desde entonces he venido evolucionando con la plataforma de desarrollo. Desde 2006 inicié con SharePoint.

La pasión que siento por el valor que la tecnología aporta a las empresas me ha permitido aprender, compartir, viajar y trabajar para múltiples organizaciones dentro y fuera de México, mi país natal.

Actualmente me desempeño como un consultor especializado en SharePoint ayudando a clientes de México y USA a tomar ventaja del producto.

Desde el año 2005 he venido reflejando en texto experiencias y lecciones aprendidas mediante mi blog <http://msmvps.com/blogs/haarongonzalez> y desde



2008 de forma audiovisual a través de mi canal de YouTube <http://youtube.com/haarongo>. He tenido la suerte de ser reconocido como MVP en ASP.NET desde 2004 y MVP en SharePoint Server desde 2009.

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

Desde niño siempre me atrajo las computadoras y cuando por primera vez escribí, compile y ejecute un programa supe a que me quería dedicar. Desde entonces me ha fascinado el valor y alcance que las soluciones basadas en tecnologías pueden aportar a nuestra vida personal y profesional. Siendo un adolescente fue gracias a un familiar que de pronto un domingo cualquiera llegó a casa con una computadora Intel 80386 y un monitor monocromático como obsequio para motivar mis estudios. Desde entonces mis sábados fueron distintos pasando horas conociendo software de paquetería pero sobre todo leyendo la documentación y practicando ejemplos de programación en Turbo Pascal. Fue durante la educación media superior que un profesor de la preparatoria me mostró Visual Basic

3.0 ejecutándose sobre Windows 3.1, creó un formulario, arrastro y soltó botones y ejecuto F5, desde entonces quede enganchado. Durante mis estudios universitarios aprendí C y C++ y por mi cuenta Visual Basic 6/ASP gracias a Visual Studio 97. Fue otro maestro el que me dio la oportunidad de construir mis primeras aplicaciones cliente-servidor y web para apoyar a la universidad en procesos internos permitiéndome aprender las primeras lecciones del oficio.

¿Cuáles son tus principales actividades tecnológicas hoy en día?

Mis actividades diarias consisten en escuchar los retos y desafíos de clientes en México y USA, posicionar la oferta de valor de la plataforma SharePoint así como los servicios de la empresa en la que participo como líder de práctica.

También constantemente aprendo y comparto experiencias técnicas con programadores y administradores SharePoint dentro y fuera de la empresa.

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

Las actividades no tecnológicas normalmente suceden los fines de semana. Tengo como política no tocar la computadora durante sábados y domingos. Siendo esposo y padre de familia es esencial dedicar el tiempo a aquellas actividades personales/familiares que merecen toda nuestra atención y enfoque.

¿Cuáles son tus hobbies?

Desde niño siempre tuve facilidad para la música y hasta el día de hoy mi principal pasatiempo favorito es capturar emociones y reflejarlas en melodías musicales. He formado bandas y también trabajé como guitarrista en grupos musicales para apoyarme en mis estudios. Aquí algunas de mis melodías: <http://www.youtube.com/playlist?list=PLyOcN1jg2rPgiYoEh1LaC8OW80qcOm>

VEI. Otro pasatiempo que me fascina es poder expresar en texto y audiovisualmente experiencias técnicas y por supuesto compartirlas con otros, mi blog y mi canal de YouTube me permiten hacerlo. Y por último participar en las comunidades de usuarios como expositor en eventos es otro de los pasatiempos y actividades que me son muy gratificantes.

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

La movilidad, el aspecto social y la nube son las tendencias que están moldeando la nueva forma de trabajar en la empresa. Las personas hoy están buscando tener las mismas herramientas de comunicación que usan día a día en casa dentro de la empresa. Las nuevas generaciones tienen otros paradigmas entorno a consumir la información y aplicaciones. Una vez dicho lo anterior, yo si visualizo a SharePoint y toda su gama de soluciones como el sistema operativo del trabajador del conocimiento en la empresa, accesible desde cualquier lugar, a cualquier hora y desde cualquier dispositivo.

...La pasión que siento por el valor que la tecnología aporta a las empresas me ha permitido aprender...



Expertos en SharePoint



Soluciones integrales



Queremos ser su socio tecnológico

Windows 8

SharePoint

Windows Phone

Windows Azure



comercial@gsc.es

www.gsc.es

Novedades en las herramientas de desarrollo de Office para Visual Studio 2013

Resumen

El pasado mes de junio de 2013, Microsoft celebró en San Francisco la conferencia Build y uno de los numerosos anuncios que se produjo en la misma fue la liberación de la versión previa (“preview”) de Visual Studio 2013 (VS 2013) que incorpora algunas novedades en lo que a desarrollo para SharePoint 2013 se refiere. En este artículo daremos un repaso a las mismas, que están enfocadas al desarrollo de aplicaciones para la plataforma.

Artículo

Como parte de VS 2013 Preview, los desarrolladores disponen de serie de una nueva versión de las Office Tools para Visual Studio de manera que sin necesidad de instalar componentes adicionales se puede comenzar a crear tanto soluciones como aplicaciones tanto para SharePoint 2013 On-Premise como para SharePoint Online en Office 365. Para crear soluciones de SharePoint, se dispone (Imagen 1) de las mismas plantillas de proyecto existentes en Visual Studio 2012 (VS 2012).

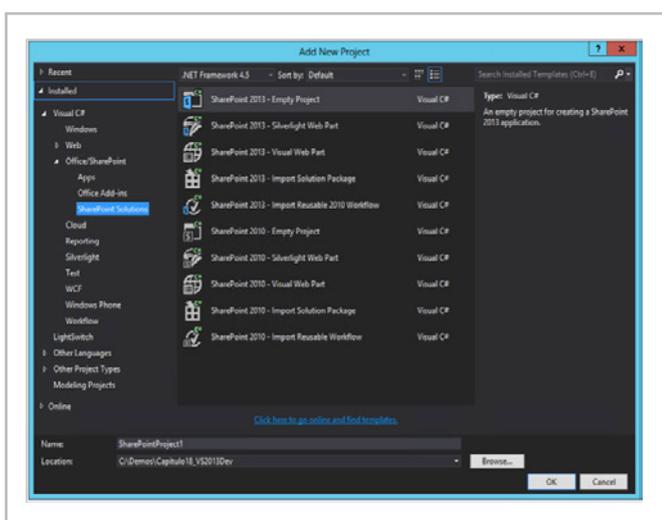


Imagen 1.- Plantillas disponibles para crear soluciones.

Lo mismo se puede decir para las plantillas de elemento de proyecto que se puede agregar en un proyecto de solución de SharePoint 2013. La Imagen 2 muestra los artefactos disponibles para agregar al proyecto.

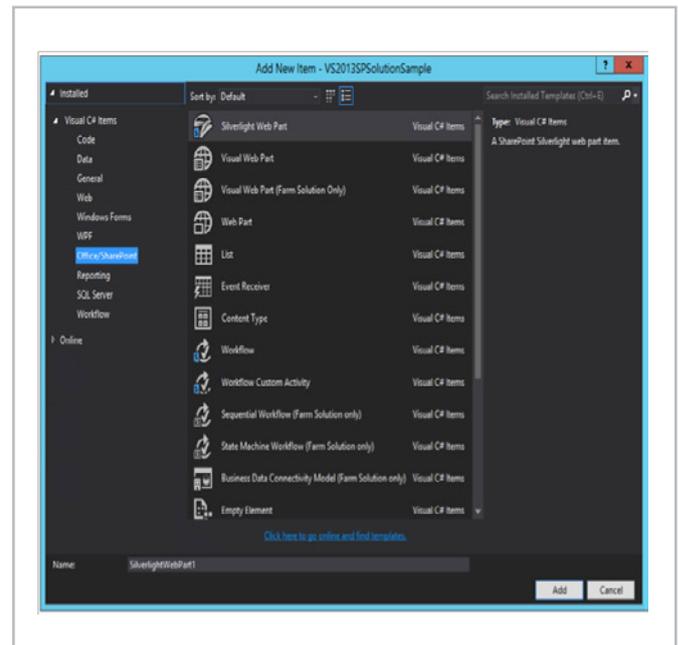


Imagen 2.- Elementos de proyecto que se pueden agregar a una solución.

Novedades en la creación de aplicaciones para SharePoint 2013

Si la creación de soluciones para SharePoint no presenta novedades, la creación de aplicaciones para SharePoint 2013 si incorpora algunos cambios interesantes en VS 2013 Preview:

- Iniciamos VS 2013 Preview y creamos un proyecto de aplicación para SharePoint 2013: “Office/SharePoint -> Apps -> Apps for SharePoint 2013”. En este punto, aparece la primera novedad en el proceso de creación de aplicaciones ya que el asistente para crear el proyecto difiere ligeramente del conocido para VS 2012. Simplemente tenemos que especificar la Url del sitio de SharePoint que vamos a usar para depurar y luego el tipo de hosting de la aplicación que se indica mediante un control de tipo “Radio Button” en lugar de una lista desplegable. Adicionalmente, disponemos de un par de enlaces para facilitar estas configuraciones.

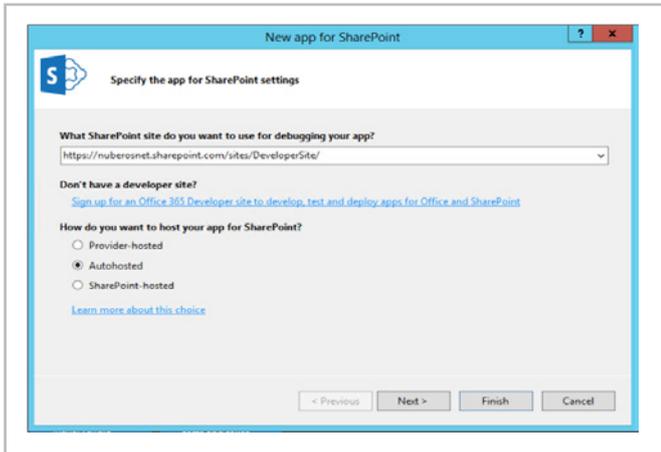


Imagen 3.- Asistente para la creación de una aplicación.

• Para continuar con el asistente, hacemos clic en el botón “Next” de forma que se presenta la segunda novedad en el desarrollo de aplicaciones para SharePoint: para aplicaciones de tipo “Autohospedadas” u “Hospedadas por el proveedor” es posible seleccionar el tipo de aplicación web que se va a crear. Como se aprecia en la Imagen 4, se puede elegir entre aplicación web de formularios ASP.NET o bien aplicación web ASP.NET MVC :-). Por lo tanto, las herramientas de desarrollo de Office para VS 2013 Preview incorporan de serie soporte directo para añadir aplicaciones web ASP.NET MVC en Aplicaciones de SharePoint. Elegimos este tipo y pulsamos “Next”.

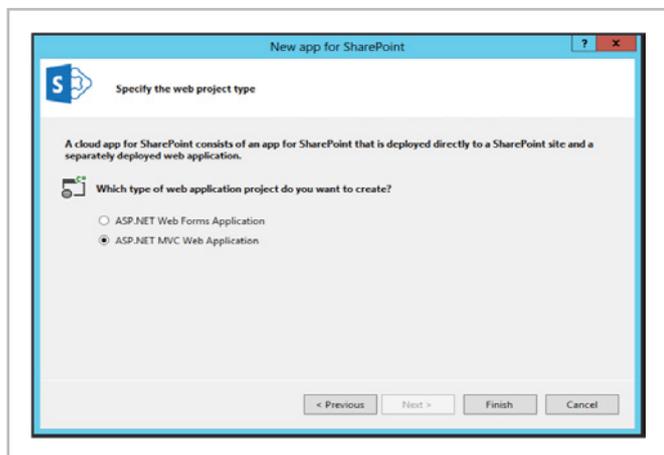


Imagen 4.- Selección del tipo de proyecto web en VS 2013 Preview.

ha creado, podemos ver que tenemos en la solución de VS 2013 Preview dos proyectos: el de aplicación de SharePoint y el proyecto ASP.NET MVC. Si inspeccionamos el proyecto de ASP.NET MVC veremos que incorpora dos novedades en cuanto a artefactos se refiere. Por un lado, disponemos de una pequeña librería JavaScript “spcontext.js” que facilita el trabajo de lectura de parámetros de la Url de la aplicación. Por otro, disponemos de un nuevo helper “SharePointContext.cs” que proporciona clases y métodos para facilitar el trabajo con tokens de contexto entre peticiones de página de la aplicación.

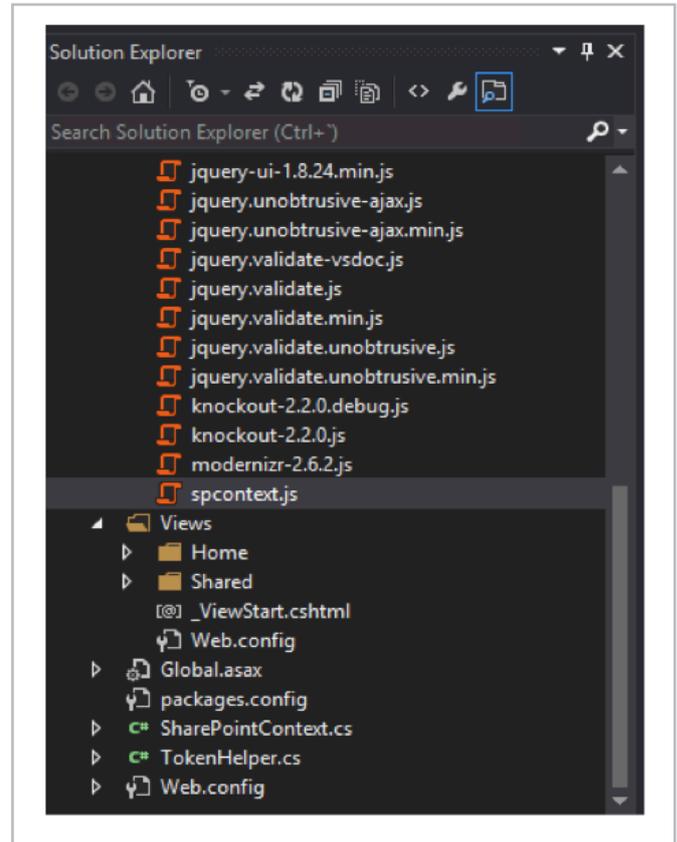


Imagen 5.- Nuevos artefactos en el proyecto web de una aplicación para SharePoint 2013.

Si revisamos el nuevo helper “sharepointcontext.cs” veremos que contiene dos clases abstractas: “SharePointContext” y “SharePointContextProvider”. La primera de las clases facilita la administración y uso de los tokens de contexto entre peticiones de página de la Aplicación. La segunda se encarga de proporcionar instancias de “SharePointContext”. Este helper está disponible de forma general para aplicaciones de tipo Autohospedado y Hospedadas por el proveedor. En cambio, la librería “spcontext.js” es específica de proyectos web de tipo ASP.NET MVC.

Para evaluar cómo utilizar las nuevas clases incluidas en el nuevo helper, podemos inspeccionar el archivo “HomeController.cs” de la aplicación que muestra cómo utilizarlas para obtener el contexto del sitio actual y crear una instancia de “ClientContext” que permita utilizar el Modelo de Objetos en Cliente (CSOM) para interactuar con el sitio host de la Aplicación:

```
User spUser = null;
var spContext =
    SharePointContextProvider.Current.GetShare
    PointContext(HttpContext);

using (var clientContext =
    spContext.CreateUserClientContextForSPHost())
```

```

{
  if (clientContext != null)
  {
    spUser = clientContext.Web.CurrentUser;

    clientContext.Load(spUser, user => user.Title);

    clientContext.ExecuteQuery();

    ViewBag.UserName = spUser.Title;
  }
}

```

Como vemos, el código define un objeto de tipo “SharePointContext” que se utiliza a continuación para definir un contexto de cliente. A partir del contexto de cliente, se accede a información relativa al usuario que está haciendo uso de la aplicación utilizando CSOM.

- Finalmente, ya estamos listos para probar la aplicación sin más que pulsar F5 de forma que tendremos una primera Aplicación de SharePoint que hace uso de ASP.NET MVC.

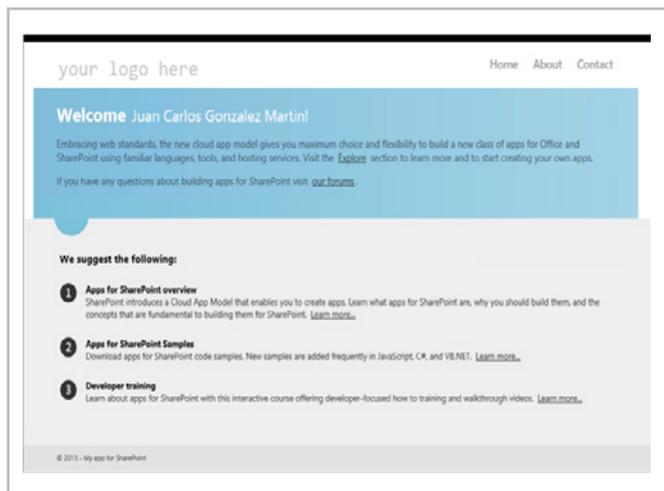


Imagen 6.- Aplicación web ASP.NET MVC desplegada y en funcionamiento.

Novedades en la publicación de aplicaciones

La publicación de aplicaciones para SharePoint 2013 con VS 2013 Preview incorpora el “Administrador de publicación” que facilita el empaquetado de los distintos componentes que la conforman de manera que esté lista para ser publicada. Esta utilidad contiene enlaces diversos a recursos que se pueden necesitar durante la publicación de la aplicación. La Imagen 7 muestra el aspecto del Administrador de publicación para aplicaciones de SharePoint de tipo “Hospedadas por SharePoint” y “Autohospedadas”. Como se puede apreciar, se dispone de un enlace “Package app” que permite generar el archivo “.app” de la aplicación. El enlace “View the Seller Dashboard” es un acceso directo al panel del vendedor.

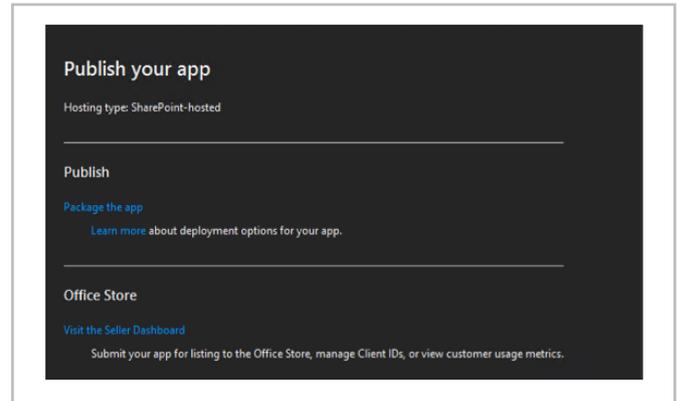


Imagen 7.- Administrador de publicación para aplicaciones “Autohospedadas” y “Hospedadas por SharePoint”.

En cambio, para aplicaciones de SharePoint de tipo “Hospedadas por el proveedor” se disponen de los siguientes elementos de configuración/información adicionales:

- Current profile, Permite indicar perfil de publicación de la aplicación web de la aplicación. Puede ser un nuevo perfil de Windows Azure Web Sites o uno existente.
- Visit the Seller Dashboard, enlace directo al panel del vendedor para publicar aplicaciones en el Office Store.

