



▶ BRANDING	3
▶ EL LADO SOCIAL DE SHAREPOINT (III)	6
▶ EXPORTAR METADATOS ADMINISTRADOS.....	10
▶ SIMPOSIO DE SHAREPOINT.....	15
▶ INTEGRACIÓN DE WINDOWS AZURE Y SHAREPOINT 2010 (I).....	18
▶ ENTREVISTA CON RICARDO MUÑOZ.....	22
▶ SHAREPOINT 2010 BI.....	24
▶ PATRÓN MVP CON SHAREPOINT 2010.....	27
▶ SITE DEFINITIONS VS. WEB TEMPLATES.....	29
▶ HERRAMIENTAS DE COMUNIDAD.....	32
▶ MODELO DE OBJETOS DE CLIENTE.....	33

Vol. 09
Septiembre 2011

Español

Comparti MOSS

REVISTA ESPECIALIZADA EN
TECNOLOGIAS SHAREPOINT

Bienvenidos a CompartiMOSS, la primera revista electrónica sobre SharePoint dirigida al mundo de habla hispana.

Editorial

Nuevo desafío completado, el número 9 de la revista esta entre nosotros ...

(Continúa en la primera página)

En esta Edición:

- ✓ Branding
- ✓ El lado social de SharePoint
- ✓ Metadatos Administrados
- ✓ Windows Azure y SharePoint
- ✓ SharePoint BI
- ✓ Patron MVP y SharePoint
- ✓ Site Definitions y Web Templates
- ✓ Modelo de Objetos de Cliente
- ✓ Más, mucho más...





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

- ✓ Fabián Imaz
- ✓ Juan Carlos González
- ✓ Gustavo Vélez

Contáctenos

fabiani@siderys.com.uy
jgonzalez@gruposodercan.es
gustavo@gavd.net

Blogs

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

Facebook

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

Escríbanos:

compartimoss@hotmail.com

Visítenos:

http://www.gavd.net/servers/compartimoss/compartimoss_main.aspx

Editorial

Nuevo desafío completado, el número 9 de la revista esta entre nosotros. En este número encontraran mucho material sobre SharePoint 2010 como se lo imaginarán y como la revista los tiene acostumbrados número a número. Pero esta edición tiene un sabor especial, tiene un artículo que nos hace recordar a los que hacemos la revista los momentos vividos en el 5 simposio latinoamericano de SharePoint. Fue un gran evento, un espacio donde tuvimos grandes charlas, experiencias y conocimientos con todas las personas que asistieron. Queremos darles las gracias a Vielka y Ricardo, organizadores del mismo por darnos la oportunidad de participar, fue un gran honor poder estar ahí durante toda una semana. Pero también fue muy grato volver a encontrarnos con amigos, colegas, hermanos de la vida que dadas las circunstancias y la distancia son pocas las veces que nos podemos ver para hablar y compartir momentos juntos. Este número se lo queremos dedicar a todas las personas que hicieron posible que el 5 simposio latinoamericano de SharePoint se llevara a cabo, gracias.

Junio, 2011

Fabián Imaz - fabiani@siderys.com.uy

Juan Carlos González jgonzalez@gruposodercan.es

Gustavo Vélez – gustavo@gavd.net

CompartiMOSS





Consejos prácticos de cómo aplicar el Branding Corporativo en SharePoint 2010 sin modificar la página maestra

..SharePoint dispone de una plantilla de sitio de publicación que está diseñada para facilitar la personalización del "Look & Feel" del sitio ajustándose a la marca de la empresa como lo es el logotipo, los colores y fuentes institucionales..

SharePoint 2010 en sus versiones Standard y Enterprise dispone de una plantilla de sitio de publicación que está diseñada para facilitar la personalización del "Look & Feel" del sitio ajustándose a la marca de la empresa como lo es el logotipo, los colores y fuentes institucionales.

En este artículo vamos a echar un vistazo de cómo utilizar esta plantilla de sitio, aplicando consejos prácticos y las mejores prácticas en la personalización del sitio y sin necesidad de modificar la página maestra directamente. Y no porque no se pueda, sino para demostrar que a nivel de las opciones de SharePoint que están disponibles en el explorador podemos modificar la visualización de nuestro portal.

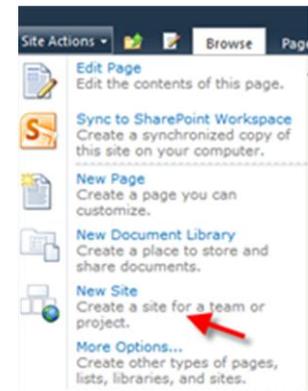
Primero debemos mencionar que si creamos de forma inicial el sitio con la plantilla de sitio de publicación no nos permitirá cambiar las plantillas de los sub-sitios que vayamos creando debajo de este y esto puede no ser deseado en un portal de colaboración. Entonces lo que vamos hacer es crear un sitio con la plantilla de Equipo y luego vamos activar la característica de publicación.

Iniciemos con los siguientes pasos:

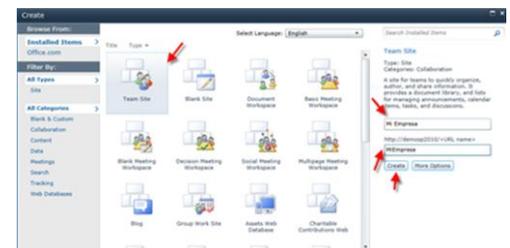
- 1) Creación de un sitio con plantilla de Equipo (Una plantilla muy utilizada que contiene elementos útiles para la creación).
 - 2) Activación de la característica de publicación a nivel de sitio.
 - 3) Creación de la hoja de estilos para aplicar nuestros propios estilos.
 - 4) Aplicación de la hoja de estilos al Sitio.
 - 5) Descubriendo los estilos que debemos de personalizar.
 - 6) Aplicación de los estilos en la hoja de estilos personalizada desde SharePoint Designer.
- Y por último las recomendaciones a tomar en cuenta en los sub-sitios que se creen posteriormente y deseamos aplicar el mismo Branding.

Creación de un sitio con plantilla de Equipo

1. Seleccione Site Actions, New Site



2. Seleccione la Plantilla Team Site, luego escriba el nombre del sitio y luego el nombre que será parte de la URL (Recomendación no deje espacios en blanco tampoco utilice caracteres especiales como '\&\$



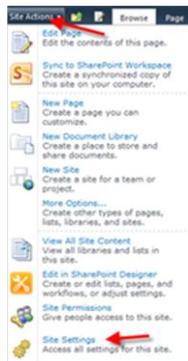
3. El resultado debería de mostrarle un listado similar al siguiente:



Activación de la característica de publicación a nivel de sitio

1. Seleccione nuevamente Site Actions y luego Site Settings





2. Luego seleccione Manage site features debajo de Site Actions



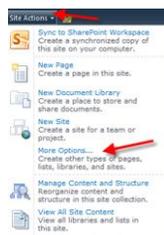
3. Presione el botón Activate sobre la opción SharePoint Server Publishing



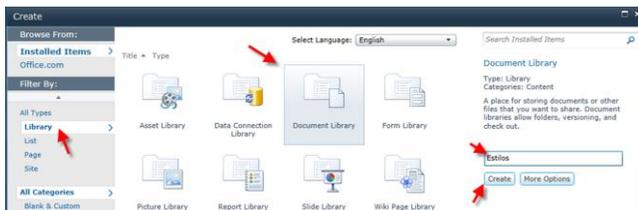
Creación de la hoja de estilos para aplicar nuestros propios estilos

Para ello vamos a crear una biblioteca de documentos para almacenar la hoja de estilos para esto siga los siguientes pasos:

1. Seleccione Site Actions, More Options



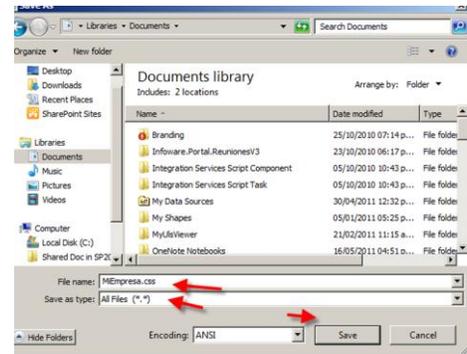
2. Luego seleccione Library, Document Library, Título escriba Estilos y luego presione el botón Create



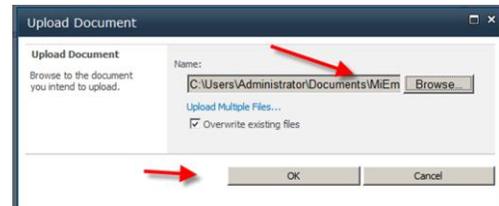
3. Luego ejecutaremos desde la ventana de Run notepad.exe para crear la hoja de estilos



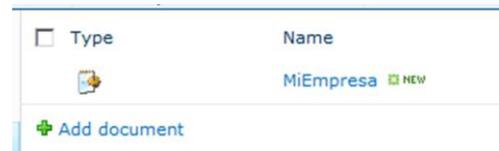
3. Luego agregue una línea /*Mi Empresa */ seleccione File, Save y luego escriba MiEmpresa.css y presione el botón Save



4. Cierre el notepad y luego vuelva al portal y seleccione Add Document, localice el documento donde lo almaceno y selecciónelo y presione el botón Ok

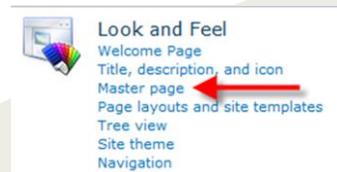


5. El Resultado debe de ser similar al siguiente:



Aplicación de la hoja de estilos al Sitio

Para aplicar la hoja de estilos es necesario que vayamos a la Configuración del Sitio para ello haga clic en Site Actions, luego Site Settings, luego haga clic en Master Page



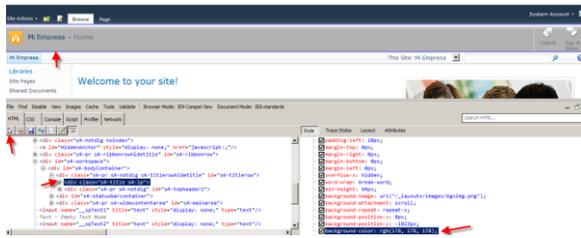


En la sección de Alternate CSS URL marque la opción "Specify a CSS file to be used by this publishing site and all sites that inherit from it" y luego haga clic sobre el botón Browse y ubique la hoja de estilos llamada MiEmpresa.css dentro de la biblioteca Estilos que estamos utilizando en este artículo



Descubriendo los estilos que debemos de personalizar

Para ello vamos utilizar el Developer Tool de IE. Sobre el navegador presiones F12



En esta imagen muestra las acciones tomadas. Seleccionamos la opción Selected by y luego seleccionamos el área esto nos dará la clase que se está aplicando y luego el estilo que se está aplicando en dicha clase. Para este ejemplo cambiaremos el color de fondo de la clase s4-title s4-lp.

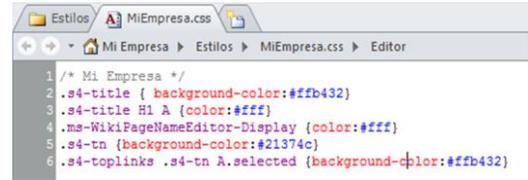
Así podremos descubrir otros estilos aplicados como el color del título del sitio en que estamos, que cambiaremos a color blanco.



Para cambiar el color de la Etiqueta Home, realizamos el mismo procedimiento arriba mencionado y al encontrar la clase la modificamos



En resumen la hoja de estilos que subimos a la biblioteca Estilos deberemos modificarla agregando las siguientes líneas:



* Esto fue modificado en SharePoint Designer. La cual puede accederse dese el Menú de SharePoint dentro de la opción Site Actions .

Por último reemplazaremos el logo predeterminado por otro. Para ello es necesario seleccionar la opción Site Actions, Site Settings y Title, Description and Icon bajo la sección Look & Feel.



Escribimos la dirección URL donde ubicamos la imagen que para este ejemplo agregamos una carpeta Imágenes dentro de la biblioteca Estilos y subimos una imagen de ejemplo. Finaliza presionando el botón OK.

El Resultado debería ser el siguiente:



Resumen final

En este artículo vimos cómo podemos aplicar un Branding Corporativo sobre un portal en SharePoint 2010 sin necesidad de modificar directamente la página maestra predeterminada. Esto no significa que no podamos o debamos modificarla pero cuando no tenemos el conocimiento necesario para hacerlo o no sea necesario pueda que esta opción nos auxilie para cambiar el look & feel de nuestro portal aplicando las mejores prácticas y opciones disponibles en SharePoint 2010.

Recomendaciones

Para los sub-sitios deberá activarse la característica de publicación y luego aplicar la hoja de estilos en el nuevo sitio para que tome la nueva personalización del sitio.

Juan Manuel Herrera
MVP SharePoint Server
<http://jmhogua.blogspot.com>





El lado social de SharePoint (Parte III)

En artículos anteriores (El lado social de SharePoint I y II), leímos las posibilidades que nos ofrecen los Perfiles de Usuarios y las capacidades sociales del mismo, así como los procesos de configuración y administración.

Con la introducción de FIM (Forefront Identity Management) como parte de la arquitectura, obtenemos múltiples mejoras en la sincronización a través de diversos orígenes, y la orientación social de los perfiles que nos ofrece la capacidad de interactuar con el contenido y las personas.

Este último artículo se basará en las posibilidades de desarrollo y de ampliación de funcionalidades. Veremos que nos ofrece el SDK y cómo podemos utilizar las propiedades de los perfiles para crear procesos de negocios que se adaptan a las organizaciones.

SDK

El amplio SDK de SharePoint nos ofrece multitud de ensamblados y espacios de nombres para poder desarrollar funcionalidades o aplicaciones que utilicen las capacidades de la plataforma. Dentro de los Perfiles de Usuarios, nos encontramos con las opciones necesarias para administrar Perfiles, aunque no con todo lo deseable. Por ejemplo, no podemos obtener los perfiles realizando una búsqueda por alguna de las propiedades definidas, tan solo podemos localizar perfiles por cuenta de usuario.

Modelo de Objetos

Para trabajar con el modelo de objetos de los Perfiles de Usuarios tenemos el espacio de nombres Microsoft.Office.Server.UserProfiles en el ensamblado Microsoft.Office.Server.UserProfiles.dll. En este encontramos la clase UserProfileManager (o OrganizationProfileManager que tiene la misma funcionalidad pero para obtener los Perfiles de Organización), que utilizaremos para obtener los perfiles de la aplicación de servicio de perfiles de usuario.

