

COMPARTIMOSS

Revista especializada en Tecnologías SharePoint



ENTREVISTA

IBERIAN SHAREPOINT CONFERENCE

MONITOREANDO SHAREPOINT CON SCOM

DISEÑO GRÁFICO EN SHAREPOINT FOUNDATION 2010

COMPARTIMOSS

Revista especializada en Tecnologías SharePoint

STAFF

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

DIRECCIÓN GENERAL

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

Contactate con nosotros

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

BLOGS

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

REDES SOCIALES

Facebook:
<http://www.facebook.com/group.php?gid=128911147140492>
Linkedin:
<http://www.linkedin.com/groups/CompartiMOSS-3776291>
Twitter:
twitter.com/CompartiMOSScom

VISÍTENOS:

<http://www.compartimoss.com>

Contenidos

03

Editorial

La revista quiere cuidar a sus autores...

04. ASP.NET MVC para SharePointeros (II)

09. Resultados y experiencias del 1º Office & SharePoint App Challenge de España

13. Páginas de Publicación en SharePoint Server 2013

La relación existente entre el historial de versiones de las Páginas de Publicación

16

Licenciamiento de SharePoint 2013

Tradicionalmente SharePoint a licenciado sus diferentes versiones de forma que todos los usuarios tienen acceso a solamente a una versión específica.

19. Mibebeyyo.com

Una implementación de un portal web público.

26. Entrevista

A Pablo Pussacq Laborde.

28. Monitoreando SharePoint Server

Con System Center Operations Manager (SCOM) – Parte 1

30. Creación de aplicaciones de Windows Phone

Con contenido de SharePoint

37. Agregarle comentarios a las páginas de SharePoint Server

39

¿Cómo se hizo CompartiMOSS?

segunda parte del artículo sobre cómo se hizo CompartiMOSS.com con SharePoint 2013

44. SharePoint Designer 2013 y Flujos de Trabajo

48. Como configurar el Panel de Refinamiento

Sobre columnas de Metadatos múltiples

51. Diseño Gráfico en SharePoint Foundation 2010

57. 1ª Conferencia Ibérica de SharePoint

59. Uso de Frameworks JavaScript en desarrollos SharePoint

La revista quiere cuidar a sus autores y con este número 16, estrenamos una página en la web dedicada a ellos (www.compartimoss.com/autores). En ella, daremos a conocer al motor que hace que en cada número tengamos artículos, a los verdaderos protagonistas que nos obligaron, indirectamente, a crecer con la versión web, y a mejorar nuestra presencia en las redes sociales con una cuenta de Twitter (@CompartiMOSScom) y un grupo en LinkedIn.

La primera versión de nuestra App para SharePoint, junto con la versión de Windows Phone y Windows 8, no es más que un pequeño paso para mejorar la visibilidad de la revista. Nuestro enfoque es que podamos leer los artículos desde cualquier formato, inclusive, en modo offline, con lo que llegarán actualizaciones de estas aplicaciones y otras nuevas que están en camino.

Aprovecho esta editorial para presentarme y agradecer la confianza depositada en mí, para continuar mi labor en la revista, dentro del staff directivo, que desde principio de este año se ha estrechado con el nacimiento de la web y las Apps. Soy Alberto Diaz Martin, MVP de SharePoint y autor de diversos artículos en esta revista y en el mundo

hispanohablante, además de un asiduo conferenciante de la tecnología .NET y en particular de SharePoint. Espero que mi incorporación sea un paso más dentro del camino que hemos trazado para CompartiMOSS y el compromiso hacia nuestros lectores y patrocinadores.

Les dejamos con el número 16, cargado de artículos con mucho valor, nuevos autores y con el apoyo de los patrocinadores, que también ayudan a que tengamos una mejor revista.

Esperamos que disfruten de este número tanto como nosotros disfrutamos creándola.

ALBERTO DÍAZ

adiazcan@hotmail.com

JUAN CARLOS GONZÁLEZ

jgonzalez@gruposodercan.es

GUSTAVO VÉLEZ

gustavo@gavd.net

FABIÁN IMAZ

fabiani@siderys.com.uy



ASP.NET MVC para SharePointeros (II)

Resumen

En este artículo seguiremos avanzando con ASP.NET MVC, introduciendo las actualizaciones y validaciones de datos.

Artículo

En el artículo previo vimos las bases de la programación de aplicaciones web en ASP.NET MVC conectadas a SharePoint mediante el modelo de objetos de cliente. La aplicación que acabamos haciendo mostraba los detalles de una lista de productos alojada en SharePoint. Sin embargo, nuestra aplicación era de sólo lectura, sin posibilidad de modificar los productos o añadir productos nuevos.

Edición en MVC

La edición de entidades de negocio (el “modelo”) en MVC suele ir dividida en dos acciones en el controlador (recordemos que el controlador es la clase donde está la lógica de las acciones de las diferentes páginas). Las dos acciones se llaman igual y la diferencia es que una se activa al realizar una petición GET y la otra una petición POST. La primera acción Edit (GET) sirve para “poner en edición”, mostrando una vista con controles de edición como las cajas de texto, opciones u otros mientras la segunda acción Edit (POST) recoge los valores editados al hacer clic en el botón “Guardar”.

Vayamos a nuestra aplicación de la entrega anterior, llamada MvcHelloWorld. La podéis descargar de <http://sdrv.ms/XRmW00>.

Añadiremos dos acciones a la clase ProductsController.cs. La primera acción es la que activará la edición de un producto basándose en su identificador id. De hecho, esta acción es la misma que la de Display de la entrega anterior: va a la clase auxiliar DBHelper a buscar el producto en SharePoint y luego muestra la vista de ese producto.

```
public ActionResult Edit(int id)
{
    var product =
    DBHelper.GetProduct(id);
    return View(product);
}
```

La segunda acción es la que recogerá los valores editados y actualizará el producto (invocando el método SaveProduct dentro de DBHelper). Todavía no tenemos este método pero lo vamos a hacer en seguida. Más adelante también veremos cómo MVC es capaz de pasarle una instancia de la clase Product a esta acción y nos ocuparemos de la misteriosa llamada RedirectToAction. Observad que decoramos esta acción con el atributo HttpPost, para indicar que sólo se tiene que activar al recibir una petición POST.

```
[HttpPost]
public ActionResult
Edit(MvcHelloWorld.Models.Product
product)
{
    if (ModelState.IsValid)
    {
        DBHelper.SaveProduct(product);
        return
        RedirectToAction("Index");
    }
    return View(product);
}
```

La edición de un elemento de lista en SharePoint mediante el modelo de objetos de cliente se realiza cargando ese elemento en una variable, con el método List.GetItemById(), cambiando las propiedades y luego llamando al ClientContext.ExecuteQuery(). Vamos a agregar el método que falta al DBHelper:

```
public static void SaveProduct(Product
p)
{
    ClientContext clientContext =
    PrepareClientContext();

    List list =
    clientContext.Web.Lists.GetByTitle("Pro
ducts");
    clientContext.Load(list);
    ListItem listItem =
```



```
list.GetItemById(p.ProductId);
clientContext.Load(listItem);

listItem["Title"] = p.Name;
listItem.Update();

clientContext.ExecuteQuery();
}
```

Enlazar la vista de edición

Hasta ahora, no tenemos la edición en marcha. Tenemos listo el controlador y la parte de SharePoint, pero nos faltan dos cosas más: la vista de edición propiamente dicha y como llamar a la acción Edit desde el listado de productos.

El segundo problema es el más fácil de resolver. En MVC las llamadas a acciones se renderizan con la sintaxis `@Html.ActionLink` de Razor. Si editamos la vista de la acción Index (Index.cshtml), veremos que estamos construyendo una lista no ordenada de elementos, y que cada elemento es un enlace a la acción Display pasándole el parámetro ID del elemento de la lista en cuestión.

```
<ul>
  @foreach (var p in Model) {
    <li>
      @Html.ActionLink(p.Name,
        "Display", new { id=p.ProductId })
    </li>
  }
</ul>
```

De la misma manera, usaremos ActionLink para generar el enlace de edición de un producto, al lado del enlace de la acción Display:

```
<ul>
  @foreach (var p in Model) {
    <li>
      @Html.ActionLink(p.Name,
        "Display", new { id=p.ProductId })
      &nbsp;&nbsp;&nbsp;
      @Html.ActionLink("Edit",
        "Edit", new { id=p.ProductId })
    </li>
  }
</ul>
```

Recordad que ActionLink tiene tres parámetros: el texto del enlace, el nombre de la acción y los parámetros de la acción. Además, en la sintaxis de Razor, todo lo que escribimos es HTML literal excepto lo que sigue a los marcadores `@`. Por eso ponemos los corchetes para que la acción Edit salga separada del enlace a los detalles del producto:

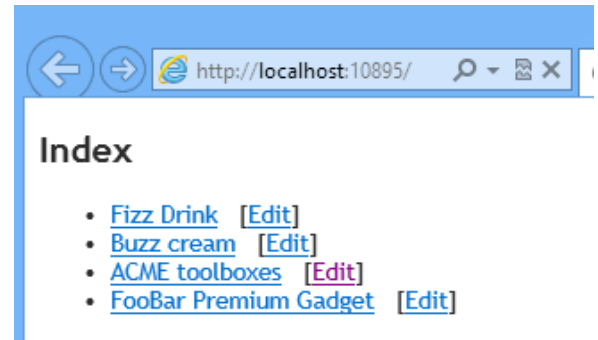


Imagen 1.- Ya podemos editar los productos.

“Scaffolding”

Nos queda pendiente realizar la vista de edición. Podemos hacerla a mano, como en la entrega anterior, o bien usando “scaffolding”. Se trata de una característica de MVC en Visual Studio que nos permite generar la vista de manera automática, ahorrando tiempo y minimizando la posibilidad de error.

Para crear la vista, igual que antes, haremos clic con el botón derecho encima de la acción del controlador y elegiremos “Add View”:

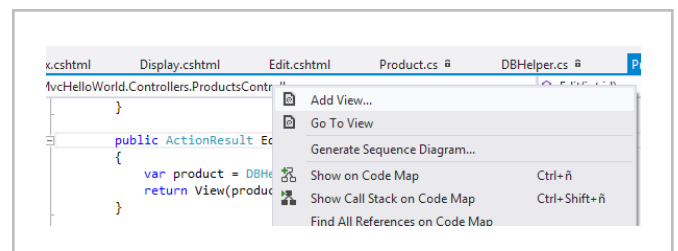


Imagen 2.- Agregar la vista a la acción Edit.

En el diálogo que se nos presenta, marcaremos la vista como fuertemente tipada (para que podamos acceder a la clase Product desde la vista) y en el apartado “Scaffold template” elegiremos “Edit”. Desmarcaremos la opción de referencias los ficheros de script, ya que no hacemos nada con ellos.

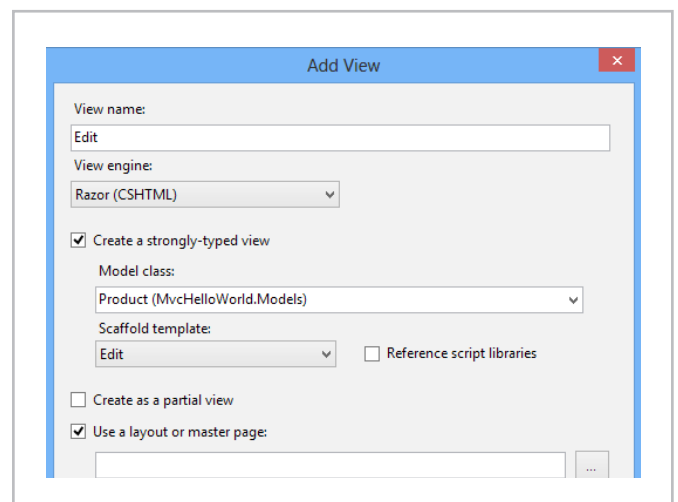


Imagen 3.- Scaffolding de la vista de edición.

La vista generada con scaffolding tiene la pinta siguiente:

```
@model MvcHelloWorld.Models.Product

@{
    ViewBag.Title = "Edit";
}

<h2>Edit</h2>

@using (Html.BeginForm()) {
    @Html.ValidationSummary(true)
    <fieldset>
        <legend>Product</legend>

        @Html.HiddenFor(model =>
            model.ProductId)

        <div class="editor-label">
            @Html.LabelFor(model =>
                model.Name)
        </div>
        <div class="editor-field">
            @Html.EditorFor(model =>
                model.Name)

            @Html.ValidationMessageFor(model =>
                model.Name)

            <p>
                <input type="submit"
                    value="Save" />
            </p>
        </div>
    </fieldset>
}

<div>
    @Html.ActionLink("Back to List",
        "Index")
</div>
```

Como podéis ver, scaffolding nos genera un formulario HTML con un botón de Submit que lo que hará es realizar una petición POST de la acción Edit (por ello tenemos la segunda acción en el controlador, activada por POST). La llamada POST enviará al controlador el valor de todos los controles que hay en la vista, en nuestro caso ID y Name del producto. El formulario está encapsulado dentro de una instrucción `using(Html.BeginForm())`.

La "magia" de MVC convertirá estos valores en una instancia de clase `Product` en la segunda acción `Edit`. Esto se realiza a través de un componente llamado `Model Binder`, que veremos en otras entregas de esta serie de artículos. Por ahora, lo que nos interesa saber es que el `Model Binder` por defecto sabe cómo mapear la vista al modelo y viceversa.

Además, tenemos un control para mostrar el resumen de las validaciones (las veremos a continuación) y los controles para cada propiedad del modelo se hacen mediante

llamadas a `EditorFor`, `HiddenFor` y `LabelFor`. Estos métodos auxiliares de `@Html` generan el control HTML adecuado para cada tipo de propiedad (cajas de texto para strings y números, checkboxes para booleanos, etc). Observad que el ID del producto se muestra oculto, pero se envía en la edición al servidor.

Si ahora ejecutamos la aplicación, veremos que podemos editar los nombres de los productos:

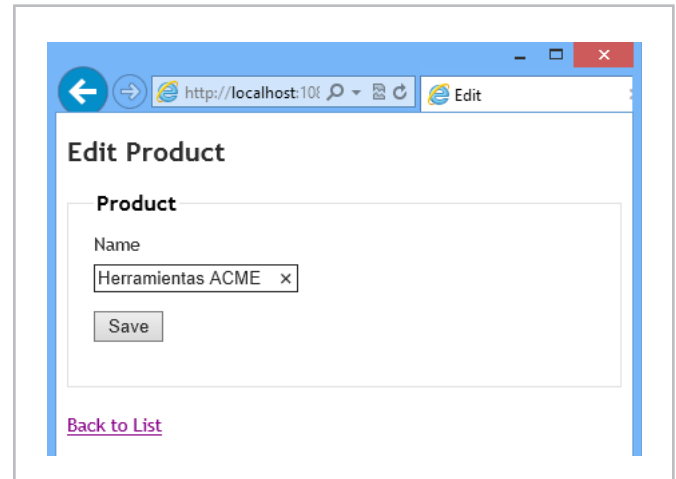


Imagen 4.- La vista de edición en funcionamiento.

Validaciones en MVC

Para que nuestra aplicación de ejemplo sea más realista, vamos a añadir la validación del nombre del producto. Pondremos una longitud máxima para esta propiedad.

Validación con DataAnnotations

En MVC, podemos extender nuestro modelo con atributos llamados `data annotations` (espacio de nombres `System.ComponentModel.DataAnnotations`). Estos atributos indican que las propiedades de la clase tienen ciertas restricciones que se validarán al crear o editar las instancias de la clase.

Vamos a añadir dos atributos a la clase `Product`: `Required` y `StringLength`, para especificar que la propiedad no puede estar vacía y que su longitud máxima es de 50 caracteres:

```
using
System.ComponentModel.DataAnnotations;
...
[StringLength(50), Required]
public string Name { get; set; }
```

Sólo con este pequeño cambio ya tenemos la validación activada. La acción `Edit` del controlador tiene una comprobación de la validez del modelo, usando la propiedad `ModelState.IsValid`, antes de proceder con el guardado del producto. Si la validación es correcta, invocamos la acción `Index` para ver el listado de los productos usando el método `RedirectToAction`. En el caso contrario, le devuelve la vista para que se muestre el mensaje de error:

```

if (ModelState.IsValid)
{
    DBHelper.SaveProduct(product);
    return RedirectToAction("Index");
}
return View(product);

```

Ejecutando la aplicación, comprobamos que la validación se está disparando correctamente. En este caso el mensaje de validación es automático y si lo queremos cambiar hay que agregar el parámetro ErrorMessage al atributo de validación dentro de la clase Product.

Edit Product

Product

Name

The Name field is required.

Save

[Back to List](#)

Imagen 5.- La validación de la edición.

Validación remota

Algunas veces no podemos validar la edición de un objeto sin realizar alguna acción externa. Por ejemplo, para validar que no haya productos con el nombre duplicado tenemos que realizar una búsqueda de todos los productos. Para dar soporte a este tipo de validaciones, MVC nos permite realizar la validación remota.

La validación remota necesita tener referenciadas las librerías de JavaScript y lo que hace es realizar una llamada JSON a una acción de nuestra elección. Si la acción devuelve true, la validación es correcta. Si no, se muestra el mensaje de error.

En nuestro caso añadiremos un controlador nuevo para la validación, llamado ValidationController, con la acción NameAlreadyExists:

```

[OutputCache(Location =
OutputCacheLocation.None, NoStore =
true)]
public class ValidationController :
Controller
{
    public JsonResult
NameAlreadyExists(string name)
    {
        var products =
DBHelper.GetProducts();
        int howMany =

```

```

products.Where(x => x.Name ==
name).Count();

        if (howMany == 0)
        {
            return Json(true,
JsonRequestBehavior.AllowGet);
        }
        else
        {
            return Json("There is
another product with the same name",
JsonRequestBehavior.AllowGet);
        }
    }
}

```

El código del controlador es muy sencillo y lo que hace es comprobar que no existan productos con el nombre que se le está intentando asignar al producto en edición. Notad también que le decimos a MVC que nunca almacene el resultado de la llamada en la caché y asegurar que siempre se llamará al servidor.

Ahora tenemos que añadir el atributo de validación Remote a la propiedad Name de la clase Product. Allí le especificamos que acción de que controlador vamos a ejecutar. El nombre y el tipo de la propiedad a validar tienen que coincidir con los de la acción validadora:

```

public class Product
{
    public int ProductId { get;
set; }

    [StringLength(50), Required]
    [Remote("NameAlreadyExists",
"Validation")]
    public string Name { get; set;
}
}

```

Por último, tenemos que agregar las referencias al JavaScript de validación antes de la instrucción @using en la vista Edit.cshtml:

```

<script
src="@Url.Content("~/Scripts/jquery.val
idate.min.js")"
type="text/javascript"></script>
<script
src="@Url.Content("~/Scripts/jquery.val
idate.unobtrusive.min.js")"
type="text/javascript"></script>

```

Ahora podemos comprobar que la edición de un producto no permite insertar nombres repetidos.

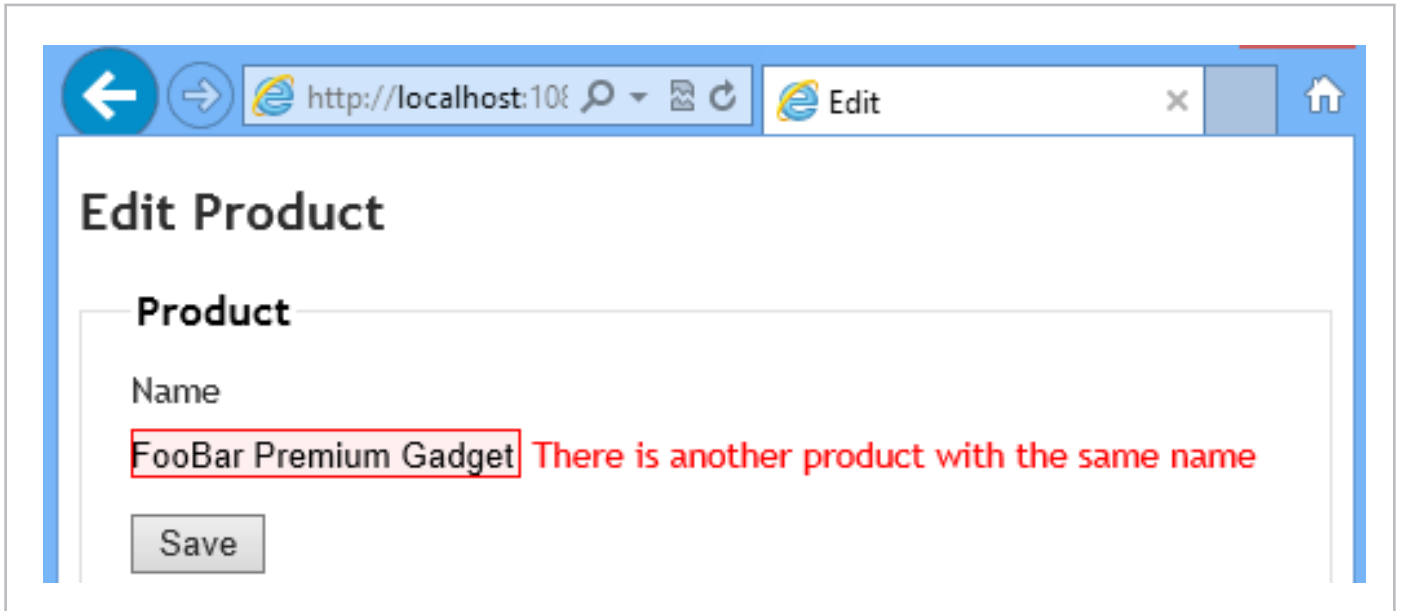


Imagen 6.- La validación remota en acción.

Conclusión

En este artículo hemos visto como editar las entidades de negocio en ASP.NET MVC y como agregar validaciones locales y remotas. En el próximo número veremos la inyección de dependencia y cómo podemos realizar pruebas unitarias con MVC. Hasta entonces, ¡feliz lectura!


El código completo de este artículo está disponible en <http://sdrv.ms/11j0cHC>.

EDIN KAPIĆ

Arquitecto SharePoint
edin.kapic@spenta.es

@ekapic
<http://www.spenta.es>

Bases de la programación de aplicaciones web en ASP.NET MVC conectadas a SharePoint


www.uruit.com.uy


Knowledge base.

Responsabilidad Social.

Gestión de ideas de innovación.


Gestión de contratos.

Gestión de desempeño.



SharePoint


Adopción (Gamification) - Integración con Dynamics CRM - Outsourcing



UruiT
Global IT Services

Microsoft Partner

Gold Collaboration and Content
Silver Application Development
Silver Customer Relationship Management



Microsoft
Most Valuable Professional

09

Resultados y experiencias del 1º Office & SharePoint App Challenge de España

Resumen

Los grupos de usuarios de SharePoint de España (SUGES, SUG.CAT, MadPoint y LevaPoint) organizaron los días 26 y 27 de abril el primer Office & SharePoint 2013 Application Challenge. La iniciativa, impulsada desde Microsoft Corporation con el apoyo a nivel organizativo de los Centros de Innovación de Microsoft de España (CIIN y CIP) y de Microsoft Ibérica, se tradujo en la realización simultánea de 3 hackatons de desarrollo de aplicaciones en las ciudades de Madrid, Barcelona y Valencia.

Artículo

Cuando se planteó desde Microsoft Corporation, y en concreto desde el área de DPE, la realización del primer App Challenge, desde los Centros de Innovación de España y los grupos de usuarios de SharePoint nos marcamos varios objetivos para trasladar el nuevo modelo de aplicaciones de Office & SharePoint a las empresas y profesionales TIC:

- Por un lado, y de forma bastante intensiva, formar a los participantes de los eventos de cada ciudad en aspectos relativos a cómo crear aplicaciones, las APIs y herramientas de desarrollo disponible, como ganar dinero llevando las aplicaciones a la nueva Office Store de Microsoft, etc.
- Por otro lado, servir como fuente generadora de ideas de aplicaciones orientadas tanto a la productividad del usuario doméstico como la del usuario más empresarial.
- Contar con referentes y expertos de la comunidad (incluidos MVPs) en el desarrollo para SharePoint y Office:
 - Alberto Díaz, Jose Quinto y Adrian Díaz en Valencia.
 - Mario Cortés y Miguel Tabera en Madrid.
 - Edin Kapic, David Martos y Juan Carlos González en Barcelona.
- Finalmente, poner en práctica los conocimientos adquiridos durante la formación para implementar las ideas de aplicaciones y posteriormente publicarlas en el Office Store.

Una vez pasado el evento, se puede concluir que el primer challenge ha sido todo un éxito en cuanto a número de asistentes (más de 70 participantes entre las tres ciudades), las ideas de aplicaciones presentadas (3-4 por ciudad) y el trabajo realizado por los mentores y organizadores del evento en cada ciudad encargados de impartir las

sesiones de formación y de ejercer como mentores de los distintos equipos que estuvieron trabajando en sus ideas de aplicaciones.



Imagen 1.- Asistentes al App Challenge de Barcelona.

¿Por qué crear aplicaciones para Office y SharePoint?

Aunque el nuevo modelo de aplicaciones para Office y SharePoint es novedoso, desde el punto de vista conceptual no lo es tanto sobre todo pensando en los Marketplace de aplicaciones que Microsoft ya tiene disponibles para otras plataformas: Windows 8, Windows Azure y Windows Phone. Frente a estos, el Office Store se puede ver como un modelo de Marketplace mixto en el que publicar aplicaciones orientadas al usuario doméstico, al usuario profesional y organizaciones empresariales de distinto tamaño. Como comenta Alberto Díaz, MVP de SharePoint Server, en este artículo de su blog (<http://geeks.ms/blogs/adiazmartin/archive/2013/04/30/office-store-191-una-tienda-de-futuro.aspx>), en el Office Store tienen cabida aplicaciones de naturaleza y propósito múltiple como por ejemplo:

- Aplicaciones empresariales que buscan mejorar la plataforma SharePoint (On-Premise y Online en Office 365) proveyendo funcionalidades y capacidades no disponibles por defecto. Por ejemplo, en esta categoría entrarían aplicaciones generalistas necesarias en el día a día como


un CRM, utilidades de contabilidad y facturación, un ERP completo o bien aplicaciones más específicas como una pequeña gestión de proyectos, una agenda de contactos. Etc.

- Aplicaciones para SharePoint de tipo gadget que muestren información cambiante como puede ser información meteorológica, de bolsa, etc.
- Aplicaciones de productividad para la suite de Office (tanto las versiones de escritorio como las web) pensadas para facilitar el acceso y uso de información externa


a usuarios domésticos y empresariales: búsqueda de noticias, búsqueda de imágenes, uso de diccionarios en línea, etc.

- Utilidades para Office que permitan analizar e interactuar con información existente así como facilitar la toma de decisiones. Por ejemplo, las típicas aplicaciones de posicionamiento de información en mapas forman parte de esta categoría.
- Y un largo etcétera.


Featured Apps




GiftCertificates.com Gift Pass
★★★★★ (0)
GiftCertificates.com
Word 2013
FREE | Add




Pro Word Cloud
★★★★★ (1)
Orpheus Technology Ltd.
PowerPoint 2013, Word 2013
FREE | Add



Sensei Task Analyzer™
★★★★★ (7)
Sensei Project Solutions
Project 2013
12,49 € | Add




Dell Social Hub
★★★★★ (3)
Dell, Inc.
SharePoint 2013
FREE | Add

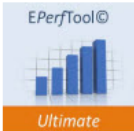


Bing Maps
★★★★★ (8)
Microsoft Corporation
Excel 2013
FREE | Add


New Apps




Collaboration Suite Quick Access
★★★★★ (0)
isolations AG
SharePoint 2013
FREE | Add




EPerfTool - Ultimate ©2013
★★★★★ (0)
Performance Plus Plus Inc
Excel 2013
834,99 € | Add



EPerfTool - Professional ©2013
★★★★★ (0)
Performance Plus Plus Inc
Excel 2013
249,99 € | Add




Exchange Rates
★★★★★ (1)
Upslide
Excel 2013
1,69 € | Add




CompartiMOSS
★★★★★ (0)
General de Software
SharePoint 2013
FREE | Add


Top Downloads




Dictionary - Merriam-Webster
★★★★★ (16)
Merriam-Webster Inc.
Excel 2013, Word 2013
FREE | Add




Bing Dictionary (English)
★★★★★ (3)
Microsoft Corporation
Excel 2013, Word 2013
FREE | Add



English Dictionary
★★★★★ (2)
STANDS4 LTD
Word 2013
FREE | Add



Diccionario Español
★★★★★ (3)
STANDS4 LTD
Word 2013
FREE | Add



Avery Templates App for Microsoft Word
★★★★★ (3)
AVERY DENNISON OFFICE PRODUCTS COMPANY
Word 2013
FREE | Add

Imagen 2.- Aplicaciones destacadas y más descargadas en el Office Store.

Sin duda, el Office Store es una buena fuente de oportunidad que puede aportar ventajas interesantes tanto a empresas desarrolladoras como a profesionales. Sólo por citar algunas, se pueden destacar las siguientes:

- Visibilidad, ya que el Office Store es un Marketplace global de manera que permitirá que las aplicaciones puedan estar disponibles en distintos países y en distintos idiomas.
- Un nuevo modelo de negocio basado en aplicaciones para Office & SharePoint apoyado en la flexibilidad a nivel

de modelos de licenciamiento ya que es posible publicar aplicaciones que sean gratuitas, que tengan versiones trial o bien que tenga un coste por usuario o por número de asientos (para organizaciones).

- Permite que desarrolladores con experiencia en tecnologías plataformas Microsoft y no Microsoft puedan crear aplicaciones y publicarlas en el Office Store.

Finalmente, como muestra del potencial del Office Store

y como ejemplo práctico de aplicación, os presentamos la aplicación de CompartiMOSS (<http://office.microsoft.com/en-us/store/compartimoss-WA104059493>).