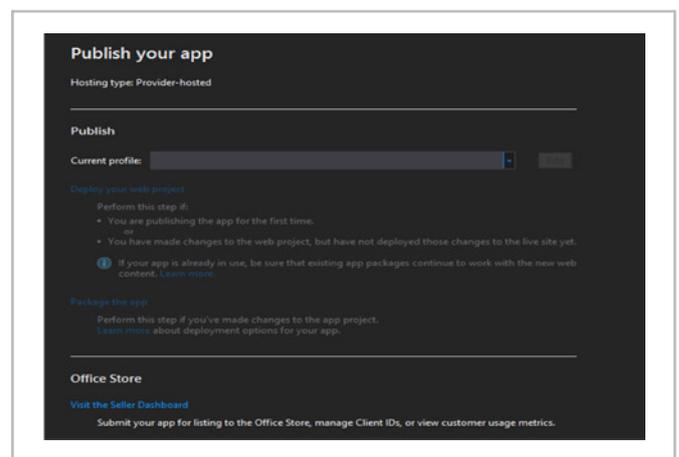


Imagen 8.- Administrador de publicación para una aplicación “Hospedada por el proveedor”.

Conclusiones

La versión preview de Visual Studio 2013 incorpora algunas novedades para la creación de aplicaciones para SharePoint 2013 On-Premise y Online en Office 365. Así, se incluye soporte directo a la creación de aplicaciones web ASP.NET MVC en aplicaciones de tipo “Autohospedado” y “Hospedadas por el proveedor”. Adicionalmente, se incorpora la utilidad “Administrador de publicación” para facilitar la publicación de aplicaciones en el catálogo corporativo o en el Office Store.

JUAN CARLOS GONZÁLEZ MARTÍN

MVP de SharePoint Server | Arquitecto de Soluciones

@jcgm1978

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



42

Creación de aplicaciones de Windows Phone con contenido de SharePoint (Parte 2)

Resumen

En este artículo continuaremos el ejemplo iniciado en el número anterior añadiendo nuevas funcionalidades a la aplicación para Windows Phone que nos muestra las tareas de SharePoint. Añadiremos validaciones de datos, selectores de fecha y mapas de Bing.

Artículo

En el anterior número de CompartiMOSS (nº 16) utilizamos las plantillas de Visual Studio Windows Phone SharePoint Application Templates para crear una aplicación de Windows Phone que trabajase con una lista de tareas para SharePoint.

Mediante el uso del asistente de creación, pudimos hacer una App perfectamente funcional que permitía crear, editar y eliminar tareas de la lista de SharePoint. En este artículo vamos a trabajar para ampliarla y dotarla de nuevas funcionalidades para dejarla más presentable. Añadiremos a la App validaciones de datos, selectores de fecha, y mapas de Bing, etc.

Es imprescindible seguir los pasos de la primera parte del artículo para crear la App, ya que los siguientes apartados la utilizan.

Preparando el entorno de desarrollo

Como recordatorio, para preparar el entorno es necesario el siguiente software:

- Si vas a trabajar con el SDK de Windows Phone 8:
- Microsoft Visual Studio 2012.
- Windows Phone SDK 8.0.
- <http://www.microsoft.com/en-us/download/details.aspx?id=35471>.
- Microsoft SharePoint SDK for Windows Phone 8:
- <http://www.microsoft.com/en-us/download/details.aspx?id=36818>.
- Si vas a trabajar con el SDK de Windows Phone 7:
- Microsoft Visual Studio 2010 (no están soportadas en VS 2012).
- Windows Phone SDK 7.1:

- <http://www.microsoft.com/en-us/download/details.aspx?id=27570>.
- Microsoft SharePoint SDK for Windows Phone 7.1:
- <http://www.microsoft.com/en-us/download/details.aspx?id=35475>.

Una vez que hayamos instalado el SDK de Windows Phone y el SDK de SharePoint para Windows Phone, aparecerán dos nuevas plantillas en Visual Studio:

- Windows Phone Empty SharePoint Application.
- Windows Phone SharePoint List Application.

Si no dispones de Visual Studio, el SDK instalará un Visual Studio Express for Windows Phone. Actualmente, estas plantillas están diseñadas para trabajar únicamente con C#. No están disponibles en Visual Basic.NET. Las plantillas para el SDK de Windows Phone 7 funcionan únicamente con Visual Studio 2010 y Visual Studio 2010 Express for Windows Phone.

Más información sobre la preparación del entorno puede ser encontrada en este enlace:

<http://msdn.microsoft.com/en-us/library/jj163943.aspx>

Validación de datos

Cuando se crea una aplicación de SharePoint para Windows Phone, los formularios de edición (EditForm.xaml) y creación (NewForm.xaml) tienen algunos campos sobre los que se realiza una comprobación de tipo de datos, como los selectores de usuario (Imagen 1).

...Una gran y poco conocida característica de SharePoint 2013 y de SharePoint Online son las columnas de geo localización...

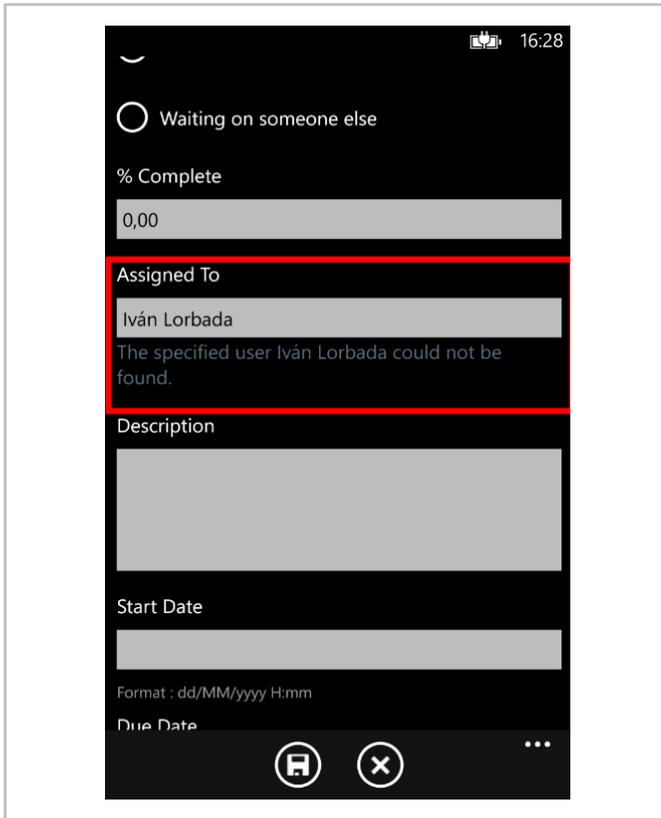


Imagen 1.- Los selectores de usuario buscan el usuario y lo validan. Del mismo modo que los selectores de usuarios de SharePoint

También son validados campos numéricos y de fecha. En la Imagen 2 se intenta escribir un texto en la fecha y salta una validación. Esto es debido a una propiedad `ValidatesOnNotifyDataErrors` del objeto `Binding` que hay en la propiedad `Text` del control `TextBox` en el que se escribe la fecha.

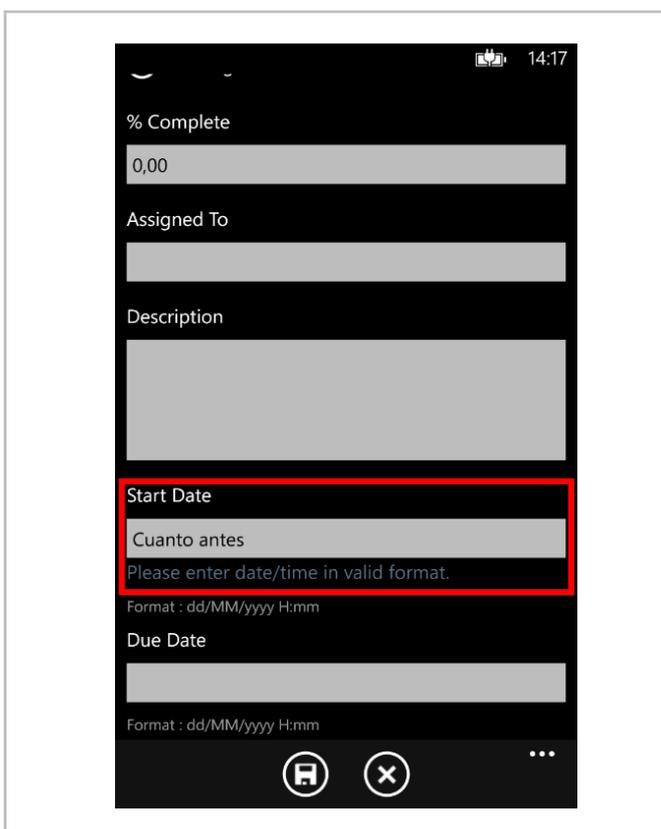


Imagen 2.- Campos como la fecha tienen validación.

Si esta propiedad se establece en `True` se realiza la comprobación al sacar el foco del campo de texto. Si es `False`, el usuario no tendrá constancia de que los datos no son válidos hasta que intente salvar los datos, momento en el que la clase base de la que deriva `EditViewModel` hace la validación de los valores.

La propiedad, que se puede encontrar en el `NewForm.xaml` y el `EditForm.xaml` es la subrayada a continuación:

```
<StackPanel Margin="0,5,0,5" Grid.Row="10">
  <TextBlock TextWrapping="Wrap"
    Style="{StaticResource PhoneTextNormalStyle}">Start
  Date</TextBlock>
  <TextBox Height="Auto"
    Style="{StaticResource TextValidationTemplate}"
    FontSize="{StaticResource PhoneFontSizeNormal}"
    Name="txtStartDate"
    Text="{Binding [StartDate],
    Mode=TwoWay,
    ValidatesOnNotifyDataErrors=True",
    NotifyOnValidationError=True}"
    TextWrapping="Wrap" />
  <TextBlock FontSize="16"
    TextWrapping="Wrap"
    Style="{StaticResource PhoneTextSubtleStyle}"
    Text="{Binding DateTimeFormat}"
  />
</StackPanel>
```

Pero hay una serie de validaciones propias de la lógica de negocio de cada empresa que pueden ser añadidas en el formulario de SharePoint pero que el asistente de creación de la App no es capaz de reproducir. Por ejemplo, imaginemos que la política de la empresa no permite poner un porcentaje completado menor al 50%. Habría que implementar una validación en la App para que el usuario no pudiera guardar una tarea con este porcentaje a menos de 50.

En las aplicaciones diseñadas según el modelo MVVM (Imagen 3), la validación de datos se suele administrar en la capa de datos (es decir, en el componente `Model`). En los proyectos creados a partir de la plantilla de aplicación de lista de SharePoint para Windows Phone, el mecanismo extensible para la validación de datos se ha subido una capa y se ha implementado en el componente `ViewModel`, a fin de que los desarrolladores puedan administrar la validación de datos con mayor facilidad.

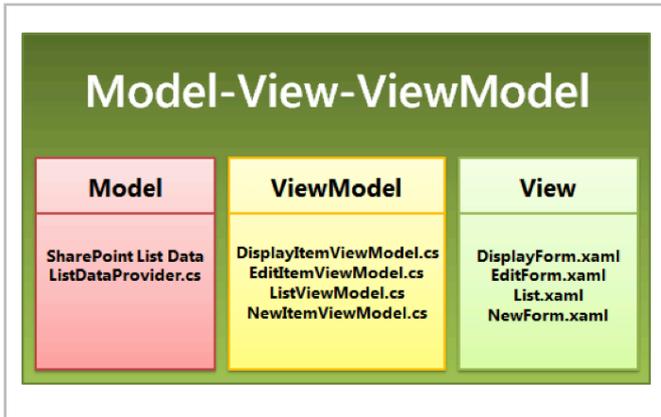


Imagen 3.- Estructura de los archivos en la plantilla "Windows Phone SharePoint List Application" siguiendo el patrón MVVM.

Por lo tanto, en los proyectos basados en la plantilla que estamos usando, el lugar más apropiado para el código personalizado que valida los datos introducidos por el usuario es en las clases ViewModel. En cuanto a la validación de datos, las clases EditItemViewModel y NewItemViewModel (las clases asociadas a los formularios en los que se suelen modificar y actualizar datos de listas) proporcionan una implementación abierta de un método de validación (denominado Validate), que invalida el método de validación base de la clase de la que derivan estas dos:

```

/// <summary>
    /// Code to Validate values for a
    field with given name
    /// </summary>
    /// <param name="fieldName">Name of
    the list Field</param>
    /// <param name="value">Value of the
    list field</param>
    public override void Validate(string
    fieldName, object value)
    {
        base.Validate(fieldName, value);
    }

```

Este método ofrece a los desarrolladores un mecanismo práctico para agregar una lógica de validación personalizada dirigida a campos individuales. El enfoque general es comprobar el valor del argumento fieldName transferido al método Validate para identificar el campo que desea asociar al código de validación personalizado. Podemos, por ejemplo, usar una instrucción switch en la implementación de este método para proporcionar una lógica de validación específica de varios campos en el formulario de edición (EditForm.xaml).

Para ello, hacemos doble clic en el archivo EditItemViewModel.cs y añadimos estas directivas:

```

using System.Globalization;
using System.Text.RegularExpressions;

```

Vamos a hacer un validador que compruebe que no se rellena el campo "% complete" con menos de 50% y que la fecha "Due Date" sea superior a la fecha "Start Date". Para ello, hay que sobrescribir el método Validate con lo siguiente:

```

public override void Validate(string
fieldName, object value)
{
    string fieldValue = value.ToString();
    if (!string.IsNullOrEmpty(fieldValue))
    //Permitiendo los campos en blanco
    {
        bool isProperValue = false;

        switch (fieldName)
        {
            case "PercentComplete":
                // Nos aseguramos que el
                dato es superior a 50.
                int percent;
                isProperValue =
                Int32.TryParse(fieldValue, out percent);
                if (isProperValue)
                {
                    if (percent < 50) // Si
                    menos de 50% lanzamos error de validación.
                    {
                        AddError("Item[PercentComplete]",
                                "Debe
                                introducir
                                un valor
                                superior a
                                50%.");
                    }
                }
                else
                {
                    // La restricción no
                    aplica si es mayor que 50%.
                }
                RemoveAllErrors("Item[PercentComplete]");
            }
            break;
            case "DueDate":
                // Nos aseguramos que Due
                Date es posterior a Start Date.
                DateTime dueDate;
                isProperValue =
                DateTime.TryParse(fieldValue,
                                CultureInfo.
                                CurrentCulture,
                                DateTimeStyles.AssumeLocal, out
                                dueDate);
                if (isProperValue)
                {
                    DateTime startDate;
                    isProperValue =
                    DateTime.TryParse((string)this["StartDate"],
                                CultureInfo.CurrentCulture,
                                Dat

```

En los formularios de edición (EditForm.xaml) y creación (NewForm.xaml) los campos de fecha son campos de texto plano. Es incómodo rellenar fechas en este formato y estamos sujetos a tener errores. Vamos a sustituir estos controles TextBox por controles DatePicker de Windows Phone.

Para realizar esta tarea, es necesario el Windows Phone Toolkit, un proyecto de Codeplex que proporciona controles muy útiles para trabajar con Windows Phone. Puedes consultar información del proyecto en <http://phone.codeplex.com> El proyecto debe ser instalado en el proyecto de Visual Studio vía NuGet utilizando el siguiente enlace: <https://nuget.org/packages/WPtoolkit> Una vez lo hayamos añadido al proyecto, aparecerán nuevas referencias a Microsoft.Phone.Controls y Microsoft.Phone.Controls.Toolkit (Imagen 5)

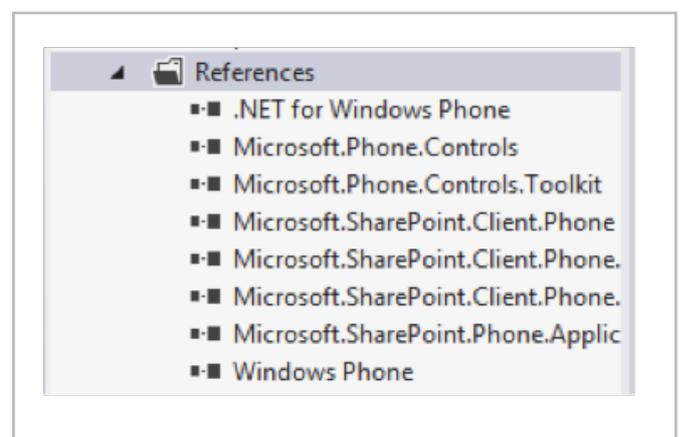


Imagen 5.- Referencias al Windows Phone Toolkit.

Ahora, en el explorador de la solución, hacemos doble clic a EditForm.xaml. En el tag <phone:PhoneApplicationPage> añadimos el namespace que distinguirá los controles del ensamblado del toolkit. Quedará de la siguiente forma (subrayado el nuevo cambio):

```
<phone:PhoneApplicationPage
x:Class="TareasSharePointOnline.EditForm"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:phone="clr-namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone"
xmlns:toolkit="clr-namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone.Controls.Toolkit"
xmlns:shell="clr-namespace:Microsoft.Phone.Shell;assembly=Microsoft.Phone"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
```

```
eTimeStyle
s.AssumeLocal,
out
startDate);

if
(dueDate.CompareTo(startDate) > 0)
{
RemoveAllErrors("Item[DueDate]");
}
else
{
AddError("Item[DueDate]",
"La fecha de
finalización debe ser posterior a la
fecha de
inicio.");
}
}
break;
default:
// No hay que añadir más
validaciones.
break;
}
}
// Y procedemos a la validación por
defecto de la clase base.
base.Validate(fieldName, value);
}
```

Como podemos ver, AddError recibe el nombre de una propiedad y el mensaje de error a pintar en caso de que hayamos considerado que la información introducida en un campo no es válida. RemoveAllErrors quitará todos los posibles errores de validación del campo, en caso de que el usuario haya corregido el valor introducido. La Imagen 4 muestra estos errores. Añadir selectores de fecha

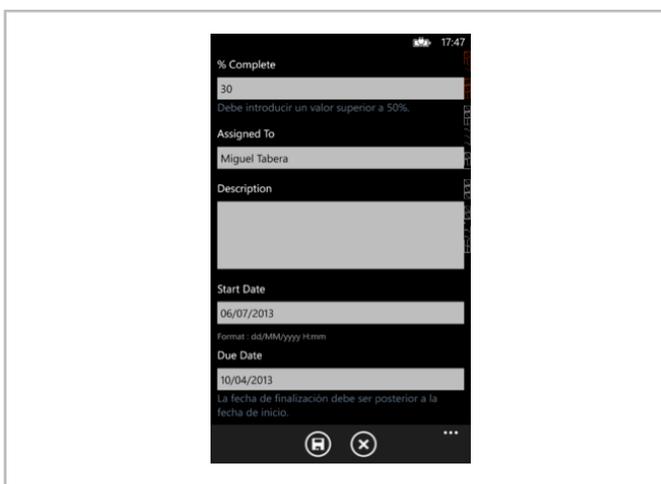


Imagen 4.- Errores de validación personalizados que acabamos de implementar.

```
xmlns:mc="http://schemas.openxmlformats.org/
markup-compatibility/2006"
  mc:Ignorable="d" d:DesignWidth="480"
d:DesignHeight="696"
  FontFamily="{StaticResource
PhoneFontFamilyNormal}"
  FontSize="{StaticResource
PhoneFontSizeNormal}"
  Foreground="{StaticResource
PhoneForegroundBrush}"

SupportedOrientations="PortraitOrLandscape"
Orientation="Portrait"
  shell:SystemTray.IsVisible="True" x:Name
= "EditPage">
```

```
Name="txtDueDate"
Text="{Binding [DueDate],
Mode=TwoWay,
ValidatesOnNotifyDataErrors=Tru
e,
NotifyOnValidationError=True}"
TextWrapping="Wrap" />
<TextBlock FontSize="16"
TextWrapping="Wrap"
Style="{StaticResource
PhoneTextSubtleStyle}"
Text="{Binding DateTimeFormat}"
/>
</StackPanel>
```

A continuación hay que localizar los StackPanel que contienen las fechas StartDate y DueDate. En este proyecto están seguidas. El código es el siguiente (Puede haber alguna diferencia en los márgenes y colocación si usas el SDK de WP7 o el de WP8):

Debemos reemplazar todo este código por lo siguiente:

```
<!--StartDate-->
<StackPanel Margin="0,5,0,5" Grid.Row="10">
  <TextBlock TextWrapping="Wrap"
    Style="{StaticResource
PhoneTextNormalStyle}">Start
Date</TextBlock>
  <TextBox Height="Auto"
    Style="{StaticResource
TextValidationTemplate}"
    FontSize="{StaticResource
PhoneFontSizeNormal}"
    Name="txtStartDate"
    Text="{Binding [StartDate],
Mode=TwoWay,
ValidatesOnNotifyDataErrors=Tru
e,
NotifyOnValidationError=True}"
    TextWrapping="Wrap" />
  <TextBlock FontSize="16"
TextWrapping="Wrap"
    Style="{StaticResource
PhoneTextSubtleStyle}"
    Text="{Binding DateTimeFormat}"
  />
</StackPanel>
<!--DueDate-->
<StackPanel Margin="0,5,0,5" Grid.Row="11">
  <TextBlock TextWrapping="Wrap"
    Style="{StaticResource
PhoneTextNormalStyle}">Due
Date</TextBlock>
  <TextBox Height="Auto"
    Style="{StaticResource
TextValidationTemplate}"
    FontSize="{StaticResource
PhoneFontSizeNormal}"
```

```
<!--StartDate-->
<StackPanel Margin="0,5,0,5" Grid.Row="10">
  <toolkit:DatePicker Header="Start
Date"
    Value="{Binding [StartDate],
Mode=TwoWay}">
</toolkit:DatePicker>
</StackPanel>
<!--DueDate-->
<StackPanel Margin="0,5,0,5" Grid.Row="11">
  <toolkit:DatePicker Header="Due Date"
    Value="{Binding [DueDate],
Mode=TwoWay}">
</toolkit:DatePicker>
</StackPanel>
```

Si ejecutamos la aplicación y editamos un elemento, veremos que ya aparece el selector de fecha (Imagen 6).