Obtener un Perfil de Usuario

Para obtener el perfil de un usuario tenemos la opción de localizarlo por la cuenta del usuario con el siguiente código.

```
UserProfile upUser = null;
var site = SPContext.Current.Site;
var serviceContext =
SPServiceContext.GetContext(site);
UserProfileManager upm = new
UserProfileManager(serviceContext);

upUser =
upm.GetUserProfile("DOMINIO\cuenta");
```

Propiedades del perfil

Como sabemos, los perfiles tienen propiedades que podemos leer programáticamente.

```
var value =
upUser["Phone"].Value.ToString();
```

También podemos hacer actualizaciones de dichas propiedades.

```
upUser["HomePhone"].Value =
"nnnnnnnnnn";
upUser["Mobile"].Value = "nnnnnnnnnn";
u.Commit();
```

Obtener los cambios recientes

Una de las opciones que nos ofrece el modelo de objetos, es poder obtener los últimos cambios que se han realizado en un perfil. Esta nos permite conocer que valores han cambiado en un perfil, por ejemplo, un cambio de Administrador o de Departamento.

```
using (SPSite site = new
SPSite("http://tusitio.com"))
{
    ServerContext context =
ServerContext.GetContext(site);
    UserProfileManager profileManager =
new UserProfileManager(context);
    // Obtener los cambios hace 5 dias
    DateTime startDate =
DateTime.UtcNow.Subtract(TimeSpan.FromDays(5
));
    UserProfileChangeQuery changeQuery =
new UserProfileChangeQuery(false, true);
    UserProfileChangeToken changeToken =
new UserProfileChangeToken(startDate);
```

... Este último artículo se basará en las posibilidades de desarrollo y de ampliación de funcionalidad es...





```
changeQuery.ChangeTokenStart = changeToken;
changeQuery.Anniversary = true;
changeQuery.SingleValueProperty = true;
changeQuery.MultiValueProperty = true;
changeQuery.DistributionListMembership = true;
changeQuery.SiteMembership = true;
UserProfile profile =
profileManager.GetUserProfile(@"DOMINIO\cuentaUsuario");
UserProfileChangeCollection changes =
profile.GetChanges(changeQuery);

foreach (UserProfileChange change in changes)
{
    Console.WriteLine(change.EventTime.ToString());
    // Obtiene los cambios en las propiedades
    if (change is UserProfilePropertyValueChange)
    {
        UserProfilePropertyValueChange
propertyChange = (UserProfilePropertyValueChange) change;
        Console.WriteLine(string.Format("Property
{0} - {1}", propertyChange.ProfileProperty.Name,
propertyChange.ChangeType.ToString()));
    }
    // Obtiene los cambios en el Membership
    else if (change is
UserProfileMembershipChange)
    {
        UserProfileMembershipChange
membershipChange = (UserProfileMembershipChange) change;

        Console.WriteLine(string.Format("Membership{0}-
{1}", membershipChange.MemberGroup.DisplayName, membershipC
hange.ChangeType.ToString()));
    }
}
}
```

Social

Otro espacio de nombre importante y que nos ofrece las características sociales es **Microsoft.Office.Server.SocialData** del ensamblado **Microsoft.Office.Server.UserProfiles.dll**. Este espacio de nombre dispone de las clases y métodos necesarios para administrar la información social de los perfiles. Veamos cómo podemos obtener la información etiquetada por un usuario, usando la clase **SocialTagManager**.

```
using (SPSite site = new SPSite("http://tusitio.com"))
{
    SPServiceContext context =
SPServiceContext.GetContext(site);
```

```
UserProfileManager myUserProfileManager = new
UserProfileManager(context);
UserProfile myUserProfile =
myUserProfileManager.GetUserProfile(false);
SocialTagManager mySocialTagManager = new
SocialTagManager(context);
SocialTag[] tags =
mySocialTagManager.GetTags(myUserProfile);
Console.WriteLine("Tags for user:");
foreach (SocialTag tag in tags)
{
    Console.WriteLine(tag.Term.Name + " " +
tag.Uri.AbsoluteUri);
}
}
```

Activity Feed o Suministro de noticias

El espacio de nombre del suministro de noticias, **Microsoft.Office.Server.ActivityFeed**, permite obtener las actividades de un usuario y agregar actividades. La clase **ActivityManager** es la encargada de obtener o crear eventos y actividades para un usuario. Por ejemplo, podrías obtener los mensajes de Twitter de un usuario e insertarlos en el suministro de noticias para que se puedan consultar por los compañeros de este.

```
public ActivityEvent CreateActivity(UserProfile
ownerProfile, UserProfile publisherProfile, ActivityType
activityType, string linkText, string linkUrl)
{
    //Obtienes el propietario y el publicador.
    Entity ownerMP = new
MinimalPerson(ownerProfile).CreateEntity(activityManager)
;
    Entity publisherMP = new
MinimalPerson(publisherProfile).CreateEntity(activityMana
ger);

    //Creamos la actividad.
    ActivityManager am = new ActivityManager();
    ActivityEvent activityEvent =
ActivityEvent.CreateActivityEvent(am,
activityType.ActivityTypeId, ownerMP, publisherMP);

    activityEvent.Name = activityType.ActivityTypeName;
    activityEvent.ItemPrivacy = (int)Privacy.Public;
    activityEvent.Owner = ownerMP;
    activityEvent.Publisher = publisherMP;

    //Creamos un link para la actividad.
    Link link = new Link();
```





```

link.Href = linkUrl;
link.Name = linkText;
activityEvent.Link = link;

activityEvent.Commit();

return activityEvent;
}

```

Servicios Web

Además del Modelo de Objetos de servidor, tenemos disponibles los servicios web de SharePoint para administrar los perfiles. Este servicio web nos permite acceder a la información de los perfiles de SharePoint desde aplicaciones cliente, tipo Silverlight o Windows Form.

Servicio http://<server>/_vti_bin/UserProfileService.asmx para administrar perfiles de usuario

Servicio http://<server>/_vti_bin/UserProfileService.asmx para obtener los cambios de un perfil

Servicio http://Site/_vti_bin/SocialDataService.asmx para obtener información social de los perfiles

Obtener un perfil

Básicamente los métodos del servicio web son similares a los que tenemos en el modelo de objetos, veamos un ejemplo de cómo se obtiene un perfil de usuario usando el servicio web.

```

UserProfileWebService.localhost.PropertyData[] properties
=
myService.GetUserProfileByName ("DOMINIO\cuentaUsuario");
for (int i = 0; i < properties.Length; i++)
{
    Console.WriteLine (properties[i].Name);
    Console.WriteLine (properties[i].Value);
}

```

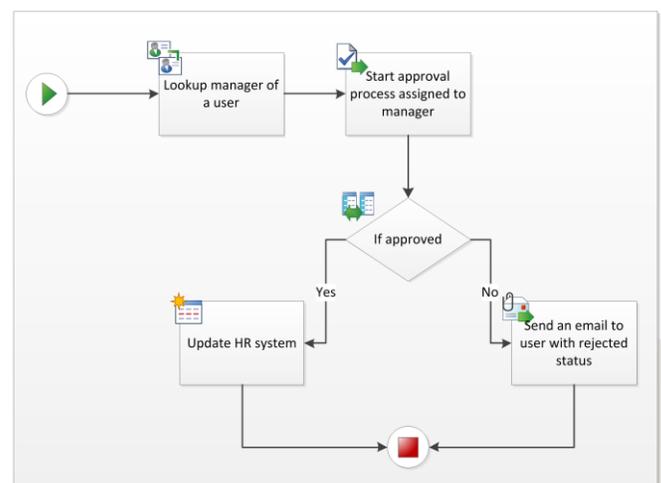
Usando las propiedades de los perfiles

Pero, ¿Qué utilidades le podemos encontrar a tener información empresarial en los perfiles?

En un solo ejemplo, mostraré el uso de estos valores para crear procesos empresariales usando Workflows. Supongamos que tenemos configurado nuestra sincronización de perfiles para que obtenga la información del Directorio Activo de la empresa y de la aplicación de Recursos Humanos, mediante el uso de un BCS (Business Connectivity Services). La sincronización de las propiedades de los perfiles quedaría tal como se muestra en la siguiente tabla.

Propiedad	Origen
Administrador (Manager)	Recursos Humanos
Teléfono	Recursos Humanos
Móvil	Recursos Humanos
Nombre	Directorio Activo
DNI – Identificador Nacional	Recursos Humanos
....

Nos piden que creamos un proceso de negocio para la solicitud y aprobación de las vacaciones. Este flujo de trabajo necesita que cuando un empleado solicite sus vacaciones, estas se asignen a su responsable (Administrador) para aprobación y una vez que estén aprobadas se envíe una actualización de las mismas a la aplicación de Recursos Humanos. Véase el siguiente diagrama.





Lo primero que tenemos que hacer es obtener el responsable del usuario que ejecuta el proceso, obteniendo el valor de la propiedad Manager de su perfil, si no tuviéramos al responsable en los perfiles no sabríamos quien tiene que aprobar las vacaciones de cada empleado.

Una vez que tenemos a este responsable, le asignamos un proceso de aprobación, y si este es satisfactorio, utilizamos un BCS, enlazado con el proceso que actualiza las vacaciones en la aplicación de Recursos Humanos. Para este proceso, nos hace falta una propiedad que nos permita identificar al usuario en la aplicación de Recursos Humanos, y, aunque parezca obvio, en la mayoría de los casos no vamos a poder utilizar el email o a la cuenta del usuario en el Directorio Activo. Para esto, tenemos una propiedad sincronizada que es el DNI (Documento Nacional de Identidad en España) que es único y la podemos utilizar para guardar la información de vacaciones en la aplicación.

Búsqueda de perfiles

Como hemos visto, la clase UserProfileManager no tiene un método que nos permita obtener los perfiles de usuario de un determinado grupo, por ejemplo, los que el campo Departamento tenga el valor 'Comercial'. Sin embargo, podemos utilizar el servicio de búsqueda (http://<server>/_vti_bin/search.asmx) para realizar consultas en el ámbito de Personas y así poder obtener los perfiles que coinciden en una búsqueda.

```
using (var servicio = new SearchService.QueryService())
{
    string keywordQueryTemplate = "<?xml version='1.0'
encoding='utf-8' ?> " +
    "<QueryPacket xmlns='urn:Microsoft.Search.Query'
Revision='1000'> " +
    "    <Query domain='QDomain'>" +
    "        <SupportedFormats>" +
    "
<Format>urn:Microsoft.Search.Response.Document.Document</
Format> " +
    "    </SupportedFormats> " +
    "    <Context> " +
    "        <QueryText language='es-ES'
type='STRING'>query_text_placeholder</QueryText> " +
    "    </Context> " +
    "    <Properties> " +
    "        <Property name='AccountName' />" +
    "        <Property name='JobTitle' />" +
    "        <Property name='Department' />" +
    "        <Property name='Rank' />" +
```

```
"        <Property name='Title' />" +
"        <Property name='Author' />" +
"        <Property name='Size' />" +
"        <Property name='Path' />" +
"        <Property name='Description' />" +
"        <Property name='Write' />" +
"        <Property name='HitHighlightedSummary' />" +
"        <Property name='HitHighlightedProperties' />" +
"    </Properties>" +
" </Query> " +
"</QueryPacket>";

servicio.Credentials =
System.Net.CredentialCache.DefaultCredentials;
string queryString =
keywordQueryTemplate.Replace("query_text_placeholder",
"Department:\Comercial" SCOPE:\Personas");
string queryResults = servicio.Query(queryString);
var kk = servicio.QueryEx(queryString);
var kk2 = XElement.Parse(queryResults);

Console.WriteLine(queryResults);
```

El servicio de búsqueda nos permite realizar consultas por campos indexados de los perfiles de usuarios, y obtener aquellos usuarios que coinciden con nuestra consulta. Ahora sólo tenemos que obtener los UserProfile de cada uno de los usuarios que se obtienen en el resultado de la búsqueda.

Conclusiones

El SDK de SharePoint nos abre las puertas a un mundo de posibilidades de mejoras y desarrollo de nuevas funcionalidades. En el lado social podemos administrar perfiles de usuarios y los perfiles de organización, gestionar el etiquetado de los usuarios e, incluso, integrar estas dos capacidades y obtener las actividades de un usuario en su red social almacenada en su perfil, por ejemplo, Twitter, y hacer las actualizaciones de su actividad social de la empresa junto con esta red social.

No veamos a SharePoint como una red social tipo Facebook, en la que lo principal son las amistades, sino que tenemos la capacidad de socializar el contenido empresarial con las personas, que son el principal valor de una empresa.

Alberto Diaz Martin

MVP SharePoint Server

Arquitecto de Soluciones en General de Software

<http://www.gsc.es/>

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

@adiazcan





SolidQ Managed Metadata Exporter: Herramienta para exportar Metadatos Administrados de SharePoint 2010

... nueva característica llamada Metadatos Administrados, la cual pretende dar un paso más para convertir este gestor de contenidos empresarial, en un sistema gestor de conocimiento...

Presentación

Como parte de la tarea habitual que asumimos los MVPs, siempre he considerado fundamental la formación y promoción de nuevos "personajes" . En este caso, les presento al amigo José Quinto Zamora, que viene trabajando en nuestro equipo desde antes de terminar su Carrera Universitaria. Seguramente cada vez más encontrarán publicaciones con su nombre: realmente valen la pena. Un saludo gigante. Daniel A. Seara.

Resumen

En SharePoint 2010 aparece una nueva característica llamada Metadatos Administrados, la cual pretende dar un paso más para convertir este gestor de contenidos empresarial, en un sistema gestor de conocimiento. Básicamente se trata de un sistema de taxonomía o repositorio común donde almacenar metadatos jerárquicamente que nos servirán luego para etiquetar y clasificar las páginas, documentos, y elementos de listas. El principal inconveniente que trae en su primera aparición es que no se pueden exportar estas taxonomías para pasarlas de un entorno a otro. Para esto SolidQ ha diseñado una herramienta que es capaz de exportar estos metadatos. Esta nota técnica pretende mostrar cómo se ha implementado dicha herramienta.

Introducción

Hace un año aproximadamente me surgió la necesidad de exportar los términos de un conjunto de términos de metadatos administrados para importarlos en otro entorno de desarrollo, mi sorpresa fue cuando me di cuenta que esta opción no estaba disponible en SharePoint 2010. Sin embargo sí existe la posibilidad de importar estos metadatos desde un fichero .csv con un formato propio [1]. Entonces la herramienta SolidQ Managed Metadata Exporter lo que hace es permitir seleccionar un conjunto de términos de los Metadatos Administrados y lo exporta en formato .csv para

luego poder importar dicho fichero en otra aplicación de servicio de metadatos administrados.