[aspx?redir=0&fromAR=1](#)). Se trata de una aplicación de SharePoint que permite visualizar los distintos números de la revista publicados hasta ahora y acceder a los mismos:

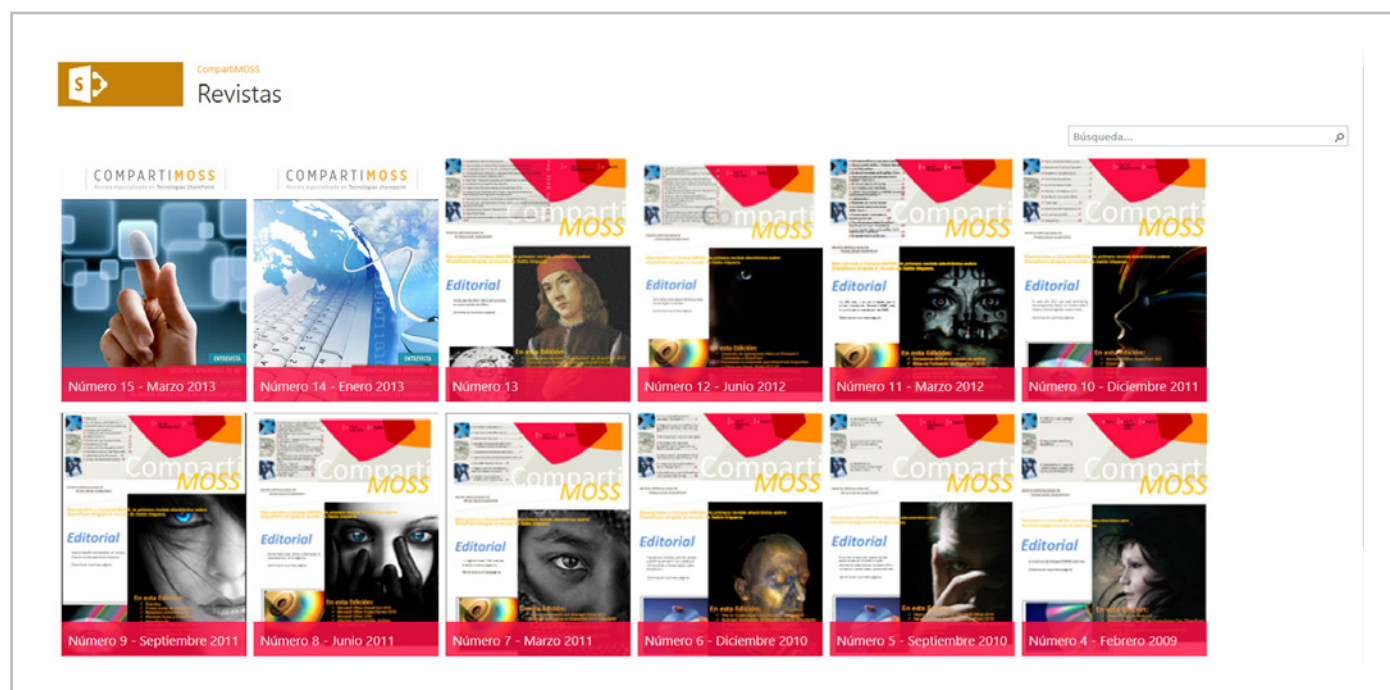


Imagen 3.- Aplicación CompartiMOSS publicada y disponible en el Office Store.

Aplicaciones e ideas presentadas

En cada ciudad, la 2ª jornada del Office & SharePoint App Challenge estuvo enfocada a la presentación de ideas de aplicaciones y al desarrollo inicial de las mismas para generar una primera versión de aplicación. Las aplicaciones qué se presentaron en cada ciudad son las siguientes:

- Madrid, aplicaciones e ideas presentadas:
 - **Beezy Profiles**, creada por un equipo de personas de Spenta, se trata de una aplicación social que permite qué los usuarios de SharePoint puedan integrar los datos de su perfil social con datos procedentes de otras redes sociales como por ejemplos LinkedIn.
 - **Business Apps Framework**, se trata de una idea de aplicación expuesta por integrantes de Online TIC. Se trata de una aplicación qué de forma dinámica es capaz de generar distintos elementos (formularios, listas, botones de acción, etc) para un cierto escenario de negocio como por ejemplo la gestión de oportunidades empresariales.
 - **SharePoint Health Dashboard**, aplicación creada por un equipo de Raona. Se trata de un panel que permite de manera rápida y en tiempo real tener una idea general de varios indicadores del servidor (usuarios, ancho de banda, ocupación de listas, etc.). Se trata de una aplicación de tipo SharePoint-Hosted y experiencia de usuario de tipo Inmersive.
- Barcelona, aplicaciones e ideas presentadas:
 - **Localizador de direcciones para Word**, idea presentada por un grupo de estudiantes de ingeniería informática. Se trata de una sencilla aplicación de tipo

“Panel de tareas” para Word qué permita recoger datos de direcciones en documentos Word, ubicarlas en un mapa y mostrar información de detalle de la dirección. Además, en el caso de encontrar varias coincidencias para una dirección, la aplicación muestra información de las mismas en un listado y permite que el usuario pueda elegir la opción qué sea más adecuada en función de la dirección recogida del documento Word.

- **Aplicación de Excel para cuadrar balances**. Se trata de una aplicación de tipo “Panel de tareas” para Excel qué permite recoger información de un balance económico para, mediante las correspondientes operaciones de cálculo, permitir que el usuario pueda cuadrar las cuentas de forma automatizada.
- **Comparador de configuración de seguridad entre diferentes sitios de SharePoint**. La idea de esta aplicación, de SharePoint, es ofrecer un comparador/clonador de seguridad entre colecciones de sitios.
- **Player de videos HTML5 para SharePoint**, idea de aplicación presentada por un equipo de Raona. Se trata de crear una aplicación qué permite visualizar en un player HTML los videos subidos a una biblioteca desplegada a nivel de aplicación. Al mismo tiempo, desde la aplicación se pueden cargar videos.
- Valencia, aplicaciones e ideas presentadas:
 - **Aplicación de gestión de capacidades de lista**, para conocer los datos relativos a las capacidades de las listas desde el punto de vista de un DBA. En este caso, se trata de una idea de aplicación con un gran

potencial de uso, y también con una alta complejidad en cuanto a desarrollo.

- **Visor de cámaras de seguridad.** Se trata de una aplicación que muestra un panel de cámaras IP recogiendo para ello la información almacenada en una lista. Esta aplicación se encargaría de enviar notificaciones PUSH a dispositivos móviles. La aplicación, presentada por un equipo de Encamina, está compuesta por un módulo de visualización de cámaras IP en internet (Publicas, proporcionando URL de conexión, o privadas especificando modelo, usuario y contraseña).
- **Aplicación para formularios de Contacto,** presentada por un equipo de SolidQ. Se trata de una aplicación que permite crear formularios de contacto dinámico en base a propiedades de configuración. Los formularios se crean a partir de los datos de configuración aportados en una App Part.
- **Aplicación de integración entre Project Server y SharePoint Server** para generar y mostrar Dashboards.

En total, en el App Challenge se expusieron en torno a 12 ideas de aplicaciones algunas de las cuáles están siendo desarrolladas en estos momentos para ser publicadas en el Office Store.

Nota: Si tienes una idea de aplicación para Office & SharePoint, estás a tiempo de poder acceder a una serie de incentivos que Microsoft Corporation tiene para aplicaciones de calidad destinadas al Office Store. Si estás interesado en acceder a los mismos o tienes dudas sobre cómo poner en práctica una idea de aplicación, puedes contactarme por medio de Twitter (@jcgm1978) o correo electrónico (jgonzalez@gruposodercan.es).

Primer Office & SharePoint 2013 Application Challenge

Ganadores por cada ciudad y a nivel nacional

En cada ciudad, los mentores y organizadores eligieron una aplicación ganadora que permitió a los creadores de la misma llevarse los premios disponibles (<http://www.suges.es/Paginas/Premios-Office-App-Challenge.aspx>).

- Ganador global y también ganador en Madrid: Beezy Profiles. Esta aplicación será presentada en la 1ª Iberian SharePoint Conference (<http://www.iberiansharepointconference.com/es>) que se celebrará en Madrid el próximo 10 de octubre.

- Ganador en Valencia: Aplicación para conocer los datos relativos a las capacidades de las listas desde el punto de vista de un DBA.
- En cuanto a Barcelona, el jurado optó por no elegir un ganador ya que el tiempo destinado a desarrollar aplicaciones se dedicó fundamentalmente a resolver dudas, ver ideas de aplicaciones y cómo enfocar el desarrollo de las mismas, etc.

Agradecimientos

El éxito de una iniciativa como el Office & SharePoint App Challenge depende del trabajo y colaboración de muchas personas. En este sentido, el éxito de este primer evento (en el futuro habrá más) ha sido posible gracias a todos los participantes, colaboradores, patrocinadores y organizadores del App Challenge:

- Microsoft y los Centros de Innovación Microsoft de España como organizadores e impulsores de la iniciativa.
- Los grupos de usuarios de SharePoint de España presentes en el challenge a través de los mentores y organizadores de cada ciudad.
- Las empresas patrocinadoras y colaboradoras en el challenge, que han hecho posible disponer de premios y regalos para los asistentes: Microsoft, Encamina, SolidQ, General de Software, Beezy, Campus MVP, Ediciones Eni, AvePoint, Pasiona, Plain Concepts.
- Y por supuesto, no pueden quedar en el olvido los más de 70 participantes en el App Challenge que además de poner a prueba a los mentores en la formación han sido capaces de generar 3-4 ideas de aplicaciones en cada ciudad, algunas de las cuáles se convertirán en aplicaciones que serán publicadas en el Office Store.

Conclusiones

La primera edición del Office & SharePoint App Challenge celebrada en Madrid, Barcelona y Valencia ha permitido acercar a profesionales y empresas desarrolladoras el nuevo modelo de creación de aplicaciones para la suite de productos de productividad de Microsoft. El evento ha permitido formar a los más de 70 participantes en cuestiones diversas sobre la creación y publicación de aplicaciones así como la generación de ideas y la implementación de las primeras aplicaciones.

JUAN CARLOS GONZÁLEZ MARTÍN

MVP de SharePoint Server

Arquitecto de Soluciones en el CIIN

@jcgm1978

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

www.ciin.es

Páginas de Publicación en SharePoint Server 2013

Resumen

Este artículo pretende analizar funcionalmente la relación existente entre el historial de versiones de las Páginas de Publicación, el estado de la aprobación de contenido y los diferentes métodos y características que propone SharePoint para la publicación de páginas.

Aunque las imágenes ilustrativas corresponden a la versión 2013 no existen diferencias significativas con la versiones 2007 y 2010.

Artículo

Cuando creamos un sitio de publicación en SharePoint se activa la característica Publicación de SharePoint Server automáticamente y se crea una biblioteca de Páginas. Una de las características más importantes de dicha biblioteca es la configuración de versiones, donde podremos:

- Requerir aprobación de contenido: Los documentos o páginas permanecerán en estado borrador hasta ser aprobados.
- Crear versiones siempre que se modifique un documento o página.
- Especificar qué usuarios verán los borradores o versiones secundarias.
- Obligar a desproteger los documentos antes de ser modificados.

Cuando se activa la Aprobación de contenido, se agrega una columna llamada “Estado de aprobación” que indica el estado en el que se encuentra la página. Esta columna puede tener uno de los siguientes valores:

- Borrador.
- En espera.
- Rechazado.
- Aprobado.
- Programado.

Cuando se activa el historial de versiones se agrega una columna adicional llamada “Versión”.

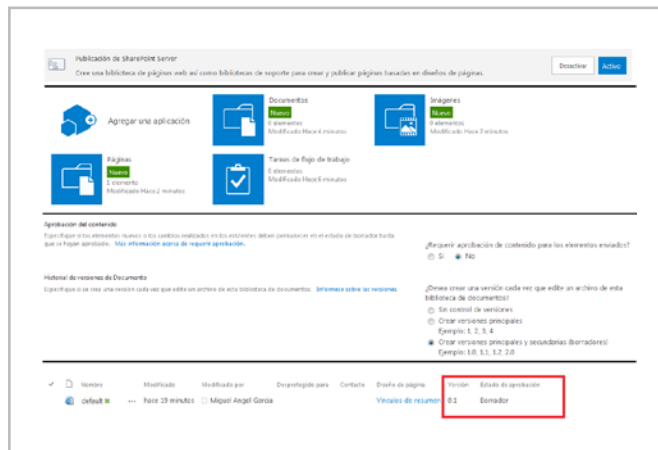


Imagen 1.- Columna Versión tras activar el historial de versiones.

Publicar páginas

La modificación del contenido de la página debe diferenciarse de la modificación de las propiedades aunque en ambos casos afectan al estado de aprobación. Existen diferentes métodos para modificar primero y publicar después las páginas:

- Menú ECB: Dependiendo del estado de la página, las acciones disponibles son diferentes.
- Pestaña “Archivos” de la Cinta: Una vez seleccionada(s) la(s) página(s) aparecen las acciones disponibles.
- Si desplegamos la página existen tres pestañas en la Cinta:
 - Navegar.
 - Página.
 - Publicar.

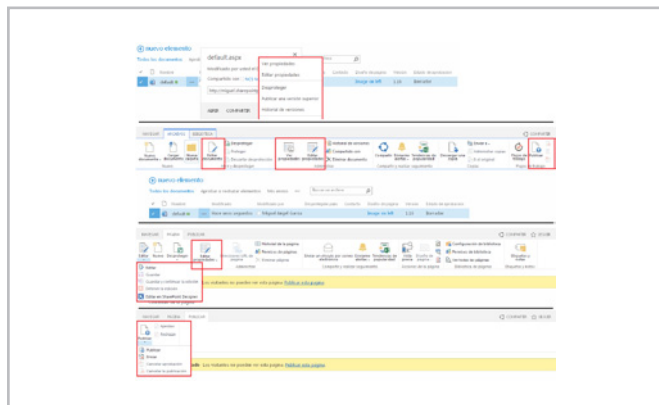


Imagen 2.- Opciones de la Cinta para la edición y publicación de páginas.

Estas tablas resumen los “Estados de aprobación” por los que pasa una página utilizando los diferentes métodos arriba señalados.

Estado de Aprobación: Borrador - Versión: X.Y			
Desde	Método	Estado Final	Versiones
Pestaña Navegar	Publicar esta página / este borrador	Aprobado	Versión principal X+1.0
Pestaña Página	Editar – Guardar	Borrador	Versión borrador X.Y+1
	Editar – Guardar y continuar la edición		
	Editar – Detener la edición – Guardar los cambios		
Pestaña Publicar	Editar Propiedades – Editar propiedades de SEO	Borrador	Sin cambios
	Editar – Detener la edición – Sin guardar los cambios		
Pestaña Publicar	Publicar - Publicar	Aprobado	Versión principal X+1.0
	Publicar - Enviar	En espera	Sin cambios
Menú ECB	Publicar una versión superior	En espera	Sin cambios
Pestaña Archivos	Publicar	En espera	Sin cambios
Menú ECB	Editar Propiedades - Aceptar	Borrador	Versión borrador X.Y+1
Pestaña Página	Ver propiedades - Aprobar Rechazar - Borrador	Borrador	Sin cambios
Pestaña Archivos	Ver propiedades - Aprobar Rechazar - Pendiente	En espera	Sin cambios

Estado de Aprobación: En Espera – Versión: X.Y			
Desde	Método	Estado Final	Versiones
Pestaña Navegar	Aprobar la página / los cambios	Aprobado	Versión principal X+1.0
Pestaña Página	Editar – Guardar	Borrador	Sin cambios
	Editar – Guardar y continuar la edición		
	Editar – Detener la edición – Guardar los cambios		
Pestaña Publicar	Editar Propiedades – Editar propiedades de SEO	En Espera	Sin cambios
	Editar – Detener la edición – Sin guardar los cambios		
Pestaña Publicar	Cancelar Aprobación	Borrador	Sin cambios
	Aprobar	Aprobado	Versión principal X+1.0
Pestaña Publicar	Rechazar	Rechazado	Sin cambios
	Rechazar	Rechazado	Sin cambios
Menú ECB	Cancelar Aprobación	Borrador	Sin cambios
Pestaña Archivos	Aprobar Rechazar – Aprobado	Aprobado	Versión principal X+1.0
Pestaña Archivos	Aprobar Rechazar – Rechazado	Rechazado	Sin cambios
	Aprobar Rechazar – Pendiente		
Menú ECB	Editar Propiedades	Borrador	Versión borrador X.Y+1
	Ver propiedades - Aprobar o Rechazar - Aprobado		
Pestaña Página	Ver propiedades - Aprobar o Rechazar - Rechazado	Rechazado	Sin cambios
Pestaña Archivos	Ver propiedades - Aprobar o Rechazar - Pendiente	En espera	Sin cambios

Estado de Aprobación: Rechazado – Versión : X.Y			
Desde	Método	Estado Final	Versiones
Pestaña Página	Editar – Guardar	Borrador	Versión borrador X.Y+1
	Editar – Guardar y continuar la edición		
	Editar – Detener la edición – Guardar los cambios		
Pestaña Publicar	Editar Propiedades – Editar propiedades de SEO	Rechazado	Sin cambios
	Editar – Detener la edición – Sin guardar los cambios		
Pestaña Publicar	Publicar - Publicar	Aprobado	Versión principal X+1.0
	Publicar - Enviar	En espera	Sin cambios
Menú ECB	Publicar una versión superior	En espera	Sin cambios
Pestaña Archivos	Publicar	En espera	Sin cambios
Menú ECB	Editar Propiedades	Borrador	Versión borrador X.Y+1
Pestaña Página	Ver propiedades - Aprobar o Rechazar - Aprobado	Aprobado	Versión principal X+1.0
Pestaña Archivos	Ver propiedades - Aprobar o Rechazar - Rechazado	Rechazado	Sin cambios
Pestaña Archivos	Ver propiedades - Aprobar o Rechazar - Pendiente	En espera	Sin cambios

Estado de Aprobación: Aprobado – Versión: X.0			
Desde	Método	Estado Final	Versiones
Pestaña Página	Editar – Guardar	Borrador	Versión borrador X.1
	Editar – Guardar y continuar la edición		
	Editar – Detener la edición – Guardar los cambios		
Pestaña Publicar	Editar Propiedades – Editar propiedades de SEO	Aprobado	Sin cambios
	Editar – Detener la edición – Sin guardar los cambios		
Pestaña Publicar	Cancelar Publicación	Borrador	Versión previa X-1.Y+1
	No publicar esta versión	Borrador	Versión previa X-1.Y+1
Menú ECB	Cancelar Publicación	Borrador	Versión previa X-1.Y+1
Pestaña Archivos	Editar Propiedades	Borrador	Versión borrador X.1
Pestaña Página	Ver propiedades - Aprobar Rechazar - Aprobado	Aprobado	Sin cambios
Pestaña Archivos	Ver propiedades - Aprobar Rechazar - Pendiente	En espera	Versión previa X-1.Y+1
Pestaña Archivos	Ver propiedades - Aprobar Rechazar - Rechazado	Rechazado	Versión previa X-1.Y+1

Historial de versiones

La característica del control de versiones, permite Ver, Eliminar y Restaurar cualquiera de las versiones principales y secundarias de las páginas guardadas en el historial de versiones. Restaurar una versión anterior principal o secundaria de una página, cambia el “Estado de aprobación” a “Borrador” e incrementa la versión borrador X.Y+1.

En el caso de las versiones principales publicadas existe la opción adicional “No publicar esta versión”. Cuando se utiliza la opción, el “Estado de Aprobación” cambia a “Borrador”. Si no existe una versión secundaria superior a la versión principal, se incrementa la versión previa X-1.Y+1, por el contrario, si existe una versión secundaria superior a la versión principal pasará a ser la nueva versión.



Imagen 3.- Historial de versiones para una página.

Desprotección de documentos

Cuando se configuran las versiones de una biblioteca se puede especificar que para modificar las páginas, necesariamente deben estar desprotegidas. Independientemente de esta configuración, un usuario que tenga los permisos necesarios puede desproteger o proteger una página. Cuando una página se desprotege inmediatamente su “Estado de aprobación” cambia a “Borrador” y se incrementa la versión borrador X.Y+1. Cuando una página está desprotegida las modificaciones no generan nuevas versiones borrador. Es posible, no obstante, generar versiones borrador conservando la página desprotegida.

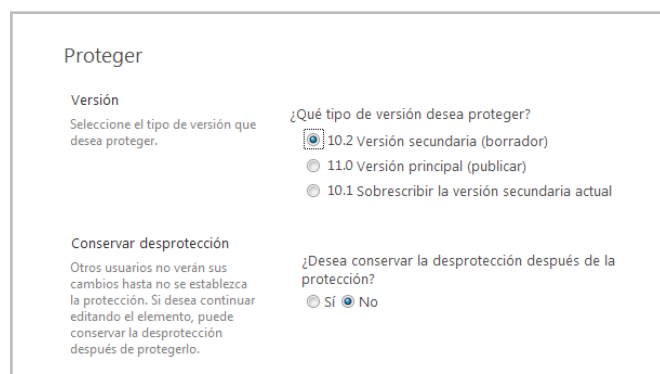


Imagen 4.- Opciones para desproteger una página.

Cuando una página está desprotegida se aplica esta tabla:

Acción	Estado final	Protección	Versiones
Antes de desproteger: Borrador, En Espera o Rechazado Versión X.Y			
Después de desproteger: Borrador – Desprotegido Versión X.Y+1			
Descartar Desprotección	Antes de desproteger	Protegido	Versión previa X.Y
Proteger - Versión secundaria (borrador) – NO conserva desprotección	Borrador	Protegido	Mantiene la versión X.Y+1
Proteger - Versión secundaria (borrador) – SI conserva desprotección	Borrador	Desprotegido	Versión siguiente X.Y+2
Proteger - Versión principal (publicar)	En espera	Protegido	Mantiene la versión X.Y+1
Proteger - Sobrescribir la versión secundaria actual – NO conserva desprotección	Borrador	Protegido	Sobrescribe la versión X.Y
Proteger - Sobrescribir la versión secundaria actual – SI conserva desprotección	Borrador	Desprotegido	Sobrescribe la versión X.Y y se mantiene la versión X.Y+1
Antes de desproteger: Aprobado – Versión X.0			
Después de desproteger: Borrador – Desprotegido Versión X.1			
Descartar Desprotección	Aprobado	Protegido	Versión previa X.0
Proteger - Versión secundaria (borrador) – NO conserva desprotección	Borrador	Protegido	Mantiene la versión X.1
Proteger - Versión secundaria (borrador) – SI conserva desprotección	Borrador	Desprotegido	Versión siguiente X.2
Proteger - Versión principal (publicar)	En Espera	Protegido	Mantiene la versión X.1

Programar la publicación de páginas

La programación de publicación de páginas está disponible en SharePoint en las Bibliotecas de documentos aunque no en las listas. Simplemente se trata de activar una de las características de la Biblioteca de documentos: “Administrar la programación de documentos”. Esta configuración requiere que la configuración de versiones principales y secundarias (borradores) esté activada y la aprobación de contenido.

Una vez activada esta característica en la biblioteca se agregan dos columnas que nos ayudan a verificar el estado de los documentos: Fecha de inicio programada, Fecha de finalización programada.

SharePoint no se basa en la Fecha de inicio programada (o Fecha de publicación) ni en la Fecha de finalización programada (o Fecha de caducidad) para verificar si un documento o página se debe mostrar o no a los usuarios autorizados. De hecho, se basa en la columna “Estado de aprobación”. Al habilitar la programación de elemento en una biblioteca de documentos, SharePoint hace uso de eventos y temporizadores para cambiar el Estado de aprobación en forma automática en base a la fecha de inicio programada y a la fecha de finalización programada.

Modificado por	Desprotegido por	Contacto	Estado de página	Versión	Estado de aprobación	Fecha de inicio programada	Fecha de finalización programada
Miguel Ángel García			Image on left	9.3	Programado	08/05/2019 7:10:09	08/05/2019 7:15:00
Miguel Ángel García			Image on left	10.0	Aprobado	08/05/2019 7:10:09	08/05/2019 7:15:00
Miguel Ángel García			Image on left	9.3	Borrador	08/05/2019 7:10:09	08/05/2019 7:15:00

Imagen 5.- Vista de páginas de publicación de acuerdo a la columna Estado de aprobación.

Esto es lo que sucede con las columnas “Estado de aprobación” y “Versión” si se utilizan las fechas de inicio y finalización:

a. Si un usuario aprueba una página (con cualquiera de las diferentes opciones disponibles que hemos visto)

en una fecha anterior a la que figura en “Fecha de inicio programada” el “Estado de aprobación” cambia a “Programado” y la versión no se modifica.

b. En la fecha y hora de la programación de inicio: si el estado de aprobación es “Programado”, cambia a “Aprobado”, el número de versión principal se incrementa X+1.0. No se producen cambios si el estado de aprobación fuese “Borrador”, “En espera” o “Rechazado”.

c. En la fecha y hora de programación de finalización: si el estado de aprobación es “Aprobado”, cambia a “Borrador” y el número de versión vuelve a la anterior X-1.Y.

Flujos de trabajo de aprobación

Adicionalmente podemos configurar un flujo de trabajo de aprobación sobre la biblioteca de páginas que modificará la columna “Estado de aprobación”. Este flujo distribuye una página para su aprobación. Los aprobadores pueden aprobar o rechazar a la página, reasignar la tarea de aprobación a otros usuarios o solicitar cambios en la página.

Al crear el flujo de trabajo, en la sección “Opciones de inicio” es necesario marcar la opción “Iniciar este flujo de trabajo para aprobar la publicación de una versión principal de un elemento.” Esta configuración disparará el flujo de trabajo cuando un usuario intente publicar una versión superior con cualquiera de los métodos arriba indicados. En la página que configura propiamente el flujo de trabajo es importante marcar las opciones que actualizarán la página según las acciones realizadas en el flujo:

- Finalizar con el primer rechazo.
- Finalizar cuando cambia el documento.
- Habilitar aprobación de contenido.

La columna “Estado de aprobación” permanecerá “En espera” hasta que el flujo de trabajo de aprobación finalice. La acción de aprobar por parte de los aprobadores, cambiará el “Estado de aprobación” de la página a “Aprobado”, incrementando la versión principal X+1.0. Si se rechaza o cambia el documento, el “Estado de aprobación” de la página cambiará a “Rechazado” manteniéndose la misma versión.

Finalizar con el primer rechazo	<input checked="" type="checkbox"/> Rechace automáticamente el documento si lo rechaza cualquier participante.
Finalizar cuando cambia el documento	<input checked="" type="checkbox"/> Rechace automáticamente el documento si se cambia antes de completarse el flujo de trabajo.
Habilitar aprobación de contenido	<input checked="" type="checkbox"/> Actualice el estado de aprobación una vez completado el flujo de trabajo (use este flujo de trabajo para controlar la aprobación de contenido).
<input type="button" value="Guardar"/> <input type="button" value="Cancelar"/>	

Imagen 6.- Opciones de configuración del flujo de aprobación de una página.

MIGUEL ÁNGEL GARCÍA RODRÍGUEZ

Analista SharePoint

miganggarrod@hotmail.com

<http://es.linkedin.com/pub/miguel-angel-garcia-rodriguez/34/481/108>



16

Licenciamiento de SharePoint 2013

Resumen

Tradicionalmente SharePoint a licenciado sus diferentes versiones de forma que todos los usuarios tienen acceso a solamente a una versión específica. SharePoint 2013 ha modificado este comportamiento para permitir que cada usuario tenga acceso solamente a la funcionalidad que necesita, ahorrando considerablemente en los costos de licenciamiento del sistema.

Artículo

Desde la introducción de la versión 2003 de SharePoint, el Portal siempre ha contado principalmente con dos versiones: una que provee toda la funcionalidad básica, que siempre ha sido gratis para usar (llamada “Foundation” en SharePoint 2010 y 2013), y una versión de pago (“Server”) que contiene varias capas extras que le agregan valor a la parte básica, haciéndolo más interesante para su aplicación en empresas. También tradicionalmente, esta versión “avanzada” de SharePoint ha contado con dos versiones diferentes (“Standard” y “Enterprise” en SharePoint 2010 y 2013), principalmente por razones de licenciamiento: la versión con la funcionalidad más compleja es (bastante) más costosa que la versión con funcionalidad empresarial básica.

En cuanto a licenciamiento, en SharePoint 2003, 2007 y 2010 era solamente posible crear un sistema empresarial, basado en SharePoint Server (o MOSS para 2003 y 2007) para darle acceso a todos los usuarios del Portal: Si se deseaba que algunos usuarios solamente tuvieran acceso a una de las versiones, y otros a la otra, era necesario crear dos granjas separadas, con las desventajas inherentes de costos en hardware, mantenimiento y dificultades técnicas para compartir la información.

SharePoint 2013 continúa teniendo el mismo esquema tradicional de versiones, es decir, SharePoint Foundation que es gratis y SharePoint Server con dos versiones de pago, Standard y Enterprise. Pero rompiendo con la tradición impuesta desde 2003, ahora es posible tener instaladas las dos versiones empresariales en una sola granja y darle derechos individuales a cada usuario para que accedan a una versión o a la otra. Teniendo en cuenta que una CAL (Client Access License) para la versión Enterprise cuesta aproximadamente el doble de una CAL para la versión

Standard, es recomendable que cada usuario consuma solamente las licencias necesarias para poder realizar su trabajo.

Note que las CAL de SharePoint son aditivas: para que un usuario pueda utilizar una CAL Enterprise, necesita tener obligatoriamente una CAL Standard también. Teniendo en cuenta que las dos mayores áreas de diferenciación entre Standard y Enterprise se pueden resumir en dos puntos:

- Soporte para soluciones de negocios, incluyendo servicios de Access e InfoPath
- Soporte a Inteligencia de Negocios, incluyendo servicios de Excel, PerformancePoint, Power View y Visio.

Es claro que solamente una parte reducida de toda una compañía puede necesitar acceso a la versión Enterprise, mientras que la gran mayoría de los usuarios tienen suficiente con la funcionalidad ofrecida por la versión Standard lo que, como consecuencia, reduce considerablemente los costos de licenciamiento.

Asignación de licencias en SharePoint Server 2013

La asignación de licencias a usuarios individuales en SharePoint server 2013 está basado en grupos de seguridad del Directorio Activo (AD) de Windows y una serie de comandos de PowerShell para configurar el servidor. Hay cinco categorías básicas que se pueden asignar: Standard, Enterprise, Project Server, Duet (para SAP) y WAC (para Office Web Apps). Un usuario puede pertenecer a diferentes grupos, permitiendo que, por ejemplo, tenga acceso a SharePoint Enterprise y Project Server, pero no a Office Web Apps. Usuarios que no pertenecen a ningún grupo no tienen acceso a ninguno de las categorías. Una nota importante es que el licenciamiento solamente funciona si la Aplicación Web utiliza autorización de Claims.

La infraestructura de licenciamiento no está activada por defecto cuando se instala SharePoint Server (Standard o Enterprise), por lo que hay que hacerlo expresamente utilizando los comandos de PowerShell que se explican más adelante. Cuando el licenciamiento no está activado, todos los usuarios tienen acceso a la versión instalada:

por ejemplo, si en la granja se ha instalado SharePoint Enterprise, todos los usuarios tienen acceso a la funcionalidad de Enterprise y, consecuentemente, tienen que tener las licencias CAL correspondientes. Importante para tener en cuenta es que si el licenciamiento está activado, un usuario que no esté configurado con algún tipo de licencia no tendrá acceso a ninguna de las funciones de SharePoint.

En SharePoint, el licenciamiento tiene efecto sobre cuatro tipos de elementos principalmente: WebParts, la Galería de WebParts, plantillas para crear Colecciones de Sitios y Sitios y en Bibliotecas de Documentos (especialmente en combinación con las posibilidades que ofrece Office Web Apps).