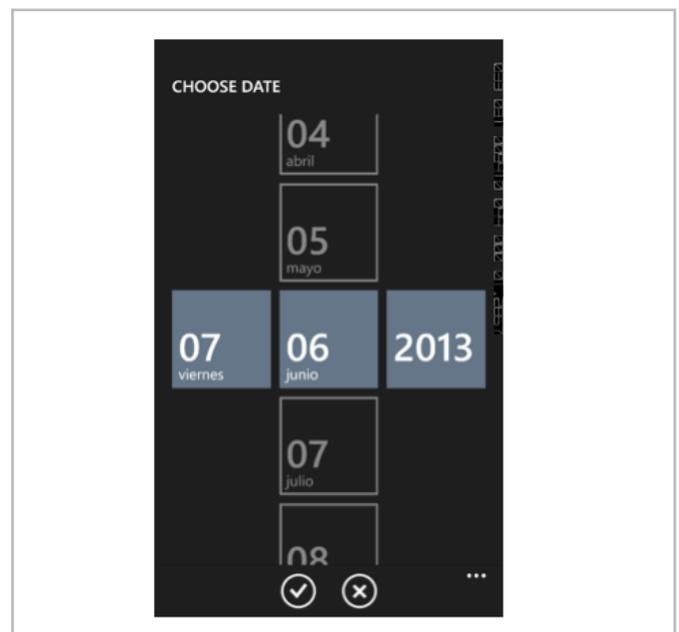


Imagen 6.- Selector de fecha en la app.



Estos pasos deben repetirse en el formulario NewForm.xaml, para tener también los selectores de fecha cuando creamos elementos.

Mapas y geo localización

Una gran y poco conocida característica de SharePoint 2013 y de SharePoint Online son las columnas de geo localización. Son columnas que almacenan coordenadas y se renderizan en un mapa de Bing dentro de SharePoint. Son poco conocidas porque están desactivadas por defecto y deben activarse en una lista vía PowerShell o vía modelo de objetos cliente.

Las App de SharePoint creadas con la plantilla “Windows Phone SharePoint List Application” permiten visualizar en el servicio de mapas del teléfono la información de estas columnas. Vamos a ver un rápido ejemplo de cómo lo hacen.

Lo primero de todo es habilitar una columna de tipo geo localización en un sitio. Microsoft lo explica paso a paso en la siguiente guía:

<http://msdn.microsoft.com/en-us/library/jj163135.aspx>

Pero hay soluciones más inmediatas para hacerlo. Yo suelo utilizar una solución de Borislav Grgić, que ha escrito un código que, una vez introducido en una “script editor webpart”, muestra un formulario en la página para crear rápidamente una columna de geo localización en una lista. Lo bueno es que, al ser código cliente, funciona a la perfección tanto en Office 365 como en SharePoint Server 2013:

<http://mysharepointinsight.blogspot.com.es/2013/05/adding-geolocation-field-to-sharepoint.html>

Para cualquiera de las dos soluciones, necesitamos generar una clave de Bing Maps. En el siguiente artículo se explica cómo hacerlo:

<http://msdn.microsoft.com/en-us/library/ff428642.aspx>

Para este ejemplo, he creado una columna de geo localización llamada “Position” en la lista de tareas que estamos usando (Imagen 7). Si visualizamos la tarea (Imagen 8), veremos que SharePoint muestra el mapa.

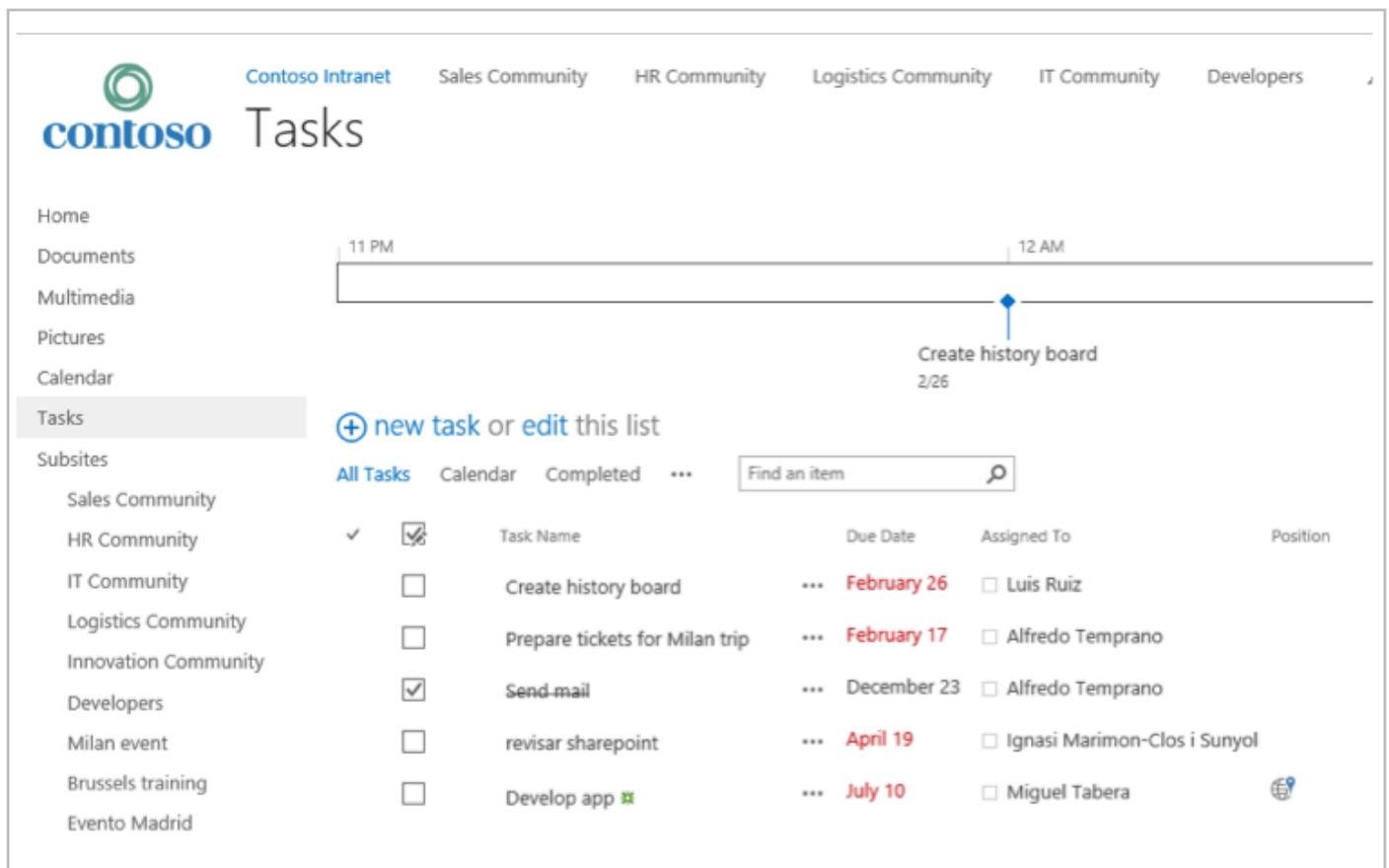


Imagen 7.- Lista de tareas con columna de geo localización.

...Las App de SharePoint creadas con la plantilla “Windows Phone SharePoint List Application” permiten visualizar en el servicio de mapas del teléfono...

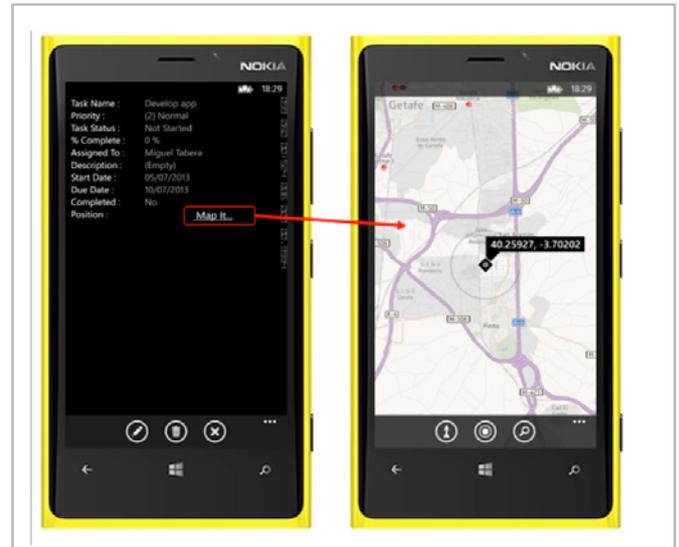
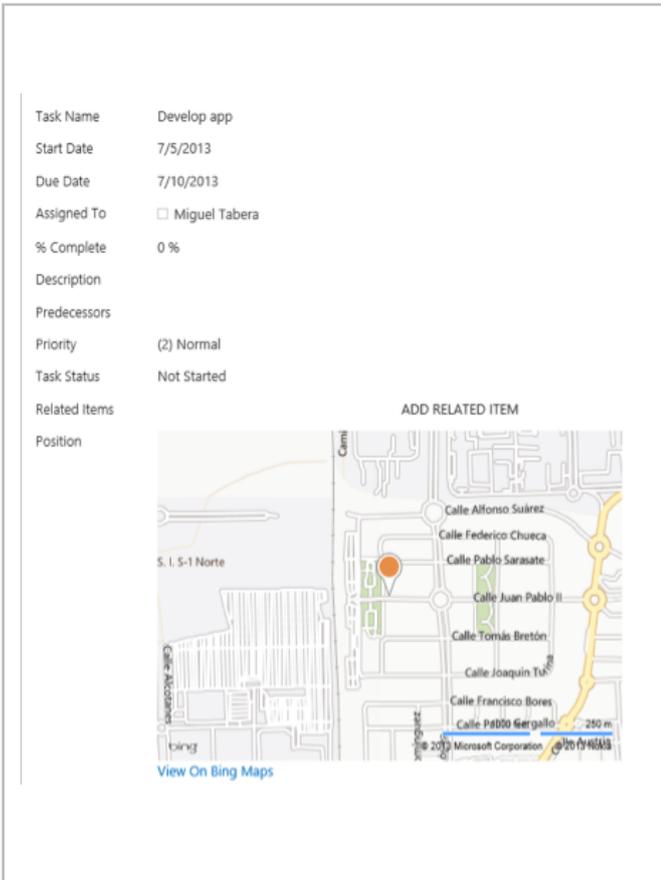


Imagen 9.- La plantilla permite visualizar columnas de geo localización por defecto, sin tener que realizar pasos adicionales.

Enlaces útiles

Como recordatorio, MSDN tiene unas excelentes guías que son muy útiles para desarrollar Apps de SharePoint para Windows Phone usando las plantillas “Windows Phone SharePoint 2013 application templates” de Visual Studio. Si estas guías no habría sido posible escribir el presente artículo. Pueden consultarse todas ellas en el siguiente enlace: [Build mobile apps for SharePoint 2013](http://www.microsoft.com/SharePoint/2013/Build-mobile-apps-for-SharePoint-2013.aspx)

MIGUEL TABERA PACHECO

SharePoint Architect
 Microsoft Active Professional
miguel.tabera@outlook.com
[@migueltabera](https://twitter.com/migueltabera)
<http://www.sinsharepointnohayparaíso.com>

Siguiendo los pasos indicados en la primera parte del artículo (CompartiMOSS nº 16) debemos crear una nueva App, teniendo en cuenta que hay que marcar la nueva columna “Position” en el asistente. Una vez finalicemos, podemos lanzar la App en el emulador y veremos que, si abrimos el ítem, se muestra un botón para visualizar en el mapa:



¿Porque asistir a la Iberian SharePoint Conference es una idea estupenda?

Resumen

El día 10 de octubre de 2013 la comunidad de SharePoint de la Península Ibérica organiza su primera conferencia de SharePoint impulsada desde las comunidades de SharePoint de la Península Ibérica: la "Iberian SharePoint Conference 2013" (www.iberiansharepointconference.com).

Desde la organización queremos que sea todo un éxito, y por eso vamos a destacar los beneficios que te aporta asistir a la conferencia, en tu faceta de profesional de SharePoint.

#1 Conectarás con otras personas de SharePoint

El beneficio más directo de la conferencia es la increíble oportunidad de establecer contactos (el llamado networking) con otros profesionales de SharePoint. En la conferencia van a estar otros desarrolladores, arquitectos, diseñadores, administradores, usuarios, jefes de proyecto y gerentes, así como empresas relevantes del sector, incluyendo a Microsoft, por lo que es una muy buena idea conectar con algunos de ellos y expandir tu red de contactos.

¿No eres una persona lanzada? Tienes muchas ventajas estando en la conferencia. Para empezar, toda la gente que va a estar allí tiene un interés común contigo: SharePoint. Ya tienes el hielo roto. Hay muchas oportunidades para establecer la conversación: en el café, en la comida, en el Ask the Experts, estar sentado al lado de alguien en las sesiones...vamos, que quien no hace unos cuantos contactos en la conferencia es porque no ha querido.

#2 Estarás al día de la oferta y de las novedades de la plataforma

Con todos los avances que ha habido con SharePoint, plataformas relacionadas y los productos que lo extienden, es muy fácil perder de vista todo el ecosistema que hay establecido alrededor de SharePoint. ¡Tranquilo! Entre la oferta de nuestros patrocinadores que verás en el espacio de exposición y los consejos de los expertos que vas a oír,

te quedará una idea mucho más clara sobre la oferta que hay en el mercado. Y si tienes más dudas, ¡pregunta, que para eso estamos!

#3 Aprenderás del mundo real

Leer sobre SharePoint es útil pero sólo hablando con la gente que ha estado en proyectos reales de implementación te enterarás de que no es oro todo lo que reluce en SharePoint. Aprende de la gente que lo ha tenido que aprender sobre la marcha y con sangre, sudor y lágrimas (bueno, sangre lo que se dice sangre....creo que no), para que tú te lo puedas ahorrar.

Aparte de aprender en las sesiones de la conferencia, tenemos un espacio llamado "Ask the Experts" donde podrás hacer preguntas a los expertos en SharePoint. Piensa en ello como en el regalo de una consultoría exprés gratuita por asistir a la conferencia.

#4 Conocerás a referentes nacionales e internaciones

La oportunidad de conocer y hablar con los referentes nacionales e internacionales de SharePoint y Office 365 es otro gran valor de una conferencia de esta naturaleza. Podrás hablar con muchos de los "grandes" (Gustavo Vélez, Michael Noel, Alberto Díaz, Joel Oleson, Edin Kapic, Juan Carlos González, etc.) del sector sobre negocio y tecnología.

#5 Te lo pasarás muy bien

Estamos poniendo mucho esfuerzo en crear un evento desde la comunidad, para la comunidad. No queremos para nada una conferencia encorsetada y académica, sino una conferencia dinámica y participativa. Queremos que tú puedas participar, con tus preguntas, sugerencias y observaciones, además de que te beneficias de los sorteos y ofertas exclusivas para los asistentes.

Y como guinda final, la conferencia se hace en Madrid, una ciudad en la que no te podrás aburrir. Puedes juntar lo útil con lo divertido y aprovechar tu visita a la conferencia para quedarte un día más (o dos) y hacer una visita turística por la capital de España.

#6 Te formarás en SharePoint

El segundo día de la conferencia, 11 de octubre, tenemos tres talleres organizados para los asistentes que quieran una formación más larga y estructurada: desarrollo, administración y negocio. Apúntate al taller que prefieras y aprovecha para hacer un curso especializado de la mano de los mejores expertos de SharePoint, que son renombrados MVPs y MCTs.

#7 Tendrás un excelente retorno de inversión

Como colofón y como último argumento para convencerte, pensamos que tendrás una relación entre el valor recibido y el coste invertido imposible de igualar. Mientras que las conferencias de SharePoint en el mundo tienen precios de 500 euros como mínimo, nosotros hemos preferido ahorrar los costes para que todo el mundo a quien le interese la conferencia pueda asistir. El precio de la entrada es de sólo 75 euros (50€ para los que hicieron el pre-registro).

Disfrutar de todas estas ventajas a este precio es muy difícil de igualar. En los tiempos que corren cualquier ahorro es bienvenido y por eso hemos querido que el precio no sea un problema.

#8 Habrá un montón de sorpresas

En una conferencia impulsada por la comunidad, siempre hay sorpresas agradables para los asistentes en la forma de sorteos de regalos, obsequios de los patrocinadores, etc. No es un incentivo para asistir a la conferencia, pero sin duda es un elemento positivo a sumar a un gran evento como el que se está gestando en estos momentos

El equipo organizador de la 1ª Iberian SharePoint Conference

www.iberiansharepointconference.com

@IberianSPConf

¡Te esperamos el día 10 de octubre en Madrid!

The infographic is a vertical stack of four colored bands (teal, orange, pink, red) on the right, and a white scroll-like structure on the left. Each scroll contains a number (01-04) and a category name. The right side lists the services and their sub-items.

- 01 CORPORATIVO**
 - Logotipo
 - Manual de identidad de marca
 - Folletería
- 02 DISEÑO WEB**
 - Maquetación HTML + CSS
 - Diseño de ecommerce
 - Diseño de CRM
- 03 IMPRESO**
 - Diseño Editorial
 - Diseño Packaging
 - Diseño de Urbanización
- 04 PUBLICITARIO**
 - Avisos publicitarios
 - Newsletter
 - Banners

nitido
GRAPHIC & WEB DESIGN

Pedro Fco. Berro 666, of 402
Tel: +598 2712 5061
www.nitido.com.uy
contacto@nitido.com.uy



51

ASP.NET MVC para SharePointers (III)

Resumen

En este artículo cerraremos la serie de programación de aplicaciones de SharePoint con ASP.NET MVC con una introducción a inyección de dependencias y pruebas unitarias.

