



REVISTA ESPECIALIZADA EN TECNOLOGÍAS **SHAREPOINT**



Entrevista An-
dré Lage

SharePoint
y Azure -
Operational
Insights

Aplicaciones
empresariales
en Office 365

Desarrollo de
Aplicaciones
con Office 365
API



Comparti MOSS

Staff

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

DIRECCIÓN GENERAL

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

DISEÑO Y DIAGRAMACIÓN

- Santiago Porras Rodríguez

Contacte con nosotros

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

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

REDES SOCIALES
Facebook:
<http://www.facebook.com/group.php?gid=128911147140492>
LinkedIn:
<http://www.linkedin.com/groups/CompartiMOSS-3776291>
Twitter:
@CompartiMOSScom

Contenido

03

Editorial

07

Desarrollo de Aplicaciones con Office 365 API

14

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

23

Entrevista André Lage

31

Generando máquinas como churros: Azure DevOps y Powershell Desired State Configuration (DSC)

37

Por un SharePoint Sostenible – Controla tus listas y bibliotecas, Conjunto de documentos

45

5 razones por las que es importante clasificar la información

04

Aplicaciones High Trust - Parte III

11

Trabajando programáticamente con tipos de contenido en C#

18

Integración Continua para Apps Office 365 con Visual Studio Online y Azure

25

SharePoint y Azure - Operational Insights

35

Aplicaciones empresariales en Office 365

42

Machine Learning: Usos y Aplicaciones

47

La importancia de diseñar y aplicar un buen uso de Gobernanza en nuestro entorno SharePoint 2013 (3^a Parte).



03

Editorial

Llegando al momento en el que SharePoint 2016 On-Premises comienza a estar disponible, podemos sacar algunas conclusiones sobre el futuro de SharePoint. Y, para ser honrados, aunque Microsoft nos ha dado algo de esperanzas de que la versión On-Premises seguirá existiendo por algún tiempo más, lo que hemos podido ver de la nueva versión no es precisamente para estar saltando de la alegría.

Desde la conferencia Ignite en marzo se conocen las nuevas especificaciones de SharePoint 2016. Si no lo ha hecho anteriormente, puede ver un resumen de lo que hay nuevo o modificado en el artículo de Bill Baer (<http://blogs.technet.com/b/wbaer/archive/2015/05/12/what-s-new-in-sharepoint-server-2016-installation-and-deployment.aspx>). Como podrá observar, los cambios entre SharePoint 2013 y 2016 son mínimos, y principalmente enfocados a infraestructura.

Esto nos presenta con un problema: nuestros clientes no tendrán ninguna motivación para moverse a la nueva versión. De igual forma, si alguna compañía que tenga planes de implementar un nuevo sistema de SharePoint On-Premises (si es que queda alguna en el mundo) y que ya disponga de licencias de 2010 o 2013, no tendrá ninguna razón para crear el sistema en 2016 teniendo que pagar por nuevas licencias.

Mejor dicho, y perdonen si hoy estamos siendo pesimistas, aunque Microsoft salvó a SharePoint de una muerte inminente hace un par de meses, el paciente sigue con pronóstico reservado y es mejor ir pensando en donde va a ser enterrado. Algo así como Silverlight, y muchas otras tecnologías brillantes que Microsoft ha dejado que se mueran solas, de pura vejez y falta de mantenimiento.

Así que, si no se habían dado cuenta, la única opción viable que nos queda es SharePoint Online en Office 365. Pero no esperen muchos cambios (o cambios radicales) en el futuro. SharePoint Online está siguiendo los pasos de Office: es un producto maduro, que permanecerá sin cambios radicales (e interesantes) por muchos años. Los tiempos de la aventura llamada SharePoint han desaparecido.

Esperamos que se diviertan tanto, como de costumbre, leyendo a CompartiMOSS, como nosotros nos divertimos haciéndola (aunque ahora nos divertimos mucho menos programando a SharePoint).

EL EQUIPO EDITORIAL DE COMPARTIMOSS

i

04

Las aplicaciones High-Trust (II)

En el último artículo hablé sobre las aplicaciones high-trust o de alta confianza (también conocidas como S2S, server-to-server) en los entornos de SharePoint 2013. Por defecto, estas aplicaciones usan la autenticación Windows. Veremos lo que hay que hacer para utilizar otro proveedor de autenticación para la app.

La identidad en una high-trust app

Tal como vimos en el artículo anterior, en las high-trust apps es la responsabilidad de la propia app establecer el token del usuario. Es decir, la app le dice explícitamente a SharePoint que usuario está accediendo a ella. Por defecto, la implementación del TokenHelper de las apps de SharePoint en Visual Studio sólo funciona para los usuarios de Windows ya que usa la clase WindowsIdentity internamente. En la mayor parte de los despliegues de una high-trust app, todos los usuarios dispondrán de su usuario de Windows así que no nos preocuparemos mucho. Sin embargo, cuando no usamos la autenticación Windows es cuando vamos a tener problemas.

Desde SharePoint 2010 existe la posibilidad de utilizar un proveedor de identidad que soporte Claims (o tokens SAML) para la autenticación en las aplicaciones web de SharePoint. De hecho, SharePoint 2013 las usa por defecto. Es relativamente fácil conectar SharePoint con un proveedor de identidad (IdP) compatible como puede ser Active Directory Federation Services (ADFS) o el gratuito pero excelente Identity Server de la empresa Thinktecture1.

El ejercicio de conectar SharePoint a un proveedor de identidad alternativo pasa por establecer el proveedor de autenticación como proveedor de identidad de confianza. Además, el proveedor de autenticación tiene que registrar SharePoint como “relying party” o consumidor de autenticación de fiar. En uno de los anteriores números de CompartiMOSS, Diego Gatti explicó como conectar SharePoint con ADFS2.

Imaginad que disponemos de un SharePoint habilitado para reconocer los usuarios “federados” identificados por un token SAML expedido por un proveedor de identidad ADFS que está disponible en la URL <https://idp.migranja.com>. Ahora queremos que la app que está alojada en miapp.migranja.com se autentique mediante identidad federada contra SharePoint.

Si hacemos una high-trust app normal como la que vimos

en el artículo anterior y la lanzamos desde SharePoint, identificados como un usuario federado, la app pedirá credenciales de Windows al ser redireccionada. Como acabo de explicar, el código estándar utiliza WindowsIdentity y la app en el web.config tiene puesto que se use el módulo de autenticación de Windows.

En las high-trust apps es la responsabilidad de la propia app establecer el token del usuario

Arreglando la autenticación de la app

¿Qué podemos hacer para arreglar la autenticación? El primer paso será cambiar la autenticación de la app de Windows a autenticación federada por SAML. Es decir, hay que sustituir en el apartado <system.web> del web.config de la app la línea de autenticación de Windows y poner esta:

```
<authentication mode=>None</>
```

Con esto le decimos a la app que no use ninguna autenticación. Pero entonces, ¿cómo va a saber la app que usuario está autenticado? Tenemos que añadir dos paquetes de NuGet que habilitan la autenticación federada e instalan las librerías necesarias:

- Microsoft Token Validation Extension for .NET 4.5.
- Microsoft ASP.NET Identity Core.

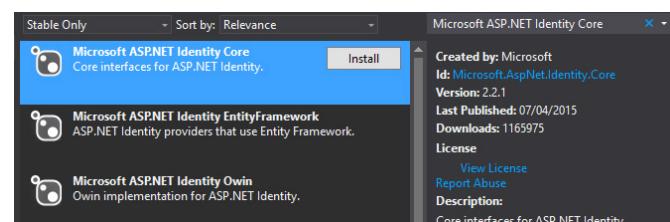


Imagen 1.- El paquete Microsoft.AspNet.Identity.Core.

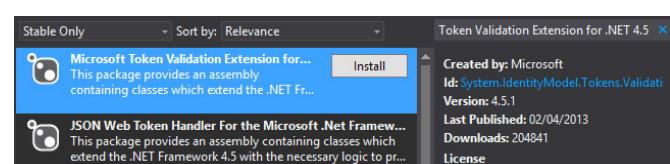


Imagen 2.- El paquete System.IdentityModel.Tokens.TokenValidationExtension.

Con estas dos librerías instaladas hay que cambiar otra vez el web.config para habilitar la autenticación federada que se instala como un módulo HTTP.

Al principio de todo del fichero, en la sección <configSections>:

```
<configSections>
<section name="system.identityModel"
    type="System.IdentityModel.Configuration.
SystemIdentityModelSection, System.
IdentityModel, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=B77A5C561934E089" />
<section name="system.identityModel.services"
    type="System.IdentityModel.Services.Configuration.
SystemIdentityModelServicesSection, System.
IdentityModel.Services, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=B77A5C561934E089" />
</configSections>
```

Justo después del elemento </system.web> hay que poner este fragmento que habilita los módulos de autenticación:

```
<system.webServer>
<modules>
    <add name="WSFederationAuthenticationModule"
    type="System.IdentityModel.Services.
WSFederationAuthenticationModule, System.
IdentityModel.Services, Version=4.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
    preCondition="managedHandler" />
    <add name="SessionAuthenticationModule" type="System.
IdentityModel.Services.SessionAuthenticationModule,
System.IdentityModel.Services, Version=4.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
    preCondition="managedHandler" />
</modules>
</system.webServer>
```

La parte más importante viene ahora. Agregamos una nueva sección en el web.config con los detalles del proveedor de identidad federado:

```
<system.identityModel>
<identityConfiguration>
    <audienceUris>
        <add value="uri:miapp.migranja.com" />
    </audienceUris>
    <securityTokenHandlers>
        <add type="System.IdentityModel.Services.Tokens.
MachineKeySessionSecurityTokenHandler, System.
IdentityModel.Services, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089" />
        <remove type="System.IdentityModel.
Tokens.SessionSecurityTokenHandler, System.
IdentityModel, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089" />
    </securityTokenHandlers>
    <certificateValidation certificateValidationMode="None" />
    <issuerNameRegistry type="System.IdentityModel.Tokens.
ValidatingIssuerNameRegistry, System.IdentityModel.Tokens.
ValidatingIssuerNameRegistry">
        <authority name="idp.migranja.com">
            <keys>
                <add thumbprint="ABCDEF123456789..." />
            </keys>
            <validIssuers>
                <add name="http://idp.migranja.com/adfs/services/trust" />
            </validIssuers>
        </authority>
    </issuerNameRegistry>
</identityConfiguration>
</system.identityModel>
```

Aquí hay que especificar cuatro parámetros que diferirán entre varios entornos de federación y las apps.

- El identificador de la app como Relying Party en el proveedor de identidad (uri:miapp.migranja.com). Puede ser un URI o una URL. Esto sirve para que el proveedor de identidad sepa que app le está pidiendo autenticación.
- La URL del proveedor de identidad (idp.migranja.com).
- La URL de la federación del proveedor de identidad (<http://idp.migranja.com/adfs/services/trust>). La app usará esta URL para realizar la autenticación.
- La huella del certificado digital con el que se firman los tokens del proveedor de identidad (ABC-DEF123456789...). Esto sirve para que la app pueda validar que el token está bien firmado y que no está falseado.

Una última edición del fichero web.config para configurar bien las cookies de autenticación:

```
<system.identityModel.services>
<federationConfiguration>
    <cookieHandler requireSsl="true" />
    <wsFederation passiveRedirectEnabled="true"
    issuer="https://idp.migranja.com/adfs/ls/"
    realm="uri:miapp.migranja.com"
    requireHttps="true" />
</federationConfiguration>
</system.identityModel.services>
```

Aquí tenemos que decirle a nuestra app la URL de login del proveedor de identidad (<https://idp.migranja.com/adfs/ls/>) y repetir el identificador de la app como realm (uri:miapp.migranja.com).

Este último punto requiere un poco más de explicación. La app primero intentará ir a la URL de federación del proveedor de identidad. Si no dispone de la cookie de autenticación, el propio proveedor nos redireccionará a la pantalla de login, después de la cual se nos redireccionará a nuestra app. El último punto en el web.config sirve para que nuestra app reconozca esa redirección y sepa extraer el token correcto.

Es relativamente fácil conectar SharePoint con un proveedor de identidad (IdP) compatible como puede ser Active Directory Federation Services (ADFS)

La modificación del código de TokenHelper

Después de todo el trabajo en el web.config, hay que modificar el fichero TokenHelper para que use la identidad de autenticación correcta. Para ello disponemos del código creado por Wictor Wilén llamado SharePointContextSaml3.

Es un fichero llamado SharePointContextSaml.cs que añadiremos a nuestra solución.

Cambiaremos el fichero TokenHelper.cs para que se declare como partial.

```
public static partial class TokenHelper { ... }
```

En el fichero SharePointContext.cs cambiaremos el constructor para inicializar el nuevo proveedor de contexto de SharePoint con SAML.

```
static SharePointContextProvider0
{
    if (!TokenHelper.IsHighTrustApp())
    {
        SharePointContextProvider.current = new
SharePointAcsContextProvider();
    }
    else
    {
        if(HttpContext.Current.User.Identity.AuthenticationType ==
"Federation")
            SharePointContextProvider.current = new
SharePointHighTrustSamlContextProvider();
        } else {
            SharePointContextProvider.current = new
SharePointHighTrustContextProvider();
        }
    }
}
```

Y finalmente en el web.config añadiremos tres parámetros de configuración.

```
<add key="spsaml:ClaimProviderType" value="SAML"/>
<add key="spsaml:TrustedProviderName" value="IdP
Migranja"/>
<add key="spsaml:IdentityClaimType" value="SMTP"/>
```

El primero especifica si usamos la autenticación SAML o FBA, ya que el código soporta las dos opciones. El segundo es el nombre del proveedor de autenticación definido en SharePoint. El último es el atributo identificador del usuario en la base de datos de perfiles. Por defecto es el correo electrónico (SMTP) pero se admiten también SIP o UPN.