En un escenario de replicar términos entre entornos de desarrollo, pre-producción y producción donde es muy interesante esta herramienta para poder exportar e importar todos los conjuntos de términos entre entornos. No obstante, una vez importamos los términos en el nuevo entorno también habrá que realizar el proceso de reasignación de GUID, ya que, si teníamos alguna columna creada con metadatos administrados lo términos importados no serán reconocidos debido a que el ID de la aplicación de servicio de metadatos administrados es distinto y los GUID internos de cada términos también serán distintos, si nos encontramos con este caso revisen este enlace [2] para solucionar el problema.

Veamos entonces algunos detalles técnicos de cómo está implementada la herramienta.

Implementación

Veamos un pantallazo de la herramienta para poder describir las partes de implementación de las que consta:



Como vemos, en primer lugar se pide un Site Collection desde el cual conectar a la aplicación de servicio de metadatos administrados (Term Store). Luego una vez tenemos seleccionado un Term Store (Managed Metadata Service Application) entonces seleccionaremos un Grupo y dentro del grupo un Conjunto de Términos (TermSet en adelante) y ya por último los





Términos y sub-términos serán mostrados en la última sección. No obstante, si seleccionamos un TermSet vamos a poder realizar más acciones, entre las más importantes están:

- Export Selected TermSet (Exportar un Conjunto de Términos a .csv)
- Add TermSet to Search Suggestion (Añadir todos los términos de dicho Conjunto de Términos a las sugerencias de búsquedas de SharePoint)



Nota: Esta funcionalidad fue añadida posteriormente por Cristian Zaragoza, véase este post para más información [3].

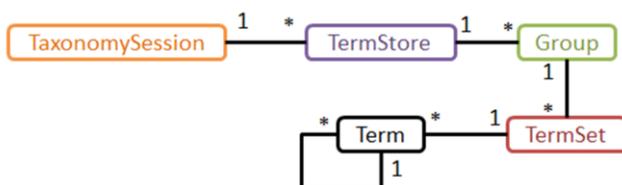
Modelo de Objetos de Metadatos Administrados (Taxonomy)

Ensamblado ' Microsoft.SharePoint.Taxonomy.dll
Espacio de nombres ' Microsoft.SharePoint.Taxonomy
Clases principales:

- Clase TaxonomySession
- Clase TermStore
- Clase Group
- Clase TermSet
- Clase Term
- Método CommitAll
- Método CreateLabel
- Método setDescription

Para ver la relación completa de clases: <http://msdn.microsoft.com/en-us/library/ee583437.aspx>

Para un mejor entendimiento, este es el esquema del Modelo de Objetos:



El siguiente código nos muestra cómo acceder a los metadatos administrado vía Modelo de Objetos:

```

Using coleccionDesitios As New
SPSite("http://miSiteCollection")
Dim sesion As New
TaxonomySession(coleccionDesitios)

'Obtengo todos los TermStores de la
TaxonomySession
For Each almacen As TermStore In
sesion.TermStores
Console.WriteLine("Almacén de términos: " &
almacen.Name)

'Obtengo todos los grupos de cada uno de
los TermStores
For Each grupo As Group In almacen.Groups
& grupo.Name)

'Obtengo todos los TermSet de cada
uno de los grupos
For Each conjuntoTerminos As
TermSet In grupo.TermSets
Console.WriteLine("
Conjunto de términos: " & _ conjuntoTerminos.Name)

'Obtengo todos los términos de
cada uno de los TermSets
For Each termino As Term In
conjuntoTerminos.Terms
Console.WriteLine("
Término: " & termino.Name)

'Si el término tiene sub
términos también los obtengo
'con la función recursiva
If termino.TermsCount > 0
Then
ImprimirSubTerminos(termino, 1)
End If
Next
Next
Next
Next
End Using

'Función recursiva para obtener todos los sub-términos
de un término
Sub ImprimirSubTerminos(ByVal termino As Term, ByVal
nivel As Integer)

'Obtengo todos los subtérminos del Term
For Each subtermino As Term In termino.Terms

'Este contador se usa para dejar tantos
espacios como
'niveles tengamos en la jerarquía
Dim numEspacios As Integer = 1
numEspacios = numEspacios * nivel
For i As Integer = 1 To numEspacios
Console.Write(" ")
Next
Console.WriteLine(" SubTérmino: "
& subtermino.Name)

'Si el sub-término tiene a su vez sub-
términos los muestro

```





```
        If subtermino.TermsCount > 0 Then
            ImprimirSubTerminos(subtermino, nivel
+ 1)
        End If
    Next
End Sub
```

A continuación, veremos cómo están implementadas la Exportación de términos y la funcionalidad de añadir a sugerencias de búsquedas (ya que me parece bastante curiosa la forma en la que se debe realizar).

Exportar Términos a .csv

Para exportar términos desde nuestros metadatos administrados a un fichero .csv tenemos el siguiente código que, mediante Modelo de Objetos Servidor de SharePoint 2010, podremos acceder a nuestros TermSets y convertirlos en .csv's, veamos cómo:

```
'Método que exportar el conjunto de términos a .csv
Private Sub ExportTermSet()
    'Objeto SaveFileDialog para seleccionar la ruta del
    disco y el nombre del fichero
    Dim saveFileDialog1 As New SaveFileDialog()
    saveFileDialog1.Filter = "Managed Metadata
File|*.csv"
    saveFileDialog1.Title = "Save an CSV File"
    'Indicamos un nombre de forma predeterminada llamado
    igual que el TermSet
    saveFileDialog1.FileName =
tvTermSets.SelectedNode.Text & ".csv"

    If saveFileDialog1.ShowDialog() = DialogResult.OK
Then
        'Si el nombre y ruta no están vacíos, continuamos
        con la exportación
        If saveFileDialog1.FileName <> "" Then
            Dim strFileName As String =
saveFileDialog1.FileName
            Dim myStream As StreamWriter = New
StreamWriter(strFileName, False)

            'Creamos el Objeto SolidQTermSetExporter que nos
            permitirá crear el .csv
            Dim exporter As New SolidQTermSetExporter

            Using SiteCollection = New
SPSite("Absolute_URL")
                Dim session = New
TaxonomySession(SiteCollection)
                Dim termStore = session.TermStores(New
Guid(tvTermStores.SelectedNode.Name))
                Dim termset = termStore.GetTermSet(New
Guid(tvTermSets.SelectedNode.Name))

                'Invocamos al método ExportTermSet con un
                TermSet como parámetro
                Dim text As String =
exporter.ExportTermSet(termset)
```

```
myStream.Write(text)
myStream.Close()
myStream = Nothing
End Using
End If
    MessageBox.Show("TermSet exported in " &
saveFileDialog1.FileName)
End If
End Sub

Public Class SolidQTermSetExporter

    'Función que data un TermSet lo convierte a
    formato .csv
    Public Function ExportTermSet(ByVal termSet As
TermSet) As String

        Dim csv As New StringBuilder()
        'La primera línea del documento .csv
        csv.AppendLine("Term Set Name","Term Set
Description","LCID","Available for
Tagging","Term Description","Level 1
Term","Level 2 Term","Level 3 Term","Level 4
Term","Level 5 Term","Level 6 Term","Level 7
Term")

        'Para cada Término escribo una línea en el
        .csv indicando todos las propiedades
        For Each topTerm In termSet.Terms
            csv.AppendLine( _
                String.Format(""{0}"", "{1}"", {2}, ""
{3}"", "{4}"", , , , , , ,", _
                    termSet.Name, _
                    termSet.Description, _
                    termSet.IsAvailableForTagging,
                    _
                    topTerm.GetDescription(), _
                    topTerm.Name))

            'Invocamos a la función recursiva
            ExportTerm que escribirá también
            'una línea en el .csv para los sub-
            términos
            For Each childTerm In topTerm.Terms
                ExportTerm(csv, _
                    childTerm, _
                    String.Format(""{0}"",
                    "", topTerm.Name))
            Next
        Next

        Return csv.ToString()
    End Function

    'Función recursiva que recorre todos los sub-
    términos de un término dado
    Private Sub ExportTerm( ByVal csv As
StringBuilder, _ ByVal term As Term, _ ByVal subT As
String)

        'Se añade el nombre del sub-término, junto con
        todos los nombres de los
        'Términos padre y ancestros de este
        subT = subT & ", " & String.Format(""{0}"",
term.Name))
```





```
csv.AppendLine(String.Format(",,{0},{1}",{
2}"), _
        term.IsAvailableForTag
ging, _
        term.GetDescription(),
        subT))

'Se recorren todos los niveles, sub-términos
de sub-términos, etc...
For Each subTerm In term.Terms
    ExportTerm(csv, subTerm, subT)
Next

End Sub
End Class
```

Como comentaba antes, este programa se debe ejecutar en el propio servidor de SharePoint 2010, ya que utiliza el Modelo de Objetos Servidor [4], si quisiéramos acceder a los metadatos administrados desde un programa cliente necesitaríamos usar los servicios web que vienen que SharePoint 2010 para esto, en concreto se llama TaxonomyClientService. Ya que no entra en el ámbito de este artículo, no entraremos en detalle, pero pueden verlo en detalle en este post [5].

Añadir Términos a las Sugerencias de Búsqueda de SharePoint

La otra funcionalidad de la que os quería hablar, es la posibilidad de añadir ciertos Términos y/o Conjuntos de Términos a las sugerencias de búsqueda, ya que en un entorno de gestión de contenido empresarial en el que también configuramos una búsqueda empresarial puede ser muy útil el tener ciertas taxonomías definidas tanto para etiquetar documentos como para poder buscar dichos documentos usando términos concretos de dichas taxonomías. Con esta herramienta podremos decidir qué términos son añadidos a las sugerencias de búsquedas de forma sencilla.

La única forma para añadir sugerencias al motor de búsquedas de SharePoint 2010 es mediante la consola de administración de PowerShell de SharePoint 2010. Por ello hemos tenido que invocar a PowerShell directamente desde el programa.

Veamos el código para ello:

```
'''Método que añade un Término a la Search Suggestion
List
Private Sub AddTermToSearchSuggestionList()
    'String que almacenará el comando PowerShell a
    ejecutar
    Dim stringCommand As String = String.Empty
    Dim listCommands As List(Of String) = New List(Of
    String)
```

```
    Dim session As New TaxonomySession(SiteCollection)
    Dim term As Term = session.GetTerm(New
    Guid(tvTerms.SelectedNode.Name))
    stringCommand = "New-
    SPEnterpriseSearchLanguageResourcePhrase -
    SearchApplication $ssa -Language en-US -Type
    QuerySuggestionAlwaysSuggest -Name " & """" &
    term.Name & """"
    listCommands.Add(stringCommand)
    RunScript(listCommands)
    MessageBox.Show("Término " & term.Name & " añadido
    correctamente.")
End Sub

'''Método que ejecuta el comando PowerShell desde un
programa de escritorio
Private Sub RunScript(ByVal listCommands As List(Of
String))
    'Crear el PowerShell RunSpace
    Dim runspace As Runspace =
    RunspaceFactory.CreateRunspace()
    'Crear el Objeto de Resultados
    Dim results As New
    System.Collections.ObjectModel.Collection(Of PSObject)
    'Abre el RunSpace
    runspace.Open()

    'Crear la tubería para pasarle el Script de
    PowerShell
    Using pipeline As Pipeline =
    runspace.CreatePipeline()

        'Añadir los cmdlets de SharePoint al Contexto
    de PowerShell
        pipeline.Commands.AddScript( _
        "[System.Reflection.Assembly]::LoadWithPartialName(""M
icrosoft.SharePoint"")")
        pipeline.Commands.AddScript("Add-PSSnapin
Microsoft.SharePoint.PowerShell")

        'Obtener la Service Application de Search vía
    PowerShell
        pipeline.Commands.AddScript("$ssa=Get-
SPEnterpriseSearchServiceApplication")

        'Añadir los comandos que pondrán las
    sugerencias en el Service Application
        'usando la variable $ssa
        For Each command As String In listCommands
            pipeline.Commands.AddScript(command)
        Next

        'Obtener el TimerJob encargado de añadir las
    sugerencias de búsquedas nuevas
        pipeline.Commands.AddScript( _
        "$timer=Get-SPTimerJob? {$_.Name -eq
        ""Prepare Query Suggestions""}")

        pipeline.Commands.AddScript("$timer.RunNow()")
        'Ejecutar el cmdlet completp
        Try
            results = pipeline.Invoke()
        Catch ex As Exception
            Throw ex
        End Try
    End Using

    'Cerrar el RunSpace
    runspace.Close()
End Sub
```





Conclusión

Respecto a los metadatos administrados, es una clara referencia de hacia donde se está moviendo SharePoint 2010, ya que al introducir un mecanismo para gestionar taxonomías [6] esto nos lleva a pensar que está siguiendo las directrices de un sistema de gestión del conocimiento [7]. Mediante el cual tendremos todo el conocimiento de nuestra empresa almacenado en un sitio centralizado desde el cual podremos clasificar, organizar, buscar y encontrar de manera eficiente cualquier conocimiento que tengamos en nuestra empresa.

Este artículo pretende dar una vista técnica a cómo está implementada la herramienta de CodePlex SolidQ Manage Metadata Exporter [8]. Pero haciendo hincapié en dos funcionalidades que no están de forma predeterminada (OOB) en SharePoint 2010, como son Exportar Conjuntos de Términos y/o añadirlos a las sugerencias de búsquedas de SharePoint 2010.

José Quinto Zamora
<http://www.solidq.com>

Referencias

[1] Formato cvs para importar Metadatos Administrados en SharePoint 2010: <http://technet.microsoft.com/es-es/library/ee424396.aspx>

[2] SP2010 - Migrating managed metadata term sets to another farm on another domain:
<http://www.cleverworkarounds.com/2011/01/09/sp2010-migrating-managed-metadata-term-sets-to-another-farm-on-another-domain/>

[3] Añadir sugerencias de búsqueda en SharePoint 2010 desde metadatos administrados:
<http://blogs.solidq.com/sharepoint/Post.aspx?ID=106&title=a%C3%B1adir+sugerencias+de+b%C3%BAsqueda+en+sharepoint+2010+desde+metadatos+administrados>

[4] Modelo de Objetos Servidor de Metadatos Administrados (Microsoft.SharePoint.Taxonomy Namespace):
<http://msdn.microsoft.com/en-us/library/ee583437.aspx>

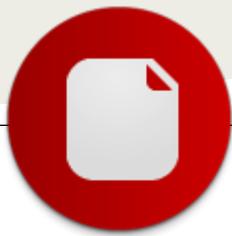
[5] SharePoint 2010 -> Managed Metadata -> TaxonomyClientService Web Service a fondo:
<http://blogs.solidq.com/sharepoint/Post.aspx?ID=35&title=SharePoint+2010+%3e+Managed+Metadata+%e2%80%93+TaxonomyClientService+Web+Service+a+fondo>

[6] Taxonomía:
<http://en.wikipedia.org/wiki/Taxonomy>

[7] Sistema gestor de conocimiento:
http://en.wikipedia.org/wiki/Knowledge_management

[8] SolidQ Managed Metadata Exporter:
<http://metadataexportsps.codeplex.com/>





Instrucciones para organizar un simposio de SharePoint



... Nuestro sueño es que el simposio se replique en muchos países de Latinoamérica de manera anual ...