Artículo

En los artículos previos vimos cómo crear aplicaciones de SharePoint con la tecnología ASP.NET MVC, introduciendo conceptos como las vistas, los controladores o las acciones. En este artículo cerraremos la serie con inyección de dependencia y pruebas unitarias, conceptos muy importantes para el desarrollo web fácil de mantener y extender.

Inyección de dependencia

El término “inyección de dependencia” (Dependency Injection, DI) y su cercano pariente “inversión de control” (IoC) son patrones arquitectónicos que resuelven el problema de acoplamiento fuerte entre componentes. Voy a intentar explicar el concepto para que quede claro.

Acoplamiento fuerte

Cuando una clase en nuestro programa necesita otra clase para realizar alguna acción, decimos que hay acoplamiento entre las dos clases. En el ejemplo siguiente, la clase ProductsController y la clase SharePointHelper están acopladas entre sí.

```
public class ProductsController : Controller
{
    public ActionResult Index()
    {
        SharePointHelper helper = new SharePointHelper();
        var products = helper.GetProducts();
        return View(products);
    }
}
```

El acoplamiento suele ser unidireccional, en nuestro caso ProductsController “sabe” como crear y pasar los datos a SharePointHelper pero SharePointHelper vive en la ignorancia más absoluta de quien lo está usando.

Pongámonos en el caso de tener que modificar este programa unos meses después. Ahora en vez de usar los datos en SharePoint decidimos almacenarlos en una base de datos NoSQL, como por ejemplo MongoDB. Tendremos que crear una nueva clase MongoDBHelper y sustituir las referencias originales de SharePointHelper por MongoDBHelper en el controlador.

```
public class ProductsController : Controller
{
    public ActionResult Index()
    {
        MongoDBHelper helper = new
        MongoDBHelper();
        var products = helper.GetProducts();
        return View(products);
    }
}
```

El problema principal es que ProductsController tiene que saber dos cosas del servicio al que está acoplado:

- La clase exacta del servicio que tiene que invocar (por lo tanto, necesita tener referencia a su ensamblado).
 - La manera de construir el servicio (pasándole los parámetros del constructor, si los hubiera).
- Esto provoca que cambiar el servicio no sea fácil y que el código del controlador tiene poco potencial de reutilización o extensión.

Inversión de control

Una de las soluciones a este problema es la inversión de control. Se llama así porque en vez de que el “usuario” de una clase (en nuestro caso, ProductsController) decida que objeto va a instanciar, el mismo objeto “le viene dado” al usuario y por tanto el “usuario” no tiene control sobre el objeto instanciado.

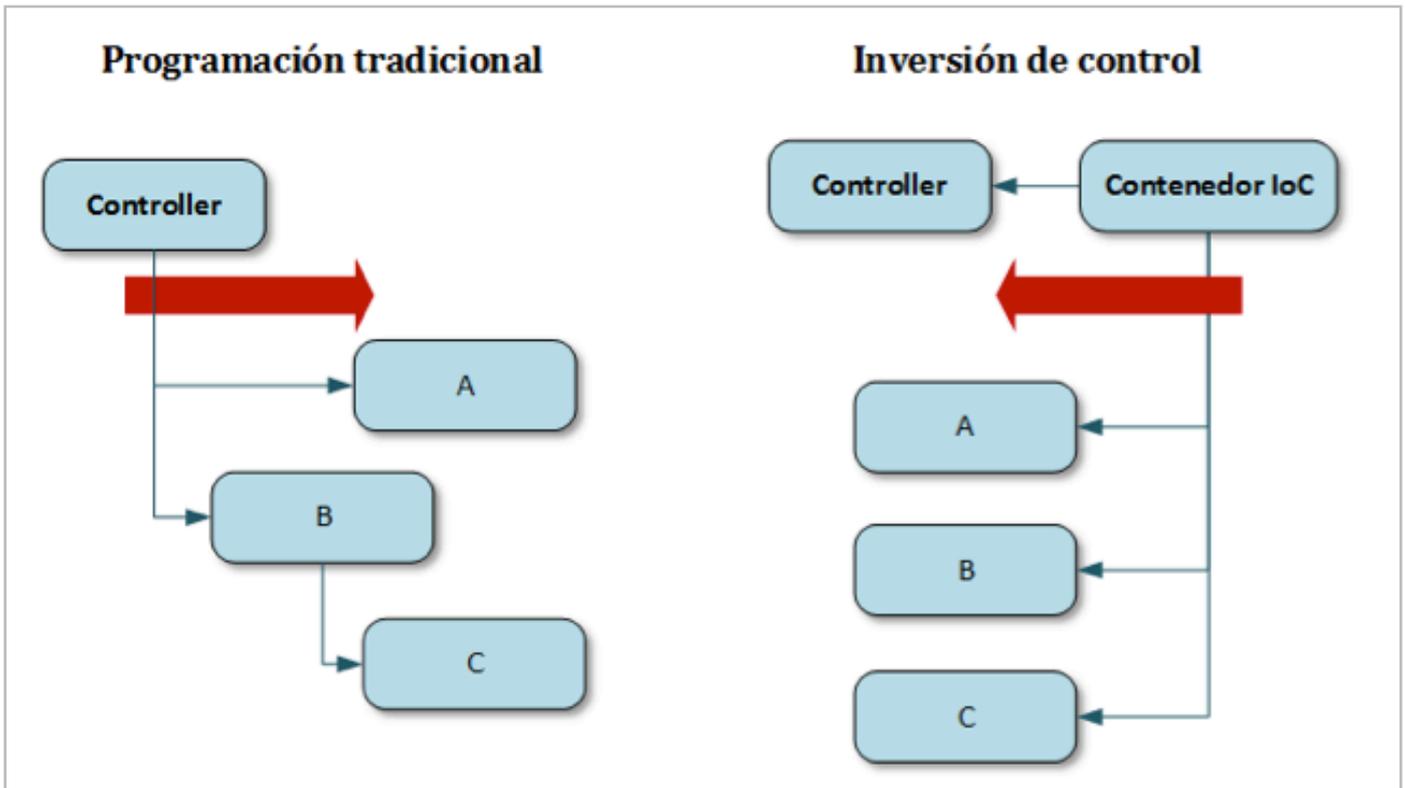


Imagen 1.- Programación tradicional versus Inversión de control

Como se ve en la Imagen 1, el usuario (Controller) cede el control sobre los objetos de los que depende a otra entidad, llamada contenedor de inversión de control (IoC). En la actualidad el contenedor es una librería especializada como Unity, AutoFac, StructureMap, Ninject o alguna de las muchas librerías de IoC disponibles. Sin embargo, en la versión más "casera" de inyección de dependencia este contenedor podría ser el método Main de una aplicación. Lo importante es entender el concepto, más allá de la implementación concreta.

¿Cómo queda ahora nuestro controlador? Al quitarle la potestad de inicializar los objetos que necesita, ahora está obligado a solicitarlos explícitamente al contenedor. Pero bueno, ¿qué va a solicitar? Pues nada menos que una interfaz. El contenedor le devolverá el objeto más adecuado que implementa esa interfaz.

En nuestro caso tenemos que sacar la funcionalidad común de SharePointHelper y MongoDBHelper en una interfaz llamada IBackendHelper.

```
{
    public Product GetProduct(int productId);
    public IEnumerable<Product> GetProducts();
    public void SaveProduct(Product p);
}
```

Ahora sólo falta habilitar la inyección de dependencia en MVC. Para ello usaremos el framework de Microsoft llamado Unity.

ASP.NET MVC y Unity

En vez de usar la librería Unity de manera aislada, nos bajaremos una adaptación de Unity para MVC llamada Unity.Mvc4 (si tenéis MVC 3, podéis usar Unity.Mvc3, es idéntico). Instalamos el paquete NuGet de Unity.Mvc4 y estamos listos para seguir.

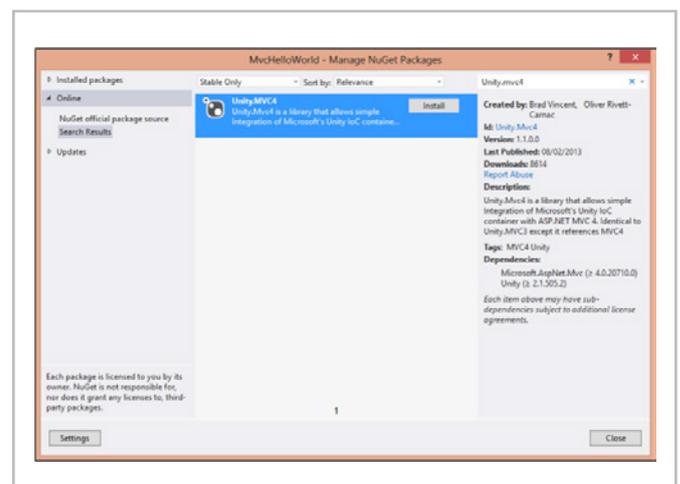


Imagen 2.- El paquete de Unity para MVC.

Lo que hace este paquete es registrar un contenedor de IoC de Unity ligado a la vida de una petición HTTP en ASP.NET MVC.



Además, nos crea una clase llamada Bootstrapper.cs donde vamos a registrar las dependencias (en nuestro caso IBackEndHelper y SharePointHelper), en el método BuildUnityContainer(). Ya tenemos todo el contenedor creado, sólo hay que añadir la línea de código que está resaltada:

```
private static IUnityContainer
BuildUnityContainer()
{
    var container = new UnityContainer();

    container.RegisterType<IBackEndHelper,
    SharePointHelper>();

    RegisterTypes(container);

    return container;
}
```

El último paso para registrar el contenedor es añadir una línea en el evento Application_Start de Global.asax para inicializar el propio Bootstrapper:

```
protected void Application_Start()
{
    AreaRegistration.RegisterAllAreas();

    Bootstrapper.Initialise();

    RegisterGlobalFilters(GlobalFilters.Filters)
    ;
    RegisterRoutes(RouteTable.Routes);
}
```

Ahora ya tenemos un contenedor inicializado con las dependencias. A continuación, toca eliminar los constructores de SharePointHelper en ProductsController y sustituirlos por una variable local _helper. Esta variable se inicializará con el valor pasado por el constructor:

```
public class ProductsController : Controller
{
    IBackEndHelper _helper;

    public ProductsController(IBackEndHelper
    helper)
    {
        _helper = helper;
    }

    public ActionResult Index()
    {
        var products =
        _helper.GetProducts();
        return View(products);
    }
}
```

Esta es la “magia” de inyección de dependencia: el controlador recibe la dependencia automáticamente sin tener que saber cómo se construye. Todo el trabajo de construir y mantener los objetos lo hace el contenedor.

Pruebas unitarias

Una vez que tenemos probado y digerido el concepto de inyección de dependencia, vamos a dotar a nuestra aplicación de ejemplo con una batería de pruebas unitarias que validarán que la aplicación funciona bien.

Las pruebas unitarias son pequeños métodos que usan las aserciones (Assert) para comprobar que se cumplen ciertas condiciones al salir del método.

El problema clásico cuando hacemos las pruebas unitarias es que resulta difícil probar componentes que acceden a servicios fuera de la aplicación como las bases de datos, SharePoint, servicios web etc. Sin embargo, al tener desacoplados los controladores de las clases “ligadas al exterior” podemos utilizar la inyección de dependencia a nuestro favor.

La idea es crear una clase “ficticia” que implemente la interfaz que el controlador necesita. En las pruebas unitarias usaremos esa clase ficticia y en el código real usaremos la real. Nuestro controlador no tiene (más bien, no debe) saber que está recibiendo una u otra.

Para comenzar, añadiremos a la solución un nuevo proyecto de tipo Unit Test Project y dentro del proyecto vamos a renombrar la clase que nos crea por defecto a ProductTests.cs. Agregaremos la referencia al proyecto de MVC para poder acceder a las clases y también a los ensamblados de System.Web.Mvc.

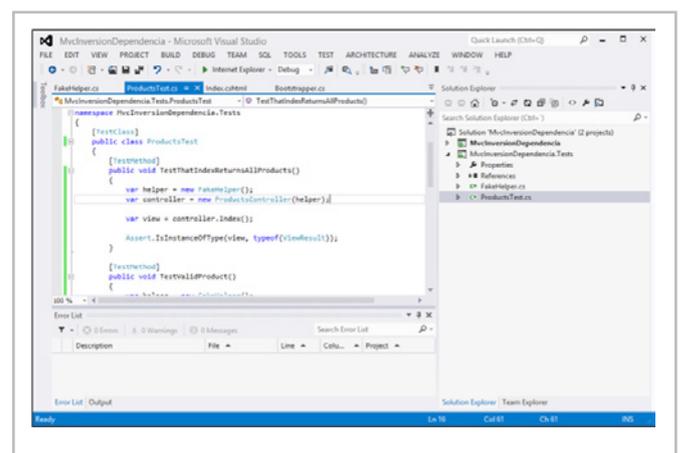


Imagen 3.- El proyecto de pruebas unitarias

El helper ficticio FakeHelper.cs que necesitamos crear será una lista de instancias de Product creadas de manera fija.

```
public class FakeHelper : IBackEndHelper
{
    List<Product> _list;

    public FakeHelper()
    {
```



```

        _list = new List<Product>();
        _list.Add(new Product() { Name =
"Product 1", ProductId = 1 });
        _list.Add(new Product() { Name =
"Product 2", ProductId = 2 });
        _list.Add(new Product() { Name =
"Product 3", ProductId = 3 });
    }

    public Models.Product GetProduct(int
productId)
    {
        var result = _list.Where(x =>
x.ProductId == productId).FirstOrDefault();
        return result;
    }

    public IEnumerable<Models.Product>
GetProducts()
    {
        return _list;
    }

    public void SaveProduct(Models.Product
p)
    {
        return;
    }
}

```

Ya estamos listos para hacer el código de las pruebas. La mayoría de pruebas unitarias siguen el llamado patrón ACT (Arrange, Call, Test):

- Primero instanciamos los objetos necesarios para la prueba (Arrange).
- Luego llamamos el método que queremos probar (Call).
- Al final, comprobamos los valores devueltos con una aserción (Test).

El primer test que vamos a hacer es comprobar que la acción Index del controlador devuelve una vista:

```

[TestMethod]
public void TestThatIndexReturnsView()
{
    // Arrange
    var helper = new FakeHelper();
    var controller = new
ProductsController(helper);
    // Call
    var view = controller.Index();
    // Test
    Assert.IsInstanceOfType(view,
typeof(ViewResult));
}

```

En esta prueba instanciamos un ProductsController pasándole como dependencia un FakeHelper. La acción Index() devuelve una vista de tipo ViewResult y al final comprobamos que la vista que se devuelve es del tipo correcto.

Otra prueba que podemos hacer es llamar a un producto concreto (p.ej. el que tiene ID 2) con la acción Display del controlador y comprobar su nombre en la vista que se devuelve. Para ello disponemos de la propiedad Model de la vista.

```

[TestMethod]
public void TestValidProduct()
{
    var helper = new FakeHelper();
    var controller = new
ProductsController(helper);

    ViewResult view = (ViewResult)
controller.Display(2);
    Product product = (Product) view.Model;

    Assert.IsNotNull(product);
    Assert.AreEqual(product.Name, "Product
2");
}

```

Para finalizar, la última prueba que vamos a hacer es consultar la acción Display pasándole un valor incorrecto y comprobaremos que el producto que devuelve la vista es nulo.

```

[TestMethod]
public void TestInvalidProduct()
{
    var helper = new FakeHelper();
    var controller = new
ProductsController(helper);

    ViewResult view =
(ViewResult)controller.Display(4);
    Product product = (Product)view.Model;

    Assert.IsNull(product);
}

```

...son patrones arquitectónicos que resuelven el problema de acoplamiento fuerte entre componentes...

Probaremos los tests yendo al menú Test/Run en Visual Studio. Si todo ha ido bien, veremos las luces verdes de pruebas finalizadas con éxito:

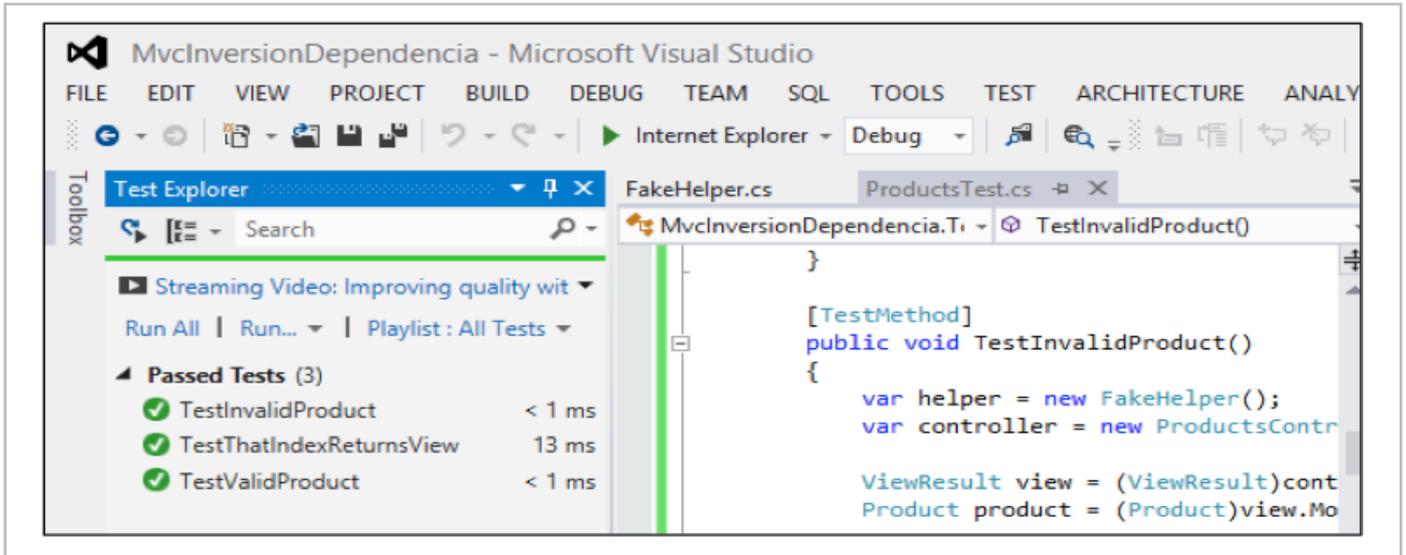


Imagen 4.- La vida es bonita con las pruebas unitarias en verde.