¡Nuestra app high-trust ya está lista para utilizar claims en su comunicación con SharePoint!

Conclusión

Hemos visto que habilitar la autenticación no Windows en las apps high-trust no es difícil pero sí que es un proceso tedioso y donde es fácil equivocarse. Si sólo una de las múltiples configuraciones que hemos tenido que hacer está mal, la autenticación no funcionará.

Con este artículo acabo la serie sobre las high-trust apps de SharePoint. Espero haberlos acercado una opción de desarrollo de la que no se habla mucho y que creo que dará buenos resultados para los nuevos desarrollos on-premises para los que sería deseable que estén preparados para su despliegue en la nube. ¡Un saludo y nos vemos en uno de los próximos números!

EDIN KAPIĆ

Arquitecto SharePoint

edin.kapic@sogeti.com

@ekapic

www.edinkapic.com

Desarrollo de Aplicaciones con Office 365 API

La plataforma de aplicaciones y servicios en la nube llamada Office 365, está siendo potenciada desde hace tiempo por Microsoft, ya sabemos que ventajas nos aporta un servicio. Principalmente nos ayuda a ahorrar costes en infraestructura y mantenimientos. A continuación, describiremos que tipo de aplicaciones podemos desarrollar y cómo funciona la API.

Tipo de Aplicaciones

En la siguiente tabla podemos ver los diferentes tipos de aplicaciones que podemos desarrollar agrupado por plataforma. Como podemos ver tenemos una gran variedad y eso nos permite interconectarnos con casi cualquier tecnología, incluso lenguajes Open Source.

MÓVIL	ESCRITORIO	WEB
iOS	Windows 8	.Net
Android	Windows 10	JavaScript
WP8		PHP , Python , etc ...

Office 365 API

A continuación, pasaremos a detallar que tipo de información tenemos disponible para su consumo, qué características tienen y cómo funciona la seguridad.

Información

La información accesible es la que podemos ver a continuación, juntamente sobre qué servicio se está consumiendo.

EXCHANGE ONLINE	AZURE AD	SHAREPOINT Y ONEDRIVE
Correos	Grupos	Archivos
Contactos		Sitios
Calendarios		

Características

Como hemos podido observar en el cuadro anterior, tenemos diferentes puntos de acceso dependiendo de la información y al servicio que estemos atacando, en el caso de

Office365 API Preview que explicaremos más adelante no es necesario. En el caso de tener diferentes tenant, necesitaremos usar el servicio Discovery que nos obtendrá el punto de acceso para obtener la información correspondiente.

En este artículo trataremos sobre cómo desarrollar con el API de Office 365 y que tipo de aplicaciones se pueden desarrollar

Aunque tenemos el Azure AD que nos proporciona la autenticación, necesitamos gestionar el token de acceso de manera separada para cada servicio antes mencionado.

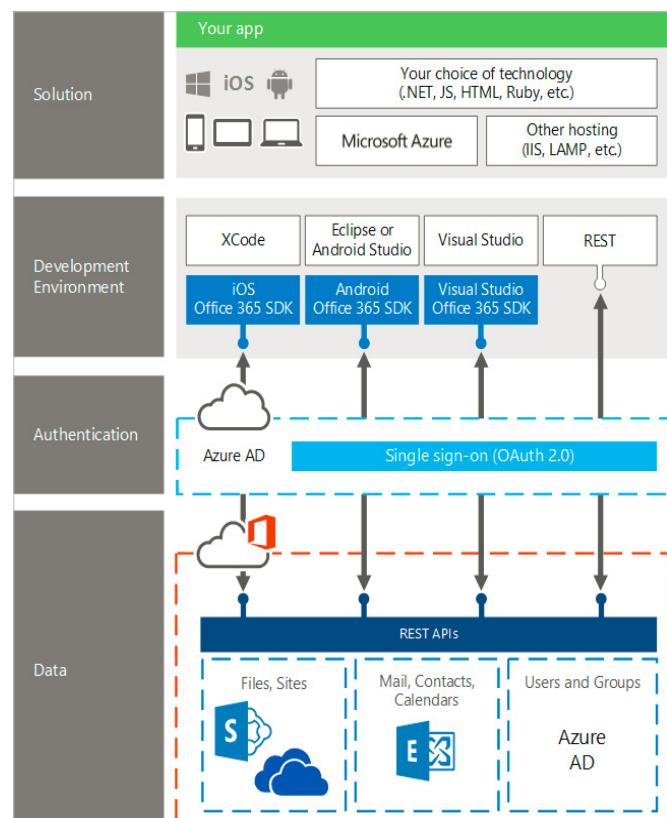


Imagen 1.- Cuadro resumen del funcionamiento de Office365 API

Seguridad

A continuación, vamos a describir que seguridad posee la API y cómo funciona el flujo de autenticación.

Autenticación

La API posee single sign on y OAuth 2.0, sistema basado en tokens y todo esto es gestionado por Azure Active Directory.

Flujo de Autenticación

- El usuario accede a una página o aplicación que consume información de Office 365, y este es redirigido a la página de autenticación de Azure.
- El usuario introduce sus credenciales y consiente en el caso que se necesiten permisos. Azure AD emite un código de autorización.
- La aplicación envía el código de autorización a Azure AD y este le devuelve el acceso y el token.
- La aplicación ya puede acceder a la información del usuario a través del token.

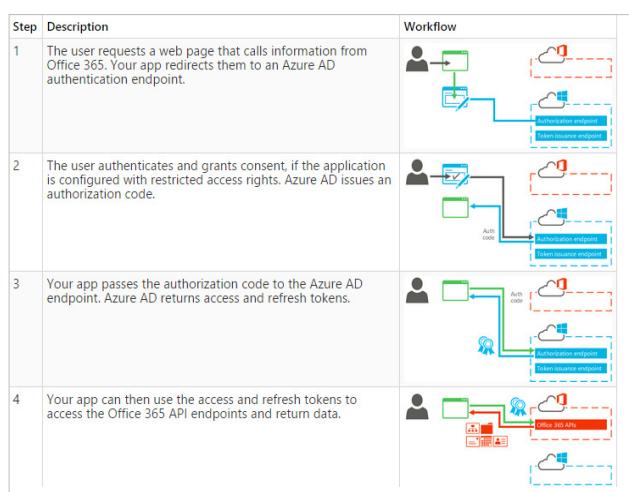


Imagen 2.- Flujo de autenticación de las APIs de Office 365.

Tenemos diferentes puntos de acceso dependiendo de la información y al servicio que estemos atacando, en el caso de Office365 API Preview

Ejemplo

- Obtenemos el token de Acceso

```
// Once the authToken has been acquired, create an outlookClient. One place to do this is inside of the
// ".then" function callback of authContext.getIdToken(...) above.
var outlookClient = new Microsoft.OutlookServices.Client('https://outlook.office365.com/api/v1.0', authToken.getAccessTokenFn('https://outlook.office365.com'));
```

- Obtenemos los contactos del usuario Ordenados por el Nombre

```
outlookClient.me.contacts.getContacts().orderBy('DisplayName asc').fetch().then(
  function (result) {
    result.currentPage.forEach(function (contact) {
      console.log(contact.displayName);
      contact.emailAddresses.forEach(function(emailAddress) {
        console.log('* ' + emailAddress.address);
      });
    });
  }, function(error) {
    console.log(error);
  }
);
```

Desarrollo Office 365

A continuación, vamos a hablar de la parte más enfocada al desarrollo, que Servicios necesitamos, Herramientas y cómo funciona la API con ejemplos.

Servicios

Estos son los servicios mínimos que vamos a necesitar para poder desarrollar una aplicación para Office 365:

- Office 365 API Setup.
- Office 365 Account:
 - Enterprise.
 - Education.
 - Developer.
 - Business.
- Azure AD.

Herramientas

Sera necesario un IDE para poder desarrollar, en este caso tenemos un abanico de posibilidades bastante amplio.

- Visual Studio.
- Eclipse / Android Studio.
- Xcode.
- Sublime.
- Etc.

API

Tenemos dos variantes, el SDK y el servicio REST.

SDK

Es una librería específica por tecnología que nos provee de un modelo de objetos para obtener la información. Tendríamos:

- Visual Studio Office 365 SDK.
- IOS Office 365 SDK.
- Android Office 365 SDK.

REST

Es un servicio multiplataforma que nos permite acceder a la información desde cualquier plataforma.

Ejemplo

- Obtenemos el token de Acceso:

```
var authContext;
var authToken; // for use with creating an outlookClient later.
authContext = new O365Auth.Context();
authContext.getIdToken("https://outlook.office365.com/")
    .then(function (token) {
        authToken = token;
        // The auth token also carries additional information. For example:
        userName = token.givenName + " " + token.familyName;
    }).bind(this), function (reason) {
    console.log('Failed to login. Error = ' + reason.message);
});
```

- Obtenemos los contactos del usuario

```
$.ajax({
    type: "GET",
    async:true,
    url: "https://outlook.office365.com/api/v1.0/me/contacts",
    headers: { "Authorization": "Bearer " + token,
               "accept":"application/json;odata=verbose" },
}).done(
    (function (data) {
        //console.log(data);
        ("#contacts").html(data);
    })
).fail(function (jqXHR, textStatus) {
    failcallback(textStatus)
});
```

Registro de una aplicación

Hay dos maneras de registrar nuestra aplicación en Azure AD para gestionar la seguridad y la autenticación. Dependiendo el IDE que usemos tendremos que usar una u otro. En el caso de Visual Studio podremos usar el modo automático o manual, en el caso del resto de IDE's será manual.

Manual

Los pasos a seguir para registrar una aplicación de forma manual en Azure AD son los siguientes:

- Registrar la aplicación.
- Configurar las propiedades.
- Seleccionar Servicio.
- Seleccionar los permisos.

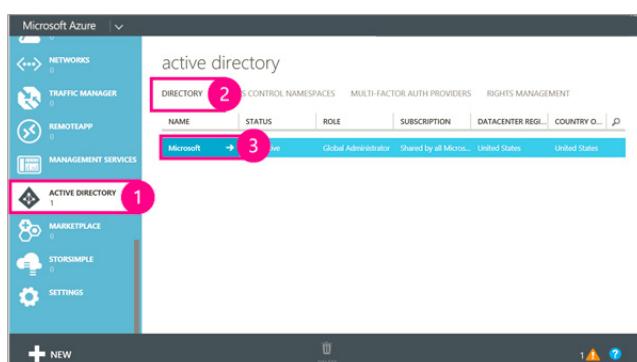


Imagen 3.- Pasos para conectar una aplicación.

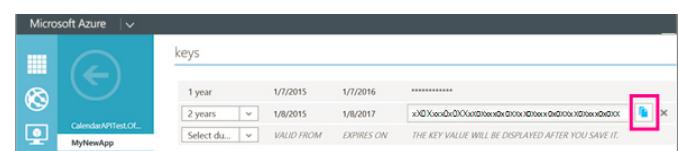


Imagen 4.- Pasos para conectar una aplicación.

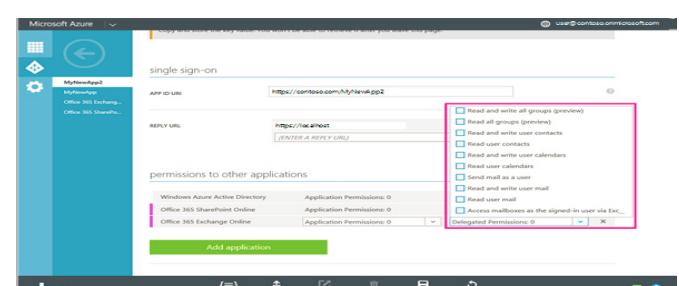


Imagen 5.- Pasos para conectar una aplicación.

A partir de la información anterior podemos ver que es posible desarrollar con cualquier tecnología

Automático

Los pasos a seguir para registrar una aplicación de forma automática por medio de Visual Studio son:

- Añadir Servicio Conectado.

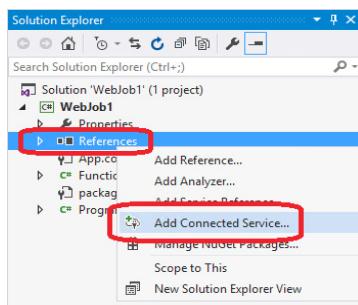


Imagen 6.- Añadir servicio conectado.

Office 365 API Preview

Esta es una de las novedades respecto a la actual API de office 365, es la nueva versión, ya es posible utilizarla, aunque no es recomendado para entornos de producción aún. Las mejoras que nos aporta esta versión son principalmente:

- Unificación de los puntos de acceso, ya no es necesario saber a qué servicio quieras consumir la información, ya que lo hace de forma transparente, además no es necesario obtener un token para cada servicio,

Ejemplo:

Llamada predefinida para obtener la información del usuario, donde ya no es necesario usar el punto de acceso del servicio.