Activando y utilizando el marco de trabajo de Licenciamiento SharePoint dispone de ocho cmdlets de PowerShell para trabajar con el Licenciamiento. Ni el Modelo de Objetos de Servidor ni el de Cliente disponen de la posibilidad para interactuar programáticamente con licencias (todo el trabajo tiene que realizarse utilizando estos comandos de PowerShell).

Activación del sistema de licenciamiento

Tres cmdlets permiten activar, desactivar y dar información sobre el sistema de licenciamiento:

Enable-SPUserLicensing - Permite activar el modelo de licenciamiento en SharePoint Server 2013

Enable-SPUserLicensing [-Confirm [<SwitchParameter>]]

- Confirm: detiene el proceso para que el administrador confirme la operación. No es obligatorio

Disable-SPUserLicensing - Permite desactivar el modelo de licenciamiento en SharePoint Server 2013

Disable-SPUserLicensing [-Confirm [<SwitchParameter>]]

- Confirm: detiene el proceso para que el administrador confirme la operación. No es obligatorio

Get-SPUserLicensing - Devuelve información sobre el uso de licenciamiento en la granja

Get-SPUserLicensing

```
PS C:\Users\Administrador> Get-SPUserLicensing
Enabled
False
```

Imagen 1.- Información sobre el modelo de licenciamiento.

Asignación de licencias

Cinco cmdlets permiten trabajar con la asignación de licencias. La asignación puede ser creada antes o después de activar el sistema (utilizando los cmdlets indicados anteriormente). De igual forma, la configuración de

asignaciones permanece intacta después de desactivar el sistema. Para utilizar estos cmdlets es necesario tener los grupos de seguridad correctamente creados en AD (aunque no necesitan tener usuarios u otros grupos para poderlos utilizar, es decir, pueden estar vacíos).

Add-SPUserLicenseMapping - Añade un mapeo entre un grupo de seguridad de AD y un tipo de licencia específico

Add-SPUserLicenseMapping -Mapping <List> [-Confirm [<SwitchParameter>]]

- Mapping: objeto (Lista Genérica) con el mapeo. Es posible mapear más de un grupo AD en una sola operación. Obligatorio
 - Confirm: detiene el proceso para que el administrador confirme la operación. No es obligatorio
- Ejemplo de uso:

```
$myMapeo = New-SPUserLicenseMapping -SecurityGroup "SPEnterprise" -License Enterprise
Add-SPUserLicenseMapping -Mapping $myMapeo
```

En el parámetro "-License" se pueden utilizar los valores dados por Get-SPUserLicense: Enterprise, Standard, Project y OfficeWebAppsEdit.

```
PS C:\Users\Administrador> $myMapeo = New-SPUserLicenseMapping -SecurityGroup "SPEnterprise" -License Enterprise
PS C:\Users\Administrador> Add-SPUserLicenseMapping -Mapping $myMapeo
PS C:\Users\Administrador> _
```

Imagen 2.- Agregar un mapeo en el modelo de licenciamiento.

New-SPUserLicenseMapping - Crea un objeto para un mapeo nuevo. Para ser utilizado en conjunción con el cmdlet Add-SPUserLicenseMapping. El cmdlet puede ser utilizado con cuatro diferentes tipos de parámetros:

New-SPUserLicenseMapping -License <String>
-SecurityGroup <String> [-Confirm [<SwitchParameter>]]
[-WebApplication <SPWebApplicationPipeBind>]

New-SPUserLicenseMapping -License <String>
-Role <String> -RoleProviderName <String>
[-Confirm [<SwitchParameter>]] [-WebApplication <SPWebApplicationPipeBind>]

New-SPUserLicenseMapping -ClaimType <String>
-License <String> -OriginalIssuer <String> -Value <String>
[-Confirm [<SwitchParameter>]] [-ValueType <String>]
[-WebApplication <SPWebApplicationPipeBind>]

New-SPUserLicenseMapping -Claim <SPClaim>
-License <String> [-Confirm [<SwitchParameter>]]
[-WebApplication <SPWebApplicationPipeBind>]

- Confirm: detiene el proceso para que el administrador confirme la operación. No es obligatorio

- **License:** Tipo de licencia, según la enumeración dada por Get-SPUserLicense. No es obligatorio
- **SecurityGroup:** El grupo de AD al que se mapea la licencia. No es obligatorio
- **WebApplication:** El URL, GUID, nombre de la Aplicación Web o instancia del objeto Aplicación Web sobre el que se aplica la licencia. No es obligatorio
- **ClaimType:** Tipo de Claim a utilizar. No es obligatorio

Remove-SPUserLicenseMapping - Permite eliminar un mapeo creado anteriormente

Remove-SPUserLicenseMapping -Identity <List> [-Confirm [<SwitchParameter>]]

- **Identity:** Identificador (Lista Genérica) del mapeo. Es posible eliminar más de un mapeo en una sola operación. Obligatorio
- **Confirm:** detiene el proceso para que el administrador confirme la operación. No es obligatorio

Ejemplo de uso:

```
Remove-SPUserLicenseMapping -Identity abcdef12-
abcd-1234-abcd-abcdef123456
```

Get-SPUserLicenseMapping - Devuelve información sobre el mapeo

Get-SPUserLicenseMapping [-WebApplication <SPWebApplicationPipeBind>]

- **WebApplication:** la Aplicación Web de la que se quiere recuperar la información. No es obligatorio

```
PS C:\Users\Administrador> Get-SPUserLicenseMapping
Identity      : a66698d3-ff1d-458e-b6a6-d71e32fb4186
License       : Enterprise
Name          : SPEnterprise
ClaimValue    : S-1-5-21-2526397879-1674642894-1278307775-2606
ClaimType     : http://schemas.microsoft.com/ws/2008/06/identity/claims/groups
id            :
OriginalIssuer : Windows
ValueType     : http://www.w3.org/2001/XMLSchema#string
Identity      : 78f56414-1a10-47e7-bde1-e8b3d0a24b4a
License       : Standard
Name          : SPStandard
ClaimValue    : S-1-5-21-2526397879-1674642894-1278307775-2607
ClaimType     : http://schemas.microsoft.com/ws/2008/06/identity/claims/groups
id            :
OriginalIssuer : Windows
ValueType     : http://www.w3.org/2001/XMLSchema#string
```

Imagen 3.- Información sobre el mapeo en el modelo de licenciamiento.

Get-SPUserLicense - Devuelve una lista con los tipos de licenciamiento aceptados (para utilizar como parámetro en el cmdlet Add-SPUserLicenseMapping)

Get-SPUserLicense

```
PS C:\Users\Administrador> Enable-SPUserLicensing
PS C:\Users\Administrador> Disable-SPUserLicensing
PS C:\Users\Administrador> Get-SPUserLicense
License
-----
Enterprise
Standard
Product
OfficeWebAppsEdit
```

Imagen 4.- Tipos de licencias aceptadas.

Verificación del licenciamiento

SharePoint mismo se encarga de mostrar un mensaje a los usuarios que intentan utilizar alguna funcionalidad para la que no tienen licencias apropiadas. Por ejemplo, en la página de SharePoint que se muestra en la imagen No. 5, se ha instalado una WebPart del tipo "Excel Web Access", que solo puede ser utilizada por usuarios con una licencia Enterprise de SharePoint 2013:

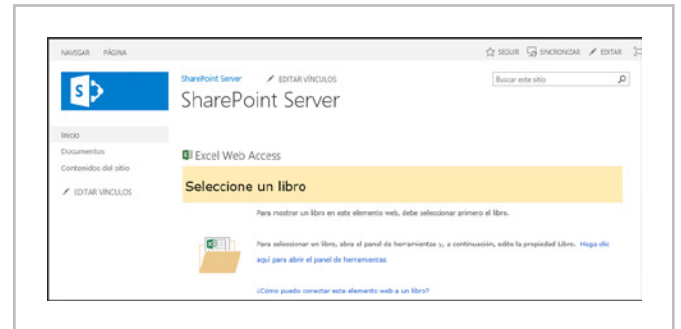


Imagen 5.- WebPart Enterprise vista por un usuario con licencia válida.

Un usuario sin la licencia Enterprise ve solamente un mensaje de SharePoint:



Imagen 6.- WebPart Enterprise vista por un usuario sin licencia válida.

Conclusiones

Poder combinar usuarios con licencias Enterprise y Standard de SharePoint Server en una sola granja ha sido no solamente un deseo, sino una necesidad urgente desde SharePoint 2003, cuando el servidor comenzó su carrera como aplicación crítica en las empresas modernas. Aunque el aspecto financiero de esta nueva funcionalidad es muy importante pues permite, potencialmente, ahorrar enormes cantidades de dinero para las empresas que utilizan SharePoint, no hay que perder de vista un factor paralelo: ya no es necesario tomar decisiones de diseño indeseables en el sistema, como por ejemplo crear granjas separadas para cada tipo de licencia e intentar reutilizar la información entre ellas. Otro aspecto importante es la flexibilidad que permite el nuevo marco de trabajo de licenciamiento, ya que se utiliza la infraestructura por defecto (AD) para manejar las licencias de SharePoint 2013.

GUSTAVO VELEZ

MVP SharePoint

gustavo@gavd.net

<http://www.gavd.net>

Resumen

El presente artículo muestra una implementación de un portal web público, Mibebeyyo.com, implementado en SharePoint 2013. Partiendo de las necesidades iniciales del cliente, se detalla como con se ha renovado y construido este portal asegurando además una buena escalabilidad y rendimiento.

Artículo

Mibebeyyo.com es un portal web de difusión de información entorno al mundo de los bebés, desde el embarazo de la madre hasta los primeros años de vida del hijo. Es el portal de referencia de las madres españolas con once millones de visitas al mes y un millón de usuarios, de los cuales alrededor de noventa mil son usuarios activos.

Se encontraban en un punto en el que no les daba respuesta acorde al crecimiento de negocio que estaban obteniendo. Necesitaban tener posibilidades de crecimiento y con la tecnología que empleaban la escalabilidad había quedado limitada, complementado con una disminución del rendimiento. Uno de los principales causantes de esa limitación era la desalineación que se producía entre el portal y las redes sociales, necesaria hoy en día para la visibilidad ante los usuarios.

Por si fuera poco, también era necesaria una reestructuración del modelo interactivo que les permitiera agilizar los procesos operacionales del departamento, ya que la gestión y administración de contenidos era considerablemente lenta.

Dado que la web siempre ha sido su canal principal de marketing, todos estos factores generaron esa necesidad de cambio. Y ahí es donde entra Birchman.

Planteamiento

Si partimos de la base de que el portal debía abarcar todos los aspectos entorno al bebé, nos planteamos una solución que permitiese darnos una “visión 360” alrededor de la madre y el bebé. La evolución de las webs en la última década ha obligado a las empresas a reciclarse si no querían perder visibilidad ante el mundo. Este es un caso concreto de esa evolución necesaria.

¿Qué es lo que necesitaba el cliente?

Dar solución a cuatro áreas bien diferenciadas:

- Advertising.
- Advisoring.
- Social.
- Tools.

El cliente necesitaba un sistema de advertising totalmente integrado con el portal que permitiese perfilar al usuario visitante acorde a la publicidad que le es mostrada.

La web se define por cubrir todo el ciclo de vida que va desde el embarazo hasta los primeros años de vida del bebé, por lo que el objetivo del equipo de trabajo es dar toda la información posible durante ese periodo, aconsejando a las madres. Este proceso de advising es el que se conseguiría mediante la publicación de contenidos, artículos e información en general, que desde fuera contemplamos como una herramienta de publishing que ofrecería información consolidada y de confianza.

Todo el portal debía girar entorno a la socialización de los usuarios, desde una valoración de un producto hasta una respuesta de un especialista sobre un tema importante en el portal. Era imprescindible conectar todas las partes ofreciendo así una experiencia de usuario completa y rica.

Paralelamente, se buscaba potenciar una serie de Tools pensadas especialmente para ayudar a la mamá. Calculadora de fecha del parto, calendario de ovulación, embarazo semana a semana... son algunos ejemplos los cuales, todos, se planteaban como desarrollos que extendían producto.

¿Qué es lo que SharePoint 2013 nos ofrece?

Conociendo el nuevo producto de Microsoft, podemos observar que nos brinda las facilidades necesarias para resolver todos los puntos que el portal requiere, ya sea directamente a través de producto o a través de extensiones personalizadas para ciertos puntos específicos.

Los cuatro pilares que nos brinda SharePoint 2013 para el proyecto son:

- Metadata + Friendly Navigation.
- Publishing Features.
- SEO Properties.
- Social Capabilities.

La navegación por URL que anteriormente tenía SharePoint, nos obligaba a navegar como el producto establecía a nivel tecnológico y podía darnos muchos dolores de cabeza al intentar asemejarlo a nuestra visión más semántica. Ahora podemos combinar un menú de navegación totalmente personalizado a través de la jerarquía de los Term Stores con las direcciones web Friendly, que nos permiten establecer los Sites, Subsites y Páginas semánticamente.

SharePoint 2013 nos permite enlazar semánticamente la arquitectura de la información del portal con el contenido que se muestra en cada sección. Conectar un contenido con una url amigable y a su vez con una sección del portal, proporciona una relación muy estrecha y lógica de la información. Este tipo de navegación semántica da un salto respecto a versiones anteriores del producto, ya que es mucho más sencilla la identificación por motores de búsqueda, pudiendo aprovechar al máximo sus características de SEO. La generación de robots.txt, junto con el SiteMap rastreable por los motores de búsqueda y cerrándolo con las propiedades META, por ejemplo, que podemos modificar en cada página, nos posicionan SharePoint como un producto solvente en soluciones de portal.

Las características de publishing nos permiten una gestión de páginas públicas y privadas, control de acceso y la posibilidad de añadir extensiones de producto, como es en este caso la adición de herramientas, como un buscador de nombres de bebés o gráficas de seguimiento del crecimiento del bebé.

Por último, si hay algo visible por lo que destaca esta nueva versión de SharePoint es por todas las características sociales que están implementadas Out of the Box. Nos permite crear toda una red social privada, muy potente y extensible, que podemos adaptar a nuestro criterio a través de extensiones y que, además, podemos darle la forma y los colores que queramos a través de restylings completos (como es el caso de Mibebeyyo.com).

Solución

ARQUITECTURA FÍSICA

Un portal público de estas características necesita entornos sólidos y fiables. Se ha establecido una granja de servidores a través de un servicio de IaaS, con la utilización de Cloud Services de Microsoft, Azure Virtual Machines. La granja propuesta se basa en una solución multirol, dividido en siete servidores. Resulta imprescindible separar los roles

de Frontal Web, Servidor de Aplicaciones, Servidor de búsqueda y Servidores de Bases de Datos.

Particularmente, cabe mencionar la complejidad de encontrar una solución óptima para los servicios de búsqueda, dada la gran afluencia de queries realizadas en una solución de portal. La topología de búsqueda propuesta pasa por, en primer lugar, un servidor de administración utilizado para todo lo relacionado con la administración de la búsqueda. Por otro lado, un entorno de indexación que se encarga de ejecutar el Crawl y tratar toda la información recogida, de forma que quede estructurada para que sea lo más eficiente posible su recuperación. Por último se ha separado tanto el índice como el procesamiento de Queries, de forma que esté particionado y repartido, consiguiendo así la respuesta y el dato en tiempo récord.

Arquitectura Lógica

¿CÓMO SE HA HECHO?

El proyecto a un nivel más lógico se ha realizado sobre SharePoint 2013, utilizando extensiones para los desarrollos a medida en Visual Studio 2012, con tecnología ASP.NET Framework 4.5 utilizando un modelo de programación de tres capas, separando así la lógica de negocio de la de datos y UI (Presentation, Logic and Data Tiers).

¿EN QUÉ SE BASA EL PORTAL?

El motor principal del portal está basado en la búsqueda, que nos permite realizar todo tipo de enlaces entre los contenidos de la web, ya sea a través de artículos relacionados con lo que estamos visualizando o resúmenes que nos dan acceso a una mayor información.

Se ha creado un buscador general, donde el usuario puede encontrar desglosado, tal y como el cliente ha especificado, contenido de todo el portal a través de varios Search Results Web Part, customizando los Display Templates según el tipo.

Las Content Search Web Parts también son parte fundamental de todo el portal, ya que prácticamente todo se estructura mediante ellas, mostrando el contenido como corresponde en cada Site y Subsites. Se trata de componentes muy potentes ya que permiten mostrar cualquier contenido dentro de una Site Collection y, por supuesto, definir cómo lo queremos mostrar, principalmente cruzando la información del término utilizado en la url amigable, con la sección actual.

“... Mibebeyyo.com es un portal web de difusión de información entorno al mundo de los bebés...”



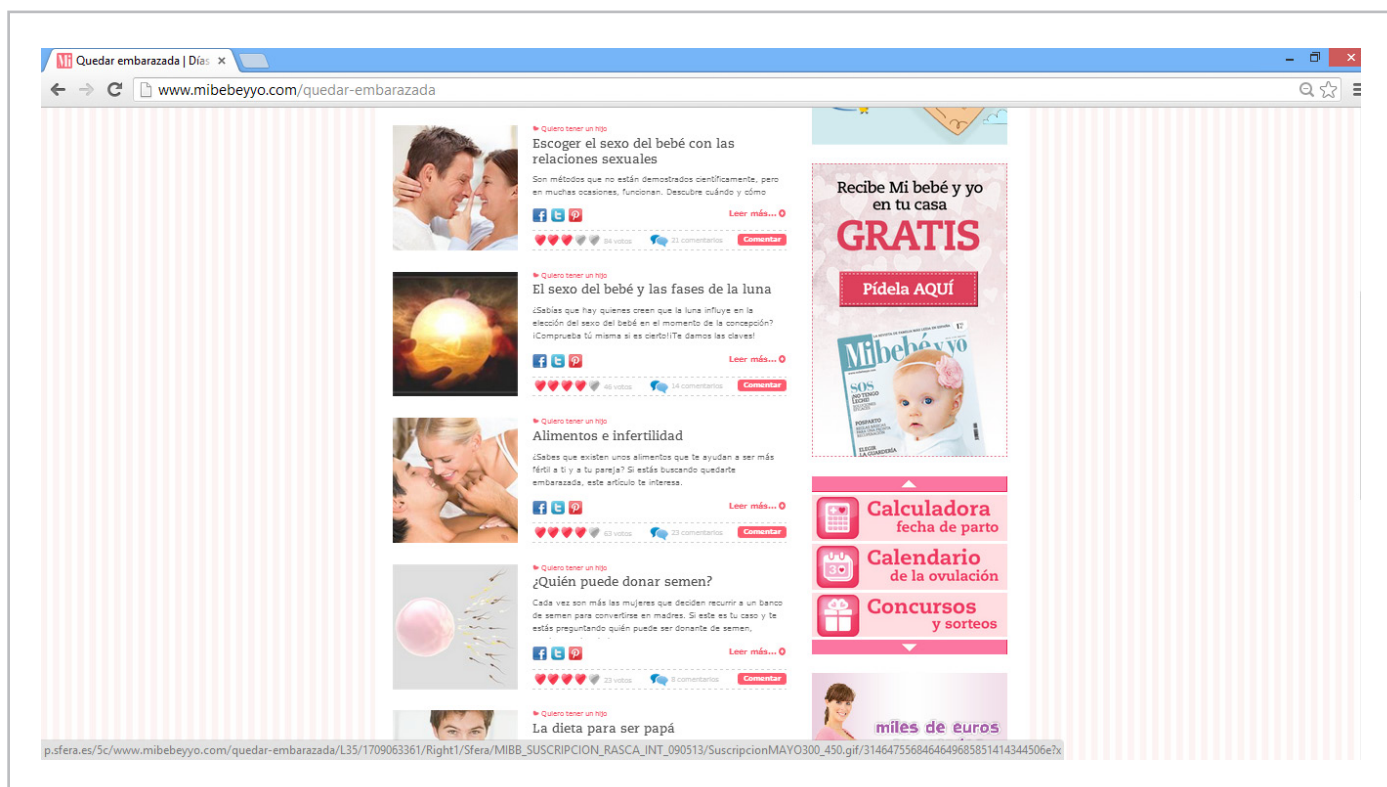


Imagen 1.- Aspecto del nuevo portal con la solución propuesta.

SEARCH ENGINE OPTIMIZACION

Dos de las principales características que nos ofrece SharePoint 2013 es el uso de URLs Friendly y Search Engine Optimization (SEO). Las características de SEO permiten la utilización de técnicas que mejoran el posicionamiento de los motores de búsqueda. Un claro ejemplo de ello es la utilización de las URL Friendly, que nos permiten olvidarnos de las clásicas direcciones web de SharePoint ("Páginas", ".aspx", "ID=1234...") y pasar a tener URLs mucho más intuitivas, simples y manteniendo las pautas que mejor se contemplan para posicionarnos mejor en las búsquedas. Más concretamente, nos permite incluso definir las variables que harán igual o distinta una url, cuando éstas se encuentren en la petición GET (...?idElemento=23). Ésta configuración resulta realmente útil de cara al rastreo por parte de los motores de búsqueda. En ésta línea, cada página podremos identificarla como potencial de pertenecer al siteMap que SharePoint nos generará periódicamente. Cuando Google rastree el portal, encontrará el XML de SiteMap con la información que nosotros hemos querido presentar.

Concretamente para las urls amigables, en el portal se encuentran cerca de 20.000 páginas, todas ellas bajo friendly urls, hecho que excede de largo los boundary limits de SharePoint en éste apartado. La solución ha pasado por distribuir la altura del árbol de la arquitectura de la información en forma de hermanos en lugar de padre-hijos. Ello implica que en lugar de distribuir el árbol de términos en un único termSet, se hayan definido tantos termGroups como primeros elementos de menú, TermSets como tantos elementos de segundo y tercer nivel, y Terms como páginas.

CACHE

AppFabric

Dentro de la estructura de capas realizada, existe una primera que gestiona las consultas de datos utilizando la caché de AppFabric. De este modo, reducimos considerablemente las peticiones a la base de datos almacenando en AppFabric nuestras listas, opciones, etc.

Ésta capa es sumamente importante y va toda apalancada sobre Windows Server, formando parte de un clúster de cuatro servidores que alimentan y se sincronizan periódicamente, ya sea con las regiones utilizadas por los desarrollos a medida como por la región por defecto utilizada en la caché distribuida de SharePoint.

Blob Cache

Un problema que puede surgir en un portal público en el que los usuarios tienen acceso para añadir contenidos es la renderización de las imágenes que se suben cuando no sabemos de qué tamaño son.

La nueva versión de SharePoint tiene una característica llamada Image Renditions (que podemos encontrar en las opciones de Look and Feel) que nos permiten tener almacenada una imagen en su tamaño completo y tener así, un número específico de variantes re-escaladas de la misma. Incluso más, cada re-escalado de imagen se guarda con el peso y definición optimizada que nos lleva a reducir drásticamente las comunicaciones y descargas de información.

Cabe decir que la utilización de ésta funcionalidad requiere

una configuración exacta y precisa de la Blob Caché y como SharePoint la gestiona.

En caso contrario nos puede llegar a generar muchos dolores de cabeza. En el caso del portal, se han definido 10 renditions distintas.

SharePoint Caché

Dada la naturaleza de portal, también se han utilizado las facilidades de caché que proporciona SharePoint. En nuestro caso se han perfilado los usuarios anónimos y los registrados, con perfiles de caché acordes a cada uno.

Por otro lado, el uso de la caché de objetos ha resultado imprescindible para reducir de forma drástica el número de transacciones entre SQL Server y los frontales, principalmente, ofreciendo así un rendimiento mucho mejor para momentos de alta concurrencia.

Social

Si tuviéramos que definir el portal en dos grandes bloques, uno sería todo el contenido de información para el usuario (artículos, consultas, expertos, advisor...), y el otro la parte social.

Se ha definido toda la parte social a través de las capacidades que nos ofrece My Site, escalado a 400.000 usuarios. En pocas palabras: Hemos creado una red social propia para Mibebeyo.com.

Esta red social consta de dos Web Applications diferentes separadas del portal principal. Una de ellas es la comunidad de usuarios, con todo lo que ello implica: el My Site del usuario, los enlaces entre usuarios (seguidores), galería de fotos propia de cada usuario...).

La otra contiene los grupos de la comunidad, siendo cada grupo una Site Collection propia en la que hemos modificado la plantilla para que permita crear Conversaciones, Eventos y Encuentros de usuarios.

Toda esta información, al estar cruzada con el millón de usuarios, se ha guardado en base de datos en lugar de en listas por una mejora requerida del performance.



Imagen 2.- Componente social de la solución.



Imagen 3.- Detalle de una comunidad.

Estos grupos son accesibles por cada miembro de la comunidad a través de búsquedas desde el muro principal. Además, se han implementado Posts, Microblogs y Followers como "amigas" utilizando el concepto de socialización a través de las opiniones (ya sea sobre artículos o sobre productos).



Imagen 4.- Acceso a grupos.

Para poder enlazar las tres Web Application hemos implementado SingleSignOn sobre autenticación FBA, donde están todos los usuarios que se registran en el portal. De este modo cuando un usuario hace Log In en la web principal o en la comunidad, si se dirige a otra de las Web application automáticamente queda loggeado y se convierte así en algo transparente al usuario final. Se ha logrado a través de las cookies de SharePoint (Federated Authentication) a nivel de Web Application y una principal que es la que indica si el usuario ha hecho Log In o no.

Todo el sistema social que se ha creado gana sentido cuando los usuarios lo convierten en algo habitual, por lo que se ha creado un sistema de Social Gaming, mediante un componente personalizado, para incentivar la participación. Este componente funciona a través de puntos y medallas, que se ganan conforme los usuarios realizan diferentes acciones en cualquier sitio de la web.

Mis puntos

Tú interactividad en el portal acumula puntos. Consigue distintivos y medallas y conviértete en Embajadora de Mibebéy.com. [Descubre cómo.](#)

MI PUNTUACIÓN...

Mis puntos: **36** Mi medalla:

Mis distintivos:

CÓMO CONSEGUIR PUNTOS...

Tu interactividad en el portal acumula puntos. Consigue distintivos y medallas y conviértete en Embajadora Mibebéy.com.

Distintivo	Mis puntos	Necesarios	Me faltan	
Mamá Participativa		3	50	47
Mamá Fotógrafa		12	50	38
Mamá Experta en Compras		0	50	50

Imagen 5.- Componente de social gaming.

CÓMO CONSEGUIR MEDALLAS...

Puntos necesarios:	Mamá Bronce	Mamá Plata	Mamá Oro	Mamá Embajadora
	50	100	150	Mención especial de la redacción sólo para Mamas Oro

Imagen 6.- Badgets o medallas.

Al inicio de este apartado hablábamos de la separación del portal en dos grandes bloques (información y socialización). Para que todo funcione, se deben unir los dos bloques de forma que el usuario vea una integración total, y no se muestre sesgada.

Para ello, la interacción que se hace desde cualquier parte de la web, queda reflejada en el propio muro de usuario. Cuando un usuario escribe un comentario en un artículo, hace una votación o una valoración de un producto, esta acción queda reflejada directamente en su muro, de modo que todas las personas que le sigan podrán verlo.

♥ Quedar Embarazada - Quiero tener un hijo

♥♥♥♥♥ 245 votos 90 comentarios [Comentar](#)

Imagen 7.- Ejemplo de interacción con la web.

Toda la parte Social del portal se compone del espacio personal (mi muro, mis amigas...), comunidades (opiniones de los usuarios externos sobre el contenido, viralidad en redes sociales como Facebook, Twitter o Pinterest...), Social Gaming y "Preguntas y Respuestas" (Sección de expertos, viralizado también en las principales redes sociales).

Imagen 8.- Espacio personal del usuario.

Esto se podría aplicar tanto en portales públicos como en una red corporativa, ya que todas estas características sociales se aplican sobre cualquier tipo de contenido, ya sean sobre bebés, coches o gestión documental de una empresa. Se trata de contenido de SharePoint que socializamos, sea cual sea.

MOBILE

Con el acceso móvil, que ya supera con creces el acceso tradicional que existe hoy en día, los diseños de los portales públicos deben ser adaptables dependiendo del dispositivo desde el que accedemos.

SharePoint 2013 permite el acceso multidispositivo por defecto, utilizando la característica de Mobile que

podemos activar a nivel de Site, y su función es presentar el contenido de forma optimizada para un dispositivo móvil. Aun así, el punto fuerte que nos ofrece el producto en ésta última versión es el sistema multidispositivo a través de Device Channels. Éstos nos permiten utilizar paneles que se muestran u ocultan dependiendo del dispositivo que solicita la información.

Device Channels en Mibebeyyo.com

Se ha hecho una asignación de los Device Channels que hemos necesitado (iPad, iPhone, Android, default mobile, default web...) y se han definido las Master Pages que corresponden a cada Device Channel.

En los Layouts de las páginas se han definido diferentes Device Channel Panels dentro de los Place Holders, cada uno correspondiente al Device Channel que queramos que muestre ese contenido. De este modo, tenemos siempre el mismo contenido, pero lo veremos de un modo u otro dependiendo desde donde accedamos.

De este modo podemos ver el mismo artículo de una forma desde nuestro ordenador portátil:



Imagen 9.- Vista normal de una página del portal.

Y de otra diferente si accedemos desde el móvil:



Imagen 10.- Vista móvil de la misma página.

RESPONSIVE WEB DESIGN

Todo el portal está realizado siguiendo los patrones de Responsive Web Design. No obstante, el uso de los Device Channels nos permite un diseño por css y javascript mucho más sencillo, ya que la división de paneles según el dispositivo nos facilita la aplicación de estilos o funciones propias de cada uno.

Conclusión

Nos encontramos ante un portal de referencia en toda Europa con mejoras operacionales de compañía cercanas al 50% y con claras expectativas de crecimiento. Se ha hecho una propuesta en SharePoint 2013 que solucionaba todas sus necesidades y que, tras meses de trabajo, ha cumplido con todas las expectativas.

Igualmente, cabe mencionar, desde un punto de vista de solución SharePoint, la gran mejora de escalabilidad que nos lleva el producto desde un punto de vista de capacidad de escalabilidad y rendimiento para concurrencias altas. De igual forma aclarar que en estos casos resulta imprescindible disponer de cachés activas, distribuidas, que preserven la salud y estabilidad de los componentes más activos de la granja.

ISMAEL CRUZ

MCITP en SharePoint 2010

MARC BÀGUENA

MCITP PRO en SharePoint 2010

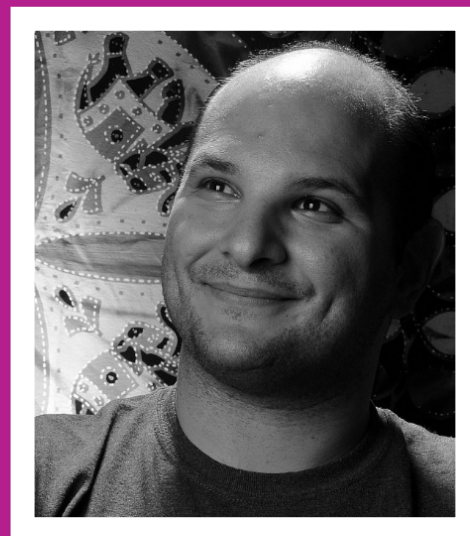
i

26

Entrevista a Edin Kapić

Me llamo Edin Kapić, nací en Mostar (Bosnia-Herzegovina) en 1978 y desde 1996 resido en Barcelona (España). Actualmente trabajo como SharePoint Architect en Spenta. Tengo una irreprimible faceta de profesor, por eso me encanta preparar y dar charlas, seminarios, cursos y escribir para compartir y diseminar el escaso conocimiento del que dispongo. También tengo suerte de trabajar en lo que me gusta, que tal como están las cosas considero que es un lujo.