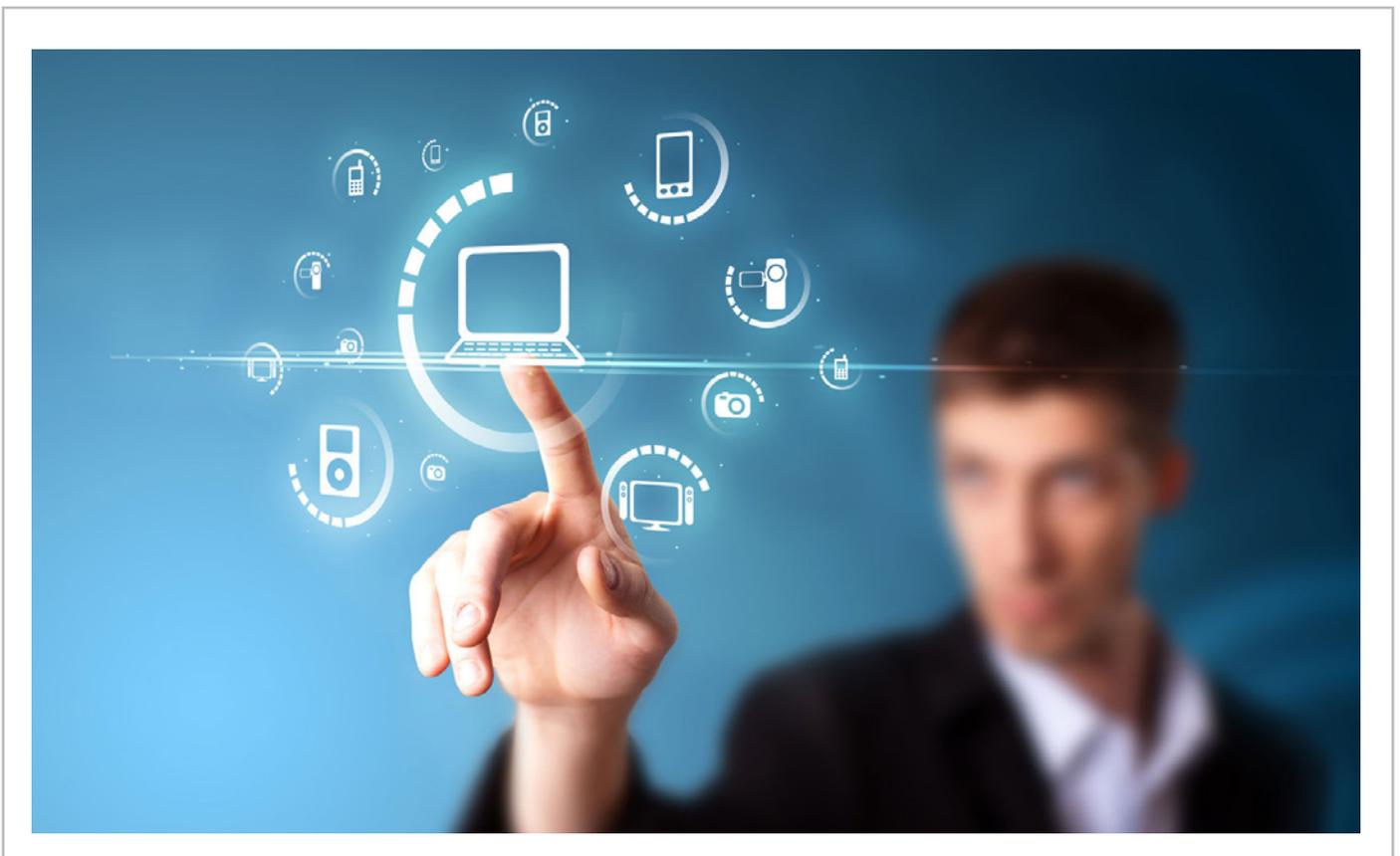
Conclusión

En este artículo hemos visto cómo separar las dependencias dentro de los controladores de MVC y cómo podemos aprovechar ese desacoplamiento para facilitar las pruebas unitarias en nuestras aplicaciones MVC. Para no extenderme demasiado, he dejado temas como la inyección de vistas o mocking de objetos en las pruebas unitarias para un posible futuro artículo, si hay suficiente interés.

Espero que hayáis disfrutado de esta pequeña serie de introducción a ASP.NET MVC para aplicaciones de SharePoint y que hayáis aprendido algo de todos ellos. El código completo de este artículo está disponible en <http://sdrv.ms/16xoioC>.

EDIN KAPIC

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



Sitios de la Comunidad

Resumen

Este artículo tiene la intención de analizar los objetos que se despliegan en un sitio de la comunidad, analizando listas, bibliotecas, páginas, elementos web y características que permiten convertir estos sitios en un foro con funciones avanzadas.

Artículo

“Un lugar donde los miembros de la comunidad conversan sobre temas de interés común. Los miembros pueden

examinar y descubrir contenido relevante si exploran las categorías, ordenan las discusiones por popularidad o eligen ver sólo los mensajes que tienen una mejor respuesta. Los miembros pueden obtener puntos de reputación al participar en actividades de la comunidad: iniciando discusiones y respondiendo a ellas, marcando las publicaciones que les gustan o indicando las mejores respuestas, entre otras”.

Así define SharePoint 2013 esta plantilla de sitio que es novedad en esta versión. Pero ¿qué significa esto exactamente?

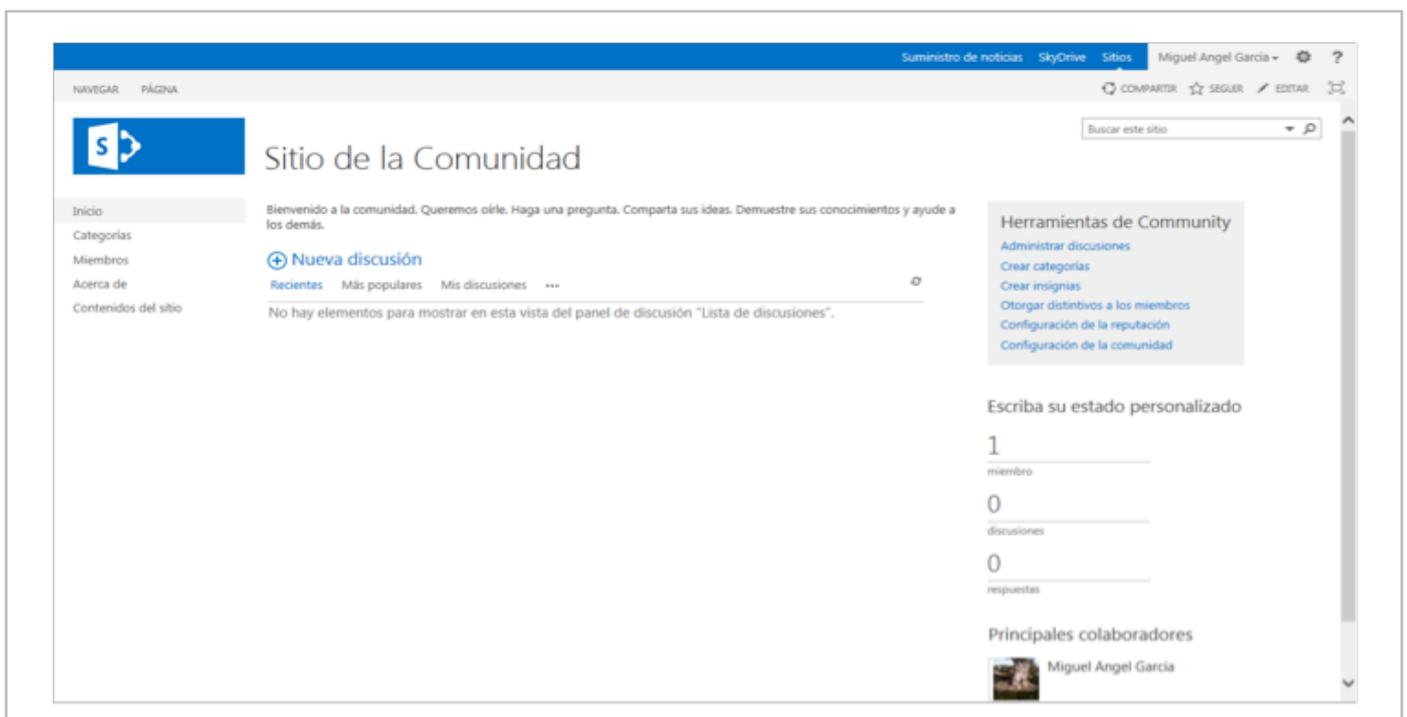


Imagen 1.- Sitio de la comunidad en SharePoint 2013.

...Cuando un administrador marca una discusión como destacada aparecerá dentro del elemento web Discusiones destacadas en la página...

Bibliotecas y Listas que se crean con esta plantilla

Listas / Bibliotecas		
Título	Plantilla	URL
Activos del sitio	DocumentLibrary	SiteAssets
Biblioteca de estilos (*)	DocumentLibrary	Style Library
Categorías	500	Lists/Categorías
Lista de discusiones	DiscussionBoard	Lists/Lista de discusiones
Miembros de la comunidad	880	Lists/Miembros
Páginas del sitio	WebPageLibrary	SitePages
Plantillas de formulario (*)	DocumentLibrary	FormServerTemplates
AbuseReports (oculta)		AbuseReports
Distintivos (oculta)		Distintivos

(*)Esta plantilla de sitio puede utilizarse en la creación de una colección de sitios (Site) o bien como un subsitio (Web) dentro de una colección. Estas bibliotecas serán creadas si se trata de una colección de sitios.

Como puede apreciarse, la estructura básica consta de tres listas que se relacionan entre sí: Categorías, Lista de discusiones y Miembros de la comunidad. La lista Categorías se basa en el tipo de contenido del mismo nombre que nos permite la utilización de cuatro columnas de datos:

Nombre columna	Tipo	Observaciones
Nombre de la categoría	Una línea de texto	
Descripción	Una línea de texto	
Imagen de la categoría	Hipervínculo o imagen	Utilizada para la vista de mosaicos
Última publicación de	Persona o grupo	

Esta lista tiene dos vistas:

- Vista de administrador (AdminView.aspx).
- Mosaicos de categoría (CategoryTiles.aspx). Esta vista utiliza la columna Imagen de la categoría para mostrar las categorías en modo mosaico.

La lista Miembros de la comunidad se basa también en un tipo de contenido con el mismo nombre que utiliza dos columnas de datos:

Nombre columna	Tipo	Observaciones
Distintivo preferencial	Búsqueda	(Relacionada con la lista oculta Distintivos)
Miembro	Persona o grupo	

Esta lista tiene 5 vistas:

- Vista de miembros (MembersAllItems.aspx).
- Principales colaboradores (TopContributors.aspx).
- Nuevos miembros (NewMembers.aspx).
- Vista de administrador (AdminView.aspx).
- Vista de un solo miembro (SingleMemberView.aspx).

Por otra parte la Lista de discusiones se basa en dos tipos de contenido (similar a una DiscussionBoard de SharePoint 2010): Discusión y Mensaje generando las siguientes columnas de datos:

...la estructura básica consta de tres listas que se relacionan entre sí...

Nombre columna	Tipo	Observaciones
Asunto	Una línea de texto	Discusión, Mensaje
Categoría	Búsqueda	Discusión (Relacionada con Categorías)
Clasificación (0-5)	Clasificación (0-5)	Discusión, Mensaje
Cuerpo	Varias líneas de texto	Discusión, Mensaje
Número de clasificaciones	Número de clasificaciones	Discusión, Mensaje
Numero de Me gusta	Número de Me gusta	Discusión, Mensaje
Pregunta	Sí o No	Discusión
Modificado por	Persona o Grupo	Discusión, Mensaje
Autor del elemento primario	Persona o Grupo	Mensaje
Última respuesta de	Persona o Grupo	Discusión

Esta lista tiene 5 vistas:

- Encadenada (Threaded.aspx).
- Plana (Flat.aspx).
- Asunto (AllItems.aspx).
- Administración (Management.aspx).
- Discusiones destacadas (FeaturedItems.aspx)

Cuando un administrador marca una discusión como destacada aparecerá dentro del elemento web Discusiones destacadas en la página Categoría.aspx.

Páginas

Además de las tres listas anteriores se crea una biblioteca de páginas con las siguientes páginas de elementos web:

- Categoría.aspx
- Acerca de.aspx
- Miembros.aspx
- Página principal de la comunidad.aspx
- Tema.aspx
- Categorías.aspx

Cada una de ellas tiene insertados determinados elementos web que permiten la correcta gestión del sitio. Se detallan a continuación dichos elementos web en cada una de las páginas.

Elementos web	Título	Lista
Página: SitePages/Categoría.aspx		
XsltListViewWebPart		Categorías
XsltListViewWebPart		Lista de discusiones
XsltListViewWebPart	Discusiones destacadas	Lista de discusiones
CommunityAdmin.webpart	Herramientas	
XsltListViewWebPart		Miembros
Página: SitePages/Acerca de.aspx		
AboutUs.webpart	Acerca de esta comunidad	
CommunityAdmin.webpart	Herramientas	
CommunityJoin.webpart	Unirse	
Dashboard.webpart	Escriba su estado personalizado	
XsltListViewWebPart		Miembros
BlogLinksWebPart		Lista de discusiones
Página: SitePages/Miembros.aspx		
XsltListViewWebPart		Miembros
CommunityAdmin.webpart	Herramientas	
CommunityJoin.webpart	Unirse	
MyMembership.webpart	Mi pertenencia	
Página: SitePages/Página principal de la comunidad.aspx		
XsltListViewWebPart		Lista de discusiones
CommunityAdmin.webpart	Herramientas	
CommunityJoin.webpart	Unirse	
Dashboard.webpart	Escriba su estado personalizado	
XsltListViewWebPart		Miembros
Página: SitePages/Tema.aspx		
XsltListViewWebPart		Lista de discusiones
CommunityAdmin.webpart	Herramientas	
Página: SitePages/Categorías.aspx		
XsltListViewWebPart		Categorías
CommunityAdmin.webpart	Herramientas	
Dashboard.webpart	Escriba su estado personalizado	
XsltListViewWebPart		Miembros

A continuación se muestran dos imágenes correspondientes a las páginas “Acerca de.aspx” y “miembros.aspx” donde se aprecia cómo se distribuyen los elementos web dentro de las zonas. Se han señalado en rojo los elementos web propios de esta plantilla de sitio.



Imagen 2.- Elementos web en la página “Acerca de.aspx”.

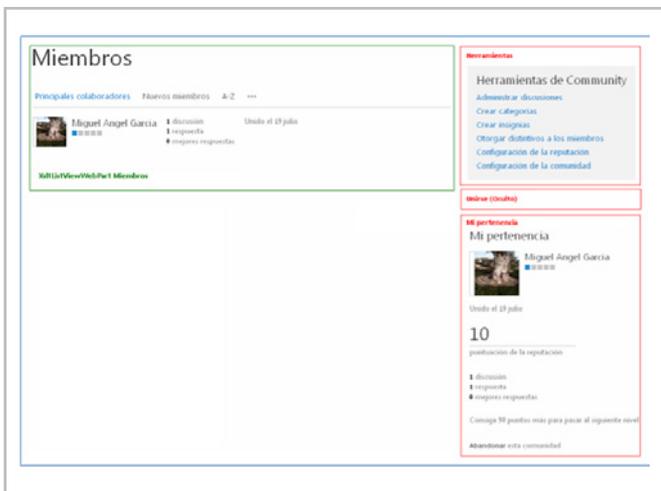


Imagen 3.- Elementos web en la página “miembros.aspx”.

Elementos web

A continuación se describen los elementos web específicos de un sitio de la comunidad. Funcionarán en los sitios de comunidad o en cualquier otro sitio que tenga activadas las funciones de comunidad.

Elemento web	Descripción
Acerca de esta comunidad	Este elemento web muestra la descripción de la comunidad y otras propiedades, como la fecha establecida.
Herramientas	Proporciona a los propietarios y administradores de la comunidad vínculos rápidos a páginas de configuración comunes y listas de contenido para administrar un sitio de comunidad.
Unirse	Permite unirse a la comunidad a aquellos que no son miembros de un sitio de comunidad. El botón se oculta si el usuario ya es miembro.
Escriba su estado personalizado	Muestra el número de miembros, temas y respuestas que hay dentro de un sitio de comunidad.
Mi pertenencia	Muestra la información de reputación y pertenencia correspondiente al visitante actual de un sitio de comunidad.

El elemento web Herramientas es una serie de vínculos que permite a los administradores desarrollar las siguientes tareas:

- Administrar discusiones. Vínculo a la vista “Management.aspx” que permite al administrador la gestión de la lista de discusiones. Adicionalmente permite marcar

como destacadas ciertas discusiones desde la pestaña Moderación.

- Crear categorías. Vínculo a la vista “AdminView.aspx” que permite agregar o modificar categorías.
- Crear Insignias. Vínculo a la tabla oculta “Distintivos” que permite agregar o modificar los distintivos.
- Otorgar distintivos a los miembros. Vínculo a la vista “AdminView.aspx” que permite desde la pestaña Moderación asignar un Distintivo a un miembro.
- Configuración de la reputación: Vínculo a la página del sistema “CommunityReputationSettings.aspx” que permite:
- Especificar si se pueden clasificar los elementos de esta lista o no.
- Elegir “Me gusta” o “Clasificación por estrellas” como la forma de clasificación del contenido.

• Habilitar sistema de puntos de logros de los miembros, especificando los valores otorgados a los miembros para una serie actividades.

• Especificar el número de puntos que necesitan los miembros para llegar a cada nivel de logro.

• Especificar si los niveles de logro aparecen como una serie de cuadros o como un título de texto que se puede personalizar.

• Configuración de la comunidad: Un vínculo a la página del sistema “CommunitySettings.aspx” que permite al administrador:

- Cambiar la fecha de inicio de la comunidad.
- Habilitar la aprobación automática para miembros que desean unirse a la comunidad.

o Permitir a los miembros de la comunidad informar de contenido inapropiado, que los administradores pueden revisar y quitar.

- Revisar entradas notificadas. Vínculo a la página entradas notificadas.aspx que permite la gestión de las entradas que han sido notificadas como contenido inapropiado.

Características

Por último analizamos las características que se activan exclusivamente con estas plantillas de sitio:



Imagen 4.- Característica de sitio de la comunidad.

Feature	ID
CommunitySite	961d6a9c-4388-4cf2-9733-38ee8c89afd4
AbuseReportsList	c6a92dbf-6441-4b8b-882f-8d97cb12c83a
MembershipList	947afd14-0ea1-46c6-be97-dea1bf6f5bae
CategoriesList	d32700c7-9ec5-45e6-9c89-ea703efca1df

MIGUEL ÁNGEL GARCÍA RODRÍGUEZ

Analista SharePoint

migangarrodd@hotmail.com

http://es.linkedin.com/pub/miguel-angel-garcia-rodriquez/34/481/108

i

59

Configurando el acceso a usuarios externos en nuestro sitio de colaboración de SharePoint Online

Resumen

En este breve artículo veremos cómo configurar una instancia de SharePoint Online de Office 365 para soportar el acceso a usuarios externos a nuestros sitios de colaboración.

Requisitos Previos

Es increíble el poder que tenemos en nuestras manos para dar soluciones técnicas a problemas de negocio de una forma rápida y sencilla. Una extranet es aquel tipo de portal que permite el acceso a personas externas a la organización para soportar un escenario de colaboración entre usuarios internos y externos. Con SharePoint Online de Office 365 podemos otorgar el acceso a usuarios externos mediante el uso de cuentas de Windows Live permitiéndonos crear una solución de extranet para nuestra empresa o clientes.

Veamos cómo podemos configurar una instancia de SharePoint Online de Office 365 para permitir la autenticación de usuarios externos con cuentas de Microsoft (Windows Live) en nuestro sitio de colaboración de SharePoint.

Supuestos

- De manera pre-definida se tiene apagado el soporte a cuentas externas.
- Se necesita acceder a la herramienta central de administración de SharePoint Online en Office 365.

Paso 1: Accedemos a la herramienta central de administración de SharePoint y seleccionamos la colección de sitios donde queremos soportar acceso a usuarios externos mediante autenticación con Windows Live.