- <https://graph.microsoft.com/beta/me>
- <https://graph.microsoft.com/beta/me/files>
- <https://graph.microsoft.com/beta/me/Messages>
- <https://graph.microsoft.com/beta/me/directReports>
- <https://graph.microsoft.com/beta/myOrganization/users>
- <https://graph.microsoft.com/beta/myOrganization/users/{user@domain}>
- <https://graph.microsoft.com/beta/myOrganization/users/{user@domain}/manager>
- <https://graph.microsoft.com/beta/myOrganization/groups>

Obtención de los mensajes del Usuario:

```
// The code sample below demonstrates how to get messages using the Office 365 unified API (Preview).
$.ajax('https://graph.microsoft.com/beta/me/Messages?$orderby=' + encodeURIComponent('DateTimeReceived desc'), {
    headers: {
        Authorization: 'Bearer <token:https://graph.microsoft.com/>',
        Accept: 'application/json;odata.metadata=none',
    }
}).then(function (response) {
    for (var i = 0; i < response.value.length; i++) {
        console.log('Message ' + response.value[i].Subject +
            " received at " + new Date(response.value[i].DateTimeReceived).toString() + '');
    }
    console.log('\n' + 'Full JSON response:')
    console.log(response);
}).fail(function (error) {
    console.log(error);
});
```

Conclusiones

A partir de la información anterior podemos ver que es posible desarrollar con cualquier tecnología, nos aporta la posibilidad de un desarrollo multiplataforma y sabemos que requisitos necesitamos para llevarlo a cabo. Lo más importante son las mejoras que están llegando y que ven-

sino que es un único token para todos.

- Se añaden más servicios, como es el caso de Office Graph, donde podremos obtener información de los usuarios y las relaciones entre los documentos y la organización.

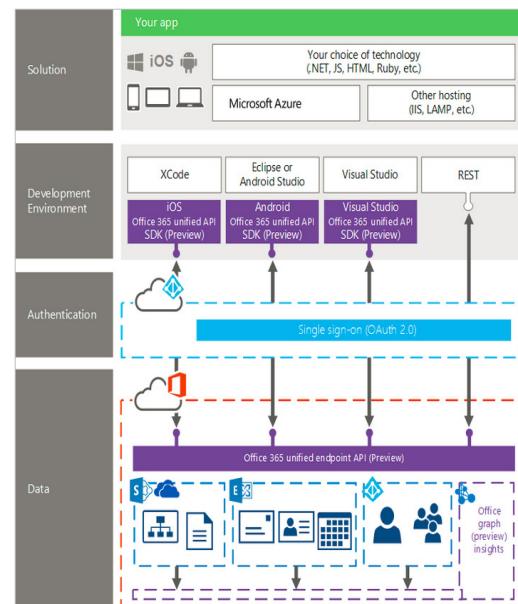


Imagen 7.- Cuadro resumen del funcionamiento de Office365 API Preview

drán más adelante y que cada vez están facilitando más el desarrollo.

RUBÉN TORIBIO

SharePoint Architect en Tokiota

@rtoribiog

i

11

Trabajando programáticamente con tipos de contenido en C#

Los tipos de contenido de SharePoint nos permiten crear listas basadas en estructuras de datos. Facilitan la replicación de cambios dentro de la colección de sitios. Por ejemplo, podemos crear listas en distintos sitios, que hereden su estructura de un tipo de contenido del sitio raíz. De esta forma, un cambio en una columna del tipo de contenido, se replica en todas las listas que lo consumen.

Crear el tipo de contenido

El primer paso es crear un nuevo tipo de contenido. Para ello trabajamos con esta función:

```
public string nuevoTipoContenido(SPSite sitio, string nombre)
{
    SPContentType viejoTC = sitio.RootWeb.ContentTypes[nombre];
    if (viejoTC == null)
    {
        SPContentType nuevoTC = new SPContentType(sitio.RootWeb.ContentTypes["Elemento"], sitio.RootWeb.ContentTypes, nombre);
        sitio.RootWeb.ContentTypes.Add(nuevoTC);
        nuevoTC.Group = "Surpoint";
        nuevoTC.Update();
        return "<span style='color: green'>Tipo de contenido " + nombre + " creado.</span><br>";
    }
    else
    {
        return "<span style='color: blue'>Tipo de contenido " + nombre + " ya existente.</span><br>";
    }
}
```

Finalmente ejecutamos esta función desde nuestro programa:

```
contenido.InnerHtml = "";
try
{
    SPSite oSiteCollection = SPContext.Current.Site;
    // Nuevos tipos de contenido
    contenido.InnerHtml += nuevoTipoContenido(oSiteCollection, "Riesgos_Planificaciones");
}
catch (Exception error)
{
    contenido.InnerHtml += "<span style='color: red'>Error: " + error + "</span><br>";
}
```

En la siguiente imagen vemos el resultado:

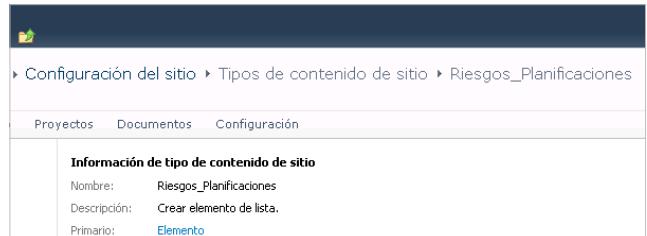


Imagen 1.– Creación de tipo de contenido.

Creación de columnas

En el segundo paso vamos a crear columnas de sitio y luego las vamos a incorporar dentro del tipo de contenido. Para ello armamos una primera función para crear columnas:

```
public string nuevaColumnaSitioFecha(SPSite sitio, string internalName, string displayName, bool obligatoriedad, string descripción)
{
    if (!sitio.RootWeb.Fields.ContainsField(internalName))
    {
        string nombreInternoNuevo = sitio.RootWeb.Fields.Add(internalName, SPFieldType.DateTime, obligatoriedad);
        SPFieldDateTime campoNuevo = (SPFieldDateTime)sitio.RootWeb.Fields.GetFieldByInternalName(nombreInternoNuevo);
        campoNuevo.Title = displayName;
        campoNuevo.DisplayFormat = SPDateTimeFieldFormatType.DateOnly;
        campoNuevo.Description = descripción;
        campoNuevo.Group = "Surpoint";
        campoNuevo.Update();
        return "<span style='color: green'>Columna " + internalName + " creada.</span><br>";
    }
    else
    {
        return "<span style='color: blue'>Columna " + internalName + " ya existente.</span><br>";
    }
}
```

Ejecutamos esta función desde nuestro programa:

```
contenido.InnerHtml += nuevaColumnaSitioFecha(oSiteCollection, "Planificado_Riesgo", "Fecha planificada del riesgo", false, "");
contenido.InnerHtml += nuevaColumnaSitioFecha(oSiteCollection, "Fecha_Planificacion_Riesgo", "Fecha de la planificación del riesgo", false, "");
```

En la siguiente imagen vemos el resultado:



Imagen 2.– Creación de columnas de sitio.

Los tipos de contenido de SharePoint nos permiten crear listas basadas en estructuras de datos.

Para incorporar las columnas dentro del tipo de contenido, creamos una nueva función:

```
public string agregarColumnaTipoContenido(SPSite sitio,
string columna, string tipoContenido)
{
    SPContentType actualTC = sitio.RootWeb.
ContentTypes[tipoContenido];
    if (!actualTC.Fields.ContainsField(columna))
    {
        SPField nuevaCol = sitio.RootWeb.Fields.
GetFieldByInternalName(columna);
        SPFieldLink nuevaColLink = new SPFieldLink(nuevaCol);
        actualTC.FieldLinks.Add(nuevaColLink);
        actualTC.Update(true);
        return "<span style='color: green'>Columna " + columna +
" agregada a " + tipoContenido + "</span><br>";
    }
    else
    {
        return "<span style='color: blue'>Columna " + columna + " ya existente en " + tipoContenido + "</span><br>";
    }
}
```

Ejecutamos esta función desde nuestro programa y luego vemos la imagen con el resultado:

```
contenido.InnerHtml +=
nuevaColumnaSitioFecha(oSiteCollection, "Planificado_-
Riesgo", "Fecha planificada del riesgo", false, "");
contenido.InnerHtml +=
nuevaColumnaSitioFecha(oSiteCollection, "Fecha_-
Planificacion_Riesgo", "Fecha de la planificación del
riesgo", false, "");
```

Información de tipo de contenido de sitio			
Nombre:	Riesgos_Planificaciones		
Descripción:	Crear elemento de lista.		
Primario:	Elemento		
Grupo:	_____		
Configuración			
<input type="checkbox"/> Nombre, descripción y grupo <input type="checkbox"/> Configuración avanzada <input type="checkbox"/> Configuración del flujo de trabajo <input type="checkbox"/> Eliminar este tipo de contenido de sitio			
Columnas			
Nombre	Tipo	Estado	
Título	Una línea de texto	Requerida	
Fecha planificada del riesgo	Fecha y hora	Opcional	Opcional
Fecha de la planificación del riesgo	Fecha y hora		

Imagen 3 – Incorporación de las columnas de sitio en el tipo de contenido.

Creación de listas

En el último paso crearemos listas basadas en el tipo de contenido en varios subsitios de nuestra colección de sitios. Seguimos estos pasos:

- Recorremos todos los sub sitios.
- Verificamos si existe la lista Riesgos (padre).
- Verificamos si existe la lista Riesgos_Planificaciones (nueva).
- Creamos la lista nueva.
- Le incorporamos el tipo de contenido.
- Le activamos el versionado.
- Le creamos un campo propio dependiente de otra lista.

```
SPWebCollection collWebsite = oSiteCollection.AllWebs;
for (int i = 0; i < collWebsite.Count; i++)
{
    using (SPWeb oWebsite = collWebsite[i])
    {
        SPList listaRiesgos = oWebsite.Lists.TryGetList("Issues y
riesgos");
        if (listaRiesgos != null)
        {
            // Nueva lista PR
            SPList listaPR = oWebsite.Lists.TryGetList("Riesgos_-
Planificaciones");

            if (listaPR == null)
            {
                SPListCollection listas = oWebsite.Lists;
                listas.Add("Riesgos_Planificaciones", "", ,
SPListTemplateType.GenericList);
                contenido.InnerHtml += "<span style='color: green'>" +
oWebsite.Title + "Lista PR creada</span><br>";
                // Tipo de contenido y versionado
                SPList listaPRNueva = oWebsite.Lists["Riesgos_-
Planificaciones"];
                listaPRNueva.EnableVersioning = true;
                listaPRNueva.ContentTypesEnabled = true;
                listaPRNueva.Update();
                listaPRNueva.ContentTypes.Add(oSiteCollection.
RootWeb.ContentTypes["Riesgos_Planificaciones"]);
                contenido.InnerHtml += "<span style='color: green'>" +
oWebsite.Title + "Lista PR - Tipo de Contenido agregado</
span><br>";
                listaPRNueva.ContentTypes["Elemento"].Delete();
                contenido.InnerHtml += "<span style='color: green'>" +
oWebsite.Title + "Lista PR - Tipo de Contenido elemento
ajustado</span><br>";
                // Creación de columna propia de la lista (no
heredadas del tipo de contenido)
                listaPRNueva.Fields.AddLookup("Riesgo",
listaRiesgos.ID, true);
            }
            else
            {
                contenido.InnerHtml += "<span style='color: blue'>" +
oWebsite.Title + "Ya existe la lista PR</span><br>";
            }
        }
        else
        {
            contenido.InnerHtml += "<span style='color: blue'>" +
oWebsite.Title + "No existe la lista Padre Riesgos</
span><br>";
        }
    }
}
```

En la imagen podemos ver la lista creada, el tipo de contenido y las columnas:

Información de la lista

Nombre:	Riesgos_Planificaciones																					
Dirección web:	http://[REDACTED]19/Lists/Riesgos_Planificaciones/AllItems.aspx																					
Descripción:																						
Configuración general																						
Título, descripción y navegación Configuración de versiones Configuración avanzada Configuración de validación Configuración de identificación de audiencias																						
Permisos y administración																						
Eliminar esta lista Guardar lista como plantilla Permisos para esta lista Configuración del flujo de trabajo																						
Tipos de contenido																						
<p>Esta lista está configurada para permitir diversos tipos de contenido. Use tipos de contenido para especificar la info otros comportamientos. Los tipos de contenido siguiente se encuentran disponibles actualmente en esta lista:</p> <table border="1"> <tr> <td>Tipo de contenido</td> <td>Visible en botón nuevo</td> </tr> <tr> <td>Riesgos_Planificaciones</td> <td>✓</td> </tr> </table> <p>Agregar a partir de tipos de contenido de sitio Cambiar orden del nuevo botón y tipo de contenido predeterminado</p>		Tipo de contenido	Visible en botón nuevo	Riesgos_Planificaciones	✓																	
Tipo de contenido	Visible en botón nuevo																					
Riesgos_Planificaciones	✓																					
Columnas																						
<p>Una columna almacena información de cada elemento de la lista. Dado que la lista permite varios tipos de contenido elemento como, por ejemplo, si la información es necesaria u opcional para una columna. Las columnas siguientes</p> <table border="1"> <thead> <tr> <th>Columna (hacer clic para editar)</th> <th>Tipo</th> <th>Usado en</th> </tr> </thead> <tbody> <tr> <td>Fecha de la planificación del riesgo</td> <td>Fecha y hora</td> <td>Riesgos_Pla</td> </tr> <tr> <td>Fecha planificada del riesgo</td> <td>Fecha y hora</td> <td>Riesgos_Pla</td> </tr> <tr> <td>Riesgo</td> <td>Búsqueda</td> <td>Riesgos_Pla</td> </tr> <tr> <td>Título</td> <td>Una línea de texto</td> <td>Riesgos_Pla</td> </tr> <tr> <td>Created by</td> <td>Persona o grupo</td> <td>Riesgos_Pla</td> </tr> <tr> <td>Modified by</td> <td>Persona o grupo</td> <td>Riesgos_Pla</td> </tr> </tbody> </table>		Columna (hacer clic para editar)	Tipo	Usado en	Fecha de la planificación del riesgo	Fecha y hora	Riesgos_Pla	Fecha planificada del riesgo	Fecha y hora	Riesgos_Pla	Riesgo	Búsqueda	Riesgos_Pla	Título	Una línea de texto	Riesgos_Pla	Created by	Persona o grupo	Riesgos_Pla	Modified by	Persona o grupo	Riesgos_Pla
Columna (hacer clic para editar)	Tipo	Usado en																				
Fecha de la planificación del riesgo	Fecha y hora	Riesgos_Pla																				
Fecha planificada del riesgo	Fecha y hora	Riesgos_Pla																				
Riesgo	Búsqueda	Riesgos_Pla																				
Título	Una línea de texto	Riesgos_Pla																				
Created by	Persona o grupo	Riesgos_Pla																				
Modified by	Persona o grupo	Riesgos_Pla																				

Imagen 4.– Creación de listas que utilicen tipos de contenido.