Con la creación del primer grupo de SharePoint en Costa Rica en el año 2007, Ricardo y yo comenzamos a trazar grandes planes para compartir nuestra pasión por la tecnología, especialmente por SharePoint.

La comunidad SharePoint Costa Rica inició realizando reuniones bimensuales en San José, Costa Rica, pero a principios del año 2008, con el apoyo de Héctor Insua y Luis Du Solier, nos lanzamos a la gran tarea de organizar el Primer Simposio Latinoamericano de SharePoint. Los cuatro queríamos crear un espacio propio donde se impartieran charlas especializadas sobre SharePoint en nuestro idioma, aprovechando los excelentes expositores latinoamericanos que tenemos y que no tienen nada que envidiarle a sus contrapartes de otras latitudes.

Después de mucho trabajo y esfuerzo de un excelente equipo hemos logrado crear un evento al mejor estilo de la Conferencia Mundial de SharePoint, pero de solamente un día de duración. En el primer simposio que realizamos en Octubre de 2008 en el Hotel San José Palacio en Costa Rica. Tuvimos dos salones dando charlas simultáneas, 9 expositores (de 3 Nacionalidades) y 200 participantes, y este año, en el Quinto Simposio realizado en Costa Rica en el Parque Pueblo Antiguo, contamos con la presencia de 330 personas, 18 expositores (de 9 Nacionalidades) y tres salones con charlas simultáneas. En este tiempo hemos logrado crecer en cantidad de asistentes así como la cantidad de Speakers de altísima calidad de toda Latinoamérica que muy amable y desinteresadamente nos regalan su valioso tiempo para participar en nuestro evento.

¿Cómo se organiza un simposio de SharePoint?

Con dos locos entusiastas de SharePoint a la cabeza. Primero decidimos la fecha, nos fijamos la meta de participantes (si queremos llegar a 200 o 300 personas) y comienza el reclutamiento de los expositores y la propuesta de charlas, esta es la parte fácil, de verdad que la comunidad de SharePoint está llena de personas a las que les gusta compartir su conocimiento. Conocer la cantidad de expositores y charlas es importante para decidir la cantidad de salones que se deben contratar. Luego empieza la ardua y complicada búsqueda del lugar que cumpla con las expectativas y sobre todo nuestras limitantes monetarias, una vez que tenemos esto conseguimos un préstamo para dar el abono del hotel y rogar para que podamos pagar de vuelta este préstamo con la venta de entradas.

Es normal que en muchos momentos sientan que no se van a vender todas las entradas o que quizás los costos sean demasiado altos sobretodo en el caso de una comunidad que es una organización sin fines de lucro y que como en nuestra experiencia los administradores de la comunidad hemos tenido que poner dinero de nuestro bolsillo.

Uno de los primeros patrocinadores que se debe reclutar es Microsoft, es importante que el evento tenga su respaldo, en Costa Rica, nos ayuda mucho el DPE o el contacto de Microsoft con las comunidades. En Costa Rica los Centros de Entrenamiento (CPLS) brinda mucho apoyo a las actividades comunitarias, así que son los primeros que deben de matricularse en el evento.





¿Qué hemos aprendido?

La experiencia facilita la organización de los eventos, pero esto lo hemos aprendido con la práctica y con prueba y error. Trabajar con hoteles es mejor, porque ellos facilitan la contratación de comida y salones, pero es importante que tenga un buen parqueo. Al seleccionar el equipo audiovisual es importante que sea de buena calidad, para poder ver bien las demostraciones y las presentaciones que tengan alta definición.

Es mejor comenzar la organización del simposio al menos unos 6 meses antes de la fecha del evento. Y comenzar el envío de la publicidad por medio de correos, medios sociales, blogs lo antes posible.

¿Hacia dónde vamos?

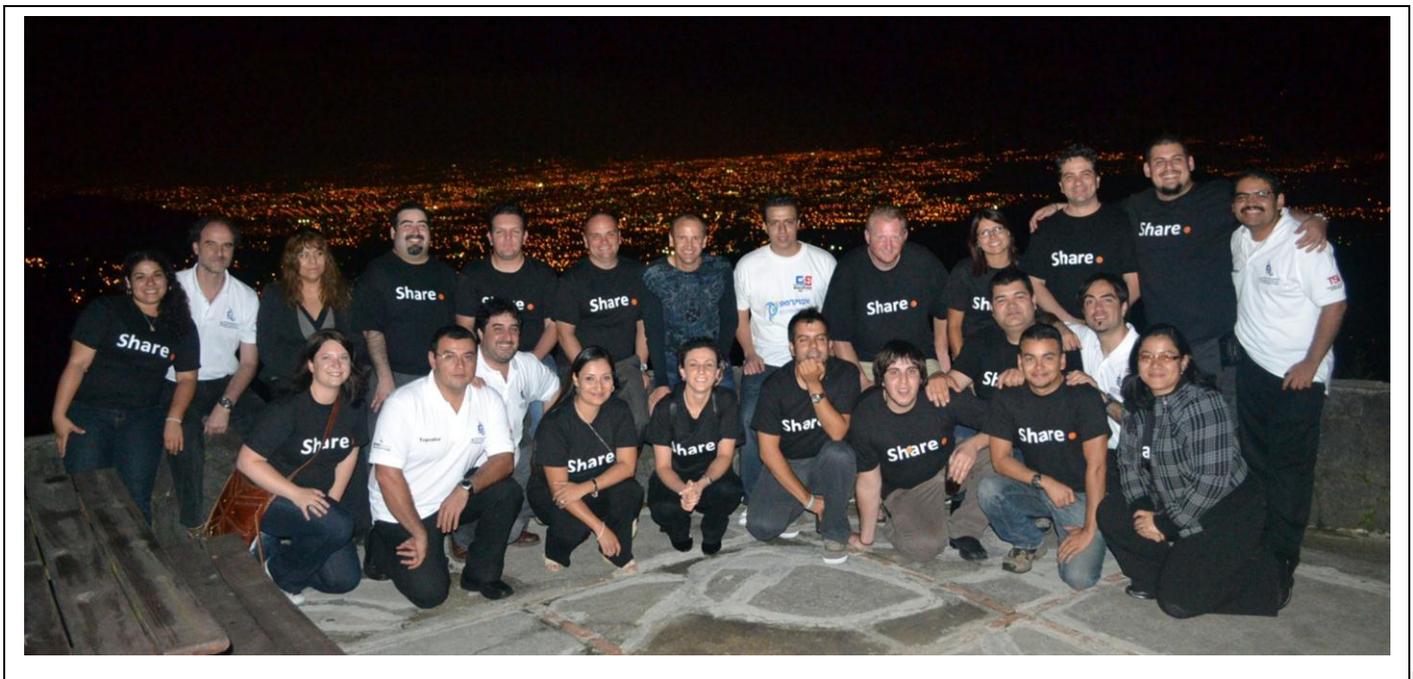
Nuestro sueño es que el simposio se replique en muchos países de Latinoamérica de manera anual, así que si usted desea que se organice uno en su país, contáctenos lo antes posible, para ayudarlo en todo lo que podamos.

Este año ampliamos el formato del simposio para incluir un taller de Desarrollo en SharePoint y otro de Infraestructura en SharePoint, de dos días de duración, para que poco a poco se convierta en una semana de SharePoint.

Pueden encontrar mayor información sobre el Quinto Simposio Latinoamericano de SharePoint en www.SharePointCostaRica.com

El sitio de la comunidad podrán encontrarlo en: www.ug.gitca.org/sites/CRC-SPS/default.aspx

Vielka Rojas
Ricardo Muñoz Monge





SHAREPOINT

quinto simposio latinoamericano

aprende colabora comparte





Integración de Windows Azure y SharePoint 2010 (I)

... un recorrido por los escenarios de integración de SharePoint y Azure como plataformas de desarrollo ...

Por un lado, SharePoint 2010 es la plataforma de colaboración empresarial que facilita la toma de decisiones en las empresas, la colaboración y compartición de conocimiento a lo largo de toda la organización e incrementa la productividad en su día a día. SharePoint no es sólo un sistema para el almacenamiento y gestión de información sino también una completa plataforma de desarrollo que permite modificarlo y adaptarlo a las más variadas condiciones exigidas por las empresas modernas. Añadido a las capacidades que incorpora por defecto la plataforma y su extensibilidad, con las nuevas herramientas para desarrollo incorporadas en Visual Studio 2010 se puede acortar los tiempos para crear soluciones innovadoras y de valor para el cliente reduciendo la curva de aprendizaje y logrando una alta productividad en el desarrollo.

Por otro lado, Windows Azure es la plataforma de Microsoft de desarrollo de "Cloud Computing" que permite crear y ejecutar aplicaciones en la nube, lanzarlas en cuestión de minutos en lugar de meses y codificar en varios lenguajes y tecnologías, como .NET, Java o PHP. Windows Azure proporciona la plataforma ideal para innovar, modelar e implementar escenarios y aplicaciones de alto rendimiento y escalabilidad, dejando de lado de problemas de redundancia, ancho de banda o limitaciones del servidor.

El objetivo de esta artículo no es otro que realizar un recorrido por los escenarios de integración de SharePoint y Azure como plataformas de desarrollo, que permiten modelar y construir elementos de funcionalidad que aprovechen el alto grado de conjunción existente entre ambas habilitando ricos escenarios de integración entre ambas, y también de negocio.

Escenarios de integración de Windows Azure y SharePoint 2010

Las posibilidades de integración de Windows Azure y SharePoint 2010 vienen determinadas a un primer nivel por quién expone los datos y como estos son consumidos. Por lo tanto, se pueden definir inicialmente dos situaciones de integración globales en función de si se están integrando datos o servicios de Windows Azure en SharePoint o viceversa. En el primer caso, se pueden consumir datos o servicios publicados en Windows Azure de acuerdo a los siguientes mecanismos:

- Mediante el uso de iFrames, que permite integrar de forma sencilla páginas ASP.NET publicadas en Windows Azure partiendo de la plantilla ASP.NET Web Role disponible en Visual Studio 2010 (Nota: Esta plantilla está disponible una vez se han instalado las Windows Azure Tools para Visual Studio 2010). Para ello basta con añadir en una WebPart de tipo "Editor de Contenido" el siguiente código HTML (Listado 1):

```
1: <IFRAME id="azureTest"
src="http://fabrikamhockeyazure.cloudapp.net/Default.aspx" scrolling="auto">
2: </IFRAME>
```

Listado 1.- Código HTML para embeber una aplicación ASP.NET de Azure en SharePoint.

Como ventajas de esta aproximación se pueden destacar que se trata de una integración simple y ligera, en la que no se requiere código y no se necesita desplegar ningún tipo de artefacto en SharePoint 2010. Como inconvenientes se pueden citar la pérdida de personalizaciones propias de SharePoint, el bajo nivel de integración y que no se pueda distribuir utilizando un paquete estándar de SharePoint (.WSP o "Windows SharePoint Package").

- A través de un servicio o de datos hospedados que facilite integrar de forma sencilla datos que se encuentran en Azure (por ejemplo en una base de datos, BD, creada en SQL Azure) ya sea de forma directa (usando la aplicación de servicio de BCS, la DataFormWebPart o bien desarrollos personalizados, por ejemplo) o mediante un elemento intermediario (un servicio web fuera de la nube que recupere datos de SQL Azure). Como ventajas de esta alternativa se pueden citar las siguientes: se están usando servicios existentes tanto en el lado de SharePoint (BCS) como en el de Windows Azure (SQL Azure), el modelo de programación es muy directo así como las técnicas para realizar el acceso a datos/servicios .





Como inconveniente principal de esta aproximación cabe destacar que es dependiente del servicio/infraestructura que expone los datos. La Figura 1 muestra el esquema de este escenario (patrón) de integración directo.

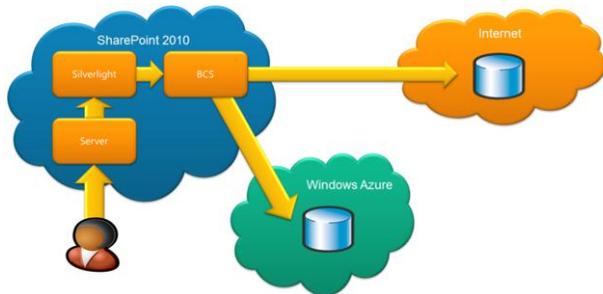


Figura 1.- Patrón de integración entre SharePoint y Windows Azure a través de servicios o datos hospedados.

- Mediante servicios personalizados publicados en Azure, que faciliten el acceso a datos de forma sencilla desde artefactos desplegados en SharePoint (aplicaciones Silverlight, WebParts, manejadores de eventos, etc) consumidores de dichos servicios. Como se puede deducir, los puntos fuertes de esta alternativa son que se basa en la generación de artefactos de distinta naturaleza desplegados en SharePoint, que permite un mayor control y la disponibilidad de múltiples puntos de entrada para integrar datos y servicios de Azure en SharePoint 2010. En cambio, como puntos débiles podemos citar que es necesario administrar estos servicios desplegados y que se necesita "tirar" más código en comparación con las otras posibilidades. La Figura 2 muestra el esquema tipo de este patrón de integración en base a un ejemplo en el que se usa jQuery como tecnología de acceso al servicio personalizado.

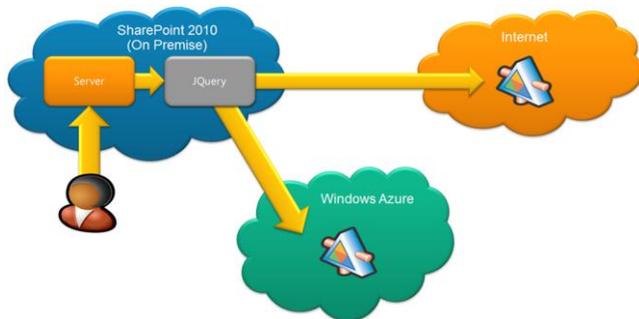


Figura 2.- Patrón de integración entre SharePoint y Windows Azure a través de servicios personalizados publicados en Azure.