Mantengo (aunque mal y tarde) mis blogs www.edinkapic.com (inglés) y spbloggedin.blogspot.com (español), así como mi cuenta de Twitter @ekapic. Soy MVP de SharePoint desde 2013 y MCT Regional Lead para España desde 2012.



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

La culpa fue de mi padre. En la segunda mitad de los años ochenta su empresa, una gran distribuidora de electricidad, le metió en la informática para probar las aplicaciones financieras que se creaban allí. Así que crecí entre libros de programación y manuales de MS-DOS, lo que sospecho que tuvo una gran influencia sobre mí.

Aprendí a programar sin ordenador, porque mis padres no se lo podían permitir, por lo que mis primeros programas fueron realizados en BASIC en el Commodore 64 prestado de un amigo mío. Incluso llegué a aprenderme el lenguaje máquina del procesador 6510. Luego di el salto al mundo PC con un 386 de la ONG Médicos del Mundo, dónde empecé a trabajar de traductor. Allí empecé a trastear con C/C++, ensamblador y Visual Basic 3. Al llegar a España y acabar la educación secundaria, decidí matricularme en la Facultad de Informática de Barcelona y así ya quedé vinculado del todo a esta apasionante mundo.

¿Cuáles son tus principales actividades tecnológicas hoy en día?

Mis actividades diarias principales son diseñar e implementar soluciones basadas en SharePoint para los clientes. Trabajo con SharePoint desde 2005 pero también hago excursiones en otras tecnologías Microsoft como Azure, Windows Phone o el presuntamente difunto Silverlight.

Además, discuto y colaboro con el resto de los compañeros en los demás proyectos de la empresa. Me mantengo fiel al principio de que cuatro ojos ven más que dos...y así sucesivamente.

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

Las actividades sociales (de las que nunca hay suficientes) y las obligaciones de mantener una casa (compra y limpieza).

Intento mantener una moderada actividad física con Pilates y caminando bastante porque el trabajo que me apasiona es, desgraciadamente, muy sedentario.

¿Cuáles son tus hobbies?

¿Además de SharePoint? Me encanta salir a navegar a vela, es una actividad que me entusiasma. Cuando no puedo navegar, me gusta hacer senderismo por los caminos de montaña o de costa. Y cuando se impone no salir de un recinto cerrado, alguna partida de juegos de mesa (tengo ya inoculado el virus del juego de mesa de Juego de Tronos a varios compañeros) o leer un buen libro.

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

Como dice un refrán mariner, "el tiempo predecir es exponerse a mentir". En la tecnología creo que pasa lo mismo, aunque no se puede negar el fuerte empuje de todo relacionados con las tecnologías cloud y la informática táctil. Lo que más me preocupa es que se pierda la noción

de que la tecnología sirve para algún fin y no es un fin en sí misma, porque viendo los fetichismos de ciertas marcas no lo tengo tan claro. En todo caso, habrá que estar preparado para cambios constantes y que van a ser cada vez más frecuentes. Está claro que no nos podemos dormir para estar al día.

"...enfoque de desarrollo orientado a aplicar diseño creado por un diseñador gráfico."

EDIN KAPIC

Arquitecto SharePoint

edin.kapic@spenta.es

@ekapic

<http://www.spenta.es>

Implemente las mejores Soluciones para SharePoint con las excelentes herramientas de KWizCom!

Con más de 40 diferentes aplicaciones, implemente **rápidamente soluciones fantásticas** que responden exactamente a sus necesidades de negocios!

www.kwizcom.com

Calendar Aggregations
Smart Forms Discussion Boards
Charts Survey Workflow
Scan Print Wiki+
Mobile Taxonomy



Descargue hoy mismo una versión de prueba GRATUITA!



Haciendo a SharePoint aún mejor!

Monitoreando SharePoint Server con System Center Operations Manager (SCOM) – Parte 1

Resumen

Para mantener un buen rendimiento en la plataforma de SharePoint es necesario contar con un buen sistema de monitoreo que pueda recolectar el estado de salud del servidor así también como del sistema operativo, para cumplir con este objetivo Microsoft nos brinda System Center Operations Manager del cual se detallan sus funcionalidades en este artículo.

Artículo

En muchas de las organizaciones que con el paso del tiempo han construido una intranet sólida basada en SharePoint, se ha convertido para muchos de los usuarios en una de las principales herramientas de trabajo del día a día, por esta razón, como parte de las responsabilidades de un administrador de la plataforma de SharePoint está el brindar alta disponibilidad a la plataforma, dar un mantenimiento adecuado, realizar tareas básicas de recuperación ante fallas como ejecución de respaldos, tanto de bases de datos como de SharePoint, replicación de servicios, entre otras tareas.

Con el fin de facilitar muchas de estas tareas SharePoint cuenta con un servicio que conocemos como el 'Health Analyzer' el cual mediante reglas de ejecución nos da a conocer recomendaciones, alertas sobre posibles errores, configuraciones erróneas, problemas que pueden afectar el rendimiento y a su vez, este servicio ofrece para algunos de los problemas corrección automatizada o posibles soluciones a errores o alertas de detección, también podemos encontrar generación de reportes y creación de reglas personalizadas para extender un poco más su funcionalidad.

Sin embargo, hay que tener presente que el Health Analyzer sigue siendo un servicio dentro de la plataforma de SharePoint, por lo que hay que ser consciente que si por alguna razón este servicio se viera afectado, nuestro monitoreo y reglas de ejecución también dejarían de funcionar.

Microsoft nos brinda dentro de la suite de System Center, una plataforma llamada Operations Manager para poder llevar a cabo monitoreo en una escala mayor, con

más alcance, complementando la administración de la plataforma al brindar los estados de salud con más detalle según las reglas de monitoreo establecidas.

Para llevar a cabo el monitoreo, System Center ofrece paquetes de administración los cuales definen de qué manera Operations Manager va a monitorear las plataformas. Sin estos paquetes SCOM puede monitorear los elementos básicos de Windows, como memoria RAM, espacio en disco, CPU, entre otros.

Cada producto Microsoft es lanzado con el MP (paquete de administración) para SCOM, como por ejemplo Hyper-V, SQL Server, NLB, entre otros.

Por supuesto, SharePoint no es la excepción, en los siguientes links podemos encontrar los paquetes para las diferentes versiones de SharePoint:

- System Center 2012 Monitoring Pack for SharePoint 2010:
 - <http://www.microsoft.com/en-us/download/details.aspx?id=34697>
- System Center Management Pack for SharePoint Foundation 2013:
 - <http://www.microsoft.com/en-us/download/details.aspx?id=35591>
- System Center Management Pack for SharePoint Server 2013:
 - <http://www.microsoft.com/en-us/download/details.aspx?id=35590>

El proceso de instalación de los paquetes se explica a continuación, como ejemplo, tome el paquete para SharePoint Foundation 2013, debido a que es requerido antes de instalar el paquete para SharePoint Server 2013.

Lo primero que debemos realizar es descargar el paquete con la extensión MSI e instalarlo en el servidor de SCOM, luego de proceder con la instalación debemos asegurarnos que los archivos se encuentran en el directorio C:\Program Files\System Center Management Packs.

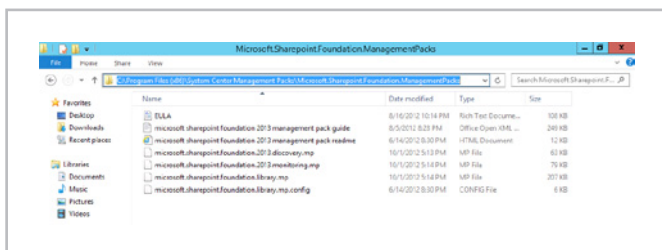


Imagen 1.- Archivos de instalación del paquete de monitores para SharePoint Foundation 2013

Posteriormente debemos importar estos paquetes a SCOM, para esto utilizaremos la consola de operaciones y debemos seleccionar el menú de Administración, posteriormente Management Packs y luego la opción de Importar, debemos elegir la dirección donde se instalaron los paquetes de administración e importarlos, como se muestra a continuación:

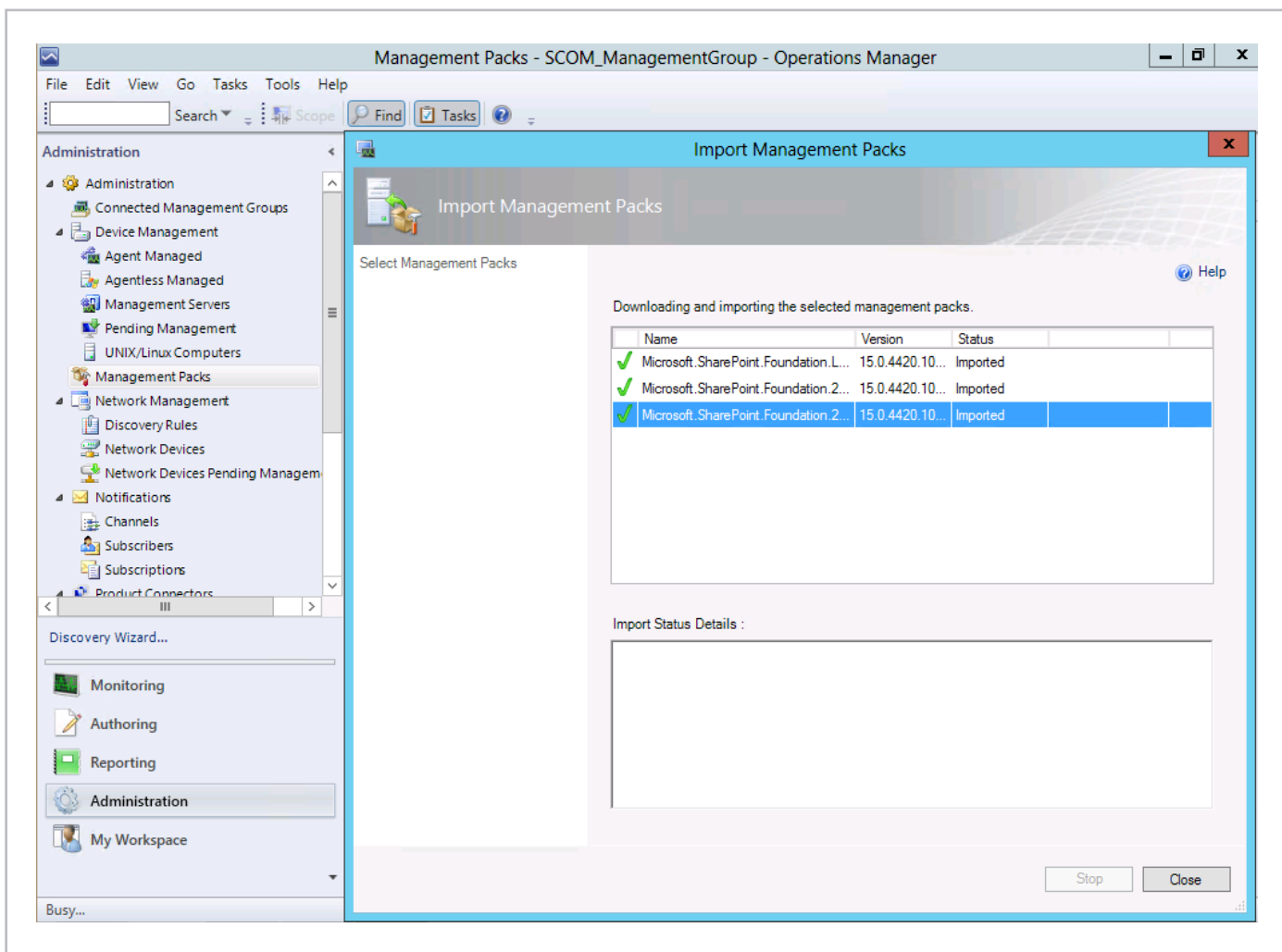


Imagen 2.- Importación de los paquetes en SCOM

Posteriormente debemos crear las credenciales de ejecución para el perfil de los monitores de SharePoint, esto lo podemos realizar en el mismo menú de administración, en la opción de cuentas, donde luego debemos establecer cuál va a ser la cuenta de dominio y la contraseña para luego asignarla al perfil, como se muestra a continuación:

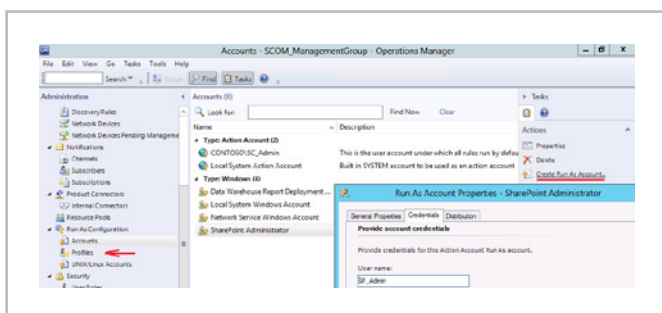


Imagen 3.- Creación de la cuenta de ejecución para los monitores de SharePoint

**Recolectar el estado de salud del servidor
así también como del sistema operativo...
Microsoft nos brinda System Center
Operations Manager**

En la segunda parte de este artículo exploraremos con mayor detalle las funcionalidades de los monitores y reglas para servicios, eventos, rendimiento, entre otros, además de los procesos de notificación.

DAVID SÁNCHEZ AGUILAR

MCITP en SharePoint Server

davidsanchezaguilard@hotmail.com

<http://davidsanchezaguilard.wordpress.com>

Creación de aplicaciones de Windows Phone con contenido de SharePoint (Parte 1)

Resumen

En este artículo aprenderemos a crear aplicaciones para Windows Phone que permitan crear, editar, borrar y actualizar elementos de listas de SharePoint. Si no tienes conocimientos sobre el desarrollo en Windows Phone no te preocupes, Microsoft nos proporciona unas plantillas de Visual Studio que harán todo el trabajo por nosotros.

Artículo

Cada vez es más importante la movilidad en nuestro día a día. Los dispositivos móviles se están convirtiendo en herramientas casi imprescindibles en nuestro trabajo. El mundo de las aplicaciones móviles permite a los usuarios empresariales poder acceder a sus sistemas, documentos e información estén donde estén.

Los usuarios de SharePoint tienen un gran aliado en Windows Phone ya que permite, de manera nativa, acceder a librerías documentales de SharePoint desde el hub de Office. Además, gracias a las aplicaciones de Word, Excel, PowerPoint y OneNote, es posible leer y editar estos documentos ubicados en SharePoint.



Imagen 1.- El hub de Office de Windows Phone permite acceder a nuestros documentos.

Pero hay numerosas ocasiones en las que no nos basta con poder consultar los documentos de SharePoint. Nuestras intranets están cargadas de otra serie de datos, páginas y listas que puede ser necesario consumir desde nuestro teléfono. Y es en ese punto en el que surge la necesidad de desarrollar una aplicación que obtenga información de SharePoint.

Es posible que un desarrollador de SharePoint no tenga una habilidad muy avanzada desarrollando en plataformas móviles. Afortunadamente, Microsoft ha creado unas plantillas de Visual Studio que nos ayudarán, mediante una serie de asistentes, a desarrollar aplicaciones que permitan trabajar con datos de SharePoint 2013, las Windows Phone SharePoint Application Templates.

Estas plantillas permiten crear en cuestión de minutos una aplicación de Windows Phone que permita trabajar (lectura, edición, creación y borrado) con los datos de una lista. Por ejemplo, se puede desarrollar en menos de 5 minutos (y con pocos conocimientos sobre Windows Phone) una aplicación para trabajar con una lista de tareas de SharePoint.

Preparando el entorno de desarrollo

Para preparar el entorno es necesario el siguiente software:

- Si vas a trabajar con el SDK de Windows Phone 8:
 - Microsoft Visual Studio 2012.
 - Windows Phone SDK 8.0:
 - <http://www.microsoft.com/en-us/download/details.aspx?id=35471>
 - Microsoft SharePoint SDK for Windows Phone 8:
 - <http://www.microsoft.com/en-us/download/details.aspx?id=36818>
- Si vas a trabajar con el SDK de Windows Phone 7
 - Microsoft Visual Studio 2010 (no están soportadas en VS 2012).
 - Windows Phone SDK 7.1:
 - <http://www.microsoft.com/en-us/download/details.aspx?id=27570>
 - Microsoft SharePoint SDK for Windows Phone 7.1:

- <http://www.microsoft.com/en-us/download/details.aspx?id=35475>

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

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

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

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

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

APIs para trabajar en desarrollo móvil para SharePoint 2013

Lo primero es hacer un breve repaso a las distintas API que tenemos disponibles en SharePoint 2013 y que nos permiten trabajar con sus datos en cualquier plataforma (no sólo web). Es posible desarrollar soluciones externas al servidor SharePoint usando el modelo de objetos cliente .NET, el modelo de objetos JavaScript y la API REST.

- Client-side object model (CSOM):
 - Para aplicaciones que usen C#.
 - Para aplicaciones Silverlight y Windows Phone.
- Javascript object model (JSOM):
 - Muy sencillo para aplicaciones web y javascript de Windows 8.
 - Útil para desarrollar apps de SharePoint para el Marketplace.
- REST:
 - API independiente de la plataforma basada en estándares como OAuth y OData.

En este artículo, usaremos el modelo de objetos cliente (CSOM) de .NET para desarrollar aplicaciones para Windows Phone. De todas maneras, no estamos limitados sólo al desarrollo móvil en Windows Phone.

Gracias a las API de JavaScript y de REST de SharePoint 2013, es posible crear aplicaciones para otra serie de terminales (Android, iOS, BlackBerry...), interactuar con los sitios de SharePoint ejecutando scripts en otras aplicaciones web, o integrar con otros sistemas que soporten capacidades REST. Más información puede encontrarse en este enlace: <http://msdn.microsoft.com/en-us/library/jj163228.aspx>

Proyecto de Visual Studio basado en la Windows Phone Empty SharePoint Application template

Si creamos un proyecto basado con la Windows Phone Empty SharePoint Application template, la estructura será muy similar a otros proyectos de Windows Phone, con la diferencia que se añaden referencias a las DLL instaladas por el Windows Phone SharePoint SDK:

- Microsoft.SharePoint.Client.Phone.dll
- Microsoft.SharePoint.Client.Phone.Auth.UI.dll
- Microsoft.SharePoint.Client.Phone.Runtime.dll

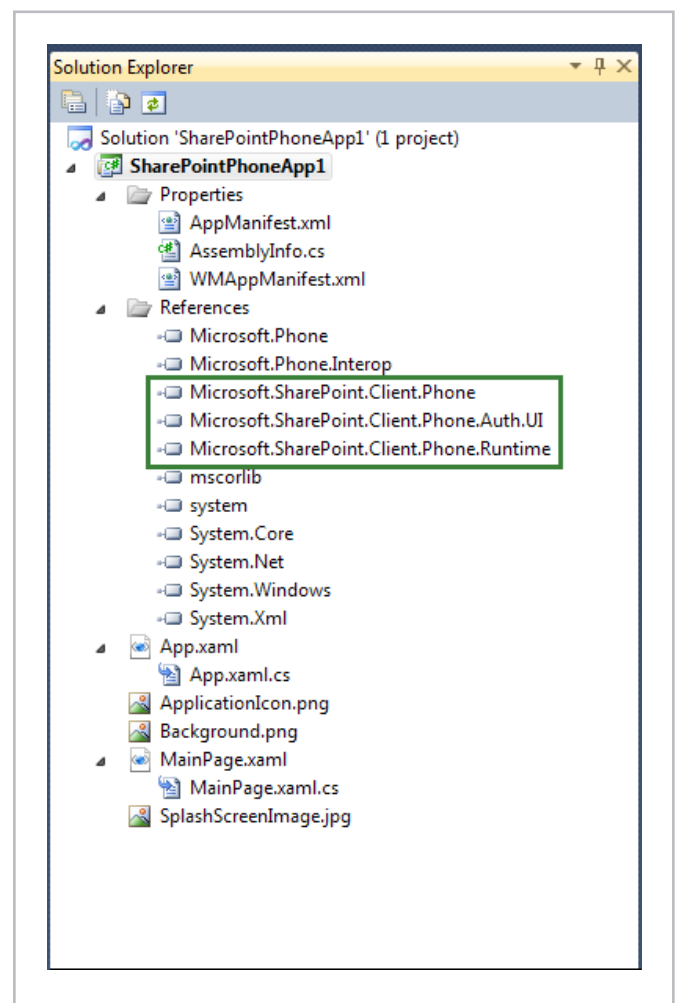


Imagen 2.- Estructura del proyecto "Windows Phone Empty SharePoint Application".

Los archivos de un proyecto basado en la plantilla Windows Phone Empty SharePoint Application son los archivos estándar de una aplicación de Silverlight de Windows Phone. El archivo MainPage.xaml contiene las declaraciones de XAML que constituyen la interfaz de usuario (UI) de la aplicación. El archivo MainPage.xaml.cs contiene el código para implementar lógica para soportar operaciones y eventos en la interfaz de usuario. El archivo App.xaml representa a la aplicación. El archivo App.xaml.cs incluye código para controlar los eventos de ciclo de vida de la aplicación.

Proyecto de Visual Studio basado en la Windows Phone SharePoint List Application template

La **Windows Phone SharePoint List Application template** es la más potente de ambas aplicaciones. Esta aplicación está preparada para desarrollar de una forma rápida una aplicación que permita consultar, editar, crear y eliminar datos de una lista en concreto de SharePoint. Al crear un proyecto con esta plantilla, un asistente nos ayudará a configurar correctamente la aplicación para dejarla ligada a una lista de SharePoint y comenzar a trabajar. El asistente preparará el código necesario para poder lanzar directamente la aplicación y que funcione correctamente sin tener que escribir ni **una línea de código**.

Los archivos generados por este proyecto son los que se muestran en la Imagen 3.

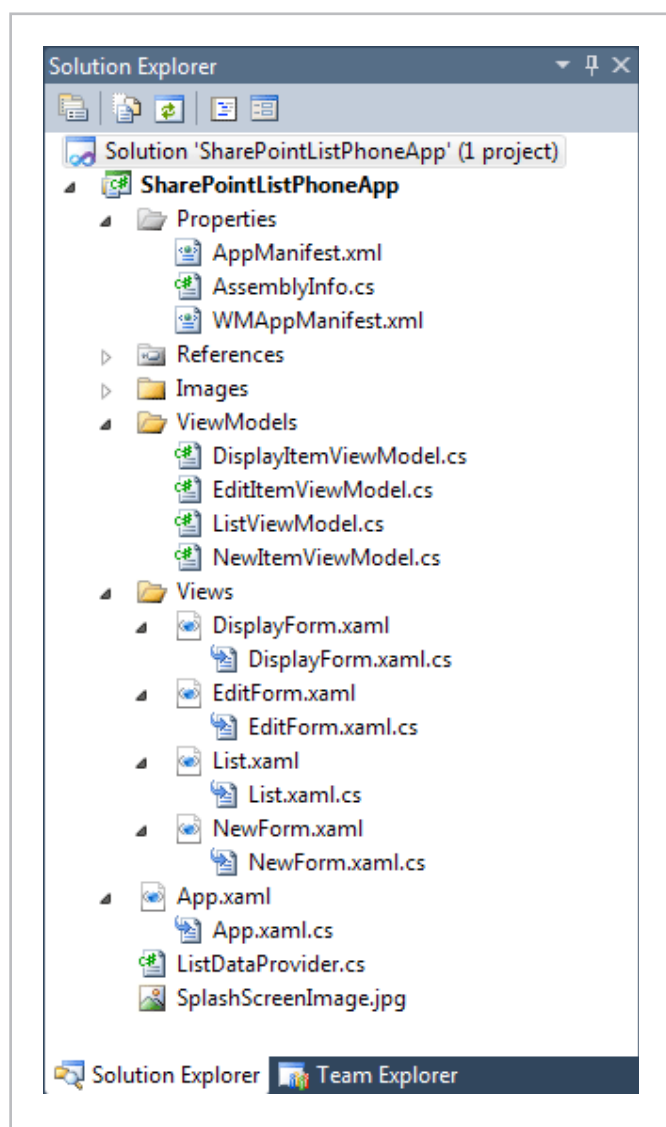


Imagen 3.- Estructura del proyecto "Windows Phone SharePoint List Application".

Los archivos importantes de esta solución son:

- **App.xaml:** representa a la aplicación de Windows Phone. Incluye declaraciones de elementos relacionados

con la aplicación, así como eventos de ciclo de vida de la aplicación como `Application_Deactivated` y `Application_Closing`.

- **App.xaml.cs:** archivo de code-behind de `app.xaml`. Podemos escribir código en este archivo para administrar el almacenamiento offline (local) de datos.
- **ListDataProvider.cs:** contiene el código para obtener acceso a datos de SharePoint y proporciona acceso a la sintaxis de consulta en el que se basan las distintas vistas de lista de la aplicación.
- **List.xaml:** formulario que muestra los distintos ítems de la lista, análogo a la vista All Items de SharePoint. Se renderizan los elementos mediante un control Pivot.
- **List.xaml.cs:** archivo code-behind de `list.xaml`. Contiene el código de métodos y eventos para los botones del formulario, como el botón Nuevo y el botón Actualizar.
- **DisplayForm.xaml:** formulario que muestra un elemento, análogo al formulario de display de SharePoint. Muestra el elemento con un StackPanel.
- **DisplayForm.xaml.cs:** archivo de code-behind de `DisplayForm.xaml`. Contiene el código de métodos y eventos para los botones del formulario, como el botón Nuevo y Actualizar.
- **EditForm.xaml:** formulario que edita un elemento, análogo al formulario de edición de SharePoint. Muestra el elemento con un StackPanel.
- **EditForm.xaml.cs:** archivo de code-behind de `EditForm.xaml`. Contiene el código de métodos y eventos para los botones del formulario, como el botón Guardar y Cancelar.
- **NewForm.xaml:** formulario que crea un elemento, análogo al formulario de nuevo elemento de SharePoint. Muestra el elemento con un StackPanel.
- **NewForm.xaml.cs:** archivo de code-behind de `NewForm.xaml`. Contiene el código de métodos y eventos para los botones del formulario, como el botón Guardar y Cancelar.
- **ListViewModel.cs:** origen de datos para el archivo `List.xaml`.
- **DisplayItemViewModel.cs:** origen de datos para el archivo `DisplayForm.xaml`.
- **EditItemViewModel.cs:** origen de datos para el archivo `EditForm.xaml`. Se puede escribir código en este archivo para validar los datos introducidos por el usuario al editar un elemento.
- **NewItemViewModel.cs:** origen de datos para el archivo `NewForm.xaml`. Se puede escribir código en este archivo para validar los datos introducidos por el usuario al crear un elemento.

El patrón de diseño MVVM

Las plantillas Windows Phone SharePoint List Application template, al igual que otras plantillas de Windows Phone,

generan un proyecto que sigue el patrón de diseño Model-View-ViewModel (MVVM). Este patrón de diseño es una forma de organizar el código en un proyecto en capas, que pueden ser desarrolladas, probadas y modificadas de forma independiente. Es particularmente efectiva en el diseño de aplicaciones de Windows Presentation Foundation (WPF) y Silverlight.

En el presente artículo, supondremos que se tiene al menos un mínimo conocimiento de este patrón. Si no es así, puede consultarse el siguiente enlace, en el que se explica el funcionamiento completo de este patrón de diseño adaptado a las plantillas de SharePoint: <http://msdn.microsoft.com/en-us/library/jj163209.aspx>

Básicamente, los tres componentes o capas principales del patrón MVVM son la Vista, el Modelo y el Modelo de Vista. En los proyectos basados en la plantilla Windows Phone SharePoint List Application template, estos componentes son implementados por los distintos archivos que hemos repasado anteriormente, la Imagen 4 muestra la distribución.

“... Los usuarios de SharePoint tienen un gran aliado en Windows Phone...”

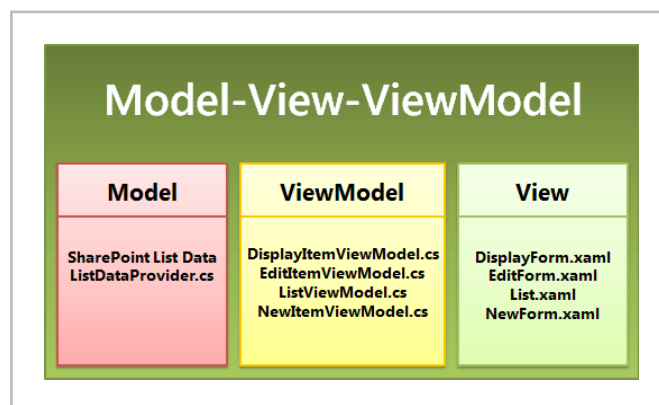


Imagen 4.- Estructura de los archivos en la plantilla “Windows Phone SharePoint List Application” siguiendo el patrón MVVM.

Creando una aplicación para trabajar con una lista de tareas en diez minutos

El último punto que veremos en la creación de una aplicación de Windows Phone para trabajar con una lista de tareas de SharePoint. Aunque parezca sorprendente, se tarda 10 minutos y no es necesario tener conocimiento previo sobre el desarrollo en Windows Phone.

Para ello, crearemos un proyecto desde Visual Studio con la plantilla Windows Phone SharePoint List Application (Imagen 5).