Conclusión

En esta primera versión del artículo hemos visto como trabajar con tipos de contenido en forma programática, no declarativa. En próximos artículos veremos algunos temas más avanzados. Espero les haya resultado de utilidad.

JUAN PABLO PUSSACQ LABORDE

SharePoint MVP

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

KWizCom Forms App

**Solución para
formularios 100%
nativos en SharePoint**

**Mejore a SharePoint,
no lo reemplace...**



www.kwizcom.com

Introducción a los Grupos de Office 365 – Trabaja como una red - Parte IV

La funcionalidad de Grupos de Office 365 facilita el trabajo colaborativo entre personas de una organización dotándoles de funcionalidad de valor añadido como muro conversacional, experiencia de usuario completamente integrada en Outlook Web App (OWA) en Office 365 y capacidades adicionales como calendario de grupo, sitio de SharePoint Online dónde ir dejando los documentos del grupo, etc. Además de la experiencia de usuario disponible en OWA, es posible trabajar con Grupos de Office 365 desde Outlook 2016.

Nota: Los Grupos de Office 365 no están desplegados de forma global de Office 365, sino que actualmente sólo pueden ser habilitados y utilizados en tenants de Office 365 en los que se haya habilitado la configuración "First Release".

La funcionalidad de Grupos de Office 365 facilita el trabajo colaborativo entre personas de una organización

Trabajo con Grupos de Office 365 desde Outlook 2016

Una de las características que más se solicitó a Microsoft para los Grupos de Office 365 era la de poder interactuar con los mismos desde el cliente Outlook sin tener que pasar por OWA. Afortunadamente, Microsoft ya tenía en mente su implementación y desde hace varias semanas tenemos la posibilidad en Outlook 2016 (Se espera que también sea posible en Outlook 2013) de trabajar con Grupos siempre y cuando nos descarguemos la versión del cliente disponible en un tenant de Office 365 configurado en modo First Release. Para comenzar a trabajar con Grupos de Office 365 desde Outlook 2016:

- Necesitamos en primer lugar instalar Outlook 2016 desde el portal de Administración de Office 365 y a continuación configurar en el mismo una cuenta de correo electrónico de Exchange Online. Una vez configurada la cuenta, lo primero que nos encontraremos en relación a los Grupos de Office 365 en Outlook 2016 es que para un buzón de Exchange Online en Office 365 se muestra una carpeta "Grupos" que al ser desplegada visualiza todos los Grupos que están

disponibles en el tenant y de los que nuestro usuario de Office 365 forma parte.

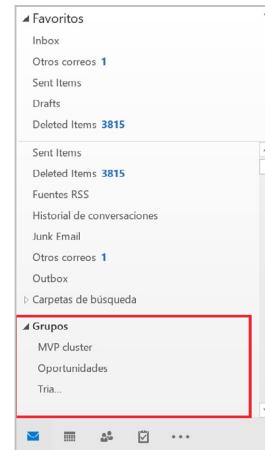


Imagen 1.- Sección de los Grupos de Office 365 en Outlook 2016.

- Si hacemos clic en uno de los Grupos, veremos cómo Outlook nos muestra los mensajes intercambiados en el mismo y además se habilitan una serie de opciones en la Ribbon de Outlook 2016 para poder trabajar con el Grupo seleccionado. También para el Grupo veremos los integrantes del mismo.

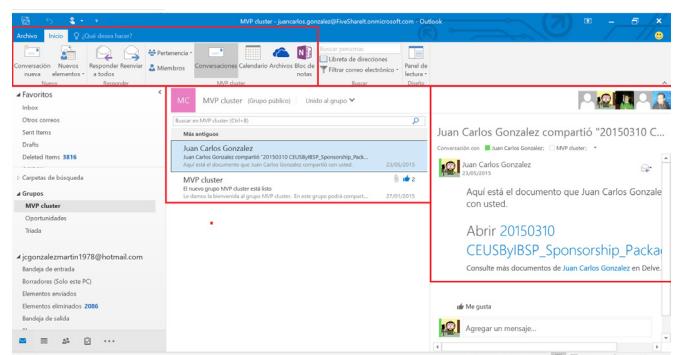


Imagen 2.- Experiencia de trabajo con Grupos de Office 365 en Outlook 2016.

Creación de un Grupo de Office 365 en Outlook 2016

El proceso de creación de un nuevo Grupo de Office 365 desde Outlook 2016 es bastante directo y sencillo:

- Para crear un nuevo Grupo desde Outlook, simplemente seleccionamos el nodo "Grupos", hacemos clic con el botón derecho del ratón y a continuación pulsamos sobre "Crear Grupo". En la ventana que se abre podremos crear un nuevo Grupo indicando su

nombre y si es público o privado.

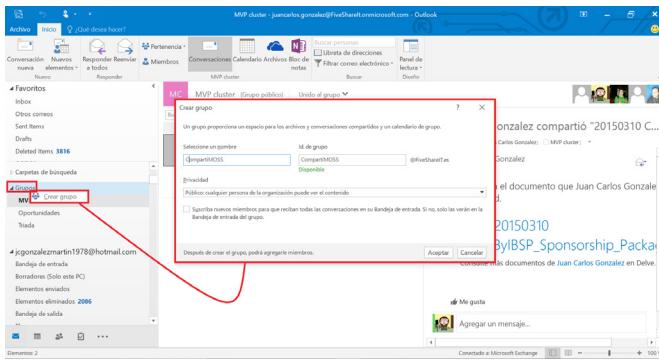


Imagen 3.- Creación de un Grupo de Office 365 desde Outlook 2016.

- A continuación, podremos añadir nuevos miembros al Grupo y también una imagen y descripción para el mismo. En el caso de la imagen, para añadirla es necesario hacer uso de OWA ya que por el momento no es posible añadirla desde Outlook 2016.

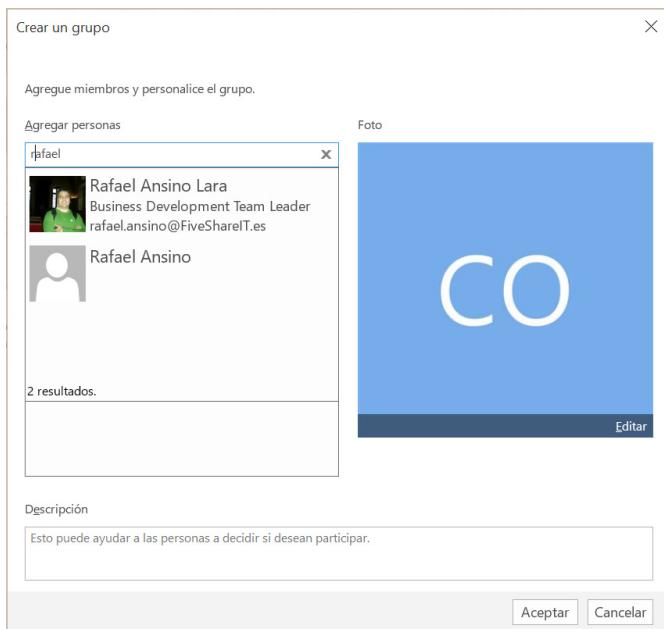


Imagen 4.- Añadiendo miembros al Grupo de Office 365.

- Una vez concluye el proceso de creación del nuevo grupo, este se muestra bajo Grupos tanto en Outlook 2016 como en OWA.

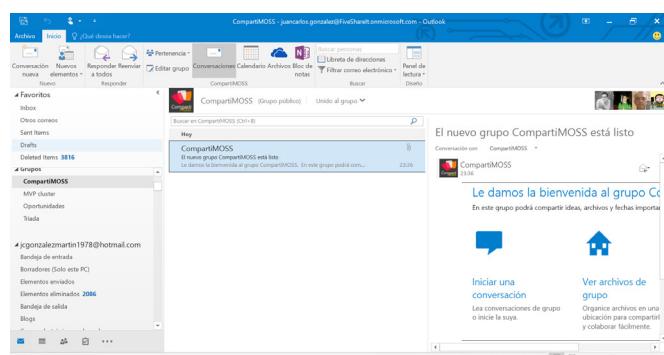


Imagen 5.- Detalle del Grupo creado desde Outlook 2016.

- A partir de aquí, podemos iniciar conversaciones en el Grupo directamente desde el cliente Outlook 2016 sin más que realizar la típica acción de escribir un correo electrónico que en este caso será enviado al

buzón de correo generado para el Grupo.

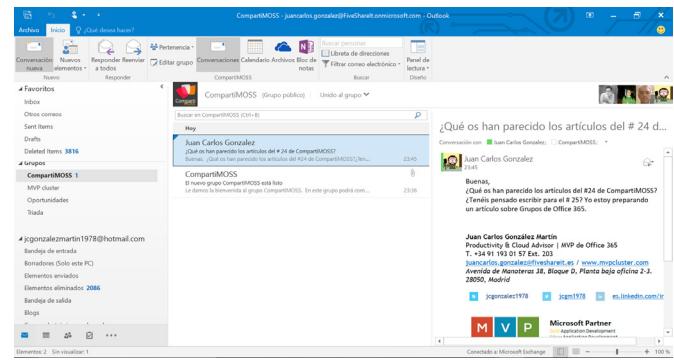


Imagen 6.- Ejemplo de conversación creada en el Grupo.

Creación de Conversaciones en Grupos de Office 365

En esta sección vamos a ver en detalle cómo crear conversaciones en el seno de un Grupo de Office 365 haciendo uso de Outlook 2016:

- Para crear una nueva conversación en el Grupo, basta con hacer clic en “Nueva Conversación” en Outlook para que se abra la ventana estándar de creación de correo electrónico para iniciar una conversación en un Grupo.

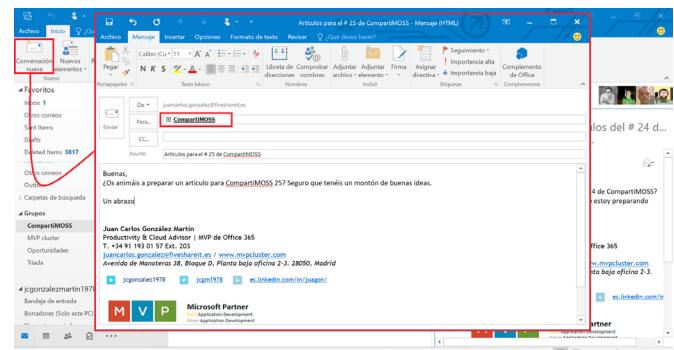


Imagen 7.- Creación de una nueva conversación en el Grupo haciendo uso de la opción “Conversación nueva” de la cinta de Outlook 2016.

- Por supuesto, podemos iniciar una conversación en cualquier momento desde Outlook sin más que en el campo “Para” del correo a enviar especificar la dirección del Grupo que tiene la forma <Nombre_Grupo>@<DominioOffice365>.

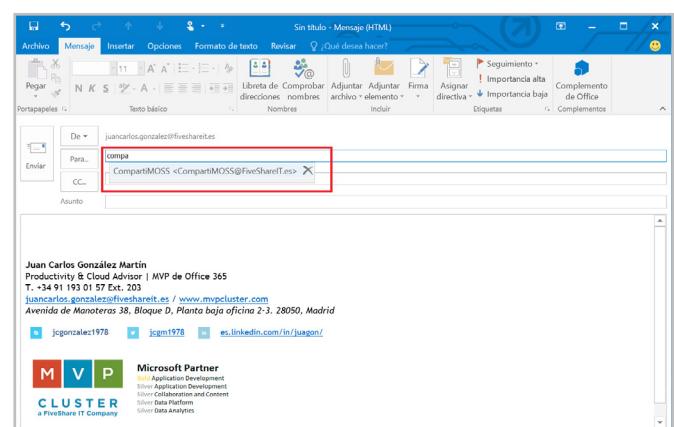


Imagen 8.- Inicio de una conversación en un Grupo desde la ventana de nuevo mensaje de correo electrónico.

- Esta forma de crear conversaciones mediante mensajes de correo electrónico habilita que usuarios ajenos al Grupo puedan enviar mensajes al mismo siempre y cuando hagan uso de la dirección de correo del Grupo que se ha indicado. Previamente, es necesario habilitar esta opción a nivel de Grupo haciendo uso de la opción “Editar grupo” de la cinta.

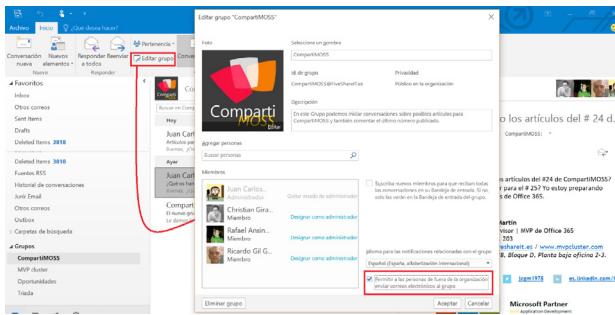


Imagen 9.- Configuración del Grupo para que usuarios externos puedan enviar mensajes al Grupo.