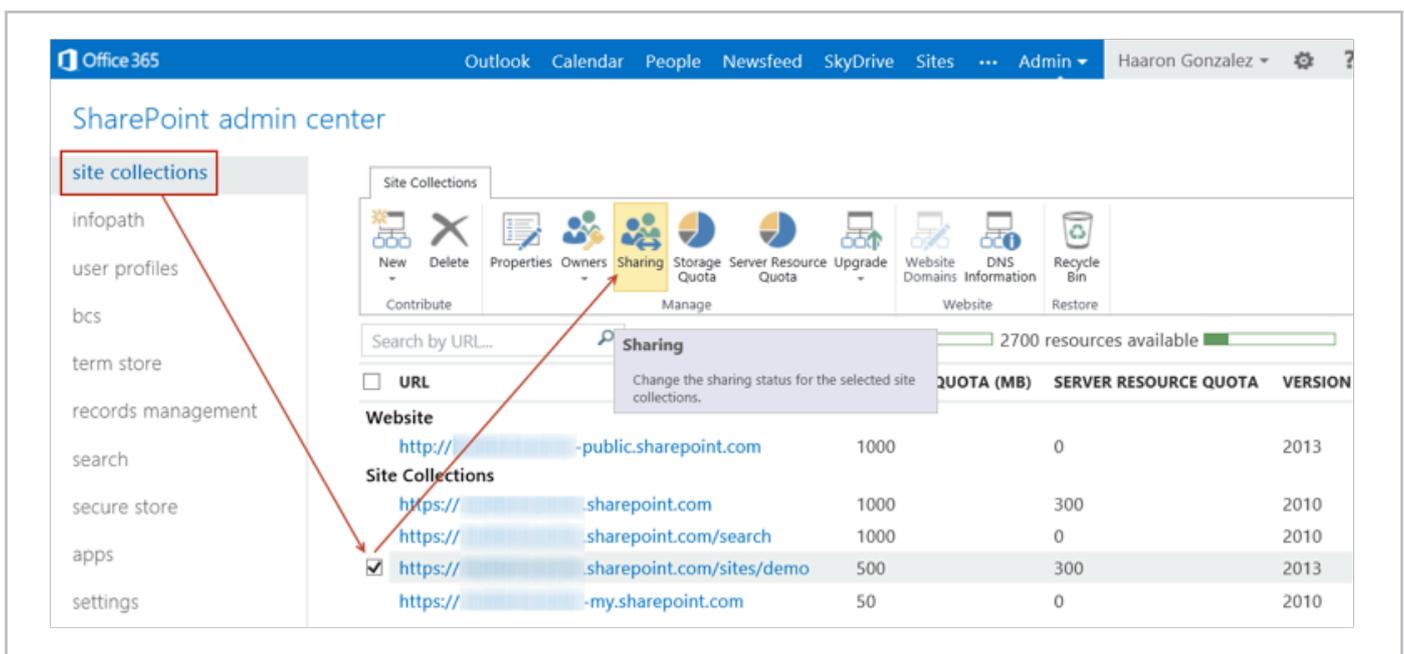


Imagen 1.- Configuración del acceso a usuarios externos en un tenant de Office 365.

...elegimos Microsoft Account y procedemos a autenticarnos con nuestra cuenta de Windows Live...

Paso 2: Seleccionamos como los usuarios podrán invitar a otros usuarios externos a acceder a nuestra colección de sitios.

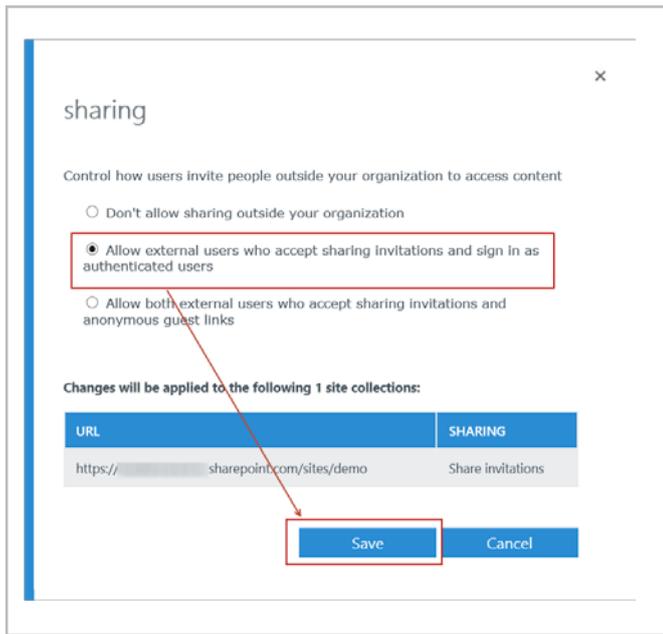


Imagen 2.- Detalle de configuración del acceso a usuarios externos en el tenant de Office 365.

Paso 3: Accedemos a nuestra colección de sitios y compartimos con algún externo.

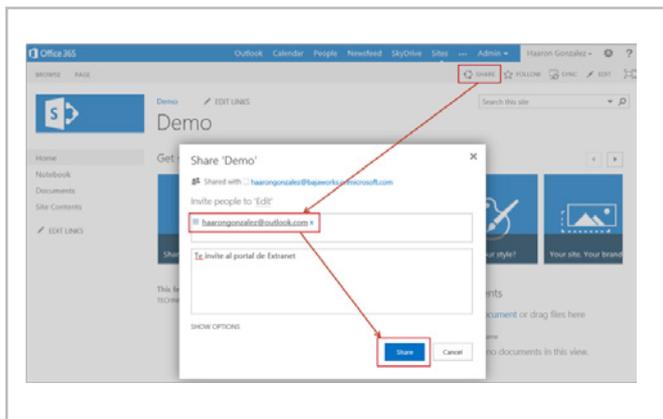


Imagen 3.- Invitación a un usuario externo en un sitio de SharePoint Online.

Un mensaje de Sistema nos informa que se ha compartido el sitio con un usuario externo.

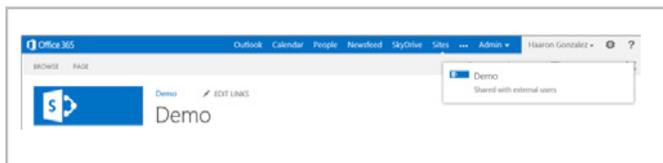


Imagen 4.- Mensaje informativo indicando que el sitio ha sido compartido.

Paso 4: El usuario externo recibirá un correo electrónico de bienvenida al sitio en su cuenta de Windows Live y procedemos a autentificarnos.

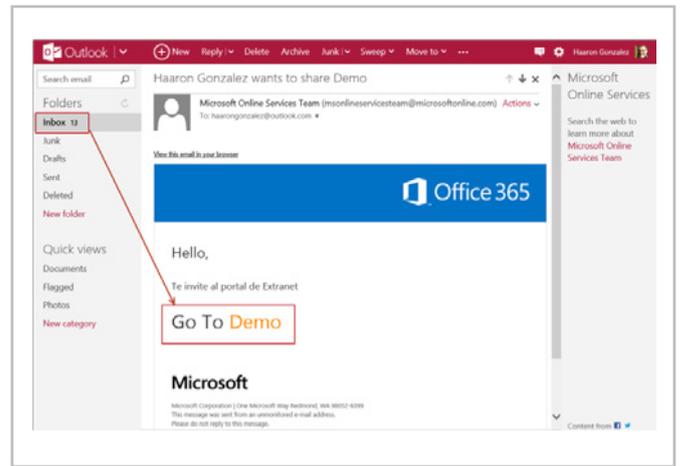


Imagen 5.- Mensaje de correo electrónico enviado al usuario externo

Al dar clic en el sitio de colaboración una página de bienvenida al servicio de SharePoint Online nos muestra los tipos de cuentas con las que nos podemos firmar. Obviamente elegimos Microsoft Account y procedemos a autentificarnos con nuestra cuenta de Windows Live.

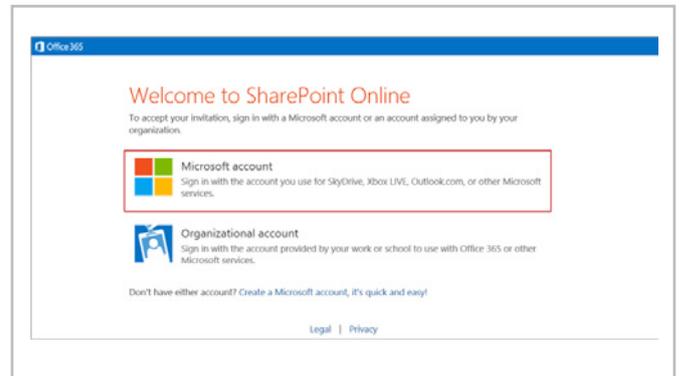


Imagen 6.- Acceso al sitio compartido por medio de un Live Id.

HAARON GONZALEZ

MVP SharePoint

haarongonzalez@mvps.org

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

<http://msmvps.com/blogs/haarongonzalez>

<http://youtube.com/haarongo>



Como configurar el Content Type Hub de SharePoint 2010

Resumen

En la mayoría de las organizaciones en las que se implanta SharePoint, en especial en las grandes organizaciones, lo normal es que se tenga la necesidad de definir más de una colección de sitios para cubrir los requerimientos de la organización. En estos casos será necesario que los tipos de contenidos definidos sean accesibles desde todas las colecciones de sitios existentes. En el siguiente artículo se verá que funcionalidad aporta SharePoint 2010 para llevar a cabo esto y cómo configurarla.

Artículo

En la mayoría de las organizaciones en las que se implanta SharePoint, en especial en las grandes organizaciones, lo normal es que se tenga la necesidad de definir más de una colección de sitios para cubrir los requerimientos de la organización. En estos casos será necesario que los tipos de contenidos definidos sean accesibles desde todas las colecciones de sitios existentes.

Hasta SharePoint 2007 no era posible hacer lo descrito anteriormente de una manera cómoda por lo que se hacía necesario definir todos los tipos de contenidos en todas las colecciones de sitios, bien a través de la interfaz de usuario o bien realizando un desarrollo que permitiera duplicar los tipos de contenidos en todas las colecciones de sitios a partir de las definiciones hechas en algunos de ellos. En ambos casos se tendrán problemas derivados de la complejidad del mantenimiento. En el caso del desarrollo, además, la complejidad es inherente al mismo.

Content Type Hub

A partir de SharePoint 2010 se incluye "Out Of The Box" una característica denominada Content Type Hub que, haciendo uso del servicio de metadatos administrados, permite definir una localización en la que se centralizan las definiciones generales de las columnas de sitio y de los tipos de contenidos y gestionar la réplica de los mismos en las colecciones de sitios de la granja.

En el presente artículo se verán las características generales de ésta y cómo se realiza la configuración básica de la misma en una granja con múltiples colecciones de sitios. Para ello, se ha de suponer que previamente se

ha configurado correctamente el servicio de metadatos administrados ya que, aunque necesario, no entra dentro del objetivo del presente artículo.

Definición del Content Type Hub

Lo primero que se debe hacer es crear una nueva aplicación web sobre la que se creará posteriormente la colección de sitios que representará el "Content Type Hub". Se podría utilizar cualquier colección de sitios existente para definir el "Content Type Hub" pero de cara a una mejor estructuración, configuración y mantenimiento de los tipos de contenidos se recomienda que la colección de sitios esté aislada y sea utilizada exclusivamente para contener la definición de este.

Una vez creada la aplicación web, se creará sobre ella la nueva colección de sitios que, como se dijo anteriormente, contendrá la definición del Content Type Hub. Se propone poner como nombre del sitio raíz de la colección "Content Type Hub" o "Concentrador de Tipos de Contenidos", si se prefiere utilizar un símil en castellano, de forma que sea fácilmente identificable. Para el sitio principal de la colección de sitios se recomienda utilizar al menos la plantilla de Team Site o Sitio de Grupo.

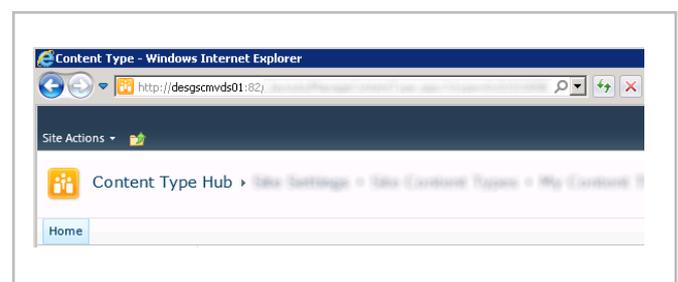


Imagen 1.- Sitio de grupo del Content Type Hub.

Este requerimiento es debido a que es necesario que se activen una serie de características relacionadas con el servicio de metadatos administrado y que, por ejemplo, si se selecciona una plantilla en blanco no se activan dichas características. Debe hacerse hincapié en que las características mencionadas no se lista en las características de sitio ni de colección de sitio y, por tanto, de ahí la necesidad de escoger al menos la plantilla indicada. Sin embargo, esta situación es reversible ya que es posible

activar desde PowerShell las características implicadas por medio de su GUID en caso de que se haya seleccionado una plantilla errónea accidentalmente. En concreto se refiere a la característica TaxonomyFieldAdded

Ahora que ya se ha creado la colección de sitios, se debe acceder a la administración central de SharePoint 2010 para configurar los parámetros que indiquen que la nueva colección de sitios es el Content Type Hub. Para realizar esto es necesario acceder a la sección de administración de aplicaciones de servicio y editar las propiedades de la aplicación de servicios de metadatos administrados.

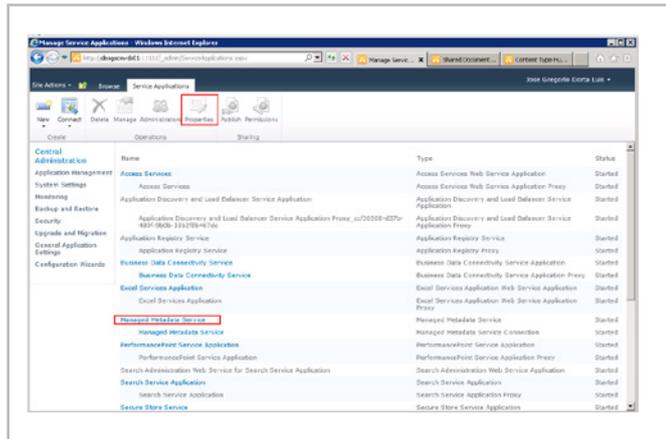


Imagen 2.- Acceso a las propiedades de la aplicación de servicio de metadatos administrados.

En este punto aparecerá un formulario en el que se deberá indicar la URL de la colección de sitio que definirá el Content Type Hub en el apartado indicado para ello. Una vez configurado, si se editan las propiedades, se verá como en la siguiente figura.

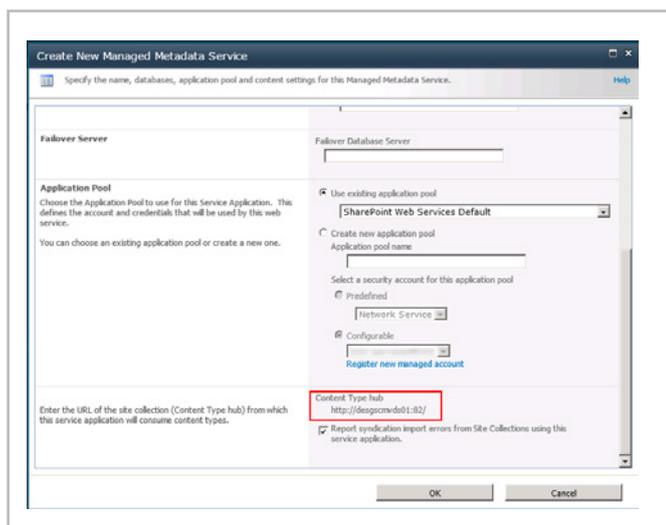


Imagen 3.- Configuración del Content Type Hub.

Posteriormente se deberán editar las propiedades del proxy de conexión de la aplicación de servicio de metadatos administrados para indicar que se requiere que los tipos de contenido sean accesibles desde la galería de tipos de contenidos existente en la colección de sitios indicada para el Content Type Hub y que se permita la publicación de los

mismos y sus actualizaciones desde esa galería en todos los sitios y subsitios existentes en el resto de colecciones de sitios.

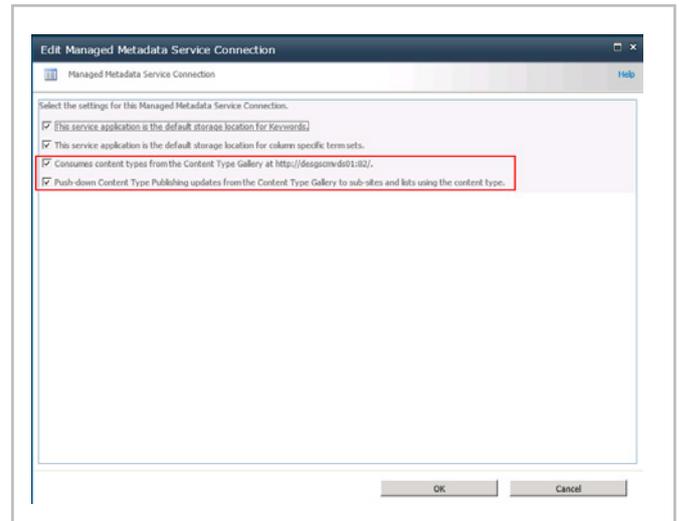


Imagen 4.- Propiedades del proxy de la aplicación de servicio de metadatos administrados.

Publicación de tipos de contenidos

Una vez se han realizado los pasos anteriores se está en disposición de acceder al sitio raíz de la colección de sitios del Content Type Hub y comenzar a definir las columnas de sitio y los tipos de contenidos que se hayan considerado en la organización. En la configuración de cada tipo de contenido definido se encontrará una opción que permite publicar, o republicar si ya fue publicado con anterioridad, dicho tipo de contenido en el resto de colecciones de sitios existentes.

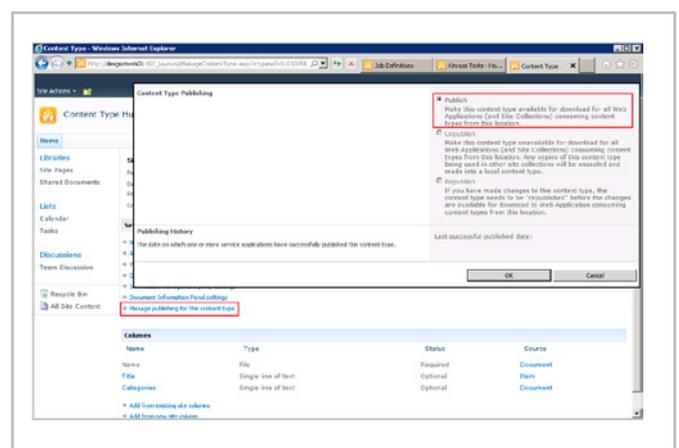


Imagen 5.- Configuración de publicación de un tipo de contenido en el Content Type Hub.

Posteriormente se deberá ir a la configuración del sitio y, una vez ahí, ir a la sección de administración de la colección de sitios. En ella se encontrará una opción denominada "Publicación de los tipos de contenidos" en la que se deberá indicar que se quiere que se refresquen todos los tipos de contenidos en la siguiente actualización.

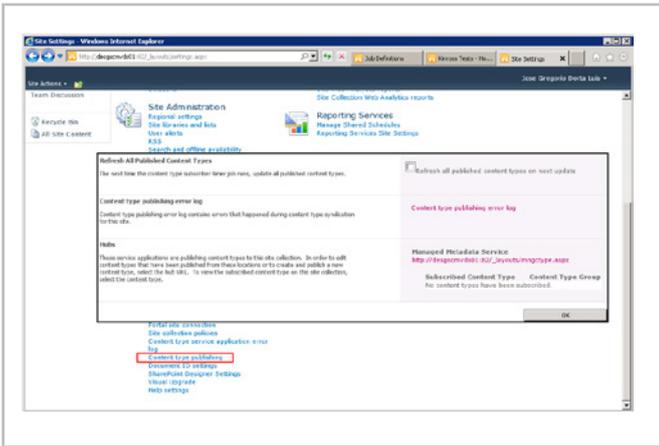


Imagen 6.- Configuración de la publicación de los tipos de contenido.

En esta misma pantalla se podrá comprobar lo errores producidos a la hora de la publicación de los tipos de contenidos estando dicha opción disponible en todas las colecciones de sitio, independientemente de que se trate del Content Type Hub. Además, en el caso de las colecciones de sitios que se suscriben al Content Type Hub mostrará los tipos de contenidos que han sido publicados en dichas colecciones de sitios como se puede comprobar en la figura anterior.

Trabajos del temporizador (Timer Jobs)

Para que se lleve a cabo la publicación de los tipos de contenidos en las diferentes colecciones de sitios, el Content Type Hub se apoya en dos trabajos del temporizador o Timer Jobs:

- Content Type Hub.
- Content Type Suscriber.