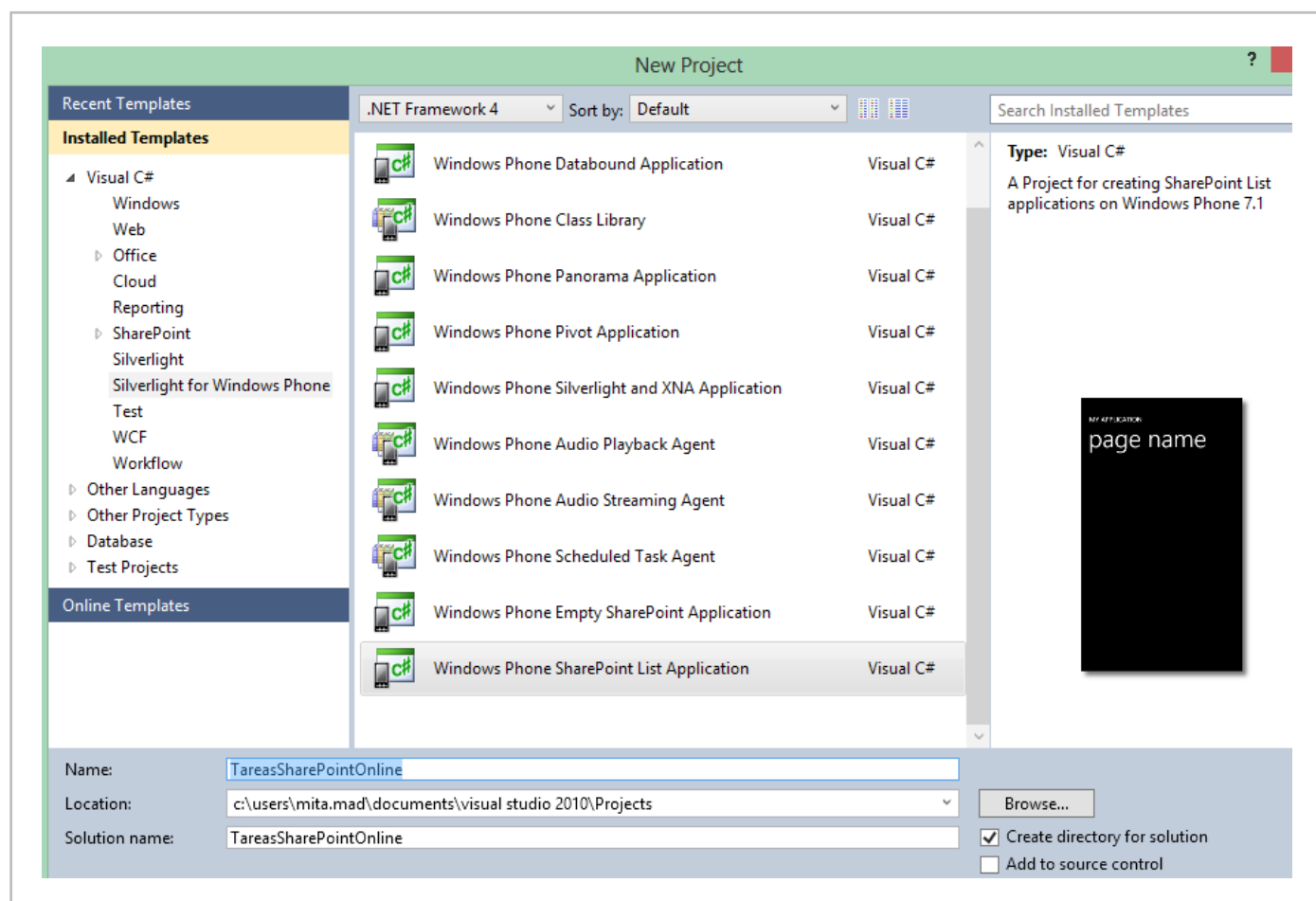


Imagen 5.- Nuevo proyecto de “Windows Phone SharePoint List Application”.

Cuando pulsemos el botón de creación, saltará el asistente. En el primer paso, escribiremos la URL del sitio de SharePoint (online u on-premises). Aparecerán las listas del sitio. Seleccionamos la lista de tareas que queramos (Imagen 6) y pulsamos Next:

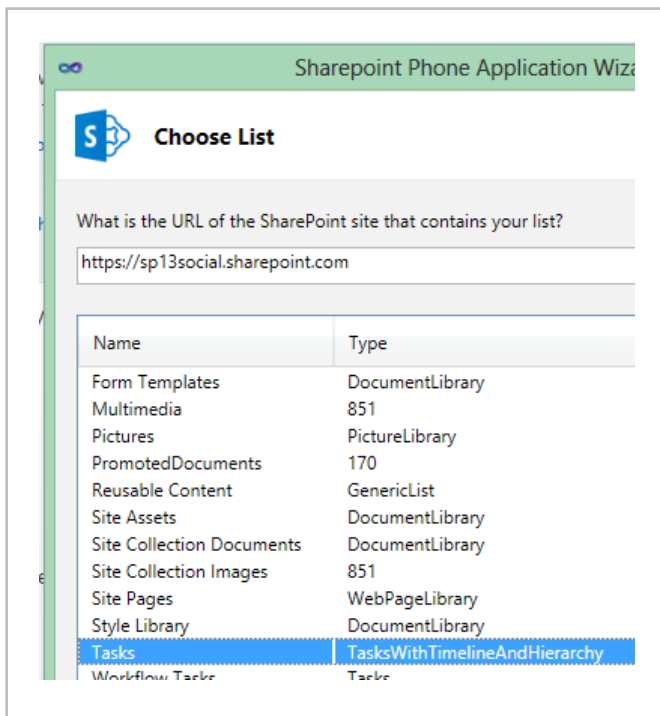


Imagen 6.- Paso 1 del asistente: Selector de sitio y lista.

En el siguiente paso debemos seleccionar las vistas de la lista que queremos que se muestren en la aplicación. Si hemos creado vistas personalizadas se podrán seleccionar también. Seleccionamos las vistas que queramos (Imagen 7) y hacemos clic en Next:

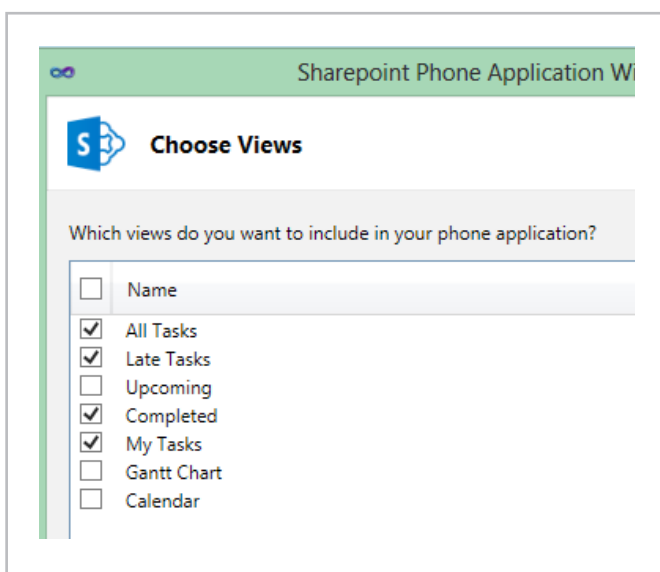


Imagen 7.- Paso 2 del asistente: Selector de vistas.

En el siguiente paso, hay que seleccionar las operaciones que queremos realizar, en este caso seleccionaremos todas (Imagen 8) y pulsamos Next.

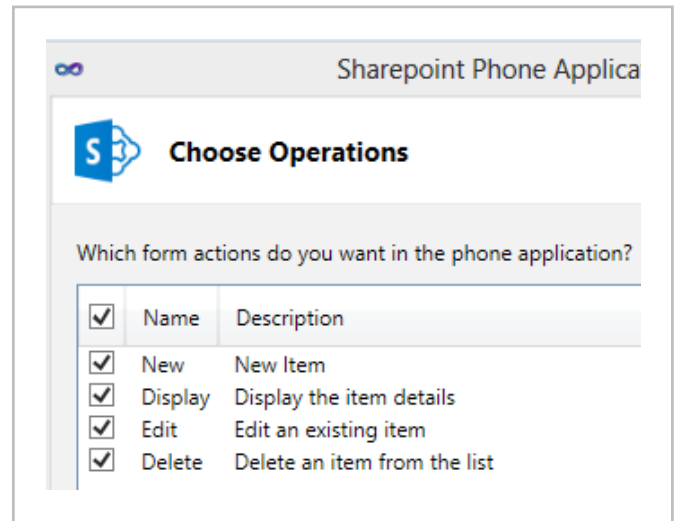


Imagen 8.- Paso 3 del asistente: Selector de formularios.

Y por último, debemos seleccionar qué columnas de la lista queremos mostrar en los formularios del paso anterior (Imagen 9) y hacer clic en Next y Finish.

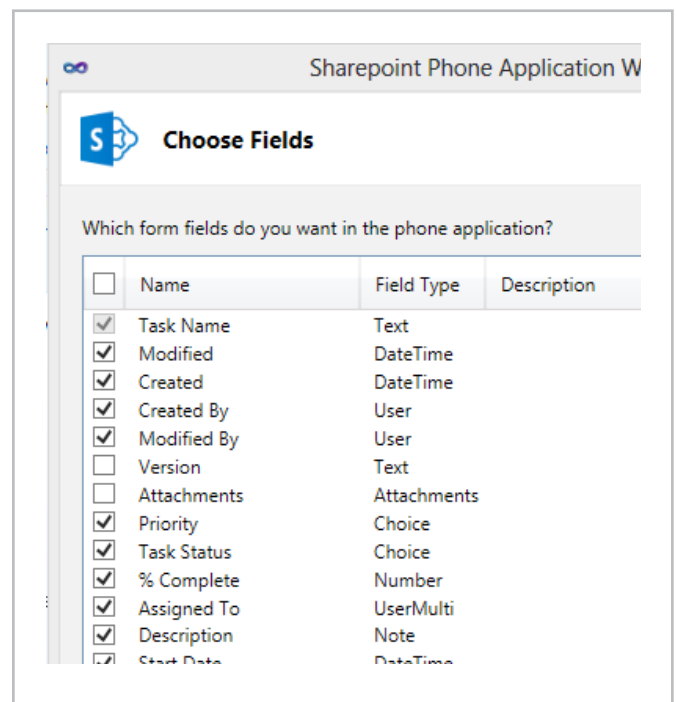


Imagen 9.- Paso 4 del asistente: Selector de columnas.

Una vez hayamos acabado y finalizado el asistente, Visual Studio creará el proyecto de Windows Phone con todos los datos que hemos pasado. El proyecto ya está preparado para acceder a SharePoint y contiene todo el código, consultas y formularios necesarios:

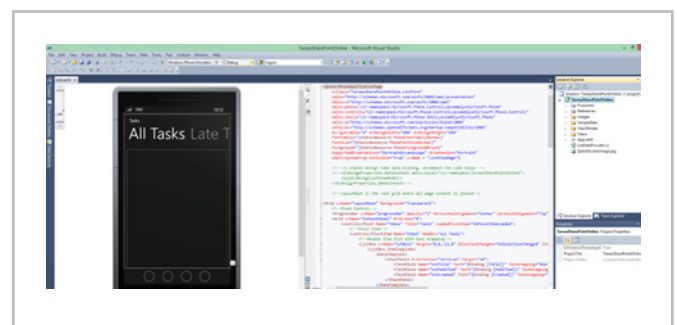


Imagen 10.- Vista del proyecto de Visual Studio.

Si lanzamos el proyecto, veremos que la app está lista para usar. Lo primero que nos pedirá es el login (en este caso en Office 365), como se puede ver en la Imagen 11.

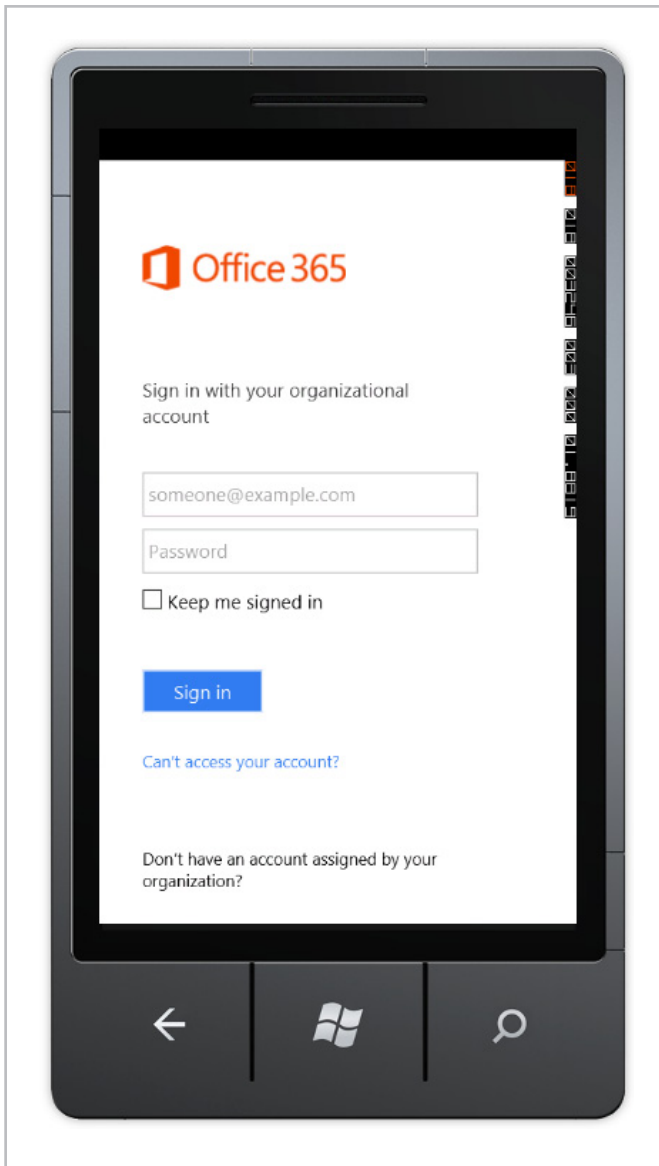


Imagen 11.- Login de la aplicación.

A continuación se muestran las distintas vistas en varios pivot. Podemos movernos de unos a otros, tal y como se muestra en la Imagen 12.



Imagen 12.- Las distintas vistas de la lista de tareas de SharePoint.

Si pulsamos en un elemento, se mostrará el formulario para visualizarlo. Si en este formulario pulsamos el botón eliminar lo borraremos. Si pulsamos el botón editar, se abre el formulario de edición. Si en una de las vistas pulsamos el botón de nuevo se mostrará el formulario de creación. Los distintos formularios se visualizarán como los que vemos en la imagen 13.

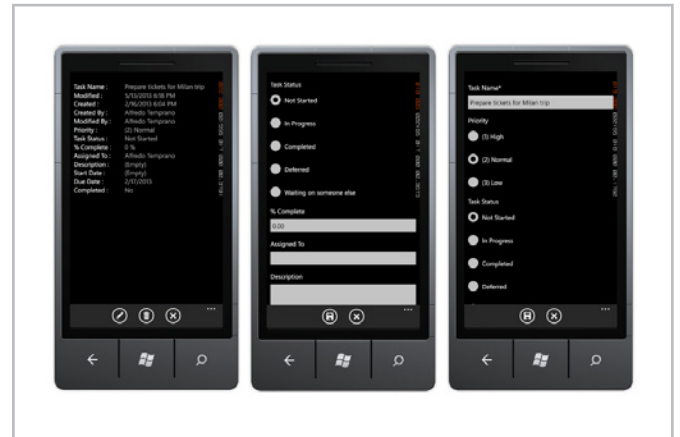


Imagen 13.- Formularios de visualización, creación y edición.

Como se ve en el ejemplo, en tan sólo 10 minutos se puede generar una aplicación totalmente funcional sin escribir ni una línea de código. Una vez que se tenga, hay aspectos que seguro que nos interesa cambiar (selectores de fechas, mostrar coordenadas en mapas, añadir más formularios, validaciones de datos....), para los que habrá que tocar el código y los formularios. Varios de estos aspectos los veremos en el próximo número.

Si no puedes esperar, quieres investigar estas funcionalidades y quieres ampliar más conocimientos que los que trataremos en el siguiente artículo, hay guías excelentes en MSDN. Si estas guías no habría sido posible escribir el presente artículo:

- Build mobile apps for SharePoint 2013 (<http://msdn.microsoft.com/en-us/library/jj163228.aspx>)
- How to: Set up an environment for developing mobile apps for SharePoint (<http://msdn.microsoft.com/en-us/library/jj163943.aspx>)
- Overview of Windows Phone SharePoint 2013 application templates in Visual Studio (<http://msdn.microsoft.com/en-us/library/jj163786.aspx>)
- Architecture of the Windows Phone SharePoint List Application template (<http://msdn.microsoft.com/en-us/library/jj163209.aspx>)
- How to: Create a Windows Phone SharePoint 2013 list app (<http://msdn.microsoft.com/en-us/library/jj163259.aspx>)
- How to: Store and retrieve SharePoint list items on a Windows Phone (<http://msdn.microsoft.com/en-us/library/jj163143.aspx>)

- How to: Implement business logic and data validation in SharePoint 2013 app for Windows Phone (<http://msdn.microsoft.com/en-us/library/jj164107.aspx>)
- How to: Support and convert SharePoint 2013 field types for Windows Phone apps (<http://msdn.microsoft.com/en-us/library/jj163213.aspx>)
- How to: Customize list item queries and filter data for Windows Phone apps (<http://msdn.microsoft.com/en-us/library/jj163224.aspx>)
- How to: Customize the user interface of a SharePoint 2013 list app for Windows Phone (<http://msdn.microsoft.com/en-us/library/jj163113.aspx>)
- How to: Use multiple SharePoint 2013 lists in a Windows Phone app (<http://msdn.microsoft.com/en-us/library/jj163289.aspx>)
- How to: Configure and use push notifications in SharePoint 2013 apps for Windows Phone (<http://msdn.microsoft.com/en-us/library/jj163784.aspx>)
- Integrating location and map functionality in SharePoint 2013 (<http://msdn.microsoft.com/en-us/library/jj163135.aspx>)

- How to: Create a mobile app in SharePoint 2013 that contains data from an external data source (<http://msdn.microsoft.com/en-us/library/jj164046.aspx>)
- How to: Integrate maps with Windows Phone apps and SharePoint 2013 lists (<http://msdn.microsoft.com/en-us/library/jj163813.aspx>)
- How to: Build search-driven mobile apps with the Navigation and Event Logging REST interfaces (<http://msdn.microsoft.com/en-us/library/jj163303.aspx>)

Nos vemos en el próximo número.

MIGUEL TABERA PACHECO

SharePoint Architect

Microsoft Active Professional

miguel.tabera@outlook.com

@migueltabera

<http://www.sinsharepointnohayparaíso.com>



Expertos en SharePoint



Soluciones
integrales



Queremos ser su socio tecnológico

Windows 8

SharePoint

Windows Phone

Windows Azure



comercial@gsc.es

www.gsc.es

Agregarle comentarios a las páginas de SharePoint Server 2010

Resumen

El artículo muestra una forma sencilla para incorporar comentarios en las páginas de un sitio de SharePoint, pudiendo interceptarlos antes de que queden públicos para los demás usuarios.

Artículo

Hace algún tiempo, trabajando en un portal de intranet en el cual se utilizó la plantilla de sitio de publicación, me encontré con el siguiente requerimiento: El cliente nos manifestó la necesidad de contar con un sub sitio de noticias organizacionales. Estas noticias deberían ser visualizadas en diferentes páginas de la intranet, como por ejemplo la home. Para esto se había desarrollado una WebPart que listaba todas las noticias con su imagen y una breve descripción, y al hacer clic en el título de una de ellas, se redirigía al detalle de la misma.

Hasta este punto el requerimiento no era nada fuera de lo común, el problema surgió cuando se nos manifestó la necesidad de que cada noticia en su detalle, contara con un área en la cual los usuarios pudieran realizar comentarios.

Adicionalmente se sumó la necesidad de incorporar un control del texto ingresado por el usuario. El mismo debía ser evaluado con la intención de minimizar la aparición de palabras inapropiadas a la hora de mostrar al público el comentario. A partir de esta definición, se nos solicitó contar con una lista que contenga todas aquellas palabras que la organización entendía inapropiadas, con el objetivo de poder administrar cada una de ellas.

Por lo tanto, el texto ingresado por un usuario sería analizado, verificando que ninguna de las palabras definidas como inapropiadas exista en el texto, en el caso de que se encontrara, el comentario se eliminaría automáticamente.

Luego de tener la definición completa del requerimiento, y de analizar cuál era la mejor alternativa para satisfacer el mismo, se presentó la opción de utilizar las capacidades sociales que SharePoint Server 2010 provee, muchas de ellas utilizan el servicio de sincronización de perfiles, puntualmente utilizan la base de datos Social Tagging, que se define a la hora de crear el servicio antes mencionado.

Imagen 1.- Profile Services – Sección Social Tagging.

Luego de haber definido que íbamos a trabajar con las características sociales provistas por SharePoint, se buscó aprovechar el Web Control SocialCommentControl que nos proveer la funcionalidad de comentarios.

A partir del mismo se pasó a definir cómo se iba a incorporar dicho Web Control en las páginas de noticias, así como también cómo se iba a trabajar para llevar a cabo el control del texto basado en la lista de palabras inapropiadas antes mencionada.

La solución que finalmente se implementó derivó en la creación de un Web Control que extendía la funcionalidad out of the box del control SocialCommentControl, con la intención de incorporar la capacidad de analizar y controlar el texto ingresado.

Cabe destacar que se debió referenciar una serie de dlls de modo de tener a disposición las características sociales de este.

En la siguiente imagen se muestran las referencias necesarias, para poder contar con las características sociales antes mencionadas:

“... forma sencilla para incorporar comentarios en las páginas de un sitio de SharePoint”

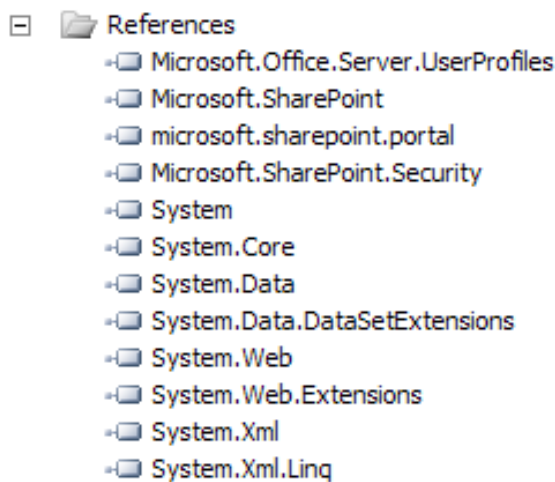


Imagen 2.- Ensamblados necesarios.

Hasta este punto se había construido un Web Control que extendía el SocialCommentControl y que permitía utilizar todas las funcionalidades out of the box que este provee. Por lo tanto, para lograr contar con el chequeo de palabras inapropiadas se sobrescribió el método RenderComment el cual se encarga de “dibujar” el comentario en pantalla, de esta forma logramos interceptar el proceso de presentación de los comentarios, pudiendo efectuar el control del texto.

A continuación se presenta un extracto de código que presenta la idea general:

```
public class SocialCommentWebControl :
    SocialCommentControl
{
    protected override void
    RenderComment(StringBuilder strBuilder,
        Microsoft.Office.Server.SocialData.SocialComment
        comment,
        int sequence,
        int page)
    {
        if
        (comment.Comment.Contains("Palabra a
        controlar"))
            comment.Delete();
        else
            base.RenderComment(strBuilder,
            comment, sequence, page);
    }
}
```

Cabe destacar que el código es un ejemplo y que no está implementado el acceso a la lista de palabras inapropiadas. Por último y como cierre, se construyó una WebPart con el objetivo de poder maximizar la reutilización del nuevo control. Pudiendo utilizar este, tanto en los layouts de noticias (como indicaba el requerimiento inicial del cliente), como también en cualquier WebPart Zone en la cual tuviéramos la necesidad de contar con comentarios.

JUAN PABLO CAPDEVILA

jcapdevila@pragmaconsultores.com

<http://www.pragmaconsultores.com>

MVA

Microsoft Virtual Academy

Boost your Microsoft technology
skills with **free training**.

i

39

¿Cómo se hizo CompartiMOSS.com? – Parte II

Resumen

Esta es la segunda parte del artículo sobre cómo se hizo CompartiMOSS.com con SharePoint 2013. Podremos ver cómo crear diseños de página e incluir campos del tipo de contenido o WebParts de una forma muy sencilla.

Además también veremos cómo crear plantillas para elementos de WebParts usando para ello HTML y JavaScript en vez de los archivos XSLT de SharePoint 2010, con lo que podremos controlar todo el código generado y aplicarle nuestras personalizaciones de una forma sencilla.

Artículo

En esta segunda entrega, veremos las secciones que nos quedaron pendientes para terminar un diseño de un sitio de publicación de SharePoint 2013, los diseños de página y las plantillas de elementos, y veremos que crear un diseño personalizado es ahora más fácil que nunca y, que con poco esfuerzo y bastante conocimiento de HTML y JavaScript, podremos tener terminado nuestro sitio, listo para que los editores de contenido comiencen a enriquecerlo.

Crear Diseños de página

Al igual que hacíamos en SharePoint 2010, podremos personalizar nuestros diseños de página para crear plantillas que estructuren la información, de una forma acorde con nuestro diseño. De esta forma, para CompartiMOSS hemos implementado las siguientes plantillas

- Home
- Plantilla para detalles de un Número
- Plantilla para detalles de un Artículo
- Plantilla para detalles de un Autor
- Plantilla para detalles de un Partner

Para crear un nuevo Diseño de página, hay que seleccionar la opción “Editar diseños de página” desde el “Administrador de diseños” y posteriormente seleccionar la acción “Crear un diseño de página” como se ilustra en la imagen 1.

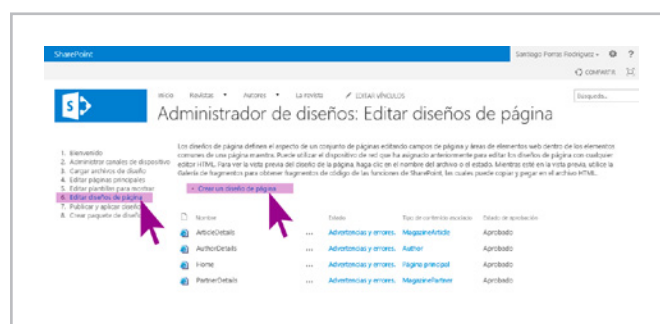


Imagen 1.- Listado con diseños de página y pasos a seguir para crear uno nuevo.

Una vez hayamos seleccionado la opción “Crear un diseño de página” se abrirá un cuadro de diálogo donde habrá que establecer un nombre para la plantilla y seleccionar una MasterPage y el tipo de contenido que se mostrará en las páginas con esta nueva plantilla.

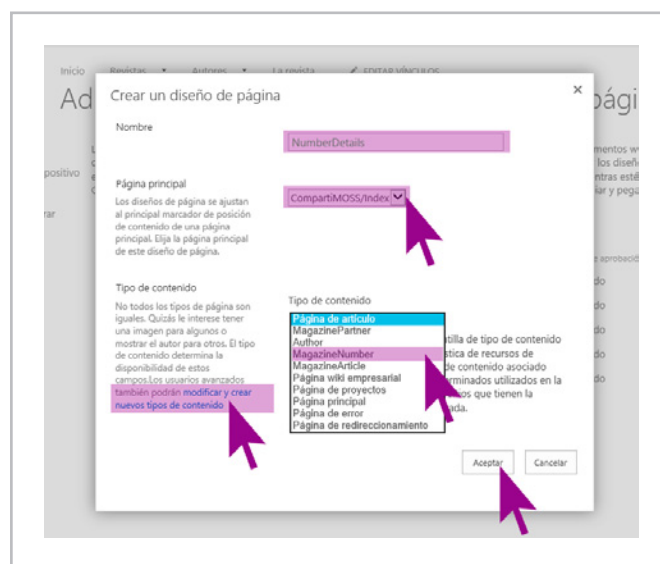


Imagen 2.- Cuadro de diálogo para crear un diseño de página.

Como podéis ver en la imagen 2, para CompartiMOSS hemos creado algunos tipos de contenido “Author”, “MagazinePartner”, “MagazineNumber” y “MagazineArticle” con sus correspondientes campos personalizados que nos ayudaran a relacionarlos entre sí y

que nos permitirán mostrar la información necesaria para cada uno de ellos.

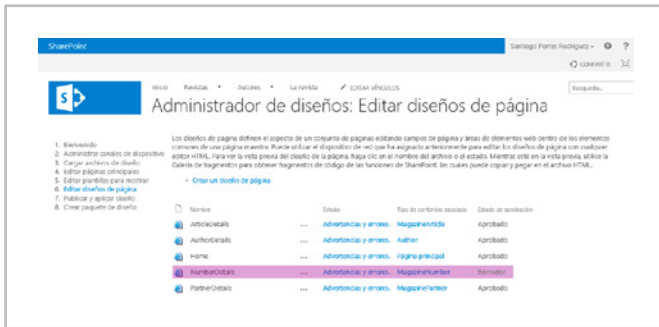


Imagen 3.- Listado de diseños de página disponibles tras crear NumberDetails.

Al aceptar y tras unos segundos, la nueva plantilla aparecerá en la lista como hemos visto en la imagen 3, y ya podremos empezar a editarla en nuestro editor de HTML preferido, en este caso, he usado WebMatrix 3 que va fenomenal para estas tareas.

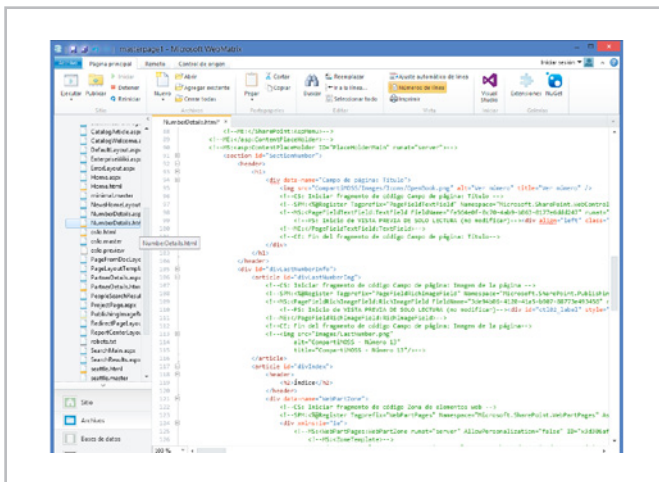


Imagen 4.- Edición del diseño de página NumberDetails en WebMatrix 3.

NOTA: Tenemos que tener en cuenta varias opciones:

- Para editar un diseño de página debemos tenerlo desprotegido.
- Sólo es necesario que desprotejamos el archivo HTML.
- Para poder usarla la primera vez deberemos haberla protegido antes, si no aparecerá disponible para seleccionar.

En estos momentos, es posible editar la plantilla desde su HTML pero, ¿cómo podemos añadir nuevos campos, WebParts u otros elementos? Esta tarea es muy sencilla y se realiza de la misma forma que para insertar elementos en la Masterpage.

1. Seleccionar el diseño de página.
2. Seleccionar "Fragmentos de código" de la vista previa del diseño de página.

3. Seleccionar el elemento que se quiera insertar, configurar si es necesario.
4. Copiar código generado y pegar en la plantilla de página desde el editor de HTML en la ubicación donde se deba mostrar según el diseño que hayamos realizado.