De la misma forma que se pueden consumir datos y servicios de Windows Azure en SharePoint, se puede producir la situación inversa de manera que se facilite que servicios publicados en Azure o servicios externos (que aprovechen componentes de la plataforma de Cloud Computing como el Windows Azure AppFabric) consuman datos de SharePoint 2010 a través de los servicios web que expone la plataforma, la API REST, servicios web personalizados o bien el modelo de objetos en cliente. La Figura 3 recoge el patrón de integración para este escenario:

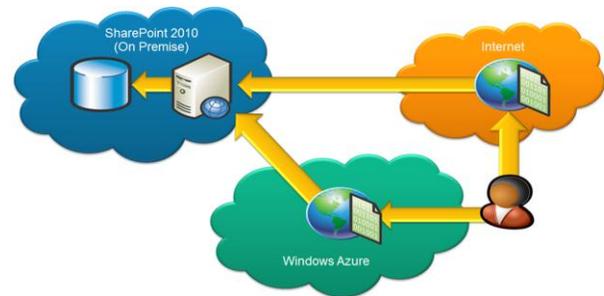


Figura 3.- Patrón de integración de datos de SharePoint en Windows Azure o en servicios externos.

Opciones de integración entre Windows Azure y SharePoint 2010

A partir de los escenarios y posibilidades de integración entre ambas plataformas, se pueden enumerar una serie de opciones de integración entre las mismas que pasan por:

- Desde SharePoint 2010, es posible integrar datos o servicios de Azure por medio de los siguientes mecanismos:
 - El modelo de objetos en cliente de SharePoint (.NET, Silverlight o ECMAScript) para interactuar con datos y servicios de Windows Azure por un lado y datos de listas y bibliotecas de SharePoint por otro.
 - A través de los Business Connectivity Services (BCS) para modelar la integración de datos y servicios de Azure en SharePoint por medio de tipos de contenido externos y listas externas.
 - Microsoft Silverlight para construir interfaces de usuario ricas que integren servicios o datos de Azure.
 - WebParts (visuales o estándar) de SharePoint que consuman servicios o datos de Azure.



- Formularios InfoPath a través de la definición de conexiones de datos a servicios web publicados en Azure o datos almacenados en SQL Azure.
- Manejadores de eventos y flujos de trabajo que permitan integrar datos o servicios de Azure en la lógica de negocio que implementen.
- Mediante las búsquedas federadas de SharePoint que permitan incluir datos de Azure.
- Desde Windows Azure, se pueden consumir datos de SharePoint a través de:
 - El uso de los servicios web de SharePoint para interactuar con sitios, listas, usuarios y otros elementos propios de la plataforma.
 - El uso de la API REST de SharePoint para interactuar con datos de listas de SharePoint.
 - El uso del modelo de objetos en cliente de SharePoint que permite interactuar de forma remota con datos de listas y bibliotecas de SharePoint. En este escenario, se podrían utilizar las implementaciones .NET y Silverlight.

Un primer ejemplo de integración

Para finalizar el artículo, a continuación se expone un primer ejemplo de integración simple entre ambas plataformas de acuerdo al primer escenario descrito: El uso de iFrames para visualizar en un sitio de SharePoint 2010 una aplicación ASP.NET publicada en Windows Azure. En este escenario se trata de crear una aplicación ASP.NET usando el "ASP.NET" Web Role disponible en las Windows Azure Tools para Visual Studio 2010 que permite crear una aplicación ASP.NET convencional y publicarla de forma sencilla en Windows Azure:

- La aplicación en cuestión permite mostrar datos de una base de datos (BD) de SQL Azure utilizando para ello LINQ To SQL como tecnología de acceso a datos de acuerdo al código del Listado 2 que se ha añadido al manejador de un control de tipo botón disponible en la página por defecto de la aplicación ASP.NET.

```
1: try
2: {
3:     using (AzureCustomersDataContext ctx =
4:         new AzureCustomersDataContext (
5:
6: "Server=tcp:<Servidor_SQLAzure>;Database=<BD_SQLAz
7: ure>;User
8: ID=<Usuario_SQLAzure>;Password=<Passwor_SQLAzure>;
9: Trusted_Connection=False;Encrypt=True;") )
```

```
6:     {
7:         var Customers = from c in
8: ctx.MD_Clientes
9:             select c;
10:        ctx.Log = swWriter;
11:        grdCustomers.DataSource = Customers;
12:        grdCustomers.DataBind();
13:        txtConsulta.Visible = true;
14:        txtConsulta.Text = "Consulta: " +
15:            swWriter.ToString();
16:    }
17: }
18: catch (Exception ex)
19: {
20:     txtError.Visible = true;
21:     txtError.Text ="Error: " + ex.Message;
22: }
```

Listado 2.- Acceso a datos de SQL Azure con LINQ To SQL.

Como se deduce en el código anterior, se está realizando una consulta LINQ To SQL a una BD de SQL Azure creando para ello una instancia del correspondiente contexto de datos en el que se ha indicado el string de conexión a la BD para a continuación definir la consulta a la tabla "MD_Clientes" definida en dicha BD, volcar el resultado de ejecución de la consulta en un control de tipo GridView y mostrar por pantalla la consulta T-SQL que se ha enviado al servidor de SQL Azure.

- Una vez depurada la aplicación ASP.NET, a través de Visual Studio 2010 se genera el correspondiente paquete de publicación que posteriormente será utilizado en el Windows Azure Portal (<https://windows.azure.com/>) para desplegar la aplicación en un entorno de staging o producción que hará accesible la aplicación desde Internet (Figura 4).

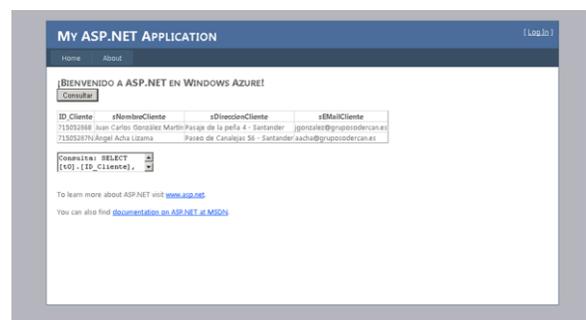


Figura 4.- Aplicación ASP.NET publicada en Windows Azure y accesible de forma global.



- La URL de la aplicación ASP.NET es la que se usará para su integración en SharePoint utilizando por ejemplo una WebPart de tipo "Visor de páginas" que añade por ejemplo en una página de WebParts creada con ese fin. Este tipo de WebPart tiene, entre otros, la URL de la página a mostrar como parámetro que es posible configurar (Figura 5).

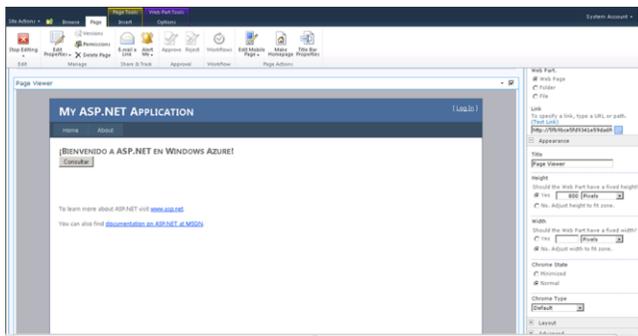


Figura 5.- Configuración de la WebPart "Visor de páginas".

- Finalmente, es suficiente con salir del modo de edición de la página y comprobar que la página embebida funciona como se espera, pero integrada en SharePoint (Figura 6).

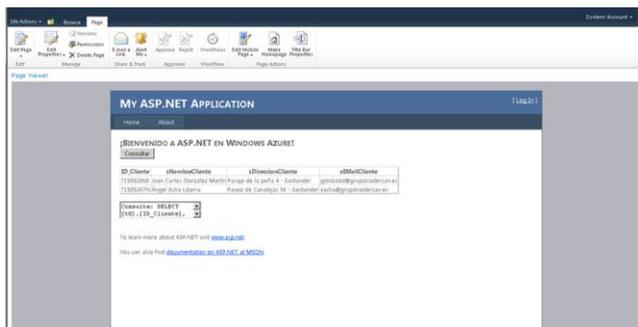


Figura 6.- Resultado de la integración de la aplicación ASP.NET de Azure en SharePoint 2010.

Conclusiones

Sin duda, SharePoint 2010 y Windows Azure son dos de las plataformas tecnológicas más potentes y flexibles ofrecidas por Microsoft para dar respuesta a diferentes requerimientos de negocio. Por separado, cada una de ellas tiene una serie de atractivos y puntos fuertes y su concepción como plataformas de desarrollo hacen que las posibilidades de modelado y construcción de soluciones de negocio sean prácticamente ilimitadas. Juntas, se incrementa aún más el espectro de escenarios que se pueden cubrir, facilitando la integración de datos y servicios desde ambos lados utilizando para ello una serie de herramientas familiares que se integran dentro de Visual Studio 2010.

Juan Carlos González Martín

Arquitecto de Soluciones, MVP de SharePoint Server
Centro de Innovación en Integración (CIIN) - Microsoft
Innovation Center Cantabria
[@jcg1978](http://geeks.ms/blogs/ciin)

Referencias

- [1] Developing Microsoft SharePoint Applications Using Windows Azure. Steve Fox. Microsoft Press.
- [2] SharePoint & Azure: Alternativas de integración. Serie de posts publicados en el blog del CIIN: <http://geeks.ms/blogs/ciin/archive/2011/06/07/sharepoint-amp-azure-alternativas-de-integraci-243-n-iv.aspx>
- [3] SharePoint and Windows Azure Development Kit. Disponible para descarga en: <http://www.microsoft.com/download/en/details.aspx?id=24398>
- [4] Get Started Building SharePoint Online Solutions. Artículo publicado por Todd Baginski y disponible en: <http://www.sharepointpromag.com/article/sharepoint-development/get-started-building-sharepoint-online-solutions>





Entrevista con Ricardo Muñoz

Biografía – Introducción



Mi nombre completo es Ricardo Muñoz Monge y nací hace 30 años en mi bello país Costa Rica, mi infancia transcurrió bastante tranquila sin muchos problemas más que las travesuras típicas de los niños y uno que otro accidente ocasionado por dichas travesuras, en esos momento mi única relación con la tecnología eran los videojuegos en consola o pc (hobbie que mantengo aun hoy en día).

En mi adolescencia inicie mi acercamiento a la tecnología tomando clases de programación en Basic y mantenimiento de computadores en el colegio lo cual

me hizo darme cuenta que la informática era lo mío, todo eso unido al hecho que mi otra carrera preferida era historia, (no sé dónde estaría ahora si hubiera tomado otra decisión) inicie mis estudios en el Instituto Tecnológico de Costa Rica.

Durante mi práctica de especialidad tuve mi primer acercamiento con SharePoint en su versión 2003 y desde ese momento no he dejado de trabajar con ella por ya más de 9 años. MI trabajo actual es como Gerente de Colaboración y Consultor Sénior en Infraestructura SharePoint para Andes IT y mi compañía LatinShare.

Cuando el tiempo entre viajes y trabajo me lo permite actualizo mi blog sobre tecnología SharePoint en www.mundomoss.blogspot.com

¿Por qué y cómo empezaste en tecnología?

Bueno creo que inicie en tecnología porque siempre me han atraído dos temas: La historia y todo lo que tenga que ver con adelantos tecnológicos, yo era uno de esos niños que intentaba armar y desarmar sus juguetes para ver sus componentes con el único problema que debes en cuando sobran algunas piezas. En el colegio tuve el primer contacto con la programación y la infraestructura, gustándome mucho más la segunda, posteriormente durante mis estudios universitarios ingrese a trabajar en un importante partner de Microsoft en Costa Rica, fui contratado desde un inicio con el objetivo de investigar las funcionalidades de SharePoint y Project Server 2003, herramientas de las cuales quede inmediatamente prendido y hoy en día son mi especialidad.

Este gran interés por la tecnología SharePoint me llevo a en conjunto con mi gran amiga Vielka Rojas a formar la comunidad de SharePoint Costa Rica, la cual inicio como una idea entre un par de locos y hoy en día es una de las comunidades de SharePoint mayor importancia a nivel latinoamericano, esto nos llevó a realizar grandes eventos como el Simposio Latinoamericano de SharePoint (www.sharepointcostarica.com) que ya va para su sexta edición, todo esto me ha permitido viajar muchísimo y conocer a grandes amigos pero sobre todo a muchísimas personas apasionadas por la tecnología así como por compartir su conocimiento.





¿Cuáles son tus principales actividades tecnologías hoy en día?

Hoy en día me desempeño como gerente del área de colaboración en la empresa Andes IT, empresa especializada en la implementación de soluciones SharePoint y Project Server. Hace mucho tiempo que deje de programar y ahora me dedico exclusivamente a la infraestructura SharePoint como arquitecto de Soluciones.

Adicionalmente desde hace unos meses inicié mi faceta como empresario en Costa Rica y tengo en conjunto con otros expertos del ramo nuestra propia compañía llamada LatinShare dedicada al desarrollo, implementación y outsourcing de soluciones de colaboración sobre plataformas Microsoft.

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

A veces no saco el tiempo que debería para actividades NO tecnologías pero principalmente me encanta viajar, me gusta conocer nuevos lugares siempre que puedo, mis últimos dos viajes han sido a Isla de Pascua y Machu Pichu dos destinos increíbles. Adicionalmente me gusta mucho leer, Jugar videojuegos siempre que puedo pero sobre todo pasar tiempo con las personas que quiero mi familia y amigos.

¿Cuáles son tus hobbies?

Bueno creo que ya es obvio que me gusta mucho la tecnología no solo en el trabajo sino también en mi vida diaria, pero adicional a esto uno de mis hobbies ha sido y será la Historia sobre todo la segunda guerra mundial, mi padre me inculco desde niño la pasión por este tema al punto de que siempre viajo con algún libro sobre el tema para mis ratos de lectura, adicionalmente me encanta conocer nuevas culturas y sus comidas soy de la idea de primero hay que probar algo por más extraño que se vea, Soy un también un gran fan de los comics americanos y europeos (sobre todo Asterix y Obelix) así como del cine.