- A partir de esta configuración, ya podremos recibir mensajes en el Grupo procedentes de correos externos al mismo.
- Además de crear nuevas conversaciones, podemos responder como integrantes de un Grupo a conversaciones existentes o bien indicar que nos Gusta la conversación haciendo uso de la opción “Me gusta” disponible a nivel de conversación.

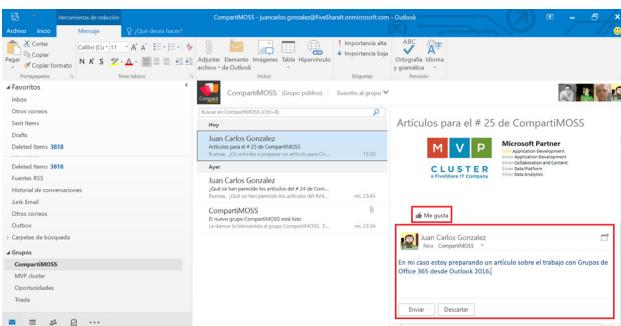


Imagen 10.- Funcionalidad de “Me gusta” y experiencia de respuesta a una conversación existente.

- Por supuesto, tanto las conversaciones con respuestas como el # de veces que se haya hecho “Me gusta” en una conversación se muestran en Outlook 2016. Para el caso de las conversaciones, la experiencia es similar a la de cualquier foro o red social de Internet en la que sobre un tema se pueden ir añadiendo comentarios y opiniones de forma anidada.

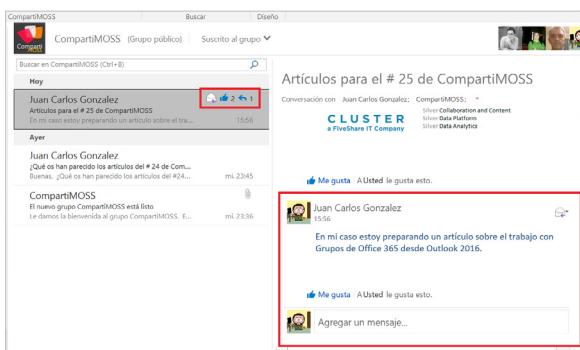


Imagen 11.- Experiencia de visualización de respuestas a conversaciones y de “Me gusta”.

Otras funciones de Grupos de Office 365 disponibles en Outlook 2016

Para finalizar este artículo, vamos a ver otras funciones de Grupos de Office 365 disponibles en el cliente de Outlook 2016:

- Si queremos recibir una copia de las conversaciones del Grupo en nuestra bandeja de entrada, basta con que a nivel de Grupo hagamos uso de la opción “Suscribirse”. De esta forma, nuestro usuario pasará de estar “Unido” al Grupo a estar “Suscripto” al Grupo.

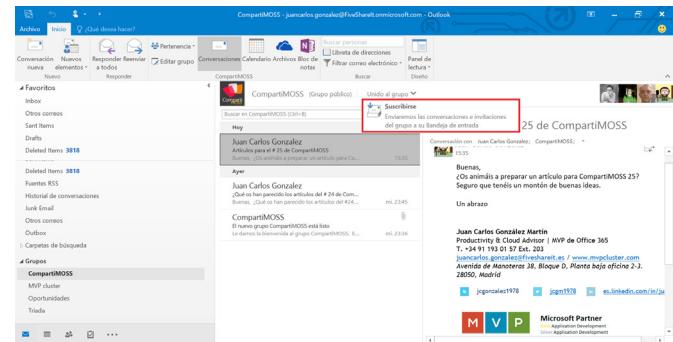


Imagen 12.- Opción “Suscribirse” en un Grupo.

- El calendario del Grupo se integra en Outlook como cualquier otro calendario y el acceso al mismo está disponible en las opciones de la cinta de Outlook disponibles para trabajar con Grupos.

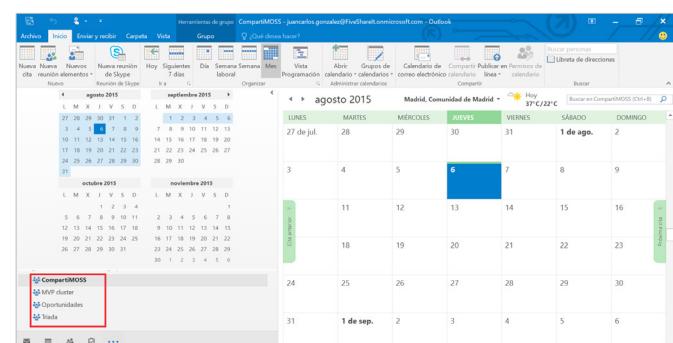


Imagen 13.- Acceso al calendario del Grupo desde Outlook 2016.

- Finalmente, si hacemos clic en Archivos o Bloc de Notas, se abrirá el navegador con el sitio asociado al Grupo de Office 365 o el Bloc de Notas por defecto del mismo. Tanto en un caso como en el otro, hay que tener en cuenta que la primera vez que se hace clic o en Archivos o en Bloc de Notas se produce el aprovisionamiento de la Colección de Sitios y del Bloc de Notas de OneNote asociado al Grupo.



Imagen 14.- Bloc de Notas asociado al Grupo.

Conclusiones

En este artículo hemos revisado como todas las características y funcionalidades de Grupos de Office 365 disponibles en OWA están también disponibles en el cliente Outlook 2016 de manera que los usuarios que habitualmente utilizan Outlook puedan participar también en los Grupos sin necesidad de hacerlo a través del explorador web.

JUAN CARLOS GONZÁLEZ MARTÍN

MVP de Office 365 | Cloud & Productivity Advisor en
MVP CLUSTER

E-Mail: juancarlos.gonzalez@fiveshareit.es

Twitter: @jcm1978

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

Web: www.mvpcluster.es



Integración Continua para Apps Office 365 con Visual Studio Online y Azure

Desde la primera versión de SharePoint, allá en 2003, los equipos de desarrollo tradicionalmente han tenido muchos problemas con el despliegue de la solución en otros entornos. Motivos hay muchos, en primer lugar la propia plataforma que fue creciendo exponencialmente el número de usuarios respecto a los desarrolladores que había, pero de la misma forma no hay que quitar responsabilidad a los equipos de desarrollo. Este hecho provocó que trabajar con SharePoint fuera un problema para muchos desarrolladores que venían de otros entornos más simples (Aplicaciones escritorio, Web,...). Ahora bien como todo buen producto, fue evolucionando y se fue adaptando a la misma forma que se hace en otras aplicaciones (con alguna peculiaridad propia de la plataforma). En este artículo vamos a ver como configurar la Build de nuestro Visual Studio Online para realizar integración continua.

¿Qué es integración continua?

La integración continua es un modelo informático propuesto inicialmente por Martin Fowler que consiste en hacer integraciones automáticas de un proyecto lo más a menudo posible para así poder detectar fallos cuanto antes. Entendemos por integración, la compilación y ejecución de pruebas de todo un proyecto. Las principales ventajas que tiene son:

- Permite detectar y solucionar problemas de integración de forma continua, evitando el caos de última hora cuando se acercan las fechas de entrega.
- Disponibilidad constante de una versión para pruebas, demos o lanzamientos anticipados.
- Monitorización constante de las métricas del código.
- Ejecución inmediata de pruebas (Unitarias, UI, Performance).

Sobre Visual Studio Online

Aunque por su nombre quizás pueda llevar a confusión Visual Studio Online (VSO) es la herramienta proporcionada por Microsoft para gestionar todo el ciclo de vida o Application Lifecycle Management (ALM) de una aplicación. Este proceso empieza mucho antes de que cualquier desarrollador escriba una línea de código y no termina cuando se pone la primera versión en producción.

Con VSO, se puede llevar la planificación del propio proyecto, aplicar una metodología ágil, separar/asignar ta-

reas, llevar el control del código fuente (bien con repositorios centralizados como Team Foundation Server o con repositorios distribuidos como Git) e incluso realizar el despliegue de nuestro desarrollo en diferentes entornos (este último aspecto es el que vamos a tratar en este artículo).

Visual Studio Online es gratis para equipos de hasta 5 desarrolladores, lo que hace que para empresas pequeñas o que acaban de empezar les sea de gran utilidad. Por lo general cuando analizamos la utilización de un producto Online respecto a uno OnPremise se valora si tienen las mismas características. En este caso Visual Studio Online tiene más características que la solución en OnPremise debido a que ya tienen instaladas todas las utilidades necesarias para integración continua, por ejemplo, para SharePoint en un entorno local necesitaríamos instalar las librerías de del servidor.

Manos a la obra

En primer lugar, partimos de la base que disponemos de una App de SharePoint del tipo Provider-Hosted. Como proveedor de alojamiento vamos a utilizar Azure. Para ello:

- Abrimos Visual Studio -> Templates -> Visual C# -> Office /SharePoint -> Apps-> App for SharePoint.
- A continuación, seleccionamos el tipo de alojamiento Provider-Hosted para nuestra aplicación.

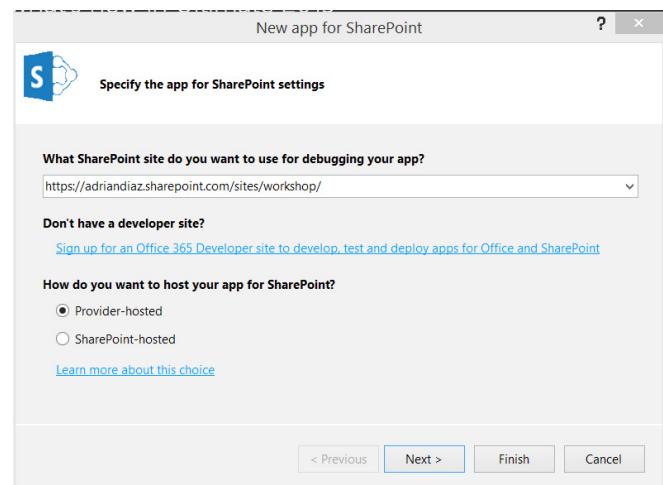
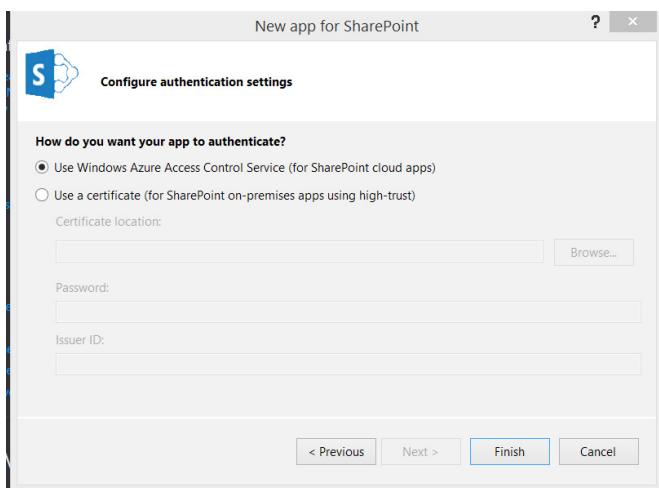


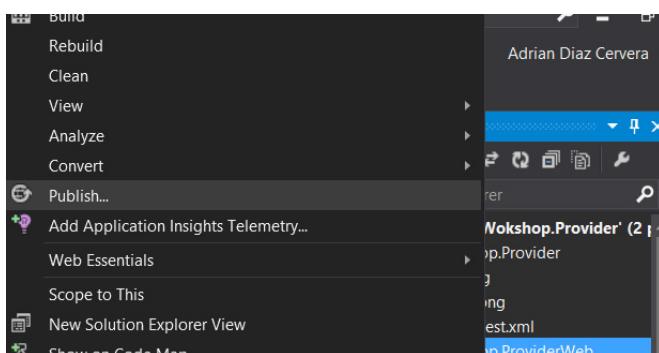
Imagen 1.- Asistente para crear una Aplicación Provider-Hosted en Visual Studio.

- Seleccionamos el tipo de autenticación que vamos a utilizar. En caso de Office 365 utilizaremos Windows

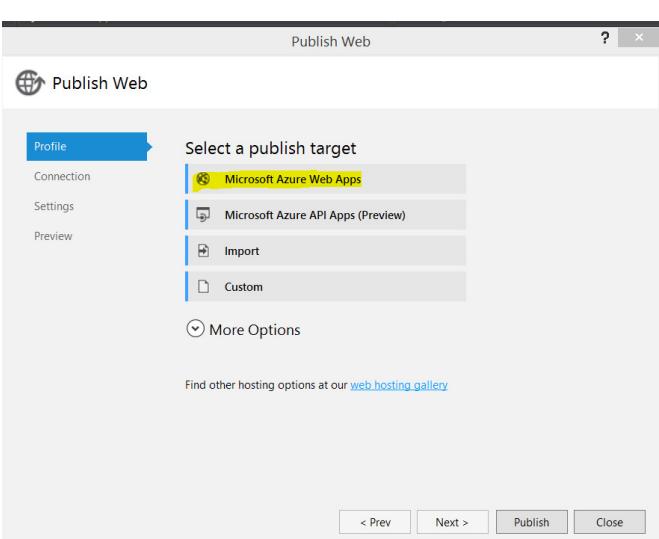
Azure Access Control Service, si estamos en un entorno On Premise utilizaremos un Certificado de una entidad autorizada.