Se encargan de la gestión de la publicación y mantenimiento de los tipos de contenidos entre las distintas colecciones de sitios existentes. En el caso del Content Type Suscriber existirá uno por cada aplicación web existente en la granja de SharePoint de la organización.

Content Type Hub		Minutos
Content Type Subscriber	SharePoint - 00	Hourly
Content Type Subscriber	SharePoint - 01	Hourly
Content Type Subscriber	SharePoint - 02	Hourly

Imagen 7.- Trabajos de temporizador para Content Type Hub.

Por lo tanto, si se desea que los tipos de contenidos publicados estén accesibles de forma inmediata en alguna de las colecciones de sitios se deberán ejecutar estos trabajos de forma manual, sin esperar a la planificación establecida en el temporizador. Si no se ejecutan por primera vez, es suficiente con ejecutar el Content Type Suscriber de la aplicación web en el que se encuentre la colección de sitios en la que se quieren publicar los tipos de contenidos.

...la publicación en el Content Type Hub serán los trabajos del temporizador los encargados de liberar las copias...

De esta forma se está en disposición de acceder a una de las colecciones de sitios y comprobar que se ha publicado el tipo de contenido. Este se publica originalmente siempre en modo de sólo lectura. Este modo puede ser alterado si se necesita que el tipo de contenido en esa colección de sitios deba tener alguna característica diferenciada del resto.

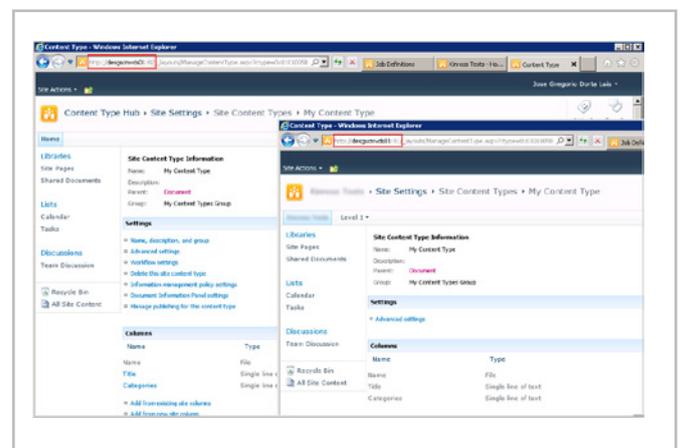


Imagen 8.- Comprobación de funcionamiento del Content Type Hub.

Habrà que tener en cuenta que estas personalizaciones sólo se verán en la colección de sitios en la que se realizó y no se trasladan al Content Type Hub y que es posible que, una vez realizadas modificaciones concretas, las actualizaciones del tipo de contenido en el Content Type Hub no se publiquen correctamente.

En el caso de que se elimine un tipo de contenido o se finalice la publicación en el Content Type Hub serán los trabajos del temporizador los encargados de liberar las copias del mismo publicados en el resto de colecciones de sitios. Esto implica cambiar su estado de solo lectura a escritura y establecer la definición del tipo como local a la colección de sitios.

JOSÉ GREGORIO DORTA LUIS
Desarrollador de SharePoint
General de Software
 @grydl

Preparados, Listos, Ya!



Migración



Governance



Calidad de datos



Mobilidad



Productividad



Su **Camino** hacia el **éxito** a
Microsoft[®] SharePoint[®] 2013
con **AvePoint**

Para saber más, visite
eu.avepoint.com/sharepoint-2013/

Puede ponerse en contacto con AvePoint para **solicitar una demo** con uno de **nuestros expertos** a través de email Ventas@AvePoint.com o teléfono 91 123 4650

Clonando SharePoint 2010

Resumen

Existen varias formas de enfrentarnos a la instalación de SharePoint, aquí pretendemos revisarlas y mostrar un caso empírico de instalación por duplicidad o clonación.

Artículo

A la hora de instalar SharePoint disponemos de varias opciones, pero: ¿cuál de ellas es la más apropiada? La instalación a través de la interfaz gráfica puede parecer sencilla y rápida aunque en ciertos escenarios puede ser poco eficiente. Revisemos que opciones tenemos y cómo podemos rentabilizar nuestro tiempo y esfuerzo.

Podríamos distinguir las siguientes modalidades a la hora de realizar una instalación:

- Instalación clásica (interfaz gráfica).
- Por medio programación (scripting PowerShell).
- Utilizando clonación.

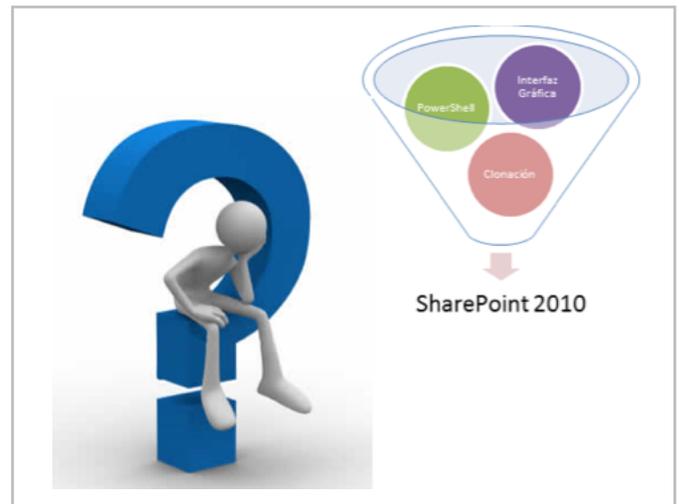


Imagen 1.- Posibilidades de instalación de SharePoint 2010

¿Qué factores podemos distinguir a la hora de seleccionar un modo u otro? Dependiendo de qué situación, el criterio de selección puede basarse en uno u otro aspecto de la instalación. Intentemos recoger los aspectos más notorios de cada proceso:

Método	Ventajas	Desventajas
Instalación clásica (interfaz gráfica)	Intuitivo, sencillo	Mucho tiempo de instalación, ejecución de cada paquete por servidor. Genera largos GUIDs en las Bases de datos. El tiempo de instalación no mejora con el número de instalaciones
Por programación (scripting PowerShell)	Personalización completa en la definición de nombres de bases de datos	Control total del script. El tiempo de instalación mejora sensiblemente entre la primera instalación y el resto.
Clonación	Perfecto para entornos virtuales donde podemos jugar con Snapshots.	Solo aplicable para instalaciones “todo en uno” o standalone. Muchos pasos, proceso detallado y ejecución minuciosa de cambios. Buena planificación y coordinación entre equipos.

Además de estos criterios nuestra experiencia marcará decisivamente en muchas ocasiones el método de instalación y aquí es donde animo a probar los 3 métodos de instalación propuestos en esta guía, pues nunca se sabe cuándo puedes precisarlos y cuan ventajosos

pueden ser en según qué situación. En esta ocasión abordare el método de clonación, los otros son mucho más conocidos y en el caso de la instalación clásica muy intuitivo y quizás uno de los más utilizados, ¿os suena el “next next next”?



Comenzamos: Clonación de una instalación de SharePoint Server 2010 Enterprise

Os voy a proponer un escenario para que se vean claramente las opciones, por su puesto las variables son infinitas, como tu imaginación.



Imagen 2.- Punto de partida para clonar una instalación de SharePoint 2010.

El escenario de partida cuenta con los siguientes componentes y elementos:

- Virtualización bajo el sistema Hyper-V.
- Máquina virtual creada con el sistema operativo Windows 2008 R2.
- Instalación en la máquina virtual de SQL 2008 R2 (instancia por defecto), una instalación limpia de SharePoint 2010 + SP1 + 2 paquetes de idioma + software extra como Visual Studio, Office, SharePoint Designer, etc.

- Todos los paquetes de seguridad aplicados.

Y las siguientes aplicaciones de servicio de SharePoint creadas y configuradas:



Imagen 3.- Aplicaciones de servicio creadas y configuradas.

Ya tenemos todos los ingredientes para comenzar nuestra aventura. A esta máquina vamos a llamarla la maquina master. La guardamos y utilizándola como plantilla creamos tantas maquinas como necesitemos. Si aplicas las direcciones IP correspondientes, una por servidor, y apagas el servidor master, te ahorraras confusiones. Aquí empieza nuestro trabajo como administradores de SharePoint!

Vamos a dividir las tareas en 2 fases de configuración y una tercera de validación.

- A. Antes de cambiar el nombre del servidor.
- B. Después de cambiar el nombre del servidor.
- C. Validación de los nuevos servidores.

Mostraré los pasos de forma muy estructurada para evitar confusión. Ahora necesitarás:

- La cuenta de la granja y la contraseña.
- El servicio "SecurityToken Service"; la frase de paso en caso de que se haya configurado alguna clave (ejemplo para Visio, Excel, etc.).

A. Antes de cambiar el nombre del servidor

Los pasos a seguir serán:

1. Añadir una nueva entrada en el Alternate Access Mapping (AAM) con el nuevo nombre de servidor.
2. Añade este nombre de servidor en la lista de backconnectionhostname en el registro (así evitaras el 401 de loopback check).
3. Cambia el nombre de la base de datos de configuración. Abre una consola PowerShell y ejecuta:
`"Stsadm-orenameserver-newservername{YourNewServerName}-oldservername {YourOldServerName}"`
4. Ahora toca cambiar el nombre del servidor (Para Windows Server en el idioma inglés: Start->Administrative Tools->Server Manager. From the Server Summary section, select "Change System Properties." Click the "Change" button from the "Computer Name). Esto requerirá un reinicio de máquina.
5. Ejecución del Sysprep.

B.Después de cambiar el nombre del servidor

Accede a la máquina virtual e intenta abrir la Administración Central. Si recibes un error del tipo "503 service unavailable" tendrás que cambiar la cuenta que utiliza la pool de aplicación de la central a "localserver". Aplica los cambios, recicla la pool y pon la cuenta original (granja). Ahora debería de poder acceder.

Deberás aplicar el mismo procedimiento para la cuenta que utiliza la pool del servicio Security Token ("SecurityTokenServiceApplicationPool").

Ahora ya deberías de poder acceder. Los siguientes pasos serian:

1. Eliminar la entrada antigua del Alternate Access Mapping (AAM).
2. Verifica que el servidor de la base de datos de configuración aparece en "servers in farm".
3. Arranca el servicio Windows "Claims to windows Token Service". Esto es debido al error de evento ID 8306.
4. Configuración SQL; asegura que has cambiado tanto el nombre como la IP en la configuración SQL. En segundo caso solo afectara para conexiones externas, pero nunca se sabe. (SQL server configuration manager > SQL server Network configuration: protocols: TCP/IP: Ip addresses). Recuerda que los cambios no se aplicaran a menos que reinicies la instancia.
5. Alias SQL: desde el cliconfg.exe (existen las 2 opciones en los servidores 64bits) aplica el nuevo nombre del servidor.
6. Los certificados del servicio "Security Token Service". Son 3 y tendrás que exportarlos de la maquinas master e importarlos en el resto.
7. Aprovisionamiento de la configuración general de las aplicaciones web. Desde una consola de SharePoint 2010 Management Shell ejecuta los siguientes comandos, tardarán un poco:

```
$shc = Get-SPServiceHostconfig
$shc.Provision()
```

8. Aprovisiona todos los servicios:

```
$services = Get-SPServiceApplication
foreach ($service in $services) { $service.provision();
write-host $service.name
```

9. Reinicio el Internet Information services:

```
Ilsreset /noforce
```



C. Validación de los nuevos servidores

Son muchas particularidades a tener en cuenta cuando duplicamos un servidor SharePoint, es fácil dejarse algo olvidado, por ello es importante revisar algunas herramientas que nos permitirán conocer el estado general de la instalación. Disponemos del log de eventos propio del servidor, el Health Analyzer de SharePoint, además probaremos las aplicaciones de servicio y la creación de una nueva aplicación web con una colección de sitios. Así aseguramos que todo responda de forma correcta.

Son muchos pasos pero ha sido divertido lograrlo, ¿te atreves a probarlo?

Conclusión

SharePoint es una plataforma multifuncional, un producto con una variedad de usos muy interesante y a la hora de instalarlo también disponemos de distintas opciones, es interesante conocerlas y atreverse con ellas, animate.

VERÓNICA TORANZO

MCITP SharePoint

veronica.toranzo.martinez@gmail.com

Deberás aplicar el mismo procedimiento para la cuenta que utiliza la pool del servicio Security Token. ("SecurityTokenServiceApplicationPool").



68

Cómo se hizo la Apps OTic-Visor de Lista de OnlineTic

Resumen

La App de "OTic - Visor de Lista" fue creada por la empresa OnlineTic SL. Se realizó dentro del Programa que lanzó Microsoft a distintos proveedores y nuestro equipo de I+D de OnlineTic, coordinado por el Arquitecto especialista de SharePoint José Antonio Fraga aceptó el reto.

Nuestra experiencia en múltiples proyectos de desarrollo y consultoría de diferentes tipos, tanto de Web Públicas, Intranets, Extranets y aplicaciones cliente, nos ha llevado a realizar una reingeniería de los desarrollos para finalizar en lo que llamamos el "Framework de OnlineTic". Este Framework nos permite ofrecer a nuestros clientes el desarrollo de aplicaciones con menor coste en tiempo y dinero.

Artículo

Los elementos del Framework de OnlineTic cubren diferentes escenarios de soluciones de aplicaciones y webs para la plataforma de SharePoint. Hemos definido numerosos elementos genéricos de forma que nos permita en el menor tiempo posible personalizar las necesidades de nuestros clientes lo que nos permite reducir hasta un 40% los costes para desarrollar aplicaciones bajo tecnología SharePoint.

Con la salida de la versión 2013 de SharePoint y con el nuevo modelo de Aplicaciones decidimos migrar una parte del Framework con dos principales objetivos:

- Aprender la nueva metodología de desarrollo, con vistas a dar mejor servicio a nuestros clientes.
- Estudiar la viabilidad de realizar productos para venderlos a nivel mundial dentro de la Store de Microsoft como nueva línea de negocio.

Aplicaciones

OTic-Visor de Lista

Publicado: 28/06/2013

Precio: GRATIS

Clasificación: ★★★★★(0)

Producto: SharePoint Server 2013 Enterprise

OTic- Visor de Lista es una WebPart de visualización de listas que permite al usuario final su personalización y configuración para proveerla de botones de acción sobre cada uno de los registros, funcionalidad no contenida en las listas de SharePoint por defecto.

Lo primero que hicimos fue migrar lo que nosotros denominamos el "OTic- Visor de Lista", que resumido en pocas palabras lo que hace es pintar un listado proveniente de la combinación "lista- vista" con el añadido de poder incluir botones de acción sobre cada registro. Esto nos es útil por ejemplo para ir a una página de detalle del elemento, lanzar procesos, realizar llamadas a "Web Services", ir a aplicaciones dentro o fuera de SharePoint mandando los parámetros oportunos. Además quisimos ofrecer filtros y la posibilidad de ordenar los registros para darle una mayor funcionalidad:

ARTÍCULO	ALMACÉN	CATEGORÍA	CANTIDAD	PRECIO POR UNIDAD	Buscar:
Naranjas	Verdulería	Productos alimenticios	2	2.99	[iconos]
Manzanas	Huerto	Productos alimenticios	3	1.99	[iconos]
Plátanos	Verdulería	Productos alimenticios	1	3.99	[iconos]
Lechuga	Mercado	Productos alimenticios	2	2.29	[iconos]
Tomates	Mercado	Productos alimenticios	4	3.49	[iconos]
Buscar en la Wikipedia	Mercado	Productos alimenticios	2	1.5	[iconos]
Ajolo	Verdulería	Productos alimenticios	2	1.99	[iconos]
Papino	Mercado	Productos alimenticios	1	2.29	[iconos]
Champiñones	Verdulería	Productos alimenticios	0.5	2.25	[iconos]
Leche	Envío a casa	Nata líquida	2	3.99	[iconos]
Queso	Envío a casa	Nata líquida	1	9.99	[iconos]
Huevos	Envío a casa	Nata líquida	2	3.5	[iconos]
Queso de cabra	Envío a casa	Nata líquida	1	3.89	[iconos]
Crema de leche	Envío a casa	Nata líquida	1	2.99	[iconos]
Yogurt	Verdulería	Nata líquida	1	4.99	[iconos]
Ternera	Mercado	Carne	10	7.99	[iconos]
Salmon salvaje	Mercado de pescado	Marisco	6	8.99	[iconos]

Imagen 1.- Solución "OTIC – Visor de lista" a migrar.

Una de las primeras decisiones a tomar fue qué tipo de Aplicación implementar:

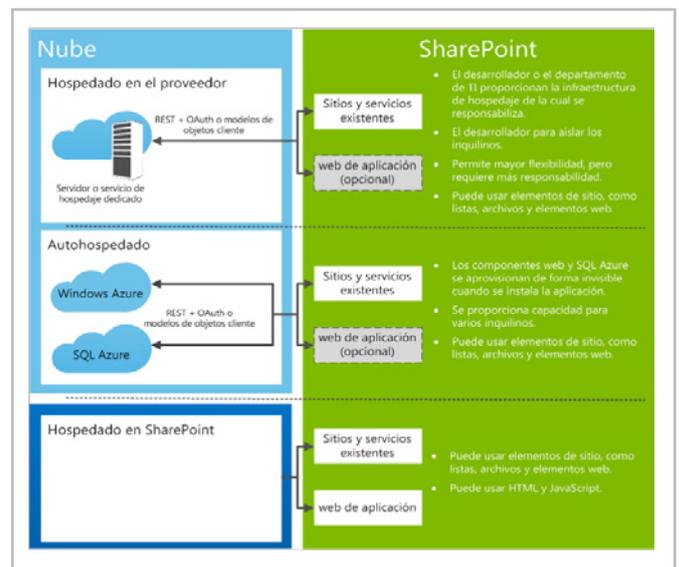


Imagen 2.- Tipos de Aplicaciones de SharePoint.

Para ello hay que tener en cuenta los requisitos de la aplicación, si se va a desarrollar en SharePoint o en otra plataforma, donde vamos a almacenar la información, etc. En nuestro caso concreto, descartamos la posibilidad de implementar una Aplicación de tipo “Hospedada en el proveedor” puesto que no queríamos tener servidores dedicados y dado que la información requerida por la APP son las listas y vistas incluidas en los sitios de SharePoint del cliente.

Para decidir entre Aplicaciones “Autohospedadas” y “Hospedadas en SharePoint”, nos decantamos por las de tipo “Hospedadas en SharePoint”, por una sencilla razón y es que las Aplicaciones “Autohospedadas” llevarán en el futuro un cargo adicional al cliente por utilizar los servicios de Azure. A cambio, todo el desarrollo tuvo que realizarse en código cliente por lo que el tiempo de desarrollo se vería incrementado frente a una Aplicación “Autohospedada”. Al decidirnos por una Aplicación “Hospedada en SharePoint” también teníamos resuelto el problema de dónde almacenar los datos de la Aplicación y es que al ser una Aplicación “Hospedada en SharePoint” automáticamente se crea un subsitio para la Aplicación, por lo que utilizaríamos este subsitio para crear las listas necesarias.

Básicamente lo que necesitamos almacenar es la configuración de cada instancia de la Aplicación, para ello, cada vez que la Aplicación se introduce en una página es necesario indicarle la lista y vista de la que se alimentará:

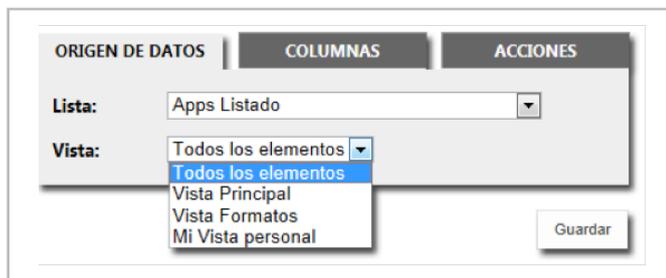


Imagen 3.- Configuración del origen de datos.