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

La tecnología cambia a veces demasiado rápido, lo cual es parte de lo interesante de nuestra profesión siempre tenemos que actualizarnos y tratar de estar al tanto de todos los avances que se dan día a día.

Hoy en día existen avances sorprendentes que hace unos 10 o 15 años nos hubieran parecido de ciencia ficción como el caso del Kinect o el iPad así como otro conjunto de tecnologías, todos estos grandes avances y cambios nos permitirán hacer más accesible la información a las masas y a las personas que actualmente no tienen dicho acceso por diversas razones económicas y sociales.

Poder predecir que nos traerá el futuro de la tecnología es muy complejo pero sé que será un futuro muy interesante y lleno de muchos retos y oportunidades para todos los que estamos apasionados con la tecnología.

(Entrevista realizada por Juan Carlos González para CompartiMOSS)





SharePoint 2010 BI

Bienvenidos a la serie de artículos sobre el uso de SharePoint en el mundo de inteligencia de negocios (BI).

En este artículo comenzaremos a ver en forma general los diferentes componentes que se usan en una solución de BI dentro de SharePoint.

En la actual versión de SharePoint la parte de inteligencia de negocios ha evolucionado de una forma exponencial con respecto a lo que puede realizar en el mundo de BI.

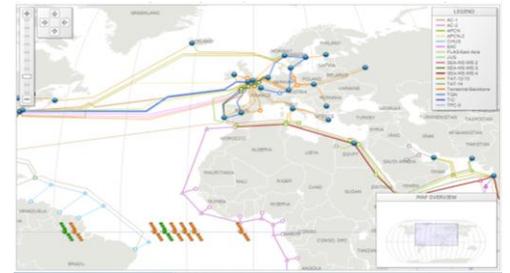
Esto es mucho dado por un lado por la incorporación de lo que conocimos como PerformancePoint 2007 Monitoring y la evolución de Excel services y la incorporación de Visio Services. Sin olvidar las mejoras a la integración de reporting services de SQL Server 2008 R2 y por supuesto el nuevo chico de la cuadra de BI Powerpivot y su integración con SharePoint.

Arquitectura de BI en SharePoint 2010

La belleza de construir una solución de BI está en la integración de los diferentes componentes que se tienen. A continuación detallamos dichos componentes:

- **SQL SERVER Database:** es la base por supuesto del repositorio de SharePoint pero además se usa como nuestro Datawarehouse central. Esto gracias a características como compresión de datos y particionamiento de tablas.
- **OLAP Services:** Se le conoce como Analysis services el cual funciona como una base de datos multidimensional que normalmente viene en base a nuestro Datawarehouse. Esto también se le conoce como manejador de cubos y es nuestra fuente principal de las soluciones como en Performancepoint por ejemplo
- **Reporting Services:** Componente que muchas organizaciones usan no solo para hacer reportes operativos sino también para realizar reportes gerenciales con indicadores, termómetros y graficas analíticas que

permiten representar la información de una forma diferente a los demás componentes como Excel services, Visio services y sobre todo Performancepoint. También se usa para realizar análisis de localización geográfico el cual muchas empresas buscan realizar para inteligencia de negocios.



- **Hojas electrónicas de Excel y Excel Services:** Componente muy importante ya que muchos de los análisis que existen dentro de las empresas están ya realizados en hojas de Excel. Excel services en esta versión de SharePoint 2010 viene muy potencializada soportando la mayoría de características del Excel cliente. Además de tener un rendering muy fiel a las hojas Excel originales.
- **Visio Services:** Este servicio es nuevo en SharePoint 2010, básicamente funciona para hacer el rendering de los archivos de Visio y sobre todo manejar la conectividad con las fuentes de información. Visio nos ayuda a crear organigramas, mapas, dashboards que contiene data que está saliendo en vivo de las fuentes y permite hacer visualizaciones como mapas estratégicos de los Balance Scorecard conectados a scorecard que realizamos en Performancepoint.



... la parte de inteligencia de negocios ha evolucionado de una forma exponencial con respecto a lo que puede realizar en el mundo de BI ...

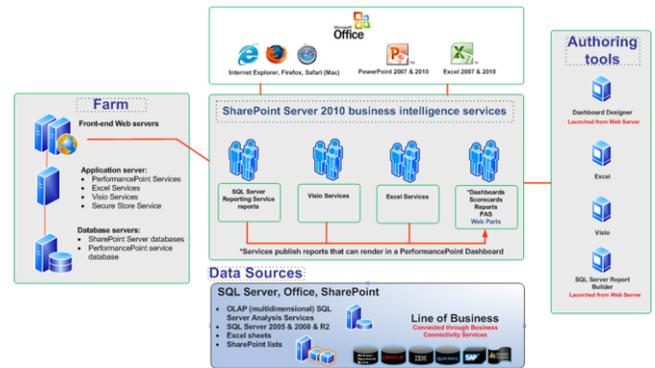




- Performancepoint services: Esto se le conocía anteriormente como Performancepoint 2007 monitoring. Propociona la funcionalidad de una herramienta de inteligencia de negocios para realizar gráficas, matrices, indicadores de desempeño, scorecards y Dashboards para los usuarios gerenciales. En SharePoint 2010 normalmente tenemos disponibles estas características como parte del BI Center dentro de las plantillas de sitios. A través de los dashboards se unen los diferentes componentes anteriores como Excel Services, Reporting services y Visio Services.



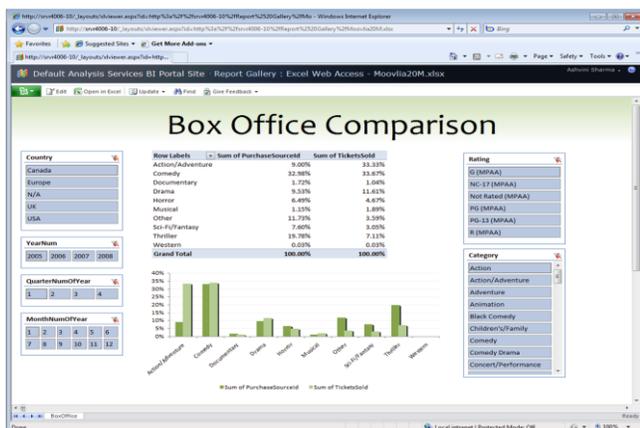
- Conector de BI Search: Este es una addin para Fast Search para poder buscar contenido en hojas electrónicas Excel, Powerpivot y reportes de Reporting Services con esto proporcionar búsquedas a los usuarios finales en base a contenido de BI.



Licenciamiento de SharePoint para BI

Es importante que las empresas tengan en cuenta que para activar estos beneficios de BI dentro de SharePoint se debe de poseer la Cal Enterprise de SharePoint. A nivel SQL se necesita la versión Enterprise si solo si va usar Powerpivot integrado a SharePoint. De lo contrario puede funcionar muy bien usando la Cal Standard. También hay que tener en cuenta que ciertas personas se necesitaran licenciamiento de Excel 2010 y Visio 2010.

- Powerpivot: El nuevo chico del BI en SQL y totalmente integrado en SharePoint. Este componente nos viene a llenar la necesidad de que los usuarios creen sus propios análisis en una forma autónoma con Excel y los compartan fácilmente usando SharePoint.



Futuro de BI dentro de SharePoint

Definitivamente la estrategia de Microsoft está basada en SharePoint y cualquier avance de la inteligencia de negocios será realizado a traves de SharePoint. Para el próximo año tendremos una nueva versión de SQL (Denali) la cual ya trae dentro de sus novedades una herramienta de análisis para usuarios finales (Proyecto Crescent) basa en tecnología Silverlight e integrada a SharePoint. También vendrá una nueva integración de Reporting Services como Servicio de SharePoint 2010, bajo la cual nos dará los beneficios como mejor rendimiento y compartir naturalmente los recursos de SharePoint y por una integración de la seguridad de SharePoint hacia nuestros reportes. Esto parte de las mejoras de Powerpivot 2.0 que tendrá novedades dentro de la integración de SharePoint y el nuevo modo tabular de SQL en Analysis Services de SQL Denali.





Además existen ya muchos productos de Microsoft tales como Project Server 2010 que usan la funcionalidad de BI Center y Excel Services para análisis de información de los proyectos administrados tales como avances de los proyectos, uso de los recursos, scorecards de los proyectos, etc.

Próximos artículos

En los próximos artículos de esta serie vamos a analizar los componentes y cómo usarlos para desarrollar nuestros dashboards necesarios dentro de SharePoint. Así que en

nuestro próximo artículo será enfocado sobre PerformancePoint services configuración, instalación y desarrollo de dashboards.

Juan M. Alvarado

Consultor de Tecnologías de BI y SQL Microsoft

GYSSA

MVP SQL SERVER

www.youtube.com/juanbizzz

juanbizzz

juanbizzz.wordpress.com

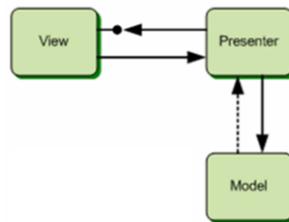




Patrón MVP con SharePoint 2010

... el patrón MVP (Model View Presenter) tiene como objetivo básico delegar las responsabilidad es a la clase Presentadora ...

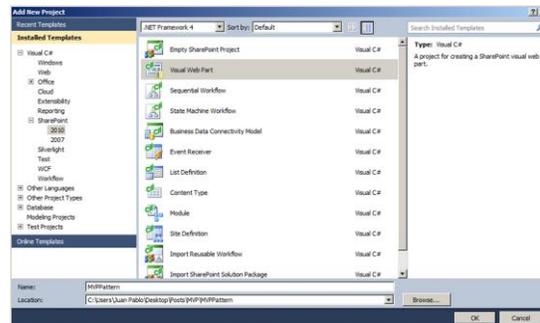
Hace ya un tiempo que estoy tratando de implementar un mecanismo para desarrollar Web Parts lo mas desacopladas posibles entre la UI y el código, y que a su vez me permita desarrollar testing de forma sencilla. En este intento por buscar mejorar el código me encontré con el patrón MVP (Model View Presenter). El mismo tiene como objetivo básico delegar las responsabilidades a la clase Presentadora, esta última actuara como intermediario entre la Vista (interfaz gráfica de usuario) y el modelo de datos. A continuación se presenta un diagrama en el cual se puede observar la idea que se menciona.



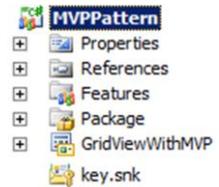
Habiendo explicado por arriba el patrón MVP volveré al foco de este artículo en el cual vamos a describir una de las formas sencillas de implementar el patrón MVP en SharePoint 2010.

Para lograr esto tomaremos como base un ejemplo sencillo que nos permitirá comprender cuál es la idea y el objetivo de utilizar MVP en conjunto con SharePoint 2010. El ejemplo no es más ni menos que una grilla que lista los datos de una lista de páginas de SharePoint 2010.

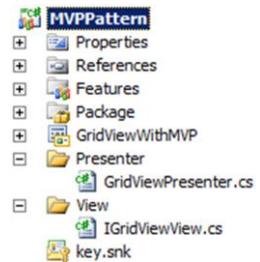
Por lo tanto, tomaremos como punto de partida la creación de una Visual WebPart como se muestra en la siguiente imagen:



Luego de creada la Visual Web Part, le cambiaremos el nombre de VisualWebPart a GridViewWithMVP, obteniendo como resultado la siguiente composición del proyecto:



A continuación se crearan dos carpetas la primera llamada Presenter y la segunda View. Creamos dentro de cada carpeta las clases GridViewPresenter y IGridViewView respectivamente. La estructura del proyecto nos debería ir quedando de la siguiente forma:



En el siguiente paso estableceremos el contrato con el cual la vista va a interactuar con el presentador. Para esto se define la interfaz con las firmas que vamos a utilizar. Por lo tanto para este ejemplo suena razonable definir las siguientes firmas:

```
public interface IGridViewView
{
    SPSPSite Site { get; }
    SPWeb Web { get; }
}
```

Como se muestra en el código, se definen dos propiedades: Site y Web, tienen como objetivo comunicar los objetos de SharePoint con el presentador.

Nota: Es importante tener en cuenta que se han definido estas dos propiedades





únicamente para el ejemplo, la interfaz debe ser moldeada en base al problema a solucionar.

Continuando con el hilo del artículo definiremos el código en el presentador. Éste, deberá tener una serie de consideraciones: la primera es que vamos a definir la propiedad `_view` de tipo de la interfaz antes mencionada, la que se encargara de contener la información de la vista (control de usuario); la siguiente, definiremos el constructor del presentador con un parámetro del mismo tipo que la propiedad y por último escribiremos el código de negocio necesario para solucionar la problemática en cuestión.

```
public class GridViewPresenter
{
    private IGridViewView _view;

    public GridViewPresenter(IGridViewView view)
    {
        this._view = view;
    }

    public DataTable LoadData()
    {
        try
        {
            ListViewByQuery listView = new
            ListViewByQuery();

            listView.List = _view.Web.Lists["Pages"];

            SPQuery query = new
            SPQuery(listView.List.DefaultView);

            query.ViewFields = "<FieldRef Name='Title'
            /><FieldRef
            Name='Author' /><FieldRef
            Name='Editor' />";
            listView.Query = query;

            SPUtility.TransferToErrorPage("Error in
            load data.");
            return null;
        }
        catch
        {
            SPUtility.TransferToErrorPage("Error in
            load data.");
            return null;
        }
    }
}
```

En cuanto al control de usuario (UserControl) de la WebPart, la idea es hacer que este implemente la interfaz antes definida. Para el ejemplo se debe implementar las dos propiedades Site y Web. La implementación se hará bajo el contexto de SharePoint por lo que podremos utilizar cualquiera de sus características, lo que nos permite hacer uso del importante `SPContext` y todas sus propiedades.

La pregunta ahora es, ¿dónde se unen todos los cabos? La respuesta a esta pregunta es en la creación del

presentar en el control de usuario. ¿De qué forma?, definiendo una propiedad `presenter` es instanciando dicha propiedad en el evento `init` del control de usuario.

```
public partial class GridViewWithMVPUserControl :
UserControl, IGridViewView
{
    #region Properties

    private GridViewPresenter presenter;

    public GridViewPresenter Presenter
    {
        get
        {
            return presenter;
        }
    }

    public new Microsoft.SharePoint.SPSite Site
    {
        get
        {
            return SPContext.Current.Site;
        }
    }

    public Microsoft.SharePoint.SPWeb Web
    {
        get
        {
            return SPContext.Current.Web;
        }
    }

    #endregion

    e) protected void Page_Load(object sender, EventArgs
    {
        gvList.DataSource = presenter.LoadData();
        gvList.DataBind();
    }

    protected override void OnInit(EventArgs e)
    {
        presenter = new GridViewPresenter(this);
        base.OnInit(e);
    }
}
```

Siendo ahora entonces la pregunta: ¿para qué nos sirven tener en cuenta estas aparentes complicaciones?