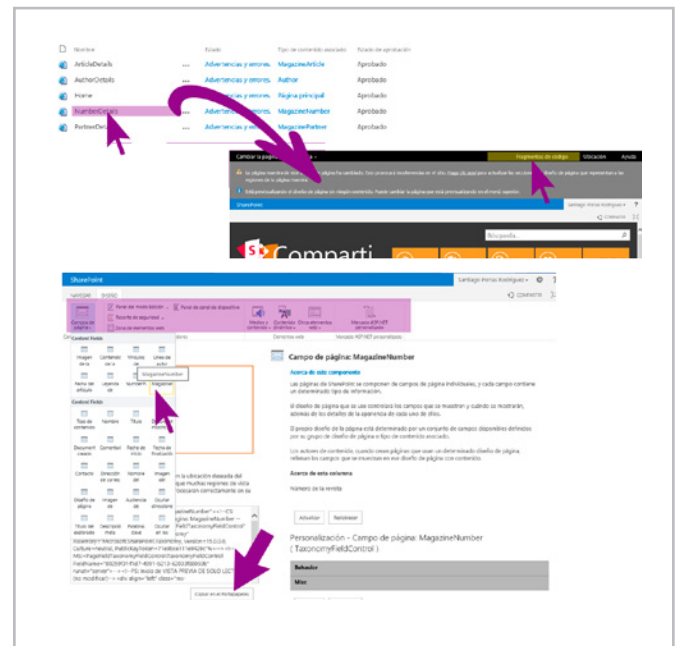


Imagen 5.- Pasos para crear fragmentos de código.

Con esto ya hemos creado el diseño de página adecuado para mostrar nuestro contenido y, tras proteger y publicar el diseño de página podemos ver el resultado deseado.

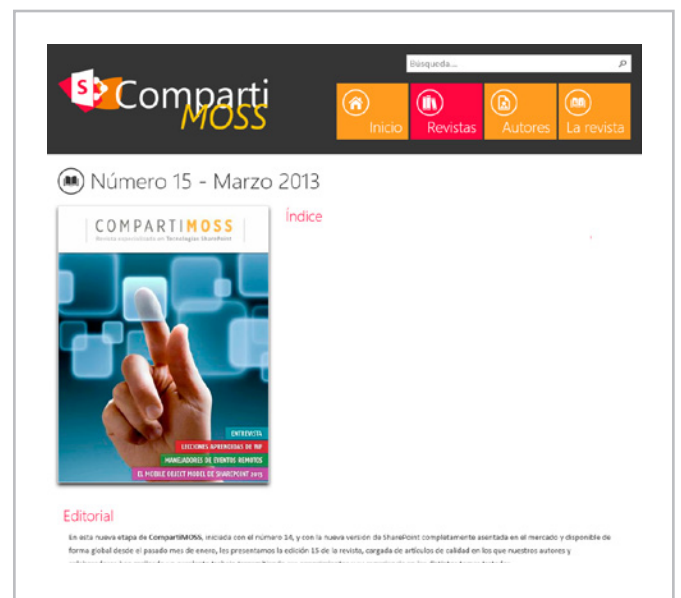


Imagen 6.- Resultado obtenido tras generar el diseño de página.

Definición de plantillas de elementos de contenido
Esta es la parte más compleja quizás porque aquí interviene muchísimo el conocimiento que podamos tener de JavaScript y, para entenderlos, quería hacer una pequeña introducción previa.

Las plantillas de elementos de contenido vienen a sustituir, dentro del nuevo paradigma de SharePoint 2013, a los ItemStyles que usábamos en versiones anteriores para establecer cómo se deben mostrar los elementos de contenido de un Web Part. En este nuevo ámbito, aunque siguen existiendo los ItemStyles por compatibilidad con SharePoint 2010, deberemos afrontar estas plantillas con HTML y JavaScript, haciendo uso de un “pseudolenguaje” de sustitución de elementos dentro de las propias plantillas.

Una vez hecha esta introducción un poco abstracta, pasemos a ver cómo generar una nueva plantilla de elementos de contenido tomando como ejemplo, el diseño de página de un número de la revista NumberDetails que contiene una lista con los artículos pertenecientes a ese número en cuestión.

html al documento que la contiene, tal y como se puede ver en la imagen 8.

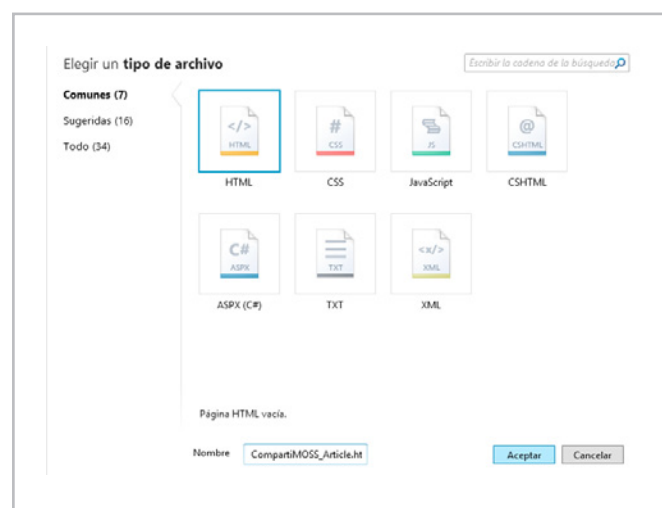


Imagen 8.- Nuevo archivo HTML CompartiMOSS_Article.html.

“... cómo se hizo CompartiMOSS.com con SharePoint 2013.”

Para crear una nueva plantilla, lo más sencillo es ir al editor de HTML que estemos usando y crear un nuevo documento HTML bajo la carpeta “Display Templates ⇒ Content Web Parts” como se ilustra en la imagen 7.

Una vez creado el archivo, si guardamos y refrescamos esta carpeta desde el editor de HTML que estemos usando, podremos comprobar cómo se crea un archivo con extensión .js de forma automática, con el mismo nombre del documento HTML que acabamos de crear para guardar la plantilla de artículo tal y como se muestra en la imagen 9. Este archivo contendrá el código que finalmente ejecutará SharePoint, pero ¡jojo! No debemos tocar este archivo.

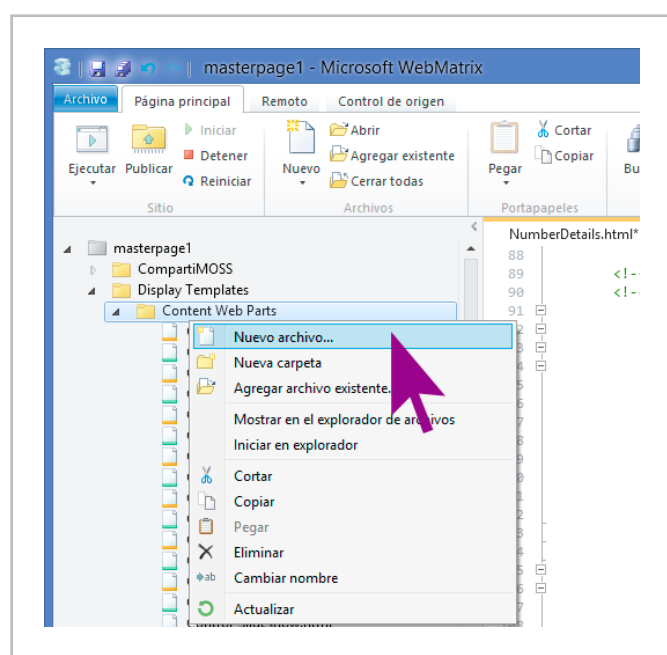


Imagen 7.- Crear un nuevo archivo en Display Templates ⇒ Content Web Parts.

Para el ejemplo de CompartiMOSS, esta plantilla se corresponderá con los artículos que contiene un número de la revista, por lo que llamaré CompartiMOSS_Article.

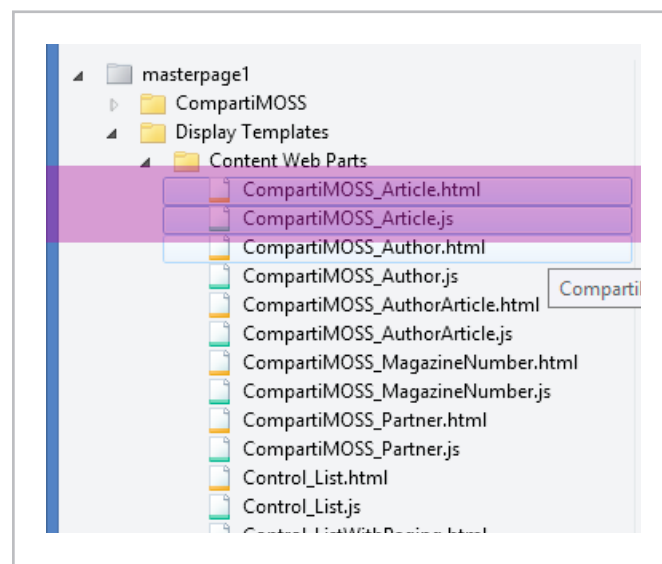


Imagen 9.- Se crea automáticamente un archivo con extensión .js.

NOTA: Tenemos que tener en cuenta varias opciones:

- Para editar una plantilla debemos tenerla desprotegida.
- Sólo es necesario que desprotejamos el archivo HTML.
- Para poder usarla la primera vez deberemos haberla protegido antes, si no aparecerá disponible para seleccionar.

El archivo HTML con la plantilla debe contener, entre otras cosas, la estructura completa de una página HTML con su sección “head” y su sección “body” donde iremos insertando nuestro código para maquetar los elementos que queremos mostrar mediante esta plantilla pero, como ya indiqué al principio, es MUY especial ya que se trata de HTML con JavaScript y un “pseudocódigo” de etiquetas que SharePoint reemplazará por el código JavaScript final y que ensamblará en el archivo con extensión .js que generó automáticamente.

En la sección “head” de la página, debe estar la definición de la plantilla junto a los parámetros que va a recibir tal y como se muestra en la imagen 10.

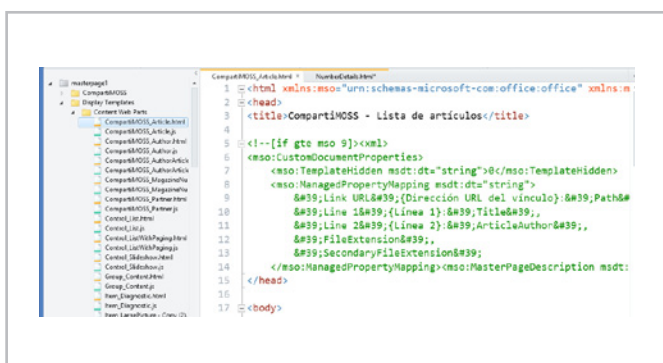


Imagen 10.- Sección “head” de la plantilla para los elementos.

En la sección “body”, debemos añadir la estructura HTML de los elementos que queramos mostrar mediante esta plantilla y, además, las etiquetas correspondientes a los campos como se puede apreciar en la imagen 11.

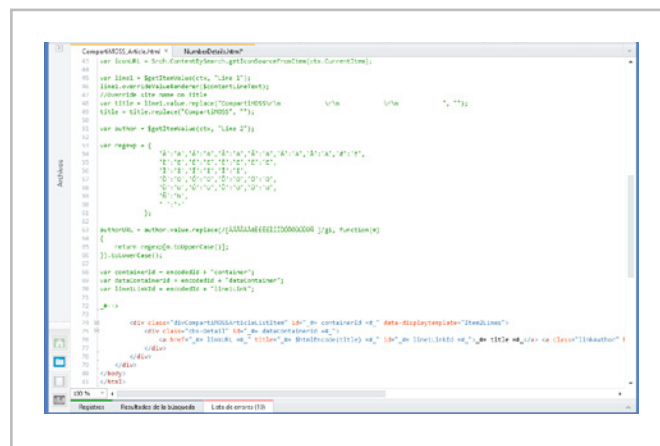


Imagen 11.- Sección “body” de la plantilla para los elementos.

A priori puede parecer muy complicado este nuevo formato de plantillas, pero rápidamente podréis comprobar que al final resulta muy sencillo generar nuevas plantillas en las que se puede controlar completamente el código HTML generado y que, además, pueden contener JavaScript para realizar operaciones, animaciones, etc.

NOTA: Una buena forma de comenzar a entender cómo funcionan estas plantillas es revisar cómo están estructuradas algunas de las que vienen por defecto y comprobar en qué se traduce el código en el archivo con extensión .js. Además, para empezar la primera plantilla se puede optar por copiar el código de una de las existentes y modificarlo a nuestro antojo.

Podemos ver las nuevas plantillas creadas en la lista desde el Administrador de diseños seleccionando la opción “Editar plantillas para mostrar” tal y como podemos ver en la siguiente imagen.

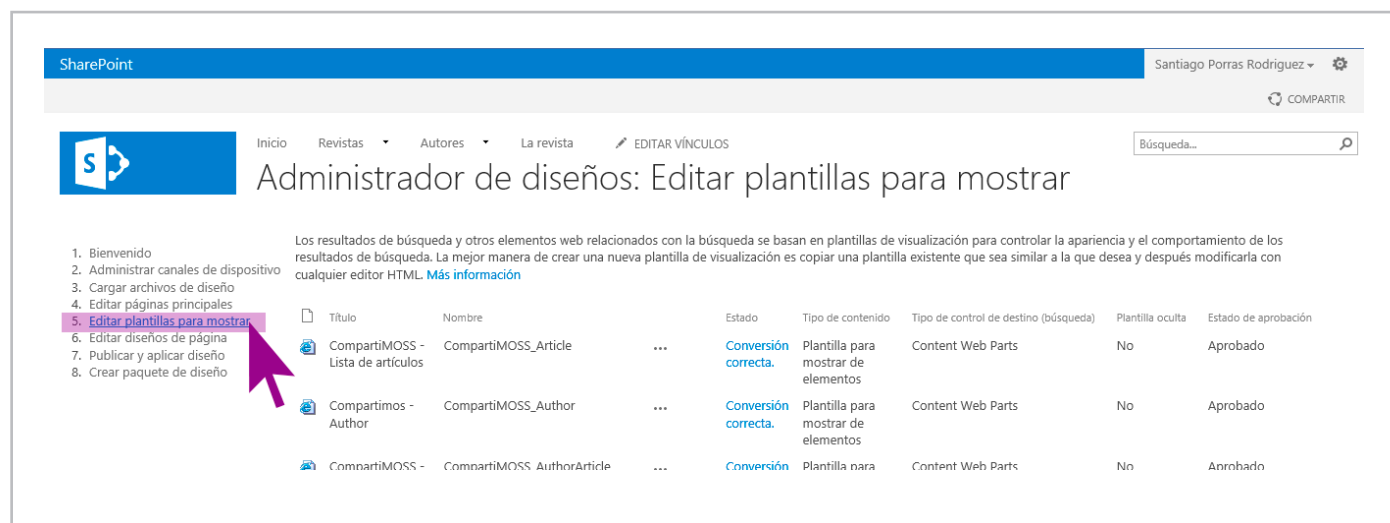


Imagen 12.- Lista de plantillas de elementos.

El resultado podemos verlo en la página de un número de la revista, donde se incluye un listado con los artículos que contiene mediante el uso de un WebPart de Búsqueda

de contenido que hace uso de esta plantilla para sus elementos.



Imagen 13.- Resultado obtenido tras generar la plantilla para los artículos.

Publicar y aplicar diseño

Finalmente sólo nos queda publicar nuestro diseño y aplicarlo al sitio. Esto lo podemos realizar mediante el enlace hacia la lista de “Páginas maestras” y que se encuentra disponible tanto desde el Administrador de diseños usando para ello la opción mostrada en la imagen 14 o, también desde la “Configuración del sitio”.

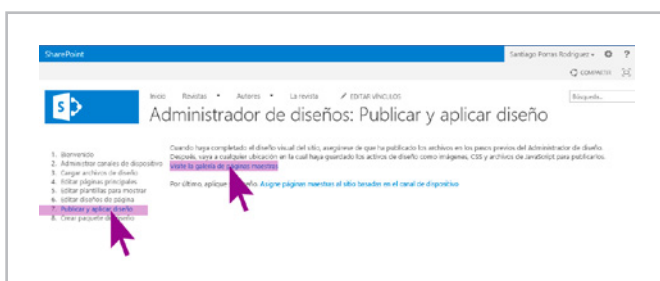


Imagen 14.- Link de acceso a los archivos de diseño.

Una vez en el listado de elementos de diseño, compuesto por los diseños de página, las páginas maestras, plantillas de elementos, ... sólo tenemos que proteger y publicar una versión superior de los ficheros que hayamos tocado, sólo los .html, ya que los generados por SharePoint se protegen automáticamente.

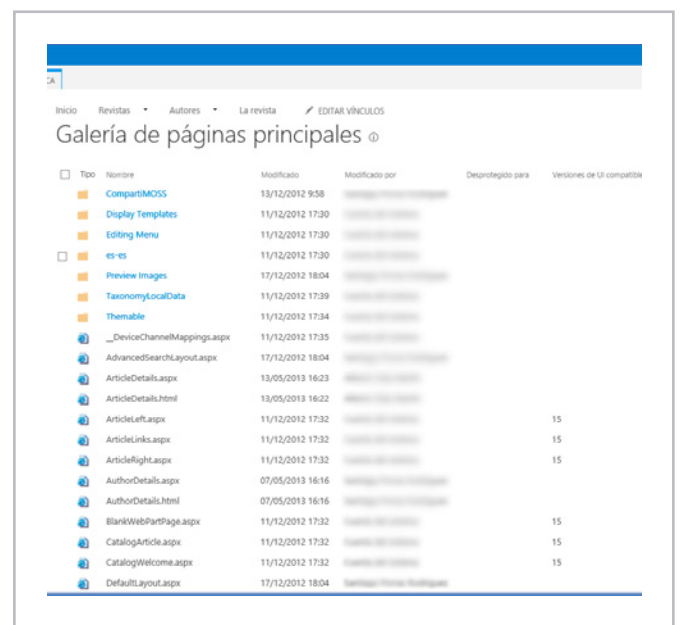


Imagen 15.- Listado de elementos de diseño.

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

Preparados, Listos, Ya!



Migración



Governance



Calidad de datos



Mobilidad



Productividad

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

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

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



44

SharePoint Designer 2013 y Flujos de Trabajo

Resumen

SharePoint Designer 2013 es la mejor herramienta para un desarrollo rápido de aplicaciones de SharePoint. A través de SharePoint Designer, tanto programadores como usuarios especializados pueden crear soluciones de SharePoint en tiempos cortos para dar respuesta a las necesidades de la organización.

Artículo

Por un lado, SharePoint Designer brinda a usuarios especializados la posibilidad de crear soluciones sin código que contengan una variedad de casos comunes, para ser trabajados en sitios colaborativos y/o Páginas Web. A través de esta herramienta se pueden brindar soluciones de inteligencia empresarial y flujos de trabajo en un entorno fácil de usar.

Por otro lado, los programadores pueden usar SharePoint Designer 2013 para lograr un inicio rápido en proyectos de desarrollo de SharePoint.

Con la llegada de SharePoint Designer 2013 se ha diversificado la manera de crear y editar flujos de trabajo. Es aquí donde encontramos la mayoría de las características nuevas.

Algunas de estas nuevas características son:

- Capacidad para crear flujos de trabajo basados en la nueva infraestructura de flujos de trabajo de .NET Framework 4.X Permitiendo a la vez que se sigan creando y editando los flujos de trabajo antiguos de SharePoint 2010
- Integración ajustada con etapas:
 - Ya no es necesario que los flujos de trabajo estén en serie.
 - Ahora todos los flujos de trabajo de 2013 están basados en estado, y es posible hacer que un flujo de trabajo salte a cualquier etapa deseada, incluso a una etapa anterior.
- El nuevo diseñador visual:
 - Una superficie de diseño completamente nueva que integra el lienzo de Visio directamente dentro de SharePoint Designer.
 - Permite crear flujos de trabajo con la acción de hacer clic y arrastrar.
 - Incluso es posible establecer las propiedades de cada acción directamente dentro del diseñador visual. No es necesario pasar de nuevo al diseñador basado en texto si no deseamos hacerlo.
- Copiar y pegar:
 - Ahora es posible copiar, cortar y pegar dentro del diseñador basado en texto.
- Bucles:
 - Ahora los bucles son una acción plenamente integrada que se puede utilizar dentro de todos los flujos de trabajo de 2013
- Llamar a la acción de servicios web:
 - Ahora se puede llamar a los servicios web REST/HTTP directamente desde dentro de los flujos de trabajo de SharePoint 2013.
- Iniciar acciones de flujo de trabajo:
 - Ahora tenemos una acción que permite generar flujos de trabajo antiguos de 2010 directamente desde dentro de un flujo de trabajo de 2013.
 - Se puede utilizar esta acción para aprovechar cualquier funcionalidad de flujo de trabajo de 2010 (por ejemplo, las acciones personalizadas antiguas).
- Empaquetado:
 - Ahora se pueden empaquetar flujos de trabajo individuales directamente desde dentro de SPD
- Variables de tipo de idioma:
 - Ahora se pueden crear variables que pueden contener una diversidad de valores
 - La combinación de esto con la nueva acción de bucle abre todo un espectro de nuevos escenarios.

Antes de comenzar a trabajar con SharePoint Designer y comenzar a realizar un ejemplo es importante conocer y tener en cuenta algunos conceptos como los siguientes:

- **Paso:** cada una de las secuencias que compone un workflow, puede haber varios pasos en un mismo flujo, y podemos elegir que se ejecuten en serie (los pasos se

ejecutan uno detrás de otro) o en paralelo (todos los pasos se ejecutan a la misma vez, por ejemplo, si queremos esperar a que varias personas aprueben un documento).

- **Condición:** parte de una condición opcional que pregunta si el paso se realiza o no.
- **Acciones:** las tareas que se ejecutarán si la condición de un paso es cierta, pueden ser varias encadenadas.
- **Variables de inicio:** unas variables que el workflow pedirá al iniciarse. Pueden ser texto, fecha/hora, número o enumeraciones.
- **Variables:** otras que definimos en el procesamiento del

workflow para nuestras necesidades.

Ejemplo de un flujo en SharePoint

OBJETIVO

Automatizar un aviso cada vez que una “Fecha de vencimiento” vaya a ocurrir. Este aviso se debe enviar a los 7 meses antes del vencimiento, luego a los 6 meses, a los 5 meses antes de vencer y así sucesivamente hasta que envíe el aviso 1 mes antes y el propio día que vence. En una lista de SharePoint, vamos a tener todos los registros, que básicamente constan de los siguientes campos:

Tipo de Producto	Marca	Producto	Fecha de emisión	Fecha de vencimiento	Registro	País de Procedencia	Casa Fabricante	Casa Representante	Status	Presentación	Comentarios	Aviso de vencimientos
------------------	-------	----------	------------------	----------------------	----------	---------------------	-----------------	--------------------	--------	--------------	-------------	-----------------------

Imagen 1.- Campos de la lista de SharePoint.

A esta lista, asociamos el workflow.

Nuestro workflow hará lo siguiente:

- Se llenará el registro en la lista, primordial que nuestro registro llene correctamente la “Fecha de vencimiento”.
- Se tomará la fecha de vencimiento para poder calcular los meses que faltan para su vencimiento.
- Se registrará la fecha de envío de la notificación en el historial del flujo (este paso es realizado más que todo para verificar que se calcula la fecha correcta).
- Se detiene el flujo hasta la fecha de envío de la notificación (este flujo se detiene para los 7 meses, 6 meses, 5 meses, etc.).
- Al llegar la fecha se envía un correo con la notificación del tiempo en que esta por vencer.
- Se registra en el historial del flujo un mensaje de cuando fue enviada la notificación.

Inicio

Se crean las variables para cada fecha de vencimiento de tipo “Fecha y Hora”:

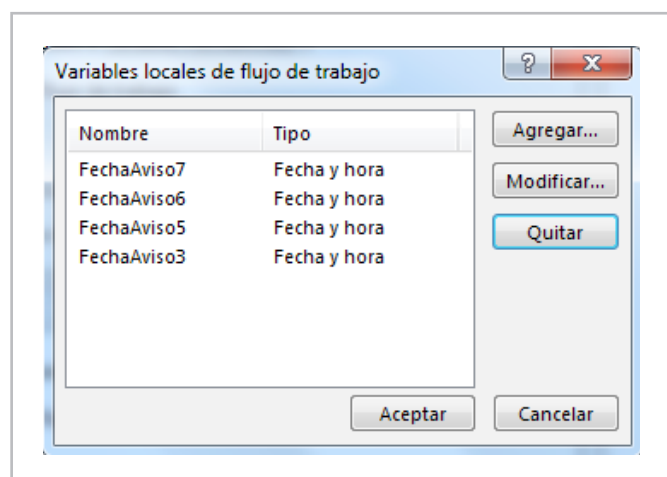


Imagen 2.- Definición de las variables necesarias.

Una vez que tenemos las variables locales creadas, comenzamos a crear los pasos para cada una de las fechas de vencimiento, en este caso, se hizo un paso “General” que contiene todos los siguientes pasos de las fechas de aviso.

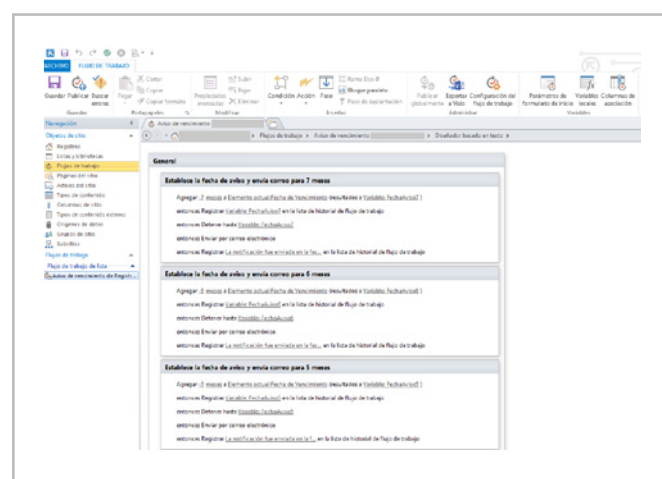


Imagen 3.- Definición de los pasos del flujo.

Para cada paso de las fechas de aviso se hizo lo siguiente:

- Se utilizó la acción de “Agregar hora a la fecha”

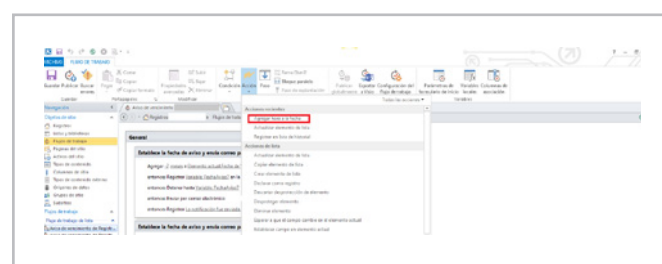


Imagen 4.- Acción “Agregar hora a la fecha”.

En esta acción se le restó 7 meses (para poder hacer la resta de los meses a una fecha de debe poner -7 meses) a la “Fecha de Vencimiento” colocada en la lista de registro y se asignó a la variable “FechaAviso7” la fecha calculada.

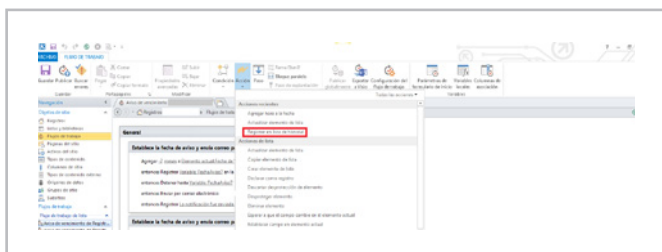


Imagen 5.- Configuración de la acción "Agregar hora a la fecha".

- La siguiente acción es registrar en el historial del flujo la fecha que se calculó:

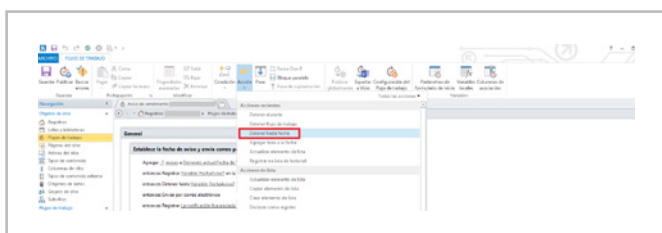


Imagen 6.- Acción "Registrar en lista de historial".

Esta acción lo que hace es que cuando arranca el flujo se calcula la fecha en que se enviará la notificación y la coloca a manera de registro en el historial (esta acción la coloqué para verificar que la fecha que se está calculando sea la indicada).

entonces Registrar Variable: FechaAviso7 en la lista de historial de flujo de trabajo

Imagen 7.- Configuración de la acción "Registrar en lista de historial".

- Luego la siguiente acción se encarga de "Detener el flujo" hasta la fecha en que se envía la notificación (en este caso la variable "FechaAviso7" tiene la fecha calculada para el envío del correo)

entonces Detener hasta Variable: FechaAviso7

Imagen 8.- Configuración de la acción "Detener hasta fecha".

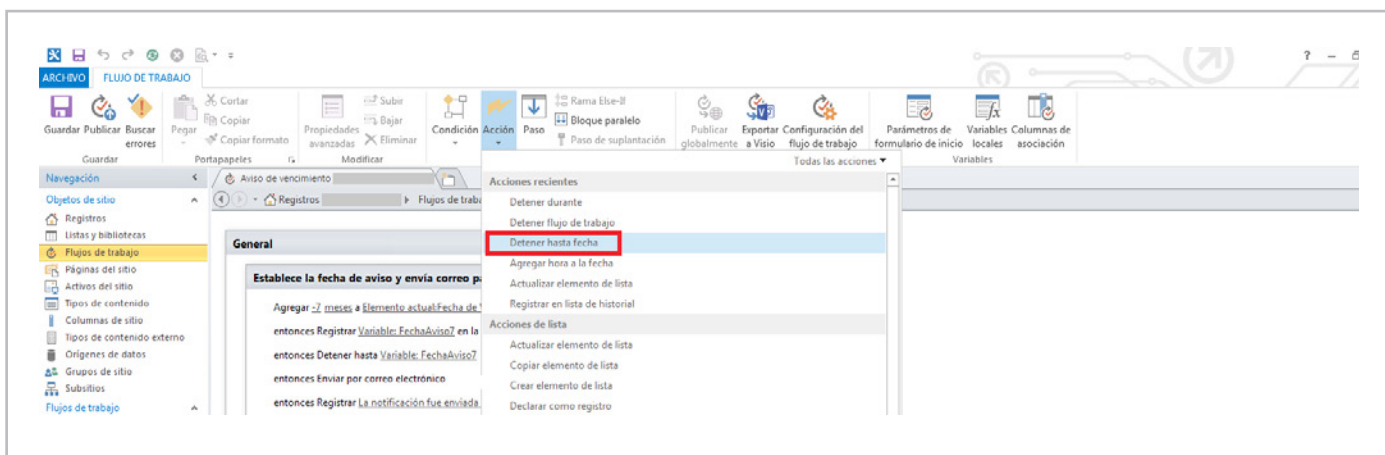


Imagen 9.- Acción "Detener hasta fecha".