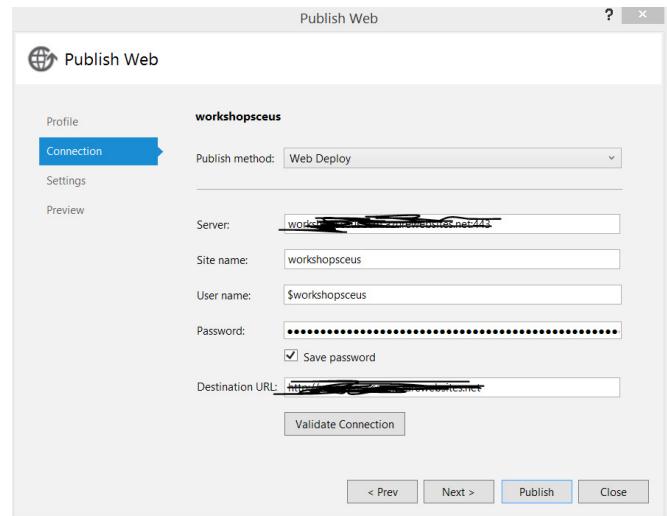
- Una vez tenemos la estructura del proyecto creada publicaremos la Aplicación MVC en un sitio de Azure. Para ello en primer lugar nos dirigimos al explorador de soluciones, se selecciona la aplicación MVC, botón derecho y pulsar sobre Publicar.



- Dentro de las opciones disponibles Seleccionaremos Microsoft Azure Web Apps.



- A continuación se crea o utiliza una Web Apps ya existente:



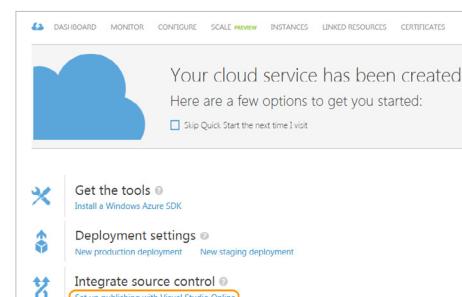
- Haga clic en Siguiente y seleccione la forma en la que se va a desplegar bien en modo depuración (debug) bien en producción (release). A continuación, se pulsa sobre Publish.

Los equipos de desarrollo tradicionalmente han tenido muchos problemas con el despliegue de la solución en otros entornos

Crear el servicio de Visual Studio Online en Azure

En la suscripción de Azure hay que crear el servicio de Visual Studio Online para permitirle acceder al repositorio de código fuente. Para ello:

- Vaya al Portal de Azure, seleccione su servicio en la nube o sitio web, o bien cree uno nuevo. Para ello, seleccione el ícono + en la parte inferior izquierda y elija Servicio en la nube o Sitio web y, a continuación, Creación rápida. Haga clic en el vínculo Configurar publicación con Visual Studio Online.



- En el asistente, escriba el nombre de la cuenta de Visual Studio Online en el cuadro de texto y haga clic en el vínculo Autorizar ahora. Puede que se le solicite que inicie sesión.

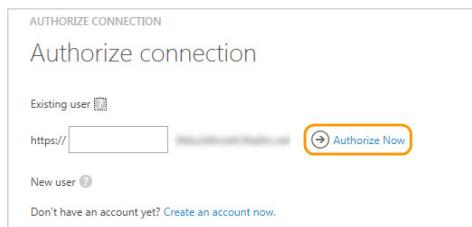


Imagen 7.- Autorización a VSO.

- En el cuadro de diálogo emergente de OAuth, elija Aceptar para autorizar a Azure a que configure el proyecto de equipo en VSO.

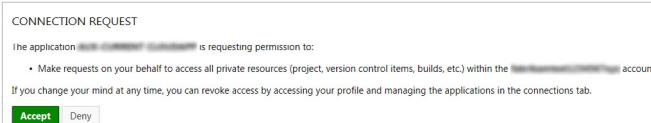


Imagen 8.- Solicitud de acceso al Repositorio.

- Si la autorización se realiza correctamente, verá una lista desplegable que contiene los proyectos de equipo de Visual Studio Online. Seleccione el nombre del proyecto de equipo que ha creado en los pasos anteriores y presione el botón de la marca de verificación del asistente.



Imagen 9.- Selección de repositorio a desplegar.

- Una vez que el proyecto se haya vinculado, verá instrucciones para registrar los cambios en el proyecto de equipo de Visual Studio Online. La próxima vez que se registre, Visual Studio Online compilará e implementará el proyecto en Azure. Para probarlo ahora, haga clic en el vínculo Proteger desde Visual Studio y, a continuación, en Iniciar Visual Studio (o el botón equivalente de Visual Studio situado en la parte inferior de la pantalla del portal).

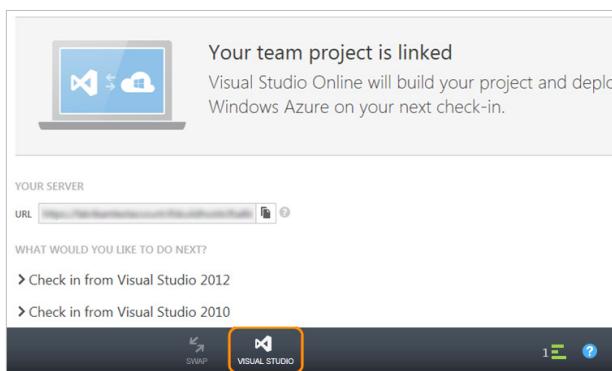


Imagen 9.- Creación del Servicio.

Tareas Previas

El punto de partida para empezar con la integración con

tinua de las Aplicaciones es el siguiente paquete de Codeplex: <https://officesharepointci.codeplex.com/>

Este paquete tiene dos carpetas:

- BuildProcessTemplates: Ficheros xml que tienen la serie de requisitos que va a realizar la build.
- DeployScript: Una serie de Script PowerShell para poder instalar la aplicación en los entornos necesarios. Estos Scripts no asumen que la aplicación ya existe.

El proyecto de CodePlex que estamos utilizando asume que la aplicación ya está registrada en tu entorno. Para las pruebas en primer lugar vamos a registrar la aplicación para ello iremos a https://sitiodeveloper/_layouts/appregnew.aspx y registrar la aplicación.

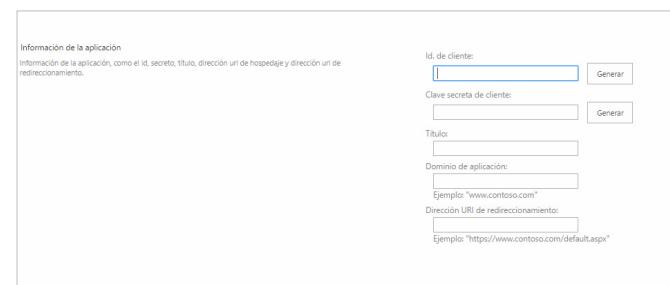


Imagen 10.- Registro de la App en el entorno de desarrollo.

Es la herramienta proporcionada por Microsoft para gestionar todo el ciclo de vida o Application Lifecycle Management

Con el ID Cliente y la Clave Secreta la introducimos en la aplicación. Por un lado dentro de los parámetros del Web.config y por otro lado en el manifiesto de la APP de SharePoint tal y como visualizamos en la siguiente imagen:

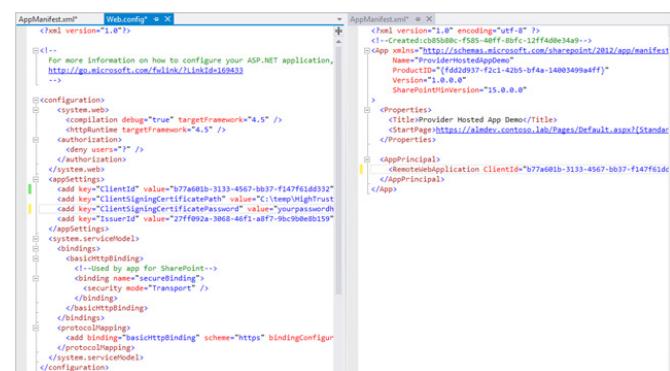


Imagen 11.- Actualización de los archivos Web.config y de manifiesto de la App.

Agregar el paquete de Codeplex dentro de nuestro Repositorio de Código

Para agregar el paquete en el repositorio de código:

- Copiar las carpetas "DeployScripts" y "BuildProcessTemplates" descargadas del Proyecto de Codeplex dentro de la carpeta del Proyecto.
- Ahora que la estructura de las carpetas esta de la manera en la que este dentro del repositorio de código

go fuente. Hacer clic con el botón derecho en el nodo del proyecto en el Explorador de control de código fuente y elegir “Add Items to Folder”

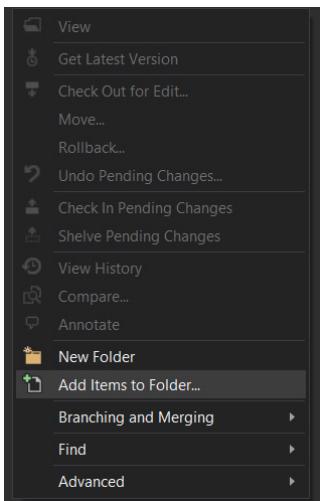


Imagen 12.- Inicio de mapeo en TFS.

- Se advierte que hay dos elementos que han cambiado en el proyecto existente. Aceptamos y elegimos todos los elementos a añadir.
 - A continuación, llega la hora de indicar los valores en los que nosotros queremos realizar la modificación. Para ello vamos a la carpeta DeployScripts/SharePointApp/Parameters Aquí es donde añadimos los valores de nuestro Office 365, SharePoint OnPremise o Web Deployment (en caso de que no sea una Web APP de Azure).

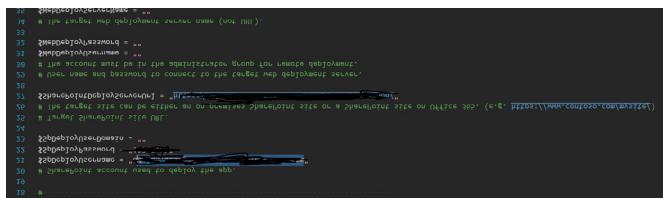


Imagen 13.- Configuración de los parámetros de despliegue.

Crear una definición de la Build

Una vez ya tenemos el servicio de Visual Studio Online funcionando en Azure, crearemos una definición de la Build en el que configuraremos la integración continua. En estas definiciones se puede configurar varios aspectos:

- Evento que provoca que se ejecute: por ejemplo cada check-in
 - Periodicidad: todos los días a una determinada hora, etc...

Es posible (a la vez que recomendable) que en cada proyecto tengamos varias definiciones, es posible que nos interese diariamente desplegar nuestros desarrollo en un entorno de preproducción, pero también es necesario por ejemplo que cada vez que se suba un cambio se testeen si se han pasado las pruebas unitarias o se han cumplido los requisitos de calidad establecidos.

- Abrimos Visual Studio, en el explorador de soluciones pulsar sobre la pestaña Team Explorer y seleccionar

sobre el ícono de Builds.

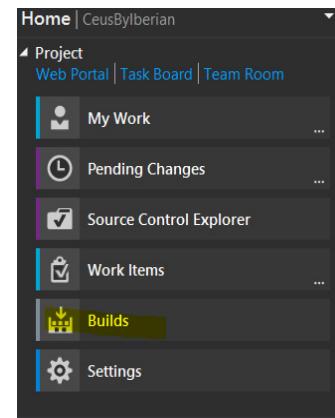


Imagen 14.- Acceso a la sección de Builds.

- Pulsamos sobre la Opción de New Build Definition:

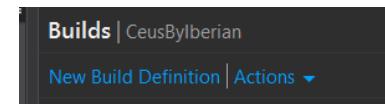


Imagen 15.- Opción New Build Definition.

- A la hora de la creación tendremos una serie de pestanas laterales en la que agregaremos la configuración de la Build.

- Trigger; Esto parámetro establece cada cuento tiempo queremos ejecutar la BUILD. Dependiendo de lo que realice la definición optaremos una solución u otra. Quizás subir a un entorno de Integración solo lo queremos realizar una vez al día, pero por el contrario sí que queremos que cada Check In que se realice se analice el código subido al gestos del código fuentes. Todas estas opciones las configuramos en la siguiente pantalla:

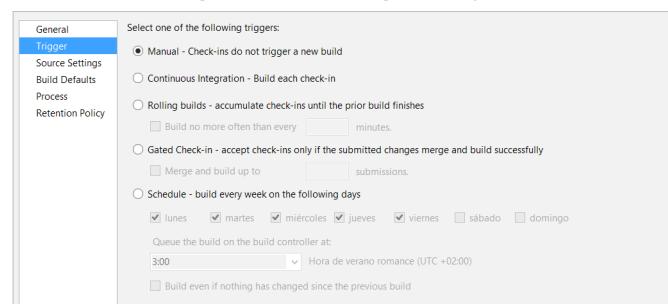


Imagen 16.- Pestaña Trigger en la configuración de la build.

- Source Settings: Agregaremos los proyectos que se van a compilar en cada Build.
 - Build Defaults: Seleccionamos el Build Controller y la ubicación donde va a estar el resultado de la build

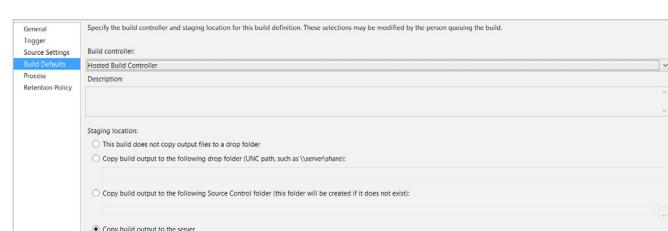


Imagen 17.- Pestaña Build Defaults en la configuración de la build.