Podemos configurar también el formato de algunos campos, como por ejemplo los de tipo fecha decirle si queremos ver la fecha y hora o sólo la fecha, o los campos de tipo búsqueda si queremos ver el texto o el ID del campo al que referencia además de la posibilidad de incluir un hipervínculo al elemento enlazado.

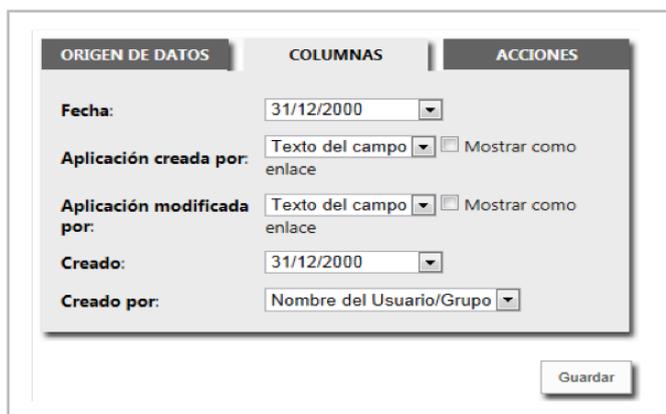


Imagen 4.- Columnas a mostrar en el visor de lista.

Y finalmente indicarle las acciones disponibles, donde podremos seleccionar alguna de las acciones predeterminadas como pueden ser “Ver elemento”, “Editar elemento”, “Flujos de trabajo”, etc; o indicarle una acción personalizada.



Imagen 5.- Configuración de las acciones

Para el caso de una acción personalizada podremos seleccionar una imagen de las mostradas en el pantallazo anterior o indicar la URL de la imagen que deseemos mostrar. El texto de la acción, la URL a la que irá la acción (podemos componer la URL a partir de las columnas de la lista, en el siguiente ejemplo lo que hará la acción es ir a una página de la Wikipedia donde la palabra buscada sea el campo “LinkTitle” del elemento en cuestión). Además podremos indicar opciones adicionales como pueden ser si queremos abrir la acción en una nueva ventana, en la misma ventana, si queremos ver la acción a la izquierda o a la derecha de la tabla, mostrar un mensaje de confirmación antes de ejecutar la acción, etc.



Imagen 6.- Configuración de una acción

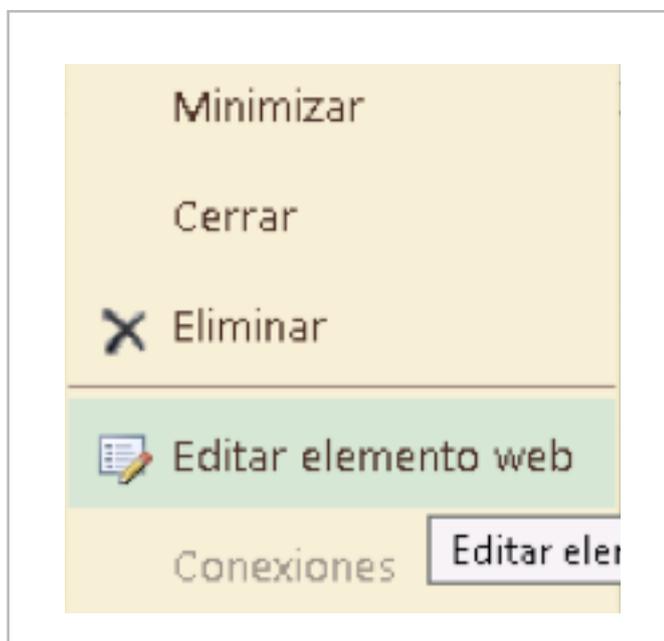
Toda esta configuración se almacena en una lista dentro del subsitio de la Aplicación, donde la configuración de cada App Part es un elemento de la lista.

Se acabó el mundo feliz, ahora empiezan los problemas ☹️ y sus Soluciones claro 😊

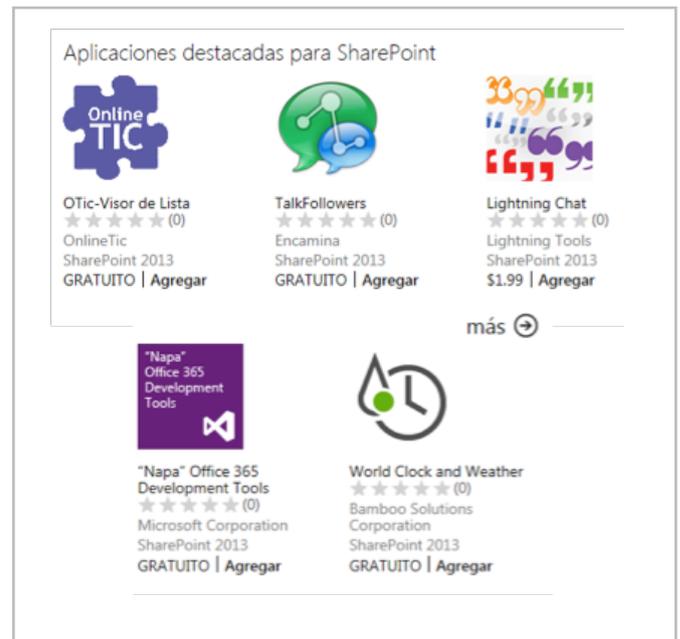
Uno de los primeros problemas que nos encontramos fue el de cómo identificar cada una de las instancias de la Aplicación puesto que cada instancia tiene su propia configuración. Para ello miramos en los Tokens de la página oficial de Microsoft (<http://msdn.microsoft.com/es-es/library/jj163816.aspx>) pero no encontramos ninguno que nos sirviera como identificador. Por lo que en una primera aproximación usamos la URL de la página en la que se encontraba la instancia de la Aplicación como identificador. Como consecuencia de esto no podíamos tener más de una instancia de nuestra Aplicación en una misma página pues cogería la misma configuración. Más adelante y después de mucho investigar, descubrimos que hay Tokens adicionales que no aparecen en la página de Microsoft y que también nos son de mucha utilidad como por ejemplo el del ID de la App Part (WPID). Usando este ID solventamos el problema y por tanto podíamos tener más de una App Part en una misma página y cada una con su configuración.

Más problemas...

Otro problema era el de la seguridad y es que no queríamos que cualquier usuario pudiera modificar la configuración de la App Part, sólo un usuario con permisos de edición sobre la página debía poder hacer modificaciones. Para este punto lo que hicimos es que las pantallas de configuración de la Aplicación mostradas anteriormente sólo estarían disponibles cuando la Aplicación se encuentra en estado de “edición” o lo que es lo mismo, cuando el IFRAME donde está contenida tiene un token denominado editMode=1, esto se consigue cuando pulsamos sobre el enlace de “Editar elemento Web” de la “WebPart”:



Al pulsar sobre “Editar elemento web” la Aplicación automáticamente recibe un token editMode=1 (en la URL del IFRAME), por lo que a la hora de hacer el desarrollo lo que hicimos es que si recibe este token muestra los parámetros de configuración y si no lo recibe directamente muestra la tabla pertinente de la “lista-vista” seleccionada. El resultado final es una Aplicación Multidioma (actualmente en Español e Inglés, próximamente será traducida a más de 5 idiomas) que se puede vender globalmente dentro de la Store de Microsoft.



Enlace a la aplicación: OTic- Visor de Lista
 Manual: <http://www.onlinetic.com/app/Oticvisor/manuales/Manual.pdf>
 Video: <http://www.onlinetic.com/app/Oticvisor/videos/Oticvisor.swf>

Seguimos trabajando...

Actualmente estamos trabajando en otros módulos del Framework, convirtiéndolos en Apps de OTic y que ya están en proceso de pruebas. Próximamente estarán en la Store!!

JOSÉ ANTONIO FRAGA SÁNCHEZ

Arquitecto SharePoint en OnlineTic

josefraga@onlinetic.es

Linkedin

<http://www.onlinetic.es>

...Al decidirnos por una Aplicación “Hospedada en SharePoint” también teníamos resuelto el problema de dónde almacenar los datos de la Aplicación...

71

Ejemplos de CAML en Client Object Model (JavaScript).

Resumen

El modelo de objetos de cliente de SharePoint ha sido una gran incorporación en SharePoint 2010. Nos permite consultar información de listas, utilizando sólo un poco de código JavaScript. No se requiere desarrollo a nivel de servidor, ni permisos especiales en la granja de SharePoint.

Artículo

Introducción

El modelo de objetos de cliente de SharePoint ha sido una gran incorporación en SharePoint 2010. Nos permite consultar información de listas, utilizando sólo un poco de código JavaScript. No se requiere desarrollo a nivel de servidor, ni permisos especiales en la granja de SharePoint. Uno de los usos más comunes, que al menos yo le he dado, es consultar información de listas, para luego desplegarla utilizando diferentes formas de presentación gráfica. Para ello, suelo utilizar consultas en lenguaje CAML. CAML tiene sus particularidades. En este breve artículo voy a presentar algunos ejemplos que me han resultado útiles.

En caso que nunca hayan utilizado Client Object Model, quizá sea conveniente que lean una introducción: <http://surpoint.blogspot.com/2011/10/introduccion-al-modelo-de-objetos-de.html>

Caso 1: empezando...

En el primer ejemplo, vamos a mostrar un caso completo. En los siguientes mostraremos sólo las variaciones. Imaginemos que necesitamos hacer una consulta simple, en donde veremos:

- Cómo ordenar elementos
- Cómo indicar qué columnas mostrar
- Cómo indicar que queremos traer sólo una fila

...La utilización de Client Object Model de JavaScript nos brinda una forma muy sencilla de consultar información y mostrarla en una página...

```
function cargar_noticia_destacada() {
    contexto = new
    SP.ClientContext.get_current();
    var lista =
    contexto.get_site().get_rootWeb().get_lists
    ().getTitle('Noticias');
    var query = new SP.CamlQuery();
    var consulta =
    '<View>' +
    // Ordenada por fecha de creación (ID)
    descendente
    '<Query><OrderBy><FieldRef Name ="ID"
    Ascending="FALSE"/></OrderBy></Query>' +
    // Una sólo noticia
    '<RowLimit>1</RowLimit>' +
    '<ViewFields>' +
    '<FieldRef Name="ID"/><FieldRef
    Name="Title"/><FieldRef
    Name="Resumen"/><FieldRef
    Name="Imagen_x0020_destacada"/>' +
    '</ViewFields>' +
    '</View>';
    query.set_viewXml(consulta);
    this.items = lista.getItems(query);
    contexto.load(items);

    contexto.executeQueryAsync(Function.createD
    elegate(this,
    this.cargaok_noticia_destacada),
    Function.createDelegate(this,
    this.cargamal));
}
```

En la siguiente función vemos cómo procesar los ítems leídos. Esto por supuesto depende de nuestra página, pero lo muestro para que tengan un ejemplo de cómo se haría.

```
function cargaok_noticia_destacada(sender,
args) {
    var itemslista =
    this.items.getEnumerator();
    while (itemslista.moveNext()) {
        var elitem = itemslista.get_current();
        var strHtmlImg =
        ...
    }
}
```

```

elitem.get_item('Imagen_x0020_destacada').get_url();
var strHtmlTxt = '<span class="h1">&gt; ' +
+ elitem.get_item('Title') + '</span><br /><br />' + elitem.get_item('Resumen');
var strHtmlBtn2 = '<a target="_self" href="javascript: AbrirVentanaModal(\' + window.location.href.replace("/default.aspx", "") + "/lists/Noticias/DispForm.aspx?ID=' + elitem.get_item('ID') + '&source=' + window.location.href + '\');" />';
var strHtmlBtn1 = "<a href='" + window.location.href.replace("/default.aspx", "") + '/lists/Noticias/' + "' />";
}
// Imagen
$("#general_modulo1_img img").attr("src", strHtmlImg);
// Título y resumen
document.getElementById("general_modulo1_titulo").innerHTML = strHtmlTxt ;
// Noticia completa
$("#general_modulo1_btn td:nth-child(2) img").each(function() {
    $(this).wrap(strHtmlBtn2);
});
// Más noticias
$("#general_modulo1_btn td:nth-child(1) img").each(function() {
    $(this).wrap(strHtmlBtn1 );
});
$("#general_modulo1_btn td:nth-child(1) img").css("border", "none");
}

```

Y finalmente nuestra rutina de error:

```

function cargamal(sender, args) {
alert('ERROR: ' + args.get_message() + '\n' +
+ args.get_stackTrace());
}

```

Caso 2: agregando una condición

En este segundo ejemplo, vamos a ver cómo agregar una condición de igual sobre un campo de tipo texto:

```

'<Query><Where><Eq><FieldRef Name="Mostrar_x0020_en_x0020_p_x00e1_g"/><Value Type="Text">Si</Value></Eq></Where><OrderBy><FieldRef Name="ID" Ascending="FALSE"/></OrderBy></Query>'

```

Caso 3: filtro por fecha actual

Supongamos ahora que queremos filtrar un campo fecha, comparándolo con la fecha del sistema. Veremos que necesitamos indicar el tipo de campo, e indicarle que no compare usando la hora:

```

'<Query><Where><Eq><FieldRef Name="Fecha"/><Value Type="DateTime" IncludeTimeValue="FALSE"><Today /></Value></Eq></Where></Query>'

```

Caso 4: filtrando entre fechas

Nuevamente compararemos con la fecha de sistema, pero esta vez, una comparación entre dos fechas:

```

'<Query><Where><And><Leq><FieldRef Name="Desde"/><Value Type="DateTime" IncludeTimeValue="FALSE"><Today/></Value></Leq><Geq><FieldRef Name="Hasta"/><Value Type="DateTime" IncludeTimeValue="FALSE"><Today/></Value></Geq></And></Where><OrderBy><FieldRef Name="ID" Ascending="FALSE"/></OrderBy></Query>'

```

Caso 5: filtro por campo calculado

En el siguiente ejemplo, mostraremos cómo filtrar cuando nuestra columna es un campo calculado:

```

'<Query><Where><Eq><FieldRef Name="Apellido_x0020_y_x0020_nombre" /><Value Type="Computed">' +
+ persona_seleccionada +
'</Value></Eq></Where></Query>'

```

Caso 6: filtro por campo de búsqueda

En este otro caso, vemos como filtrar en campos de búsqueda:

```

'<Query><Where><Eq><FieldRef Name="Usuario" LookupId="TRUE"/><Value Type="Lookup">' +
+ usuario_conectado +
'</Value></Eq></Where></Query>'

```

Caso 7: fecha ingresada por pantalla

Finalmente veremos un caso en donde necesitamos comparar contra una fecha ingresada por pantalla. El truco aquí es el formato de la fecha, el cual debemos formatear así:

```

var desde = ISODateString(Pdesde);

var hasta = ISODateString(Phasta);

```

...El modelo de objetos de cliente de SharePoint ha sido una gran incorporación en SharePoint 2010...

La función utilizada es:

```
function ISODateString(d) {

    function pad(n){return n<10 ? '0'+n : n}

    return d.getUTCFullYear()+'-'

        + pad(d.getUTCMonth()+1)+'-'

        + pad(d.getUTCDate())+'T'

        + pad(d.getUTCHours())+':'

        + pad(d.getUTCMinutes())+':'

        + pad(d.getUTCSeconds())+'Z'

}
```

...Uno de los usos más comunes, que al menos yo le he dado, es consultar información de listas...

Por último, el CAML para comparar las fechas ingresadas con las registradas en la lista:

```
'<Query><Where><And><Leq><FieldRef
Name="Fecha"/><Value Type="DateTime"
IncludeTimeValue="TRUE">' + hasta +
'</Value></Leq><Geq><FieldRef
Name="Fecha"/><Value Type="DateTime"
IncludeTimeValue="FALSE">' + desde +
'</Value></Geq></And></Where></Query>'
```

Conclusión

La utilización de Client Object Model de JavaScript nos brinda una forma muy sencilla de consultar información y mostrarla en una página de diferentes formas. En este breve artículo hemos mostrado algunas variantes del lenguaje de consulta CAML para entender cómo se aplican diferentes condiciones. Espero que les haya resultado útil. Hasta la próxima!

JUAN PABLO PUSSACQ LABORDE

SharePoint MVP

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

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

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



74

El autor destacado del
Número 16

Estrenamos sección y como comentamos en la editorial de este número, queremos premiar y reconocer el trabajo de nuestros autores. El mecanismo es sencillo y transparente, el autor del artículo que sea más leído en una edición será el autor destacado y lo anunciaremos con el siguiente número de la revista, así que son ustedes, los lectores, quienes deciden quién es el autor destacado de cada número.

Entre todos los autores destacados del año, seleccionaremos al que consideremos el mejor artículo del año de la revista, un premio al trabajo y la calidad de los artículos, que intentaremos recompensar de alguna forma.

Puesto que en la revista queremos premiar a todas aquellas personas que aportan su granito de arena escribiendo para cada número, los artículos de los Directores no podrán participar en este reconocimiento, que ya bastante tienen con poner en su currículum que dirigen una revista de habla hispana sobre SharePoint.

En el número 16 se publicaron 12 buenos artículos y, como siempre, queremos agradecer a los autores por el esfuerzo y por permitirnos publicar número a número sus conocimientos de SharePoint. Según las estadísticas de nuestro SharePoint 2013, que es la versión del sitio de

publicación de la web de la revista, el artículo más visitado corresponde a:

Adrián Díaz Cervera por su artículo **“Uso de Frameworks JavaScript en desarrollos SharePoint - Parte I”**, que al ser un artículo con varias partes supondrá todo un reto para superar en el siguiente número. Queremos agradecer y darle las felicidades a Adrián por el artículo y su trabajo en la revista.





Alberto Díaz

Alberto Díaz es Responsable de Proyectos de desarrollo en SharePoint, .NET, aplicaciones en movilidad y todo lo relacionado con tecnologías Microsoft en General de Software. Su labor en GSC es ofrecer soluciones empresariales a sus clientes y ofrecer nuevas tecnologías que ayuden a las empresas a realizar su trabajo más eficientemente, enfocado en productos 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.

Email: adiazcan@hotmail.com

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

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



Fabian Imaz

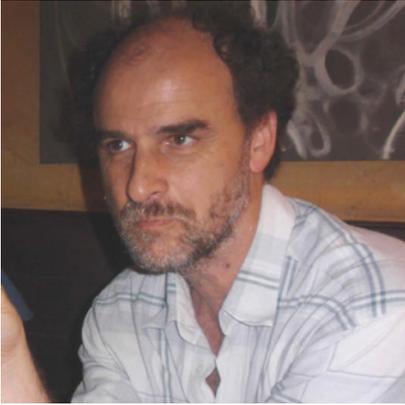
Fabián Imaz, MVP de SharePoint Server trabaja en el mundo del desarrollo de software desde hace más de 10 años, teniendo la suerte de trabajar en distintas arquitecturas y tecnologías Microsoft. Pertenece a la firma Siderys, <http://www.siderys.com> empresa de desarrollo de Software especializada en 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.siderys&bsn.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 casi 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 la plataforma SharePoint. 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) y del Grupo de Usuarios de Cloud Computing de España (CLOUDES). Desde el año 2011 participa junto con Gustavo Vélez, Fabián Imaz y Alberto Diaz en la dirección de CompartiMOSS. Hasta la fecha, ha publicado tres libros sobre SharePoint y varios artículos en castellano y en inglés sobre la plataforma.

Email: jcgonzalezmartin1978@hotmail.com

Blogs: <http://geeks.ms/blogs/ciin> & <http://jcgonzalezmartin.wordpress.com/>

Twitter: [@jcgm1978](https://twitter.com/jcgm1978).



¿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 |