La respuesta es muy sencilla y fácil de comprender, el objetivo de este patrón es la de separar las aguas teniendo el código bien ordenado y fácil de leer, ya que quitamos prácticamente el código de la vista (control de usuario) y le pasamos la responsabilidad al presentador. Esto último a su vez nos permite una gran utilidad, que es la de realizar test unitario en SharePoint.

Juan Pablo Capdevila

Pragma consultores
jcapdevila@pragmaconsultores.com





Site Definitions vs. Web Templates

... nos deja a nosotros todo el control a la hora de definir cómo va a estar estructurado (por lo menos al principio) nuestro sitio en SharePoint ...

Trabajando en un proyecto de SharePoint 2010, me surgió la necesidad de crear la estructura de un sitio mediante un proyecto de Visual Studio 2010. Esto tiene muchísimas ventajas ya que nos deja a nosotros todo el control a la hora de definir cómo va a estar estructurado (por lo menos al principio) nuestro sitio en SharePoint, y no dependemos de hacer modificaciones manuales en los diferentes entornos (Desarrollo, Testing y Producción) a la hora de instalar nuestra solución.

Con esto no teníamos dudas frente a otras alternativas (como por ejemplo Feature Stapling), pero surgió la duda si implementar un Site Definition o un Web Template. Paso siguiente fue investigar un poco las ventajas y desventajas que traían aparejados cada uno e implementar algún que otro ejemplo.

Ambos nos dan la posibilidad de definir las listas, bibliotecas, webparts, páginas, features y configuraciones que va a tener el sitio cuando lo creamos, la principal diferencia está en que los Site Definitions se alojan en %ProgramFiles%\Common Files\Microsoft Shared\web server extensions\14\TEMPLATE\SiteTemplates y los Web templates se alojan en la base de datos y a su vez están basados en un sitio web existente.

Site definitions:

Los site definitions consisten en 2 archivos almacenados en el filesystem:

- WebTemp*.xml: Que describe cómo va a mostrarse el site definition para los usuarios finales.
- onet.xml: Que contiene la definición propiamente dicha del sitio.

Cuando hacemos una instalación del site definition solo podemos instalarlo a nivel de Farm ya que al estar dentro del filesystem, esta definición va a estar disponible (por defecto) para toda aplicación en la granja.

Una gran desventaja que tienen los site definitions es que no debemos cambiar el archivo onet.xml una vez que se creó un sitio con esa definición. Esto no está soportado y sirve para evitar futuros issues, así que ya saben, no lo

hagan. Lo que sí se puede hacer es colocar toda la personalización del sitio en features aparte, y en el onet.xml solo referenciar la activación del feature.

Otra gran desventaja es que no está soportado por los servicios en la nube de Microsoft.

Un ejemplo de Site Definition es el publicado mi artículo "TIPS PARA LA PERSONALIZACIÓN DE MY SITES EN SHAREPOINT 2010" en el número 8 de la revista CompartiMOSS, donde creamos una definición de sitio para My Sites. También fácilmente pueden crear un site definition en el Visual Studio 2010 ya que podemos encontrar el tipo de proyecto en la página de "nuevo proyecto".

Web Templates:

Los web templates también constan (cada uno) de 2 archivos:

- Elements.xml: donde se define (entre otras cosas) el web template y de cual existente nos basamos.
- Onet.xml: Que contiene la información propiamente dicha del sitio.

Una vez que contamos con estos dos archivos creados, los vamos a asociar a un feature que va a ser el encargado de instalar el Template.

Este feature puede activarse en dos scopes diferentes Farm, donde el template va a estar disponible para toda la granja, y Site, donde se activa para una determinada colección de sitios.

Como dijimos antes, la gran ventaja ante los site definitions, es que los web templates no guardan ningún archivo en el filesystem por lo que son soportados en los servicios en la nube de Microsoft. También esto permite que el web template pueda actualizarse ya que el onet.xml solo se usa cuando se crea el sitio.

Otra gran ventaja es que podemos utilizar los features de publicación en un Web template, lo cual nos abre un gran abanico de





posibilidades a la hora de crear nuestra solución.

Cabe destacar que hay algunos elementos que no se pueden activar en un web template, pero pueden simularse activando (en el onet.xml del web template) otros features que usen estos elementos.

- Módulos
- Componentes
- ServerEmailFooter

Como desventajas podemos decir que nos podemos encontrar con dificultades a la hora de crear una jerarquía de sitios mediante el uso de web templates (aunque podemos encontrar workarounds mediante el uso de Powershell) y a la hora de usar variations para el multilinguaje.

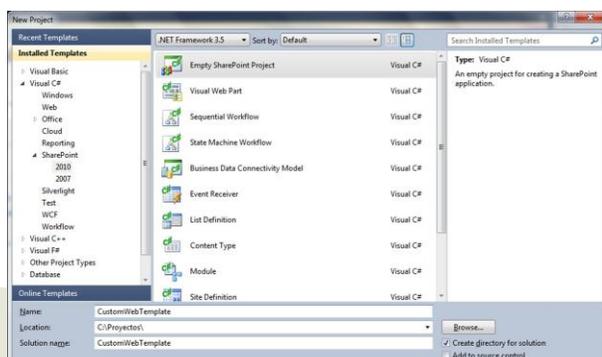
Conclusiones

Según mi experiencia y lo que recomienda Microsoft para seguir teniendo sitios compatibles con futuras versiones debemos intentar usar web templates ya que:

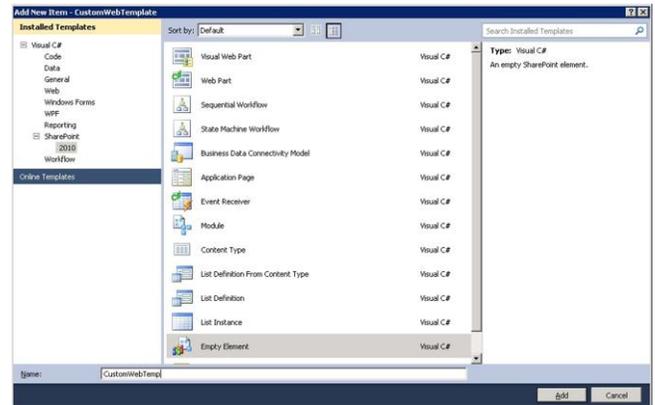
- Son fáciles de crear
- Son fáciles de instalar
- Se pueden usar en contextos donde no se tenga acceso al filesystem (por ejemplo la Nube)
- Se pueden modificar sin afectar sitios existentes.
- Se puede incluir casi cualquier cosa que pueda hacerse mediante la interfaz de usuario.

Pasos para crear un Web Template:

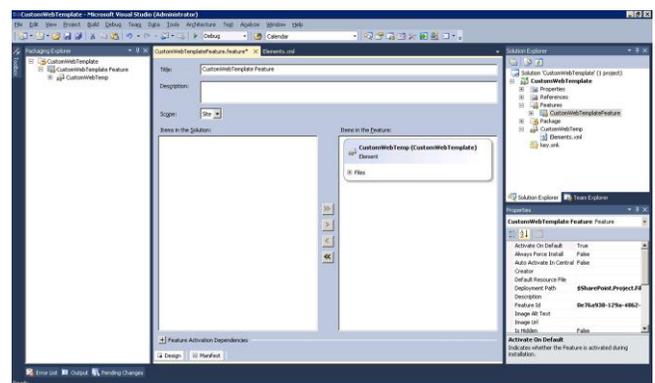
Para crear un web template lo primero que tenemos que hacer es abrir el visual studio y crear un Empty SharePoint Project. Le ponemos el nombre CustomWebTemplate y lo instalaremos como una Farm Solution.



Una vez que tenemos el proyecto vacío, agregamos un Empty Element y lo llamamos CustomWebTemp.



Al agregar el Empty Element se crea el feature al cual, según nuestro gusto le cambiamos el nombre para reflejar el Web Template que vamos a construir. Como scope, a modo de ejemplo, le colocamos Site.



En el archivo Elements.xml definimos el tipo de web template que vamos a crear, como ejemplo usamos el siguiente:

```
<?xml version="1.0" encoding="utf-8"?>
<Elements
  xmlns="http://schemas.microsoft.com/sharepoint/">
  <WebTemplate
    Name="CustomWebTemp"
    BaseTypeID="1"
    BaseTypeName="STS"
    BaseConfigurationID="0"
    Title="Web Template Personalizado"
    Description="Ejemplo de web Template"
    DisplayCategory="Custom"
    Locale="3082"
    xmlns="http://schemas.microsoft.com/sharepoint/"
  />
</Elements>
```

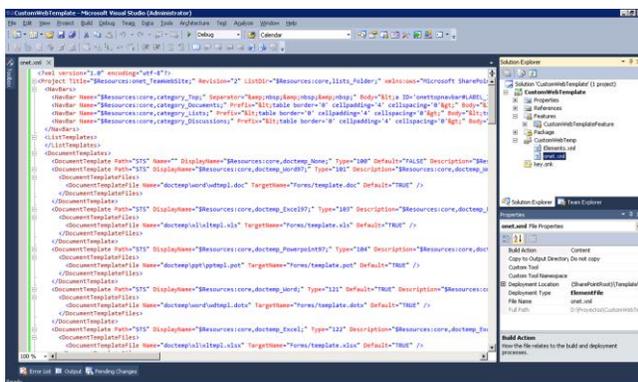




Es importante que el nombre del WebTemplate (en este caso CustomWebTemp) coincida con el nombre que le pusimos al EmptyElement.

Ahora lo siguiente por hacer es crear el onet.xml. Para esto recomiendo basarse en un onet.xml ya existente, como por ejemplo el que se encuentra en C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\14\TEMPLATE\SiteTemplates\sts\xml.

Creamos un nuevo archivo, le colocamos el nombre onet.xml y como Deployment Type colocamos ElementFile. Luego pegamos el contenido del archivo onet.xml copiado.



Aquí es donde podemos editar el onet.xml y agregar los features u otras características necesarias para personalizar el web template.

Cabe aclarar de nuevo que hay elementos que no son soportados por los web templates, por lo que deberíamos eliminar estas secciones del onet.xml a efectos de este artículo

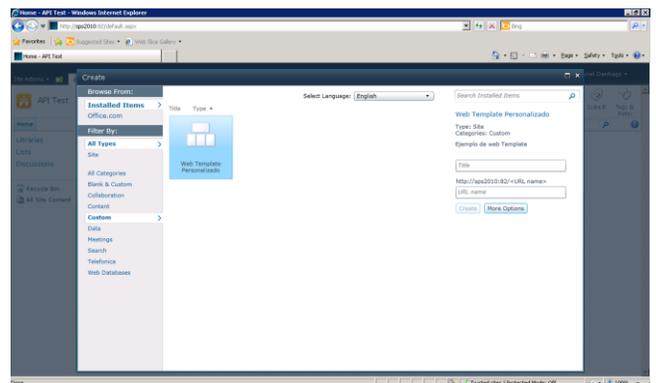
- Document templates
- Otras configuraciones que no tengan id=0

- Módulos
- ServerEmailFooter

Yo recomiendo agregar otros proyectos a la solución que tengan por ejemplo, definiciones de listas, webparts, páginas, etc. y agregar sus correspondientes Features Ids en el onet.xml del web template.

Por ultimo solo nos queda instalar la solución. Para eso presionamos botón derecho y seleccionamos deploy.

Si todo salió bien, vamos a nuestro Site Collection y podemos ver que, al crear un nuevo sitio, tenemos disponible nuestro nuevo Web Template.



Espero les haya servido!

Carlos Ariel Dantiags
carlos.dantiags@aycron.com
www.aycron.com
www.facebook.com/dantiags
 @dantiags





Herramientas de Comunidad

Visual Studio 2010 SharePoint Power Tools

Sitio:
<http://visualstudiogallery.msdn.microsoft.com/8e602a8c-6714-4549-9e95-f3700344b0d9/>

Creado por: Microsoft

Productos: Visual Studio 2010, SharePoint Foundation 2010, SharePoint Server 2010

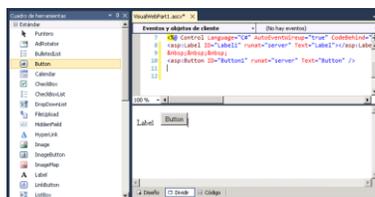
Propósito: Agrega una plantilla para crear WebParts Visuales que pueden ser utilizadas en el SandBox y una extensión que muestra errores cuando se utilizan tipos u objetos no permitidos en aplicaciones de SandBox

Instalación: Ejecute el archivo de instalación en el servidor que contiene Visual Studio 2010. Acepte las condiciones de utilización

Utilización: La plantilla para crear WebParts Visuales es una de las plantillas de Elemento de Proyecto (SPI, SharePoint Project Item). En un proyecto de Visual Studio 2010 de SharePoint 2010, vaya a Agregar - Nuevo elemento - Visual Web Part (Sandboxed)



La WebPart puede ser creada y codificada de la misma forma que una WebPart Visual para la granja:



Por ejemplo, si se utiliza una Label y un Botón como en la imagen anterior, el siguiente código puede ser utilizado:

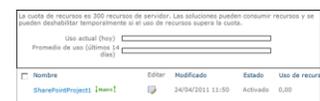
```
using System;
using System.ComponentModel;
using Microsoft.SharePoint;
```

```
namespace SharePointProject1.VisualWebPart1
{
    [ToolboxItem(false)]
    public partial class VisualWebPart1 :
        System.Web.UI.WebControls.WebParts.WebPart
    {
        protected override void OnInit(EventArgs e)
        {
            base.OnInit(e);
            InitializeControl();
        }

        protected void Page_Load(object sender,
            EventArgs e)
        {}

        protected void Button1_Click(object sender,
            EventArgs e)
        {
            using (SPSite mySite = new
                SPSite(SPContext.Current.Site.Url))
            {
                using (SPWeb myWeb =
                    mySite.OpenWeb())
                {
                    Label1.Text = "Hola " +
                        myWeb.CurrentUser.LoginName;
                }
            }
        }
    }
}
```

Después de implementar la WebPart desde Visual Studio (no olvide iniciar el "Servicio de código de espacio aislado de Microsoft SharePoint Foundation"), la WebPart se encuentra en la Galería de Soluciones del sitio seleccionado:



La plantilla convierte la página aspx de la WebPart Visual (que no puede ser instalada en el sistema de archivos de SharePoint porque es una solución de Sandbox) en código dinámico creado en tiempo de ejecución por el compilador.