- Process: En esta opción seleccionamos el Template que vamos a utilizar para esta Build, para realizar la integración continua con las APP seleccionaremos el template que nos hemos descargado de CodePlex (más adelante veremos cómo personalizar este template)



Imagen 18.- Selección del Template para la Build.

Dentro de los parámetros de este proceso de Build tendremos que añadir lo siguiente:

1. Advanced-> MSBuild Arguments: /p:ActivePublishProfile="perfil de Azure"
2. Misc-> Deployment Script Ruta donde hemos dejado el fichero SharePointApp-Delay.ps1 (descargado previamente de Codeplex)

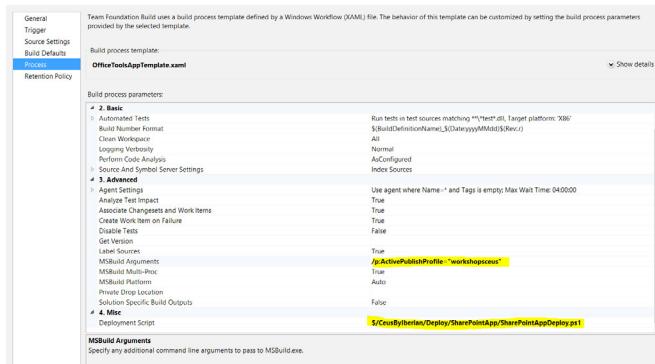


Imagen 19.- Parámetros del proceso de Build a añadir.

- A continuación, ejecutamos la build de forma manual y si todo ha ido bien se muestra una pantalla como la

siguiente:

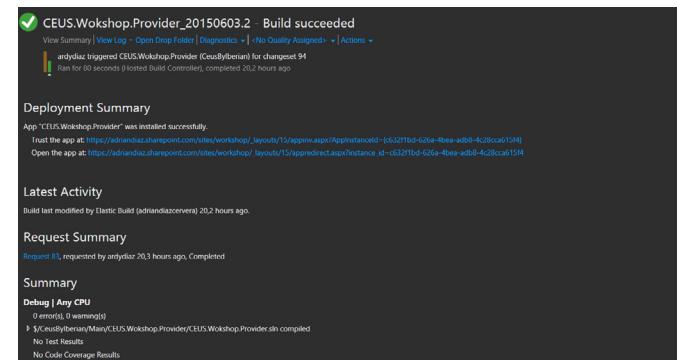


Imagen 20.- Ejecución de la Build.

Conclusión

El proceso de ALM es muy necesario en cualquier plataforma y Office 365/SharePoint no es una excepción, en este artículo hemos visto como configurar un proceso de integración continua para nuestras Apps. Este proceso, conforme a las necesidades de cada equipo y a los requisitos, se va mejorando con otras definiciones de Build, definiciones que se pueden encargar de verificar la calidad del código descrito, de pasar los Test Unitarios, pruebas de carga y/o de la interfaz.

ADRIÁN DIAZ CERVERA
SharePoint Architect at Encamina
MVP SharePoint Server
<http://blogs.encamina.com/desarrollandosobresharepoint>
<http://geeks.ms/blogs/adiazcervera>
adiaz@encamina.com @AdrianDiaz81

Entrevista André Lage

My name is André Lage, I am Portuguese but living in Basel, Switzerland.

I am Solution Manager of Swiss company call Processlynx with 25 persons over my management from different countries and languages > 8, with creation of Solutions (SharePoint, Office 365, Web Base Solutions and Mobile). I have multiple SharePoint certifications and when the opportunity is provided and time I like to share my knowledge of this platform in my blog or SharePoint/Office 365 public events.

I am SharePoint MVP since 2010 and this give me opportunity to meet and talk with some many people around the world and new friends with different views, Business approaches and capabilities.

I also love to support the Office 365 team with the transi-



tion from Full trust code to Office 365 Apps creating samples in the office 365 practices and patterns (PnP).

SharePoint was the product on fire in 2006 and 2007 was the turning point to became the intranet Product, was crazy times and start to deeply search everything about the product and their features and every time more amaze what we could do, was a fully integrated solution with Microsoft Products and other platforms and saw the growth to the Cloud (Azure Platform and Office 365) with even more capabilities in 10 years it's just incredible a total revolution and transition on how we deal with IT.

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

Spectrum was definitely the turning point, I remember exploring a Spectrum magazine with some basic command and I created a circle with a line in the middle and planet Saturn was born in my TV screen, I automatically saw the potential in my hands.

After this event computers and IT continues to be part of my life and explorer with my father other commercial applications like Lotus 123 and making my first network connection with an Olivetti in the beginnings of 90's for stock management.

Constant investment in Business solution provide me the ability to have dipper knowledge of different business where my family was involve and provide me a lot of real world Human Behavior, you can have the best tool in the world but will not work by himself "yet".

¿Cuáles son tus principales actividades tecnológicas hoy en día?

Manage Multinational teams/projects with different worldwide projects and technologies involving SharePoint, Azure, Office 365 and developing of Solution like BI tools, Office 365 Apps and transition from full trust code to Office Apps, PKI Solutions and Office 365 security validation and Java Solutions.

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

Read about history, the best way to prepare the future is to read how people react with challenges and how it's always repeating, history it's all about properties, conditions and dependencies that affects all in every decision we take, can be personal life or IT.

¿Cuáles son tus hobbies?

I love to hike with my family in the Swiss Alps and bike in the countryside.

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

I can easily see in the next 5 years a huge growth in integration of Hybrid environments/Solutions and consolidation of Products created by ISV to be prepared for the

Cloud, with the integration of on premise solution with Web API connector and multiple Identities with security associated, in the Cloud Solution like Office 365 or Azure (Security validation will be critical and with huge potential grow of market with a huge leak of real experts).

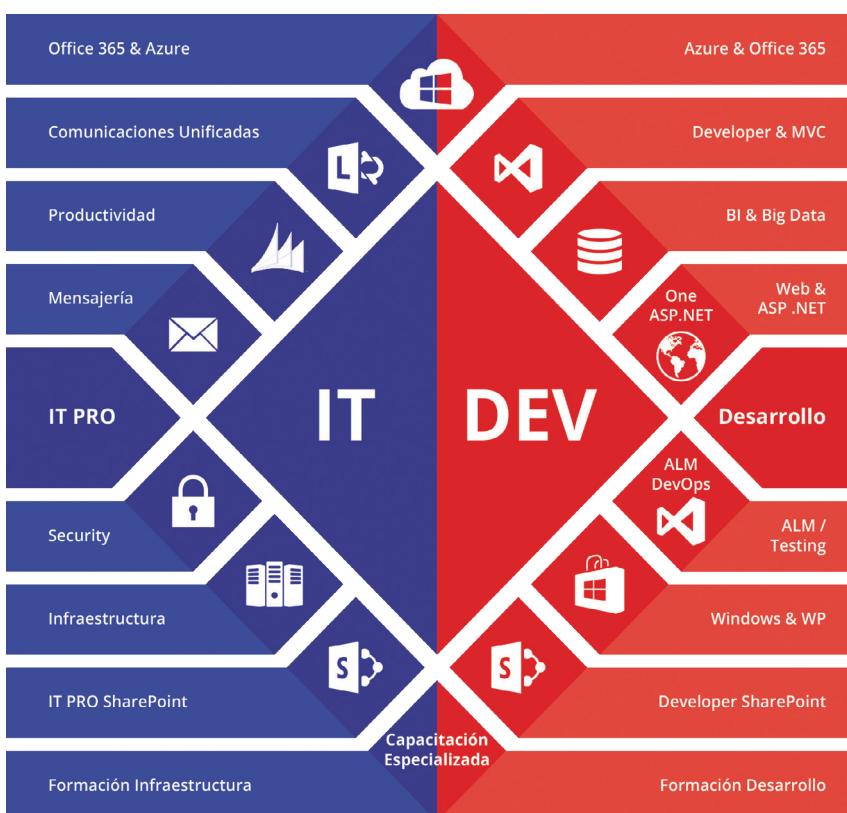
ANDRÉ LAGE

Microsoft SharePoint MVP

Blog: <http://aacleage.blogspot.ch/>

Twitter: aacleage

Expertos en Plataformas y Tecnologías Cloud & OnPremises



CLUSTER
a FiveShare IT Company



www.mvpcluster.es

i

—
24

SharePoint y Azure – Operational Insights

Muchos de los servicios ofrecidos por Azure se pueden utilizar para ampliar y mejorar el funcionamiento de SharePoint, tanto OnPremises como en la nube. Uno de los servicios que ofrece Azure es «Operational Insights». Operational Insights es una solución de Software como Servicio (SaaS) específicamente diseñada para ayudar a los grupos de infraestructura a recolectar, guardar y analizar datos de registros (logs) de virtualmente cualquier servidor Windows (y Linux), desde cualquier centro de datos o desde la nube. Con los datos almacenados es posible tomar decisiones en tiempo real sobre posibles problemas o mejoras de los equipos monitoreados.

El servicio permite recolectar y agregar información obtenida desde múltiples sistemas y centralizarla en un solo repositorio de datos para identificar fácilmente la causa de problemas operacionales. El servicio dispone de un sistema de paneles de control que se pueden configurar por medio de consultas a los datos almacenados, utilizando los paneles predefinidos, o creando nuevos a la medida.

Adicionalmente, Operational Insights permite identificar parches que no han sido instalados y análisis de virus que pueden haber accedido a los sistemas monitoreados. Otra posibilidad importante es la que el servicio ofrece para mantener un registro de los cambios introducidos en los sistemas, de tal forma que, si el equipo comienza a presentar conductas inesperadas, se puede determinar si un cambio de configuración realizado en el pasado reciente ha sido el culpable.

Muchos de los servicios ofrecidos por Azure se pueden utilizar para ampliar y mejorar el funcionamiento de SharePoint

Finalmente, el servicio permite analizar los sistemas bajo control para determinar si cumplen las buenas prácticas («Best Practices») recomendadas por Microsoft. SQL Server, SharePoint, Exchange y Hyper-V se pueden examinar de esta manera, y generar reportes que permitan mejorar la seguridad y el funcionamiento de los servidores.

Creación de un servicio de Operational Insights en Azure

Para comenzar a utilizar el servicio es necesario crear primero un Espacio de Trabajo (Workspace) de Operational Insights:

1. Desde el portal de administración de Azure (<https://manage.windowsazure.com/>), haga un login con sus credenciales.
2. Seleccione “Operational Insights” desde el menú vertical al lado izquierdo de la ventana.
3. Haga clic sobre “Create a workspace”. La ventana de “Quick Create” debe aparecer.



Imagen 1.- Creacion de un Workspace en Azure.

4. En “Account” puede seleccionar un Workspace que ya haya sido creado anteriormente, o crear uno nuevo. Asigne un nombre al Workspace (Azure comprueba que el nombre sea único). Seleccione el tipo de cuenta (“Tier”), la localización del centro de datos de Azure y la suscripción de Azure a utilizar. El “Tier” define el tipo de servicio que se va a recibir y su precio: “Free” es gratis y tiene un límite diario de transferencia de datos de 500MB y un periodo de retención de los datos de una semana; “Standard” y “Premium” son de pago, no tienen límite de transferencia y el periodo de retención es de un mes para el primero y un año para el segundo.
5. Después de utilizar el botón de “Create Workspace”, Azure crea y activa la cuenta, lo que puede durar algunos minutos.
6. Cuando el proceso de activación termine, haga clic sobre el nombre del Workspace y en la ventana que aparece, haga clic sobre “Visit your Operational Insights account”.

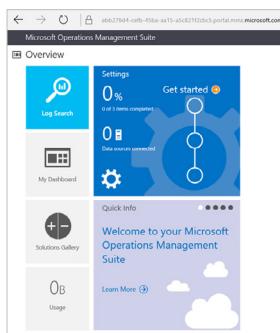


Imagen 2.- El cuadro de control del Workspace.

7. Clic sobre “Get started” en la nueva página.
8. La primera ventana del asistente permite seleccionar las operaciones que se van a monitorizar. Las opciones incluyen análisis de malware, estado de replicación de máquinas virtuales, análisis de updates, estado del backup, monitoreo de cambios y automatización de tareas repetitivas. Seleccione todas las opciones y utilice el botón de “Add selected Solutions”.

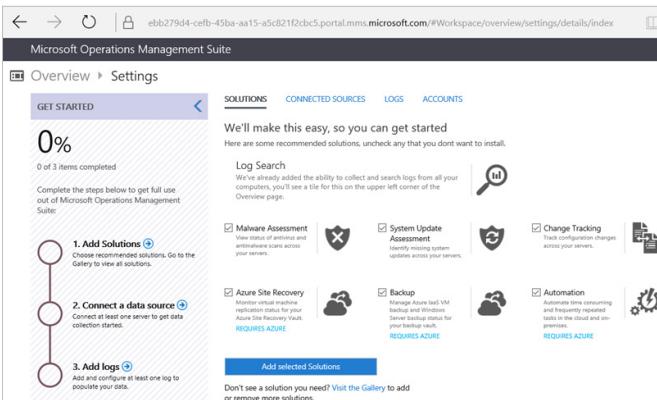


Imagen 3.- Soluciones para el Workspace.

9. Haga clic sobre “Connect a data source” en el menú vertical al lado izquierdo. Azure permite conectarse a diferentes tipos de fuentes de datos (sistemas a monitorizar): Servidores Windows de 32 y 64 bits (soporte para servidores Linux está planeado, pero no activo en el momento), Microsoft System Center Operations Manager o diferentes sistemas de datos de Azure. Como se desea conectar a Operational Insights con un servidor de SharePoint OnPremises, utilice el botón de “Download Windows Agent (64 bit)” y guarde el ejecutable localmente (MMASetup-AMD64.exe).