- Una vez que llegue la fecha de envío de correo, el flujo se reactiva y envía un correo personalizado, notificando el tiempo que le queda al registro por vencer.

entonces Enviar por correo electrónico **Cristina Quesada**

Imagen 10.- Configuración de la acción de enviar correo electrónico.

Este correo electrónico se puede enviar a algún responsable previamente cargado en la lista y se coloca el campo de la lista para que se lo envíe a "x" persona, si no sabemos de anticipado a quien se le enviará a la hora de configurar el flujo. El correo electrónico que se envía es el siguiente:

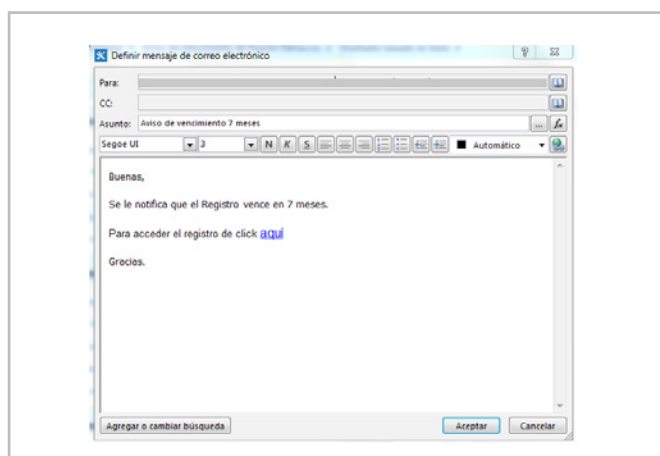


Imagen 11.- Configuración del mensaje de correo electrónico.

La llegada de SharePoint Designer 2013 se ha diversificado la manera de crear y editar flujos de trabajo

Se le indica al responsable cuantos meses tiene por vencer el registro y además se le pone un acceso para que ingrese directamente en el Registro.

- Y finalmente al enviar el correo electrónico se registra un mensaje en el historial del flujo indicando en qué fecha se envió la notificación de vencimiento.

entonces Registrar La notificación fue enviada en la fecha... en la lista de historial de flujo de trabajo

Imagen 12.- Configuración de la acción de registro en la lista de historial.

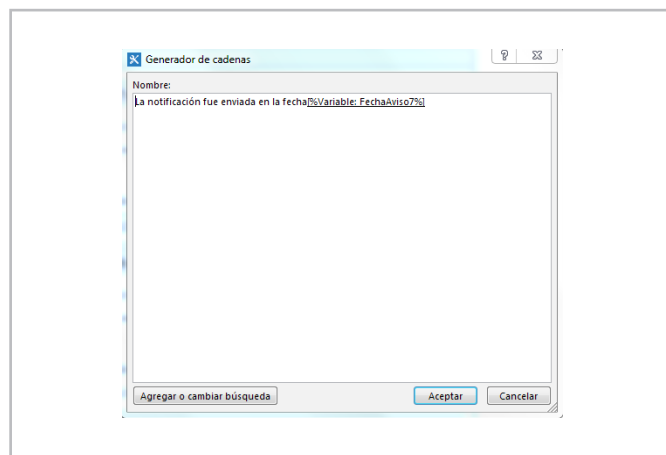


Imagen 13.- Detalle de la configuración de la acción de registro en la lista de historial.

Para los pasos siguientes de 6 meses, 5 meses, etc., se replican las mismas acciones.

CRISTINA QUESADA CALDERÓN

Consultora en Tecnologías SharePoint en Grupo CMA

cristi_q@hotmail.com

@cris_quesada

<http://cristina-quesada.blogspot.com/>





IberianSharePoint
Conference

más información:
www.iberiansharepointconference.com

1ª Conferencia Ibérica de **SharePoint**

10 de octubre de 2013
Madrid

Reserva ya tu plaza





Beezy



K2



KWizCom
Knowledge Worker Components



Microsoft | Innovation Center
CIIN - Cantabria



spenta



pasiona
consulting - development



general de
software



encamina
PIENSA EN COLORES



Plain Concepts



SolidQ



O'REILLY®



campus
MVP

48

Como configurar el Panel de Refinamiento sobre columnas de Metadatos múltiples

Resumen

En este artículo veremos cómo podemos añadir categorías de filtro al webpart del panel de refinamiento y cómo solucionar el problema de columnas de metadatos que permiten múltiples valores.

Artículo

La WebPart de Panel de Refinamiento es uno de las más utilizadas en SharePoint 2010 cuando se construyen Search Driven Applications. Esta WebPart nos permite realizar filtrados sobre los resultados de una búsqueda.

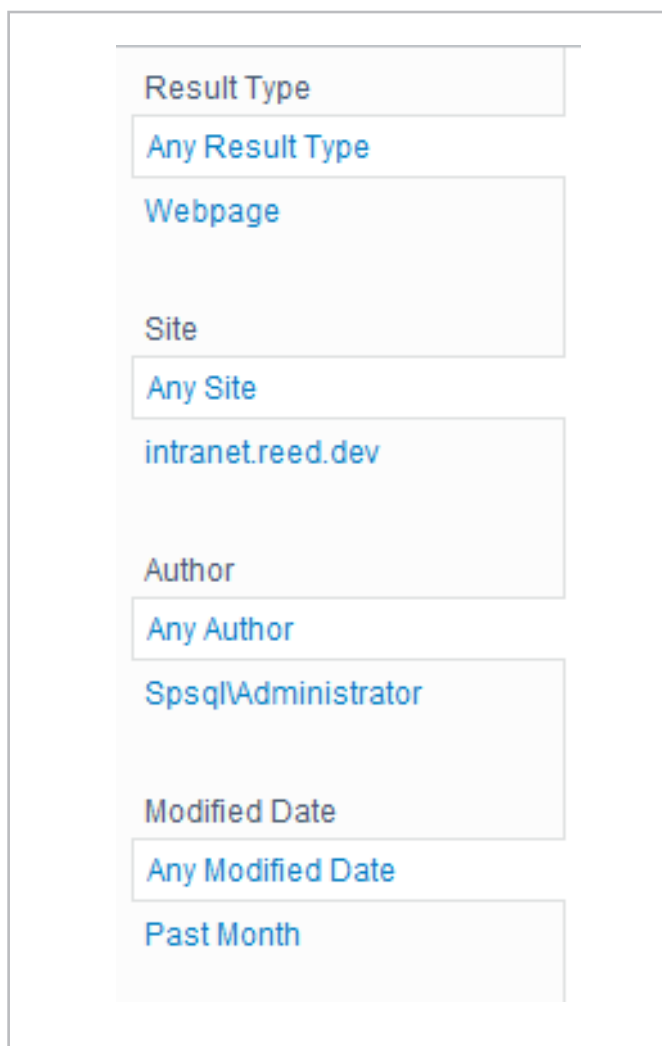


Imagen 1.- WebPart de Panel de Refinamiento

Como vemos en la Imagen 1, el panel de refinamiento trae algunos filtros por defecto: Autor, Tipo de resultado, etc.

Un requisito muy común en aplicaciones de búsqueda, es poder añadir filtros de otras columnas de metadatos, para ello hay que configurar la propiedad "Filter category definition", dentro del apartado "Refinement". Dicha propiedad es una cadena de XML con un formato específico, que permitirá definir las categorías que queremos utilizar como filtros.

Para añadir nuestra categoría, primero hemos de asegurarnos que tenemos una propiedad administrada en la aplicación de servicio de búsqueda. El siguiente código muestra como añadir una categoría de filtro para la propiedad "ContentType"

```
<Category Title="Content type"
  Description="Content type of the item"

  Type="Microsoft.Office.Server.Search.WebControls.ManagedPropertyFilterGenerator"
  MetadataThreshold="1"
  NumberOfFiltersToDisplay="4"
  MaxNumberOfFilters="20"
  SortBy="Frequency"
  SortByForMoreFilters="Name"
  SortDirection="Descending"
  SortDirectionForMoreFilters="Ascending"
  ShowMoreLink="True"
  MappedProperty="ContentType"
  MoreLinkText="show more"
  LessLinkText="show fewer" />
```

Y aquí podemos encontrarnos con el primer problema, y es que la propiedad administrada ContentType, tiene un comportamiento algo extraño.

Si no conseguimos hacer funcionar el código anterior, la solución pasa por crear una nueva propiedad administrada, digamos "ItemContentType" y mapearla a las mismas propiedades rastreadas que tiene la propiedad "ContentType". Si hacemos esto, tendremos el tipo de contenido disponible para el filtrado.

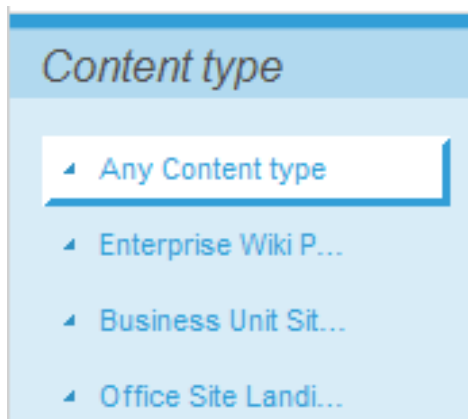


Imagen 2.- Filtro por Tipo de contenido en el panel de refinamiento.

Otro requisito muy común es que la categoría por la que filtrar, sea una columna de tipo Metadatos administrados. En este caso, un código como el anterior funcionará perfectamente, excepto en aquellos casos que la columna de metadatos admita múltiples valores, y tengamos catalogado algún documento o ítem con varios valores en dicha columna. La siguiente imagen muestra el resultado:

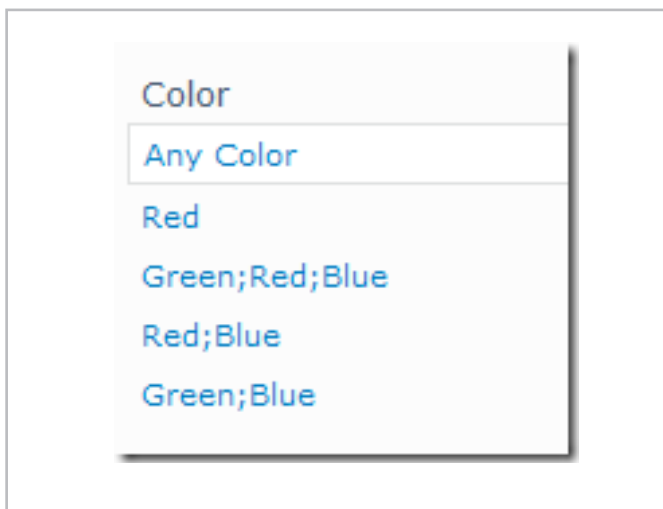


Imagen 3.- Filtro para una columna de metadatos administrados multi-valor.

Como se observa en la imagen 3, aparecen varios valores en un mismo filtro, separados por “;”, además, si se pincha sobre uno de estos filtros, no se devuelve ningún resultado.

Existen muchas referencias en internet sobre como añadir filtros al panel de refinamiento, y en algunos casos se comenta este tipo de comportamiento, sin embargo, no existen referencias sobre su solución, y en muchos casos se comenta que es un bug del producto, o que hay que desarrollar código para conseguir el efecto esperado.

Por suerte, SharePoint si tiene una solución OOB a este requerimiento. Para ello, tan solo debemos utilizar otro control generador de la categoría. Este control se define en el atributo type del XML. El control que necesitamos es:

```
Microsoft.Office.Server.Search.WebControls.TaxonomyFilterGenerator
```

Además de este control, hay otra consideración muy importante al respecto. Se trata de la propiedad a la que se mapea el control. Pero antes de ver como mapearlo, una aclaración sobre las propiedades administradas en columnas de metadatos.

Cuando se crea una columna de tipo metadato y se indexa en el buscador, SharePoint crea automáticamente 2 propiedades rastreadas, una, que podríamos llamar normal, ya que tiene el formato habitual de las propiedades rastreadas, y que sería:

ows_NombreDeLaColumna

Y una segunda propiedad, con el siguiente formato:

ows_taxId_NombreDeLaColumna

No lo puedo asegurar a ciencia cierta, pero mi sensación es que la primera columna almacena el valor en texto del metadato, mientras que la segunda, almacena el Identificador del Término en el almacén de términos.

Además de estas dos propiedades rastreadas, SharePoint crea una propiedad administrada, mapeada a la segunda rastreada (ows_taxId_NombreDeLaColumna), con el formato:

owstaxIdNombreDeLaColumna

El control TaxonomyFilterGenerator necesita que el atributo MappedProperty del XML, tenga un valor fijo: “ows_MetadataFacetInfo”. Pero además, el atributo Title será el nombre de la propiedad administrada autogenerada por SharePoint, eliminando el prefijo owstaxId. Por ejemplo, si tenemos:

	Nombre	Mapeada a
Columna de sitio	Product	
Propiedades rastreadas	ows_Product	
	ows_taxId_Product	
Propiedad administrada	owstaxIdProduct	ows_taxId_Product

El XML necesario para configurar correctamente nuestro filtro sería:

```
<Category Title="Product"
Description="Use this filter to restrict
results by a specific Product"

Type="Microsoft.Office.Server.Search.WebControls.Tax
onomyFilterGenerator"
MetadataThreshold="3"
NumberOfFiltersToDisplay="6"
MaxNumberOfFilters="20"
ShowMoreLink="True"
MappedProperty="ows_MetadataFacetInfo"
MoreLinkText="show more"
LessLinkText="show fewer"
ShowCounts="Count" />
```

Con esto resolvemos el problema, y obtenemos la funcionalidad deseada.

“... la WebPart de Panel de Refinamiento es uno de las más utilizadas en SharePoint 2010.”

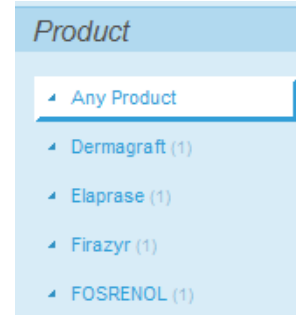


Imagen 4.- Filtro configurado sobre una columna de Metadatos que admite múltiples valores.

El siguiente enlace contiene algo de información al respecto de los controles que podemos utilizar para los filtros:

[http://msdn.microsoft.com/en-us/library/ff625183\(v=office.14\).aspx](http://msdn.microsoft.com/en-us/library/ff625183(v=office.14).aspx)

LUIS MAÑEZ

MCPD SharePoint 2010 and Microsoft Active Professional

@luismanez

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

El libro que no puede faltar,
si trabajas con SharePoint

Sharepoint 2013

de principio a fin

Gustavo Velez
Juan Carlos González
Fabián Imaz



Prólogo de Eva Ordóñez Pérez (Técnico Especialista en ventas de SharePoint y Office 365 en Microsoft Ibérica S.R.L.)

Diseño Gráfico en SharePoint Foundation 2010

Resumen

En este artículo se detalla un enfoque de desarrollo orientado a aplicar diseño creado por un diseñador gráfico en SharePoint Foundation 2010.

Requisitos Previos

En este artículo se detalla un enfoque de desarrollo orientado a aplicar diseño creado por un diseñador gráfico en SharePoint Foundation 2010. Al final del camino, se busca un sitio como el siguiente:



Imagen 1.- Ejemplo de página de inicio con diseño gráfico.

Dentro de los objetivos perseguidos, hay dos que se consideran los más importantes:

- Separar dentro de lo posible el trabajo del diseñador gráfico del trabajo del desarrollador
- Facilitar la mantenibilidad del sistema, evitando utilizar funciones menos conocidas por los desarrolladores, como crear una página maestra desde cero o utilizar código de servidor.

Principales lineamientos

- La plataforma es Foundation, no Server.
- El diseño es de un ancho fijo, diferente al estándar de SharePoint que ocupa el ancho total de la pantalla.
- Se busca preservar la ribbon porque posee funcionalidad muy importante, pero tratando de que no ocupe tanto el centro de atención.
- Existen elementos de cabecera que se repiten en

todas las páginas, lo que incluye una imagen, el usuario conectado y un menú horizontal. También un pie de página.

- La imagen de cabecera es más alta que la ribbon de SharePoint.
- Existen elementos que sólo se ven en la home page, como la efemérides, la fecha del día y el contenido en sí.
- El contenido de la home page, posee un diseño gráfico, muy diferente al estándar de SharePoint.

Enfoque técnico

- Se busca maximizar el uso de CSS.
- Se busca minimizar el uso de la Página Maestra.
- Se crea el contenido propio de la home con Cliente Object Model (Java Script) para poder formatearlo con mayor facilidad.

Página Maestra

El primer paso consiste en crear nuestra propia página maestra. En muchos casos se parte desde un página maestra básica (starter), pero nosotros preferimos copiar la estándar. Para ello debemos completar los siguientes pasos:

1. Abrimos nuestro sitio desde SharePoint Designer y nos ubicamos en la sección "Master Pages".
2. Copiamos la página "v4.master" y le ponemos un nombre como "intranet.master".
3. Luego elegimos la opción "Set as default master page".

Página de inicio con diferente diseño

Supongamos que necesitamos una página de inicio con tres o cuatro columnas. Si bien, desde SharePoint 2010, es posible elegir cualquier página como home page, vamos a optar por un método que nos dé mayor flexibilidad, siguiendo estos pasos:

1. Crear una librería de documentos llamada "Internal".
2. Creamos un archivo llamado "Intranet_Home.htm".

3. Agregar una CEWP en la home que llame a ese archivo. Ese archivo contendrá scripts y estilos que sólo se apliquen en la home page. Puro HTML, JavaScript, jQuery y CSS.

Hoja de estilos

Creamos un archivo llamado "Intranet.css" para definir los estilos que sobre-escriban los de SharePoint. Este archivo podemos almacenarlo en la librería "Internal".

Luego modificamos la página maestra (en SharePoint Server no sería necesario modificar la página maestra), agregando el siguiente código antes del "</head>":

```
<SharePoint:CssRegistration name="/sites/TU-
SITIO/Internal/Intranet.css" After="corev4.css"
runat="server"/>
```

Así queda nuestra nueva librería Internal y la librería de páginas maestras:

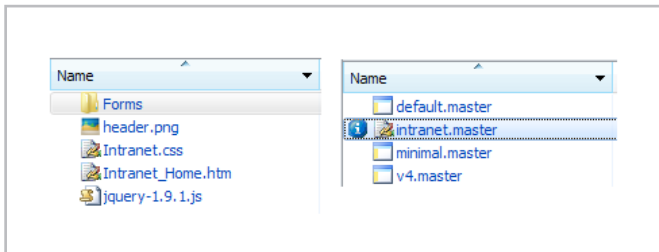


Imagen 2.- Librerías Internal y Master Pages.

También modificamos la página maestra para agregar el pie de página, siendo esta nuestra última modificación, ubicada luego de SharePoint:DeveloperDashboard:

```
<div id="intranet_footer">LA EMPRESA - RECURSOS
HUMANOS - PROYECTOS - NOTICIAS</div>
```

Lo importante, teniendo en cuenta uno de los objetivos, es que apenas hicimos dos simples cambios en la página maestra.

Estilos generales

En el siguiente código, se muestran los estilos generales que buscan:

- Crear una cabecera con diseño.
- Preservar la ribbon.

Este estilo se ubica dentro de Intranet.css:

```
/* -----
Ocultamiento de secciones
----- */
/* Oculto Acciones del Sitio */
```

```
.ms-siteactionsmenu { display: none; }
/* Ocultar Icono de navegación hacia arriba */
.s4-breadcrumb-anchor { display: none; }
/* -----
Ajustes estéticos
----- */
/* Seteo el color de fondo */
body { background-color: #828282; }
/* Aplico un ancho fijo, no 100% */
#s4-ribbonrow, #s4-workspace
{
    max-width: 1100px;
    margin: 0 auto;
    padding: 0px;
    background-color: white;
}
/* -----
Imagen de cabecera
----- */
/* Agrando la Ribbon porque la imagen es más alta
*/
#s4-ribbonrow { padding-top:37px; min-height:
133px !important; }
/* Oculto la fila de título y navegación */
#s4-titlerow { margin-top: -88px; }
/* Subo las solapas de la Ribbon para que no mo-
lesten a la imagen, especialmente si hay un logo a
la izquierda */
.ms-cui-topBar2 { margin-top: -37px; }
.ms-cui-tabContainer { border-top: #B8BABD 1px
solid; background-color: white; padding-top: 20px;
}
.ms-cui-tabBody { padding-bottom: 17px; back-
ground-image: none; }
/* Elimino el fondo del título y del top navigator */
.s4-title { background: none; }
/* Seteo mi imagen de cabecera en la ribbon row
*/
#s4-ribbonrow { background-image: url('/sites/TU-
SITIO/Internal/header.png'); }
/* Saco el fondo de la ribbon que molesta a la
imagen */
#s4-ribboncont { background: none !important; }
/* Posiciono en forma fija al top navigation */
.menu-horizontal { position: fixed !important; top:
134px !important; }
/* Subo el menú de usuario */
#RibbonContainer-TabRowRight { margin-top:
-10px; }
/* Elimino el borde superior de la ribbon */
.ms-cui-topBar2 { border: 0px; }
/* Ajustes de algunos bordes */
#s4-topheader2 { border-top: 0px !important; back-
ground: white !important; }
/* Borde derecho a la izquierda de la scrollbar
molesta */
#s4-titlerow { border-right: 0px; }
```



```

/* -----
Estilos en barra superior
----- */

/* Menú de bienvenida*/
.ms-welcomeMenu A:link {
color: #8F8F8F;
font-size: 11px;
font-family: Arial, Helvetica, sans-serif;
}

/* Colores de solapas de la ribbon, tratando de que
no sean tan visibles */
.ms-cui-tt-a, .ms-cui-cg-t { color: #B8BABD !important; }
.ms-cui-cg-t, .ms-cui-ct-ul { background: none !important; }
.ms-cui-ct-ul, .ms-cui-cg-i, .ms-cui-cg, .ms-cui-tt-a
{border-color: #F4F3F9 !important; }
.ms-cui-cg-i {border-color: transparent !important; }
.ms-cui-cg {border-color: transparent !important; }

/* -----
Estilos en Top Navigation
----- */

/* Efecto de margen izquierdo */
.s4-tn .menu-horizontal {
margin-left: 200px;
}

/* Estilo de los links */
.s4-tn .menu-item-text {
color: #FCFCFC;
font-size: 11px;
border-right: #FCFCFC solid 1px;
padding-right: 25px;
font-size: 11px;
font-family: Arial, Helvetica, sans-serif;
}

/* Estilo del link seleccionado */
.s4-tn .selected {
font-weight: bold;
}
.s4-tn .selected .menu-item {
background: none !important;
border: none !important;
}

/* Estilo del link hover */
.s4-tn .menu-item:hover {
text-decoration: none !important;
}

/* -----
Footer
----- */

#intranet_footer {
background-color: #ACACAD;
margin-top: 15px;
font-family: Arial, Helvetica, sans-serif;
font-size: 9px;
color: white;

```

```

text-align: center;
padding: 4px;
clear: both;
border-top: white solid 40px;
}

```

En la siguiente imagen, puede observarse la forma que adquiere la Ribbon:



Imagen 3.- Ejemplo de ribbon con el diseño gráfico aplicado

Sobre la imagen de cabecera

La imagen de cabecera posee algunas características importantes:

- Ocupa todo el ancho fijo: 1100px.
- Debe ser de la altura 170px (si se cambia, hay que ajustar la CSS).
- Posee un sombreado en la parte superior (como parte de la imagen).
- Posee el logo a la izquierda, dejando algo de espacio en blanco para la botonera (solapas) de la ribbon.
- La zona de abajo es el fondo de los links de la Top Navigation.

La imagen de cabecera ejemplo es:



Imagen 4.- Ejemplo de imagen para la cabecera

La home page

En la home page, trabajamos sobre estos aspectos principales:

- Nombre de usuario.
- Efeméride.
- Botón de sub sitio.
- Contenido de la home.

Buscamos algo como lo siguiente:



Imagen 5.- Particularidades de la página de inicio.

NOMBRE DE USUARIO

El nombre de usuario se ve a la izquierda con el agregado del "Bienvenido" y "> > >". Para ello trabajamos con una técnica jQuery que nos permite insertar HTML:

```
$('#zz15_Menu span').
prepend('Bienvenido&nbsp;');
$('#zz15_Menu span').append(' > > >');
```

EFEMÉRIDE

A la derecha, arriba, agregamos:

- La fecha actual obtenida con JavaScript.
- La efeméride obtenida con Cliente Object Model.

Para obtener la fecha actual, usamos código JavaScript. Para obtener la Efeméride, consultamos los datos de una lista usando Cliente Object Model, tal como lo expliqué en un artículo previo, que pueden encontrar en: <http://surpoint.blogspot.com.ar/2011/10/introduccion-al-modelo-de-objetos-de.html>.

Para insertar la efeméride (ítem extraído de una lista), usamos:

```
ExecuteOrDelayUntilScriptLoaded(cargar_
efemeride, "sp.js");
$('.ms-cui-ribbonTopBars').append($('#efemeride'));
```

Dentro del código que lee la efeméride (cargar_efemeride), se cargó previamente un div de la siguiente forma:

```
document.getElementById("efemeride").innerHTML
= strHtml;
```

El div debe existir en Intranet_Home.htm:

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

BOTÓN DE SUBSITIO

Este es un simple botón que agregamos en nuestro HTML de la siguiente forma:

```
<table id="subsitio_contenedor">
<tr>
<td width="990px"></td>
<td width="100px">
<a id="subsitio" href="/sites/blablabla.
aspx">Subsitio</a>
</td>
</tr>
</table>
```

Resultado



Imagen 6.- Botón de acceso a un subsitio

EL CONTENIDO DE LA HOME

Siguiendo el enfoque de las efemérides, cargamos todo el contenido de la home. Lo primero que hacemos es crear una tabla para nuestro contenido, con el diseño que más nos guste. Este es un ejemplo:

```
<table id="super_contenedor">
<tr>
<td>
<table class="cumple-tabla">
<thead>
<tr>
<th>Para compartir</th>
</tr>
</thead>
<tbody>
<tr>
<td><div id="cumple"></div></td>
</tr>
</tbody>
</table>
</td>
<td><strong>Zona reservada para noticias</strong>: Lorem ipsum ad ... </td>
</tr>
<tr>
<td><strong>Zona reservada para anuncios</strong>: Lorem ipsum ad ... </td>
<td><strong>Zona reservada para noticias en las sedes</strong>: Lorem ipsum ad his scripta blandit
```

```

partiendo... </td>
</tr>
</table>

```

Ocultamos la zona de webparts de la derecha:

```

$('#ctl00_MSO_ContentDiv table table td:first').
width("100%");

```

Y luego insertamos con client object model los datos que consultamos desde listas de SharePoint, asociando las clases para aplicar los estilos que nos proporcionó el diseñador gráfico.

Obtenemos algo como lo siguiente:



Imagen 7.- Ejemplo de información obtenida con Client Object Model

LOS ESTILOS DE LA HOME

Habrán notado que algunas cosas son distintas en la home, como el ocultamiento de la ribbon o la ubicación del nombre de usuario. Para ello aplicamos estilos propios:

```

<style>
/* -----
Ajustes generales
----- */
/* Ocultar Inicio Rápido */
#s4-leftpanel { display: none; }
.s4-ca { margin-left: 0px; }
/* Ocultar solapas de Ribbon */
.ms-cui-tts { display: none; }
/* Y el borde de la top navigation */
#s4-topheader2 { border-bottom: 0px !important; }
/* Nombre de usuario a la izquierda */
#RibbonContainer-TabRowRight { float: left; }

```

“... buscar tiempos de respuesta más ágiles, mejor comunicación y sobretodo una mejor colaboración.”

```

/* Efeméride a la derecha */
#efemeride {
  color: #03935F;
  font-weight: bold;
  text-transform: uppercase;
  font-size: 11px;
  font-family: Arial, Helvetica, sans-serif;
  float: right;
  margin-top: -39px;
  margin-right: 10px;
}
/* Fecha actual a la derecha */
#fecha {
  font-size: 11px;
  font-family: Arial, Helvetica, sans-serif;
  color: #8F8F8F;
  font-weight: normal;
  text-transform: none;
}
/* Botón de llamada a un subsitio */
#subsitio {
  font-size: 18px;
  font-family: Arial, Helvetica, sans-serif;
  color: white;
  background-color: #F9B10F;
  border: solid 4px white;
  border-right: none;
  padding: 5px 10px;
  text-align: center;
}
#subsitio_contenedor {
  position: fixed;
  top: 91px;
}
/* -----
Contenedor de partes
----- */
/* Estilo de la Tabla que contiene cada una de las partes (contenido) */
#super_contenedor {
  width: 100%;
}
#super_contenedor td {
  padding: 10px;
  vertical-align: top;
}

```

```

/* -----
Para compartir (estilos para mostrar cumpleaños)
----- */

.cumples-tabla th {
background-color: #288BCE;
color: white;
font-family: Arial, Helvetica, sans-serif;
font-size: 13px;
font-weight: bold;
}
.cumples-tabla td {
background-color: #CBE9F6;
}
</style>

```

Conclusión

En este artículo explicamos un camino para incorporar diseño gráfico a SharePoint. Existen diferentes formas

de encarar esta tarea. En este caso buscamos una que no requiera demasiados conocimientos específicos de SharePoint, que no “desarme” funcionalidades como la ribbon, pero que a la vez permita personalizar el diseño haciendo que no se parezca a “SharePoint”. Hasta la próxima!

Bibliografía

Real World Branding with SharePoint 2010 Publishing Sites
 SharePoint Branding 101: Branding Master Pages
 SharePoint 2010 branding. A beginners guide
 10 Worst Mistakes in SharePoint Branding
 SharePoint Ribbon CSS Tips and Tricks
 SharePoint 2010 CSS Reference Chart

JUAN PABLO PUSSACQ LABORDE

SharePoint MVP

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

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

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

The infographic is a vertical stack of four colored bands, each representing a service area. On the left, there are four numbered cards (01-04) with detailed descriptions. On the right, there are larger cards with icons and brief descriptions for each category.

- 01 CORPORATIVO**
Logotipo, Manual de identidad de marca, Paleta de colores, Manual de identidad de marca, Libros, tarjetas, Menús de identidad de marca, Folletos.
- 02 DISEÑO WEB**
Maquetación HTML + CSS, Diseño de ecommerce, Diseño de CRM, CSS, Diseño de ecommerce, Diseño de CRM.
- 03 IMPRESO**
Diseño Editorial, Diseño de Packaging, Diseño de Urbanización, Diseño de Packaging, Diseño de Urbanización.
- 04 PUBLICITARIO**
Avisos publicitarios, Newsletters, Banners, Avisos publicitarios, Newsletters, Banners.

+ DISEÑO CORPORATIVO
Logotipo
Manual de identidad de marca
Folletería

DISEÑO WEB
Maquetación HTML + CSS
Diseño de ecommerce
Diseño de CRM

DISEÑO IMPRESO
Diseño Editorial
Diseño Packaging
Diseño de Urbanización

DISEÑO PUBLICITARIO
Avisos publicitarios
Newsletter
Banners

nitido
GRAPHIC & WEB DESIGN

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



57

1ª Conferencia Ibérica de SharePoint

Resumen

Hemos reunido a los mayores expertos de SharePoint de la Península Ibérica y Latinoamérica para contarte todas sus novedades y su vinculación con otros productos Microsoft. Tendrás acceso a contenido de calidad, oportunidades para intercambiar conocimiento y experiencias con los colegas del sector de negocio y la tecnología.