La extensión funciona silenciosamente dentro de Visual Studio 2010 y filtra todos los objetos que no pueden ser utilizados en código de Sandbox. En el ejemplo anterior, si se intenta utilizar SPControls.GetWeb(), Visual Studio no muestra el objeto en su sistema de IntelliSense y si se escribe manualmente, el compilador muestra un error de compilación (si no se utiliza la extensión, Visual Studio permite ver el objeto en IntelliSense y compilarlo, y el error solo ocurrirá en el momento de utilizar la WebPart en SharePoint).

Herramientas creadas por y para la Comunidad de SharePoint...

Gustavo Velez





Modelo de Objetos de Cliente

Introducción

Trabajar con SharePoint implica conocer un conjunto de tecnologías y enfoques de programación que necesitan ser balanceados para obtener los mejores resultados y la mejor inversión del presupuesto de nuestros clientes. Muchas veces debemos combinar alternativas como:

- Configuración de SharePoint
- SharePoint Designer
- Visual Studio
- Herramientas (Excel Services o InfoPath por ejemplo)

Desde el punto de vista de programación, en la versión 2007 ya contábamos con un robusto modelo de objetos para programar en el servidor. Sin embargo el trabajo se volvía algo más complicado cuando necesitábamos interactuar con SharePoint desde el cliente, o cuando los administradores de SharePoint no nos permitían instalar componentes en el servidor, por el riesgo de desestabilizar la granja. Así surgieron gurúes y especialistas que se caracterizaban por:

- Aprovechar los servicios web de SharePoint, no muy amigables.
- Hacer maravillas con JavaScript y jQuery.
- Hacer cosas increíbles con SharePoint Designer.

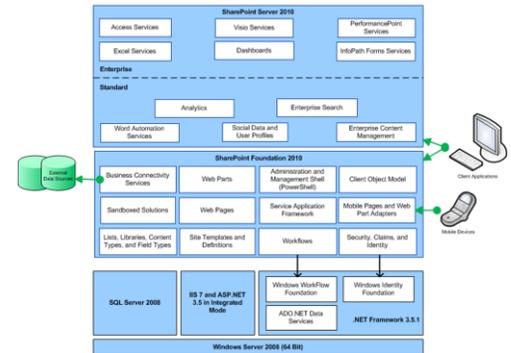
Estos métodos por momentos daban muy buenos resultados, otras veces no llegaban a convencer.

Pero esos días han terminado para el programador de SharePoint. Porque en la versión 2010 tenemos muchas mejoras y una de las que me parece más interesante (no es la única, ver imagen más abajo) es el Modelo de Objetos de Cliente de SharePoint 2010.

Sobre él, haremos hoy una breve introducción, simplemente para que el lector comience a imaginar todo lo que puede hacer...

Qué lo disfruten!

SharePoint 2010 Development Platform Stack



Fuente de esta imagen:
<http://msdn.microsoft.com/en-us/library/gg552610.aspx>

El modelo de objetos de cliente de SharePoint 2010

Lo primero que debemos saber es que el modelo de objetos de cliente de SharePoint 2010 es algo nuevo e introduce tres variantes:

- Administrado
- Silverlight
- JavaScript

Cada una de estas variantes está contenida en una librería particular:

- Administrado:
Microsoft.SharePoint.Client.dll y Microsoft.SharePoint.ClientRuntime.dll (carpeta ISAPI)
- Silverlight:
Microsoft.SharePoint.Client.Silverlight.dll y Microsoft.SharePoint.Client.Silverlight.Runtime.dll (carpeta LAYOUTS\ClientBin)
- JavaScript: SP.js (carpeta LAYOUTS)

Los tres modelos interactúan con SharePoint a través del servicio Client.svc. Los desarrolladores programamos entonces contra el modelo de objetos de cliente, pero las operaciones son enviadas en forma de lote como XML al servicio Client.svc.

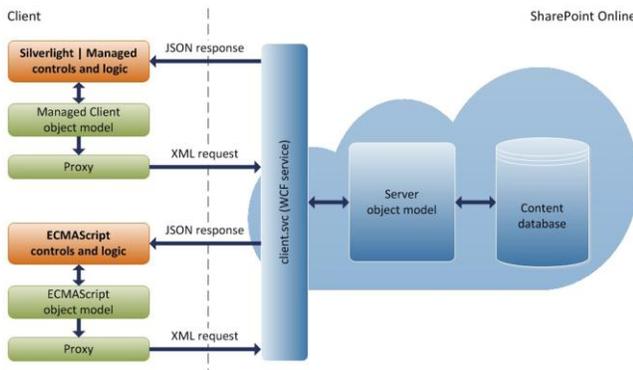
... el modelo de objetos de cliente de SharePoint 2010 es algo nuevo e introduce tres variantes...





El servicio Client.svc ejecuta las operaciones contra el modelo de objetos del servidor y los resultados son enviados nuevamente al cliente en la forma de JavaScript Object Notation (JSON).

La siguiente imagen explica cómo se implementa (en SharePoint Online por ejemplo):



Fuente de esta imagen: <http://msdn.microsoft.com/en-us/library/gg317460.aspx>

Hay mucho, mucho más. Pero prefiero detenerme acá con la teoría y pasar a un ejemplo terrenal. Ya habrá otros artículos para profundizar o ustedes tendrán tiempo de consultar MSDN o algún libro.

¿MSDN? Aquí el enlace: <http://msdn.microsoft.com/en-us/library/ee537247.aspx>.

Un caso sencillo

Hasta antes de SP 2010, acceder al modelo de objetos de SharePoint desde JavaScript no era una opción. Imaginen sólo unos minutos todo lo que podemos hacer ahora a nivel de interfaces de usuario. Adicionen a eso que en principio no necesitamos ningún permiso especial del administrador de la granja de SharePoint, lo cual también significa que el modelo de objetos de cliente es más limitado que el del servidor.

Voy a darles un ejemplo muy sencillo para que vean el potencial de este modelo. Espero en próximos artículos mostrar ejemplos más complejos, pero siempre es conveniente arrancar por algo parecido al "hola mundo".

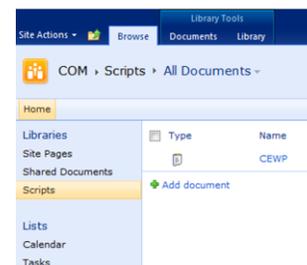
En este caso vamos a:

1. Insertar un elemento web de edición de contenido (CEWP).
2. Leer los ítems de una lista del sitio raíz utilizando el modelo de objetos de cliente para JavaScript.
3. Aplicar algo de estilo a través de CSS.

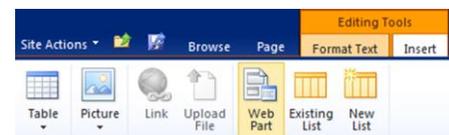
Sí, leyeron bien, vamos a leer datos de SharePoint a través de su modelo de objetos desde dentro de la simple CEWP.

Paso 1: insertando la CEWP

Vamos a repasar rápidamente como insertar una CEWP, ya que hay algunos cambios (mejoras) en SharePoint 2010. Lo más interesante es que el contenido de la CEWP lo podemos tener almacenado en un archivo de texto, lo cual facilita el mantenimiento del código. En nuestro caso vamos a agregar un archivo llamado "CEWP.txt" dentro de una librería de documentos llamada "Scripts" tal como muestra la siguiente imagen:



Luego vamos a la página principal de nuestro sitio, hacemos clic en "Page" en la cinta y luego "Edit" para editar la página. Dentro de "Editing Tools", hacemos clic en "Insert" y luego en "Web Part":

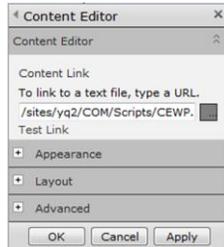


Luego elegimos "Media and Content" / Content Editor" e insertamos el elemento web:





Finalmente editamos las propiedades del elemento web, en donde completaremos la ruta y el nombre del archivo que contiene el código de nuestra CEWP.



Paso 2: codificando

Abrimos nuestro archivo "CEWP.txt" desde algún editor, como puede ser el humilde Bloc de Notas, directamente desde la librería de SharePoint, insertamos el código que nuestro a continuación y luego grabamos.

Nota: más adelante explicaré el código.

```
<div id="div1"></div>

<script type="text/javascript">

ExecuteOrDelayUntilScriptLoaded(cargar_productos,
"sp.js");

function cargar_productos() {
contexto = new SP.ClientContext.get_current();
var lista =
contexto.get_site().get_rootWeb().get_lists().getByTitle('Productos');
var query = new SP.CamlQuery();
query.set_viewXml('<View><ViewFields><FieldRef Name="Title"/></ViewFields></View>');
this.items = lista.getItems(query);
contexto.load(items);
contexto.executeQueryAsync(Function.createDelegate(this, this.cargaok),
Function.createDelegate(this, this.cargamal));
}

function cargaok(sender, args) {
var itemslista = this.items.getEnumerator();
var strHtml = "<table id='box-table-a'><thead><tr><th scope='col' id='box-table-a'>Productos</th></tr></thead><tbody>";

while (itemslista.moveNext()) {
var elitem = itemslista.get_current();
strHtml += "<tr><td>";
strHtml += elitem.get_item('Title');
strHtml += "</td></tr>";
}

strHtml += "</tbody></table>";
document.getElementById("div1").innerHTML = strHtml;
}
}
```

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

```
</script>
```

```
<style>
```

Aquí agregamos los estilos que más nos gusten. En el ejemplo, yo he tomado los estilos propuestos en este enlace:

<http://coding.smashingmagazine.com/2008/08/13/top-10-css-table-designs/>

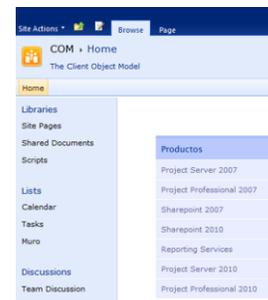
Estos estilos mapean con los identificadores creados en el código JavaScript anterior: id='box-table-a' para los <table> y <th>

Nota: no es este el lugar ideal para agregar estilos, pero prefiero no agregar complejidad en este ejemplo y que además el lector vea el potencial de la CEWP.

```
</style>
```

Paso 3: el resultado

Una vez que hayamos pegado nuestro código en la página principal de nuestro sitio, hacemos clic en "Home" y luego vemos los resultados, tal como grafica la siguiente imagen. Observen como sin ningún código de servidor hemos hecho una consulta a datos residentes en una lista de SharePoint:



Paso 4: entendiendo el código

Bien, vamos ahora a analizar un poco el código. Recuerden que estamos viendo un caso particular de modelo de objetos de cliente para javascript, el código puede variar si el JavaScript está por ejemplo en un página de aplicación, o si usamos Silverlight o una aplicación con código .net administrado.

Empecemos por analizar la primera línea importante de código:



```
ExecuteOrDelayUntilScriptLoaded(cargar_productos, "sp.js");
```

Esta es la instrucción que utilizamos para garantizar que SP.js esté cargada antes de que nuestra función comience a acceder a los datos de SharePoint a través del modelo de objetos de cliente.

Más información en <http://msdn.microsoft.com/en-us/library/ff411788.aspx>.

Lo segundo es obtener el contexto. A diferencia de Silverlight y el código administrado, no necesitamos especificar la URL:

```
contexto = new SP.ClientContext.get_current();
```

En la siguiente porción de código vamos a acceder a una lista. Si bien no podemos acceder a una lista de otra colección de sitios, si podemos acceder a una lista del sitio raíz :-), que es lo que estamos haciendo en este caso:

```
var lista = contexto.get_site().get_rootWeb().get_lists().getByTitle('Productos');
```

Luego entramos en un terreno un poco más conocido, preparamos una consulta CAML, en este caso para extraer el campo título de todos los ítems de esta lista:

```
var query = new SP.CamlQuery(); query.set_viewXml('<View><ViewFields><FieldRef Name="Title"/></ViewFields></View>'); this.items = lista.getItems(query); contexto.load(items);
```

Por último, vamos a pedir al servidor que ejecute nuestra petición. En JavaScript sólo podemos hacerlo en forma asíncrona, motivo por el cual debemos indicarle el nombre de las funciones que llamaremos si nuestra ejecución es exitosa o no:

```
contexto.executeQueryAsync(Function.createDelegate(this, this.cargaok), Function.createDelegate(this, this.cargamal));
```

Más información en: <http://msdn.microsoft.com/en-us/library/microsoft.sharepoint.client.clientruntimecontext.executeQueryasync.aspx>

En caso que nuestro código se ejecute en forma correcta, lo que haremos es recorrer la lista y mostrar los resultados en una tabla HTML:

```
function cargaok(sender, args) { var itemslista = this.items.getEnumerator(); var strHtml = "<table id='box-table-
```

```
a'><thead><tr><th scope='col' id='box-table-a'>Productos</th></tr></thead><tbody>";
```

```
while (itemslista.moveNext()) { var elitem = itemslista.get_current(); strHtml += "<tr><td>"; strHtml += elitem.get_item('Title'); strHtml += "</td></tr>"; }
```

```
strHtml += "</tbody></table>"; document.getElementById("div1").innerHTML = strHtml; }
```

En caso de error, mostramos el mensaje por pantalla:

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

Ahí termina nuestro código. Es un código simple en el que me tomé algunas licencias principalmente para que puedan ver el potencial del modelo de objetos de cliente, y como dije anteriormente, imaginar todo lo que podrían hacer...

Conclusiones

Creo que el modelo de objetos de cliente es uno de los cambios más importantes que la versión 2010 de SharePoint implementa frente a su antecesor en cuanto a sus capacidades de desarrollo. No es el único, también están las nuevas capacidades de flujos de trabajo en SharePoint Designer, Power Shell (cool), la integración con Visual Studio y las Sandboxed solutions (más cool), por nombrar sólo algunas.

Como siempre digo, SharePoint es una excelente plataforma para construir aplicaciones, y cada vez lo es más, afortunadamente para los que nos dedicamos a esto.

Hasta la próxima!

Juan Pablo Pussacq Laborde

<http://jpussacq.me/>

<http://twitter.com/jpussacq>

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



SharePoint Communities



¿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 2007 y/o Microsoft Office SharePoint 2010 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:

fabiani@siderys.com.uy
jgonzalez@gruposodercan.es
gustavo@gavd.net
compartimoss@hotmail.com