Requisitos Previos



Los Grupos de usuarios de SharePoint de la Península Ibérica (SUGES, SUG.CAT, MadPoint y LevaPoint) organizan la 1ª Conferencia Ibérica de SharePoint que tendrá lugar el 10 de octubre de este año en Madrid. Será el primer evento dedicado exclusivamente a SharePoint desde distintas perspectivas en el que participen los principales expertos nacionales e internacionales en la plataforma (MVPs y líderes de la comunidad como Gustavo Vélez, Alberto Díaz, Juan Carlos González, Edin Kapic, Mario Cortes, Fabián Imaz, Joel Oleson entre otros), y en el que los asistentes tendrán la oportunidad no solo de formarse, sino también de conocer distintas soluciones construidas sobre la plataforma.

Conferencistas

Como en toda gran conferencia, estarán presentes los mayores especialistas de SharePoint y tecnologías afines de España y Latinoamérica, así como una selección de destacados referentes internacionales. Entre estos, podemos destacar a:

- Gustavo Vélez.
- Alberto Díaz.
- Fabián Imaz.
- Mario Cortés.

- Ricardo Muñoz.
- David Martos.
- Christian Buckley.
- Juan Carlos González.
- Joel Oleson.
- Daniel Seara.
- Andre Lage.
- Luis Ruiz Pavón.
- Jethro Seghers.
- Edin Kapic.

Gracias a ellos, y a su desinteresada participación, tendremos tres tracks en paralelo, con la temática de IT, Desarrollo y Negocio.

Agenda

Por el momento no se ha conformado oficialmente la agenda, pero si os podemos contar que tenemos preparadas más de 40 sesiones, con temas tan interesantes como:

- ALM en SharePoint.
- Aplicaciones, un nuevo modelo para personalizaciones en SharePoint 2013.
- Como crear una solución ECM (cero papel) en SharePoint 2013.
- Lync comunicaciones y conectividad sin límites.
- Office 365: ¿Debo dar el salto a la nube?
- 10 Best Productivity features in SP2013 you need to know about.
- Automating SharePoint 2013 deployment: tips, tricks and scripts.
- SharePoint 2013: Mobile strategy and design.
- y muchos más que podéis consultar en la web de la conferencia.

La conferencia se ha diseñado y concebido como un punto de encuentro natural para los mayores expertos de SharePoint de la Península Ibérica y Latinoamérica, grandes referentes internacionales en la plataforma,

empresas nacionales e internacionales cuya actividad principal gira en torno al producto estrella de Microsoft y por supuesto todos los profesionales vinculados de forma directa en su día a día con la plataforma: decisores de negocio, responsables de departamentos de desarrollo e IT, arquitectos de soluciones, desarrolladores, diseñadores y profesionales de IT Pro. Todo aquel que trabaje con SharePoint tiene cabida en esta conferencia que supone una gran oportunidad para conocer no sólo a los ponentes de nivel que participarán en la misma, sino también a las empresas más importantes que trabajan en su día a día con SharePoint.

WorkShops

Complementando la conferencia, hemos preparado dos WorkShops especializados para el área de IT y Desarrollo. Estos WorkShops, basados en el training de Ignite de SharePoint 2013 creado por Microsoft, serán impartidos por MVPs de SharePoint Server. Una gran oportunidad para recibir formación de calidad de la mano de los mejores expertos en SharePoint.

Patrocinio

Para que la conferencia sea un éxito, necesitamos patrocinadores, empresas que aporten su granito de arena y colaboren con la primera de las muchas conferencias de SharePoint que seguirán en España. Si estás interesado, no dudes en contactarnos y si piensas que tu empresa puede ayudar, no dudes en trasladarle la información a la persona que se encarga del marketing porque estar en la conferencia es un MUST y se podrán abrir muchas oportunidades de negocio en la zona de patrocinadores. Tenemos varios niveles de patrocinio que se adaptan a todo tipo de empresas colaboradoras. Hemos dispuesto dos formas de contacto para tratar cuestiones relativas a patrocinio de la conferencia:

- Cuenta de Twitter: @IberianSPConf
- Correo electrónico: suges@hotmail.es

Pre-Registro

Sin duda, se trata del primer evento técnico y de negocio de esta naturaleza que se realice en España y en el que se traten todas las novedades sobre SharePoint, así como otras plataformas y tecnologías relacionadas

¿Qué esperas para pre-registrarte? ¿Sabías que las plazas son limitadas y que por acceder al pre-registro tendrás un descuento preferente en el precio del evento?

“... se ha diseñado y concebido como un punto de encuentro natural para los mayores expertos de SharePoint de la Península Ibérica y Latinoamérica...”

Reserva ya tu plaza



Información clave sobre la conferencia

- Fecha: 10 de octubre de 2013.
- Lugar: Madrid, ubicación pendiente de confirmar.
- Sitio Web de la conferencia: www.iberiansharepointconference.com
- Enlace de registro: <http://iberianspc2013-iscpage.eventbrite.com/>
- Twitter de la conferencia: @IberianSPConf

Uso de Frameworks JavaScript en desarrollos SharePoint - Parte I

Resumen

En este artículo vamos a introducir uno de los frameworks JavaScript Kendo UI que hay en el mercado. Veremos la sencillez del mismo a nivel de desarrollo, para de esta forma quitarnos el miedo que tenemos los desarrolladores de SharePoint con respecto al uso JavaScript.

Como desarrolladores de SharePoint, tenemos la ventaja del conocimiento que tenemos sobre la plataforma de manera que si le sumamos conocimientos de JavaScript obtendremos unos resultados espectaculares. Adicionalmente, el uso de este tipo de frameworks nos permite hablar crear desarrollos HTML5 de tipo “responsive design” ofreciendo soluciones que sirvan para todo tipo de plataformas.

Requisitos Previos

La llegada de la nueva versión de SharePoint obliga en cierto modo a que los desarrolladores de anteriores versiones tengan que adaptarse a los cambios del desarrollo en la Web y a familiarizarse con un lenguaje de programación que no levanta las simpatías de la comunidad .NET.

Este lenguaje como se deduce es JavaScript. En este artículo se va a introducir el framework JavaScript Kendo UI de Telerik y en un artículo posterior ExtJS de Sencha para facilitar nuestro trabajo en los desarrollos a realizar y sobre todo otorgarles una mayor calidad.

¿Qué es un Framework?

Según la Wikipedia es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, que puede servir de base para la organización y desarrollo de software.

Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio, y provee una estructura y una especial metodología de trabajo, la cual extiende o utiliza las aplicaciones del dominio.

¿Por qué utilizar un Framework JavaScript?

Si analizamos el uso que anteriormente hacíamos de JavaScript en nuestros desarrollos casi siempre se basaban en utilizar alguna librería bien para dotarnos de un efecto que nos gustaba (ejemplo del típico banner slider que está en muchísimas web), bien hacer uso del patrón MVVM haciendo uso de la librería KnockoutJS como introdujo Edin Kopic en números anteriores de la revista. O bien hacer uso de JQuery para añadir una funcionalidad específica (validar campos, ocultar los mismos, refrescar datos, etc...).

El utilizar JavaScript sin entender realmente lo que estamos desarrollando hace que nuestros desarrollos tengan falta de calidad y bugs relativos a este uso. Es decir vamos concatenando funcionalidades JavaScript a nuestros desarrollos sin saber bien que acciones estamos realizando dentro de nuestro desarrollo. Esto hace que mezclemos diferentes versiones de las librerías que estamos utilizando y da lugar a que se produzcan muchos fallos en nuestros desarrollos. Estos errores son debidos al desconocimiento sobre JavaScript como lenguaje de programación. Y lo que dentro de un desarrollo en .NET ni siquiera nos lo planteamos, en JavaScript dado que es un lenguaje muy vivo y permisivo (en tiempo de desarrollo) no dudamos en realizarlo.

El uso de un Framework JavaScript aporta como valor añadido que nos olvidemos de estos problemas a la vez que introduce una serie de patrones para otorgarle más calidad a nuestro software. Eso sí, esto no implica que no tengamos que conocer JavaScript como lenguaje de programación. Nos proporciona una abstracción de código común que provee funcionalidades genéricas que pueden ser utilizadas para desarrollar aplicaciones de manera rápida, fácil, modular y sencilla, ahorrando tiempo y esfuerzo. Estas funcionalidades se traducen en la forma de diversos tipos de Grid, Combos, Formularios, efectos en las imágenes y distintos gráficos de barras.

Aunque este tipo de componentes también los tenemos disponibles en desarrollos en .NET, la gran ventaja del uso de estos Frameworks es que están basadas en los estándares web y que son compatibles con todos los dispositivos

móviles/tabletas (IOS, Android, Windows Phone, Surface) y todos los navegadores modernos (IE8 o superior, Chrome, Mozilla). Además aportan unos patrones para realizar nuestros desarrollos lo que repercute directamente en la calidad del software.

Otra gran ventaja es que generan una interfaz de usuario muy intuitiva y con gran variedad de gráficos de barras, gauges, y diversos indicadores muy interesantes para incluir en nuestros desarrollos.

Manos a la obra

A continuación, nos centraremos en introducir estos frameworks dentro de nuestros desarrollos utilizándolos en una aplicación de SharePoint. A modo de ejemplo, crearemos un típico mantenimiento de datos maestros que permite mostrar aspectos muy útiles de estas herramientas. El resultado sería el siguiente mockup:

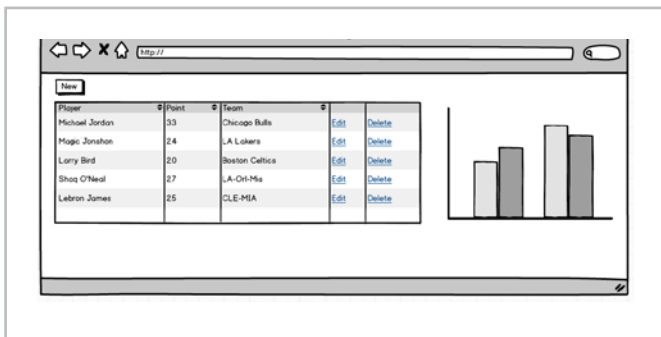


Imagen 1.- Mockup del desarrollo a implementar

En concreto, la aplicación permite mostrar en un Grid el contenido de una lista de SharePoint que contiene los máximos anotadores de la NBA, y con la posibilidad de añadir, editar y eliminar cualquiera de estos registros y a la vez visualizamos una gráfica en la que se realiza una comparación de los mismos.

Utilizando el Framework Kendo UI Telerik

Kendo es un framework JavaScript creado por Telerik (cuyos componentes dentro del mundo .NET son muy populares) con menos de un año de vida. Su funcionamiento consiste extender de forma sencilla las capacidades que están disponibles en jQuery.

Desde Kendo se puede bien usar el modelo de objetos de cliente (CSOM) JavaScript de SharePoint o bien consumir directamente la API REST. En este ejemplo vamos a utilizar la API REST por dos motivos:

- En primer lugar, poner en práctica el uso de estándares web en aplicaciones.
- En segundo lugar, demostrar que la API REST de SharePoint tiene la misma potencia que el modelo de objetos de Cliente con la gran ventaja de qué se puede usar desde plataformas Microsoft (W8, WP8) y no Microsoft (IOS, Android).

Empezando el desarrollo

Este desarrollo se va a implementar por medio de una APP de tipo SharePoint-Hosted, (aunque se podría portar sin ningún problema en WebParts clásicos):

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

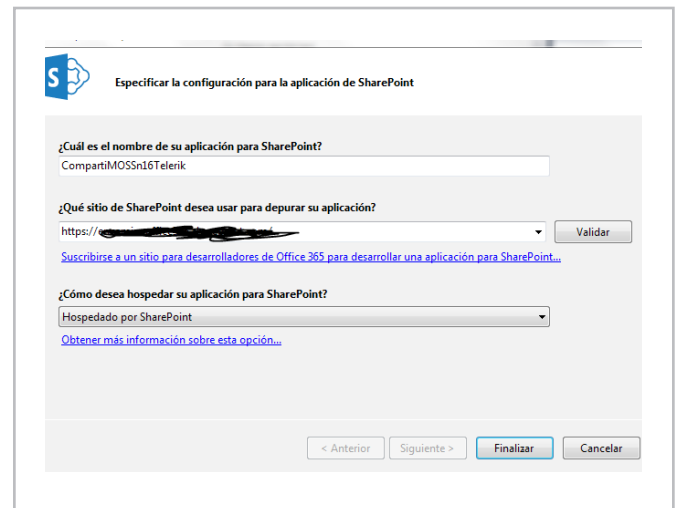


Imagen 2.- Asistente para crear la aplicación SharePoint-Hosted.

- En primer lugar, agregamos al proyecto una lista llamada Players en la que añadiremos los datos que queremos mostrar en el grid. Esta lista contendrá los siguientes campos:

Nombre para mostrar de la columna	Tipo	Necesario
Título	Una línea de texto	<input checked="" type="checkbox"/>
Scores	Número	<input checked="" type="checkbox"/>
Team	Una línea de texto	<input checked="" type="checkbox"/>
<input type="text"/> Escriba un nombre de columna nuevo o existente		<input type="checkbox"/>

Imagen 3.- Definición de la Lista Players

- Dentro de la lista le añadiremos unos datos por defecto para que una vez que carguemos el grid podamos ver algunos resultados. Para añadir los datos basta con editar el archivo "elements.xml" que se encuentra ubicado dentro de la Instancia de la lista y añadir los datos.
- La siguiente acción a realizar consiste en importar las librerías JavaScript y los CSS de Kendo (vamos a aprovechar el estilo que viene de serie y dejamos para los diseñadores la edición de estos CSS) por lo que añadimos los siguientes ficheros al proyecto: "kendo.all.min.js", "kendo.common.min.css" y "kendo.default.min.css" (estos ficheros se encuentran disponibles dentro de la descarga del propio framework).
- En la página default.aspx añadimos referencias a las librerías JavaScript y archivos .CSS de la siguiente forma:

```
<link rel="Stylesheet" type="text/css"
href=" ../Content/App.css" />
```



```
<link
href="../../Content/kendo.common.min.css"
rel="stylesheet" />
<link
href="../../Content/kendo.default.min.css"
rel="stylesheet" />
<script type="text/javascript"
src="../../Scripts/kendo.all.min.js"></script>
<script type="text/javascript"
src="../../Scripts/App.js"></script>
```

- Una vez ya tenemos las referencias a las librerías que vamos a utilizar, comenzamos con el desarrollo. Para ello, todo el código que vamos a necesitar lo incluiremos dentro de la página default.aspx y dentro del archivo App.js. En primer lugar haremos uso de un componente DataSource, qué es una abstracción para el uso de datos locales (arrays de objetos) o remotos (XML, JSON, JSONP). Este componente es totalmente compatible con operaciones de tipo CRUD (Create, Read, Update, Destroy) y proporciona apoyo tanto en el lado cliente como del lado del servidor para la ordenación, la paginación, el filtrado, agrupación y agregados.
- Abriremos el fichero App.js y definimos este componente de forma que apunte a la lista de SharePoint creada. En la definición, configuraremos el modelo que vamos a usar y personalizaremos algunas características. A continuación, se muestra la definición, configuración y uso del componente:

```
dataSource = new kendo.data.DataSource({
    type: "odata",
    transport: {
        read: {
            url: listUrl,
            type: "GET",
            dataType: "json",
            contentType:
                "application/json;odata=verbose",
            headers: {
                "accept":
                "application/json;odata=verbose"
            }
        },
        pageSize: 5,
        schema: {
            data: "d.results",
            model: {
                id: "ID",
                fields: {
                    ID: { editable: false,
                        nullable: false },
                    Title: { validation: {
                        required: true } },
                    Scores: { type: "number" },
                    Team: { type: "string" }
                }
            }
        }
    }
});
```

- Como se puede apreciar, únicamente hemos realizado la configuración "read" para el origen de datos. Conforme definamos el resto de operaciones CRUD, iremos añadiendo más configuración. Del reto de configuraciones, lo que resulta más interesante es que por un lado el número de elementos que estamos paginando en nuestro caso es 5 y por otro lado estamos realizando la definición de un esquema con un modelo de datos, en los que podemos ir añadiendo diversas características interesantes como son los campos requeridos, si permitimos editarlos, etc.

- A continuación vamos a crearnos un Grid, para ello en primer lugar nos vamos a la página "Default.aspx" y añadimos el div donde lo posicionaremos de la siguiente forma:

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

- Dentro del fichero App.js, a continuación de la definición del DataSource, introduciremos el siguiente código:

```
$("#grid").kendoGrid({
    dataSource: dataSource,
    pageable: true,
    height: 430,
    columns: [
        { field: "ID", title: "ID" },
        { field: "Title", title:
            "Player" },
        { field: "Scores", title:
            "Scores" },
        { field: "Team", title: "Team"
        }
    ]
});
```

A este tipo de grid se les puede otorgar mucho más complejidad, como agrupar por equipos, mostrar totales, etc, y solo añadiendo unas pocas de opciones, para ello consultar la propia documentación de Kendo y podéis obtener todas las posibilidades que nos brinda la herramienta. Si ejecutamos y desplegamos la APP, obtendremos el siguiente resultado:

ID	Player	Scores	Team
1	Nathan Jordan	33	Chicago Bulls
2	Shaq O'Neal	38	LA Orlando
3	LeBron James	27	Miami De
4	Kobe Bryant	33	LA Lakers

Imagen 4.- Resultado obtenido al desplegar la APP.

- Una vez tenemos este primer resultado, el siguiente paso es introducir a ambos componentes la posibilidad de

poder añadir, editar y eliminar elementos. En primer lugar iremos al fichero App.js y reemplazaremos la definición del DataSource, por la siguiente definición:

```

dataSource = new
kendo.data.DataSource({
  type: "odata",
  transport: {
    read: {
      url: listUrl,
      type: "GET",
      dataType: "json",
      contentType:
"application/json;odata=verbose",
      headers: {
        "accept":
"application/json;odata=verbose"
      }
    },
    create: {
      url: listUrl,
      type: "POST",
      dataType: "json",
      contentType:
"application/json;odata=verbose",
      headers: {
        "accept":
"application/json;odata=verbose",
        "X-
RequestDigest":
$("#__REQUESTDIGEST").val(),
      }
    },
    update: {
      url: function
(data) {
        return
listUrl + "(" + data.ID + ")";
      },
      beforeSend:
function (jqXHR, options) {
        var data =
JSON.parse(options.data);

jqXHR.setRequestHeader("If-Match",
data.__metadata.etag);

      },
      type: "POST",
      dataType: "json",
      contentType:
"application/json;odata=verbose",
      headers: {
        "accept":
"application/json;odata=verbose",
        "X-
RequestDigest":
$("#__REQUESTDIGEST").val(),
        "X-HTTP-Method":
"MERGE"
      }
    },
    destroy: {
      url: function (data)
{
        return listUrl +
("(" + data.ID + ")";
      },
      type: "DELETE",
      dataType: "json",

```

```

      contentType:
"application/json;odata=verbose",
      headers: {
        "accept":
"application/json;odata=verbose",
        "X-
RequestDigest":
$("#__REQUESTDIGEST").val(),
        "X-HTTP-Method":
"MERGE",
        "If-Match": "*"
      }
    },
    parameterMap: function (data,
type) {
      if (type === "update" &&
data["__deferred"]){
        delete data["__deferred"];
      }
      return kendo.stringify(data);
    },
    pageSize: 100,
    schema: {
      data: "d.results",
      model: {
        Id: "ID",
        fields: {
          Title: { type:
"string" },
          Scores: { type:
"number" },
          Team: { type:
"string" }
        }
      }
    }
  });

```

De esta forma, estamos consiguiendo añadirle a nuestro origen de datos que permita realizar operaciones CRUD. Para ellos hemos añadido la opción “create”, “update”, “destroy” y “parameterMap”. Esta última sirve para indicar que parámetros queremos enviar o no a SharePoint utilizando la API REST. Debido a que si enviamos los datos tal y como lo genera Kendo da un error indicando que hay un valor “__deferred” que no es admitido. Por este motivo lo que hacemos antes de enviar los datos es eliminar este valor para evitar cualquier fallo al enviar los datos.

- Una vez tenemos preparado el DataSource, con añadir una serie de opciones en el grid dispondremos de un formulario para dar de alta los jugadores, la posibilidad de editar y eliminar los mismos datos. Para ello realizamos las siguientes modificaciones en el componente Grid que habíamos implementado anteriormente:

```
$("#grid").kendoGrid({
  dataSource: dataSource,
  pageable: {
    refresh: true,
    pageSize: true
  },
  height: 430,
  toolbar: ["create"],
  columns: [
    { field: "Title",
    { field: "Scores",
    { field: "Team",
    { command: ["edit", "destroy"],
title: "&nbsp;", width: "45px" }],
    editable: "popup"

  });
```

Como se deduce, hemos modificado las columnas del grid para añadir una nueva columna command en la que podremos editar y eliminar el registro automáticamente. Adicionalmente, en la parte superior hemos creado un botón para añadir nuevos registros. Este botón puede tener dos funcionalidades bien añadir el registro directamente en el grid, o bien mostrar una ventana modal en la que en la que introducir el nuevo registros. En este caso se ha optado por la segunda opción. Una de las grandes ventajas que tiene es que cuando realizamos cualquier opción de edición de los registros estos formularios son inteligentes, en el sentido que ya incorporan la propia validación de los campos obligatorios y del formato de los mismos.

El resultado se muestra a continuación:

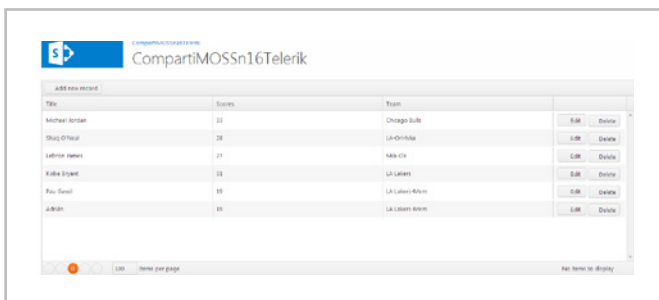


Imagen 5.- Grid resultante con operaciones CRUD.

Cuando pulsamos sobre el botón de "Add new record" vemos como se muestra la pantalla modal:

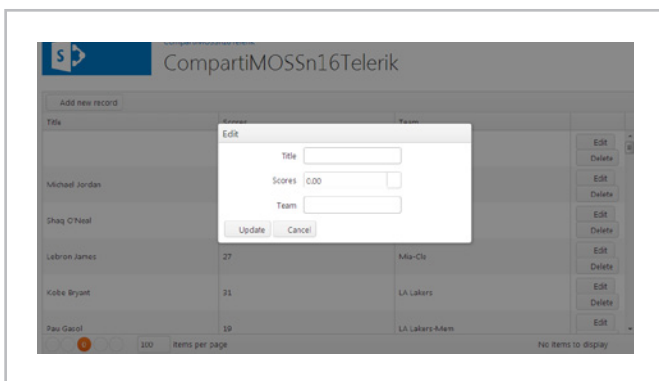


Imagen 6.- Ventana modal para añadir registros.

Una vez realizado las operaciones básicas vamos a mostrar como con el mismo datasource se puede visualizar una gráfica relativamente simple y muy sencilla de implementar. Para ello tenemos que hacer uso del Widget "Bar Chart". Antes de ponernos con el JavaScript retocamos un poco el HTML para ello añadimos las siguientes líneas de código dentro de la página default.aspx:

```
<div class="chart-wrapper" style="margin:
auto;">
  <div id="chart">  </div>
</div>
```

A continuación, modificamos el archivo App.js de la siguiente forma:

```
$("#chart").kendoChart({
  dataSource: dataSource,
  title: {
    text: "Promedio Puntos
Jugadores"
  },
  legend: {
    position: "top"
  },
  seriesDefaults: {
    type: "column"
  },
  series:
  [{
    field: "Scores",
    name: "Scores"

  }],
  categoryAxis: {
    field: "Title",
    labels: {
      rotation: -90
    },
    majorGridLines: {
      visible: false
    }
  },
  valueAxis: {
    labels: {
      format: "N0"
    },
    majorUnit: 10000,
    line: {
      visible: false
    }
  },
  tooltip: {
    visible: true,
    format: "N0"
  }
});
```

Una vez añadido estas modificaciones ya tenemos el gráfico listo en nuestra aplicación como mostramos en la siguiente pantalla:

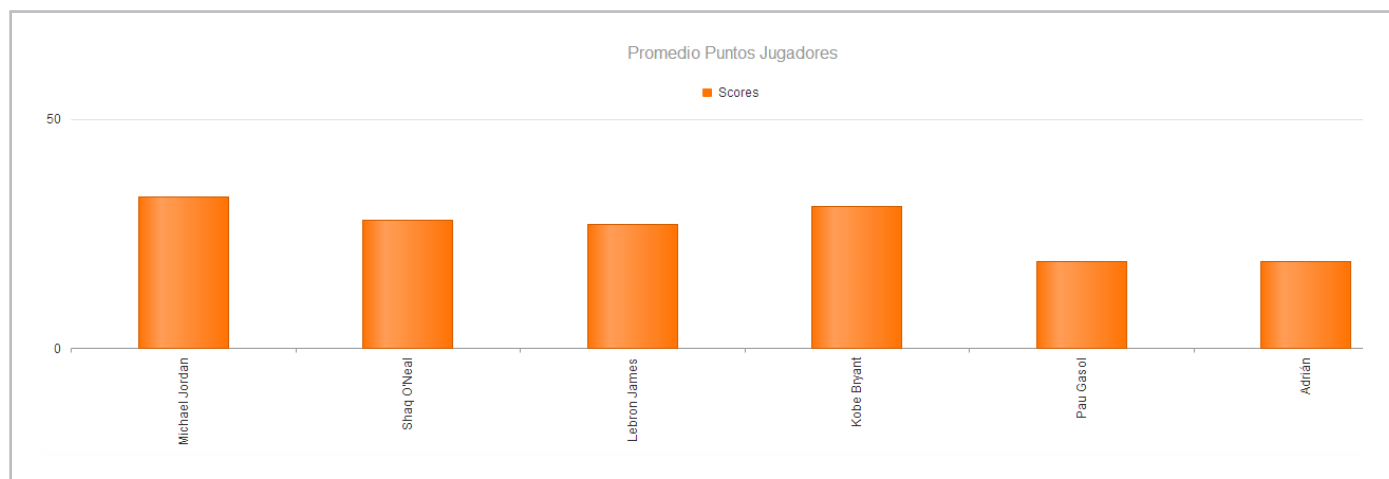


Imagen 7.- Gráfico Comparación de Puntos.

Conclusiones

Todo esto es una pequeña parte de la que contiene este Framework que es muy completo y merece la pena cuanto menos probar su utilidad. También tiene incluido la funcionalidad KnockoutJS para incluir patrones Model View View Model, incluye muchas más utilidades sencillas de utilizar y que da un salto de calidad a nuestros desarrollos y con menor esfuerzo.

Existen multitud de frameworks JavaScript, cada desarrollador puede tener preferencia por uno u por otro. Mi opinión personal es que no podría indicar cuál es el mejor framework JavaScript que hay en el mercado, cada desarrollador tiene su preferido. Pero al final lo que tenemos que tener claro es que hay que realizar el software con la mayor calidad posible y para ello el uso de cualquiera de estos frameworks es prácticamente una obligación.

Referencias:

- <http://demos.kendoui.com/>
- <http://stackoverflow.com/questions/16439721/specifying-the-fields-to-send-on-an-update-command-using-the-kendo-ui-framework>
- SharePoint 2013: Perform basic data access operations by using REST in apps
- SharePoint 2013 de principio a fin. Editorial Krasis Press, Febrero de 2013.

ADRIÁN DIAZ CERVERA

Sharepoint Architect at Encamina

MCPD SharePoint 2010 MAP y MCC 2012

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

adiaz@encamina.com

@AdrianDiaz81

“... Introducir el framework JavaScript Kendo UI de Telerik...”





Alberto Díaz

Alberto Díaz es Responsable de Proyectos de desarrollo en SharePoint, .NET, aplicaciones en movilidad y todo lo relacionado con tecnologías Microsoft en General de Software. Su labor en GSC es ofrecer soluciones empresariales a sus clientes y ofrecer nuevas tecnologías que ayuden a las empresas a realizar su trabajo más eficientemente, enfocado en productos Microsoft. Para la comunidad, ha fundado TenerifeDev (www.tenerifedev.com) con otros colaboradores, un grupo de usuarios de .NET en Tenerife, y coordinador de SUGES (Grupo de Usuarios de SharePoint de España, www.suges.es) y colaborador con otras comunidades de usuarios. Microsoft MVP de SharePoint Server desde el año 2011 y asiduo conferenciante en webcast y conferencias de tecnología de habla hispana.

Email: adiazcan@hotmail.com

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

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



Fabian Imaz

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

Sitio Web: <http://www.siderys&bsn.com>

Email: fabiani@siderys.com.uy

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

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



Gustavo Velez

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

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

Email: gustavo@gavd.net

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



Juan Carlos González Martín

Juan Carlos González, es Arquitecto de Soluciones en el Centro de Innovación en Integración (CIIN) de Cantabria, uno de los Microsoft Innovation Centers de España. Ingeniero de Telecomunicaciones por la Universidad de Valladolid, cuenta con casi 10 años de experiencia en tecnologías y plataformas de Microsoft diversas (SQL Server, Visual Studio, .NET Framework, etc.), aunque su trabajo diario gira en torno a la plataforma SharePoint. Juan Carlos es MVP de SharePoint Server desde el año 2008, coordinador del grupo de usuarios .NET de Cantabria (Nuberos.Net, www.nuberos.es) y co-fundador del Grupo de Usuarios de SharePoint de España (SUGES, www.suges.es) y del Grupo de Usuarios de Cloud Computing de España (CLOUDES). Desde el año 2011 participa junto con Gustavo Vélez, Fabián Imaz y Alberto Diaz en la dirección de CompartiMOSS.

Sitio Web: www.ciin.es

Email: jgonzalez@gruposodercan.es

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

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



¿Desea colaborar con CompartiMOSS?



La subsistencia del magazine depende de los aportes en contenido de todos. Por ser una revista dedicada a información sobre SharePoint en español, todo el contenido deberá ser directamente relacionado con Microsoft SharePoint y escrito en castellano. No hay limitaciones sobre el tipo de artículo o contenido, lo mismo que sobre el tipo de versión.

Si desea publicar algo, por favor, utilice uno de los siguientes formatos:

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

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

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

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

revista@compartimoss.com
adiazcan@hotmail.com
fabiani@siderys.com.uy

jgonzalez@gruposodercan.es
gustavo@gavd.net

| COMPARTIMOSS |