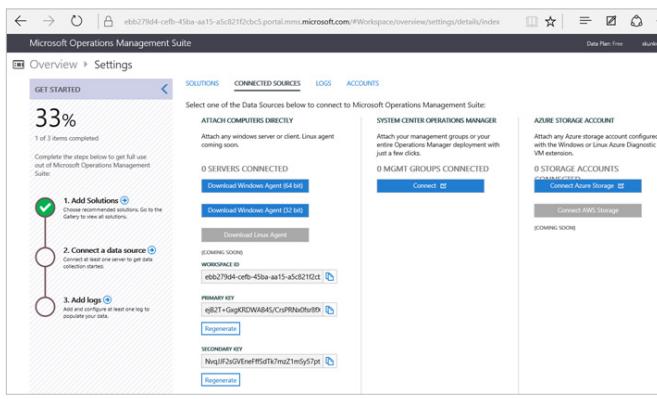


Imagen 4.- Fuentes de datos para el Workspace.

11. Haga clic sobre “Add logs” en el menú. Seleccione la casilla “Collect W3C format IIS log files” y utilice el botón “Save” (borde inferior de la ventana).

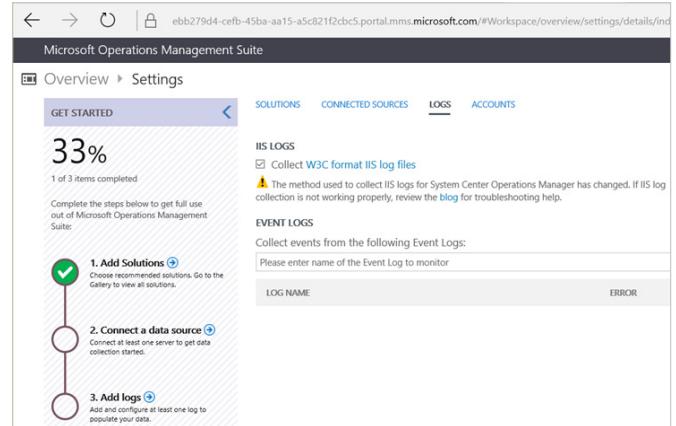


Imagen 5.- Log para monitorear en el Workspace.

Configuración de Operational Insights en un servidor de SharePoint (OnPremises)

El agente que se descargó en el punto 9 de la sección anterior se debe instalar en todos los servidores de la granja de SharePoint OnPremises.

1. Copie el archivo descargado en cada uno de los servidores de la granja, y ejecútelo localmente.
2. Acepte las condiciones de uso del agente.
3. Seleccione un directorio para instalarlo, o acepte el por defecto.
4. En la ventana de “Agent Setup Options” del asistente, seleccione “Connect the agent to Microsoft Azure Operational Insights” (“Next”).

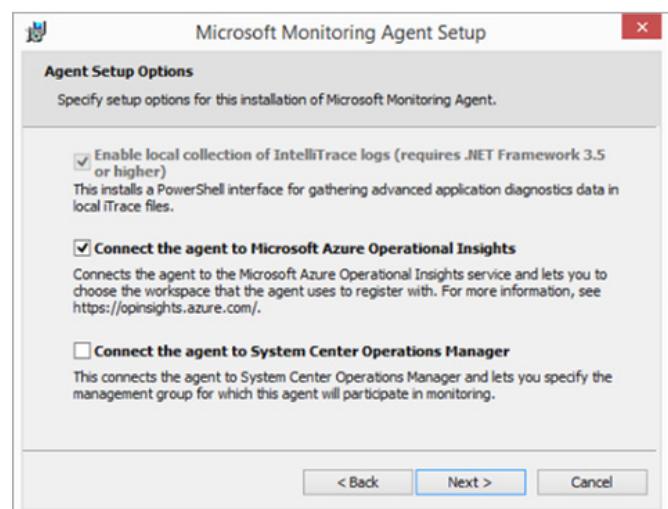


Imagen 6.- Configuración del agente.

5. Regrese al administrador de Azure, segundo punto (Figura 4) y copie los valores de “Workspace ID” y “Workspace Key” (utilice el valor de “Primary Key”) en las casillas correspondientes. El botón de “Advanced” permite configurar un proxy si la granja de Sha-

rePoint lo está utilizando.

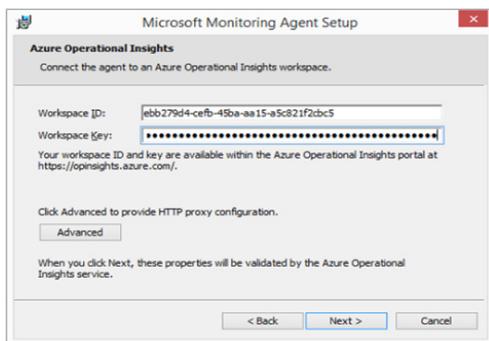


Imagen 7.- Configuración del Workspace en el agente.

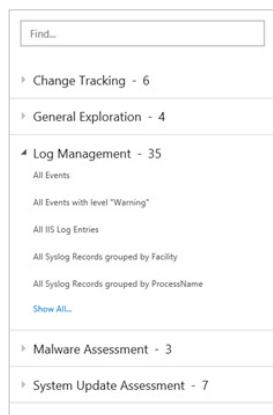


Imagen 9.- Baldosas pre-configuradas.

1. La siguiente ventana del asistente permite configurar si se desea actualizar el agente por medio de los parches mensuales de Microsoft o no. Recomendado es utilizar la opción de actualizar.
2. En la última ventana del asistente revise los datos configurados e instale el agente.
3. Abra el Panel de Control de Windows. Una nueva sección “Microsoft Monitoring Agent” ha sido instalada allí. Desde las ventanas de esta sección se puede configurar de nuevo el agente si es necesario, y se puede confirmar que está conectado a Azure.
4. Los servidores están ahora conectados con Operational Insights en Azure, pero los agentes instalados demoran algunas horas recopilando los primeros datos del monitoreo y enviándolos para análisis.

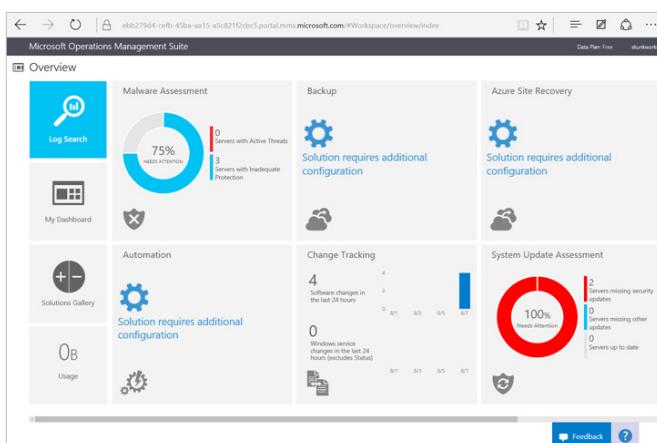


Imagen 8.- Consola de manejo de Operational Insights.

Configuración de Paneles de Control

La consola de manejo de Operational Insights ofrece acceso directo a algunos paneles de control, como por ejemplo el monitoreo de malware y de updates. Pero es posible crear varios otros tipos, más adaptados a las necesidades de monitoreo empresariales. Desde la consola seleccione “My Dashboard”. El panel de control (“Dashboard”) está compuesto de baldosas (“Tiles”), una por cada tipo de monitor que se desea configurar. Para crear nuevas baldosas, utilice el botón “Customize” en la parte inferior de la página, el que abre el menú de baldosas por defecto:

Seleccione y mueva el elemento al dashboard; cada elemento se convertirá en una baldosa individual, mostrando los datos monitoreados hasta el momento:

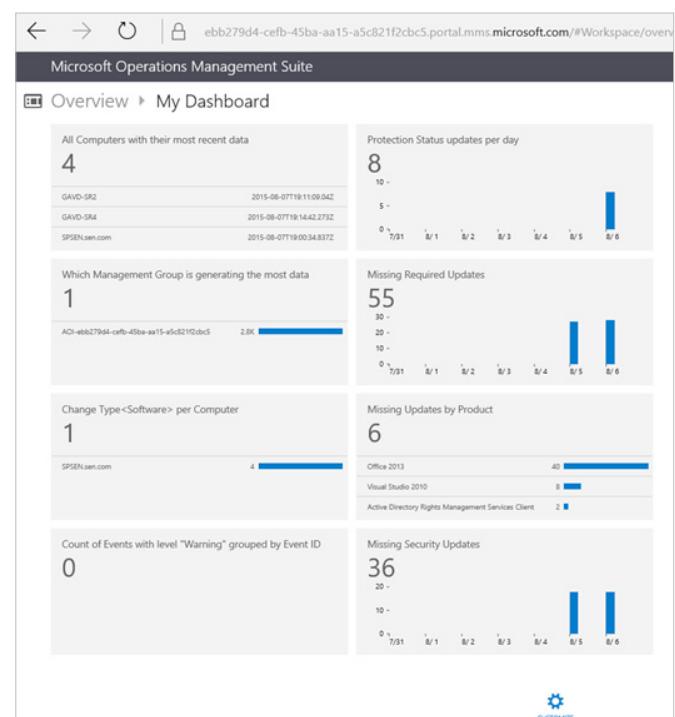


Imagen 10.- Dashboard con baldosas pre-configuradas.

Cuando hace clic sobre alguna de las baldosas, se abre una ventana con información detallada sobre las estadísticas del elemento:

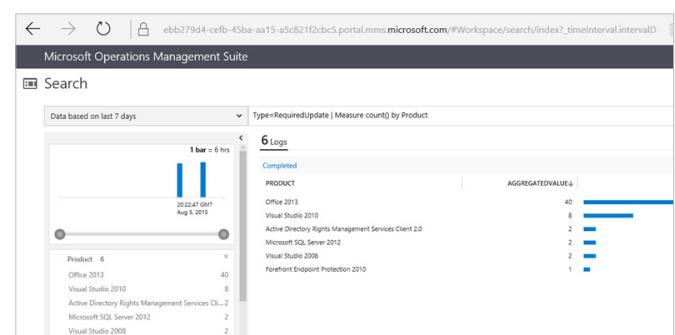


Imagen 11.- Drilldown de una baldosa específica.

Utilizando el botón de “Customize” de nuevo, se puede seleccionar cada baldosa y modificar su forma de presentar

la información o removerlo totalmente. Hay dos formas de presentación: por medio de gráficas, o por medio de números. Esta última forma de visualización permite también definir límites superior e inferior, de tal forma que la baldosa funcione como un KPI:

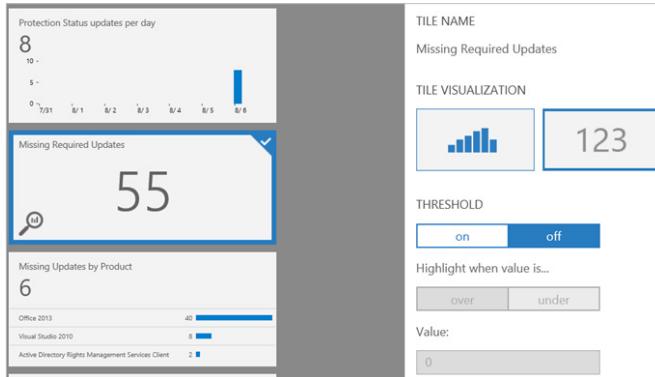


Imagen 12 - Configuración de una baldosa

Finalmente, las baldosas se pueden organizar en el dashboard moviéndolas con el ratón.

Búsqueda

La consola de Operational Insights tiene un sistema de búsqueda bastante poderoso que permite filtrar la información almacenada de forma rápida y precisa. Para iniciar una búsqueda, utilice el botón con la lupa ("Log Search") en la página principal de la consola. Una nueva página abre con una casilla en donde se puede crear la consulta. Por ejemplo, utilizando la consulta "Type=ConfigurationChange" se pueden ver todos los cambios de configuración registrados en todo el tiempo de monitoreo almacenado:

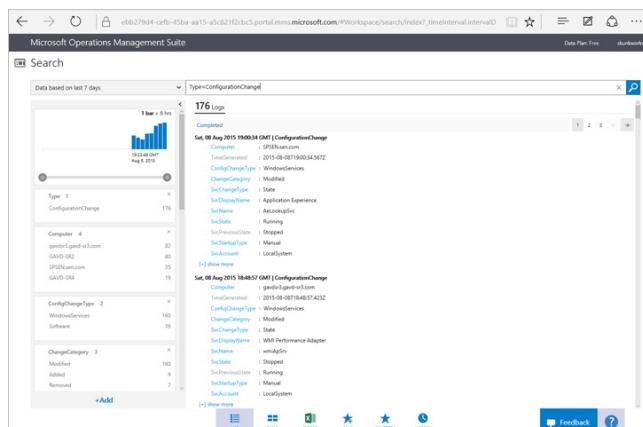


Imagen 13.- El sistema de búsqueda en acción.

Las consultas se pueden refinar: por ejemplo, para encontrar cuando se registró el último cambio de configuración en cada uno de los computadores monitoreados, la consulta se puede filtrar como “Type=ConfigurationChange | Measure Max(TimeGenerated) by Computer”

Las consultas de búsqueda pueden ser bastante complejas. En MSDN se puede encontrar un documento que menciona todos los parámetros posibles de utilizar y una descripción de la sintaxis para las consultas (<https://azure.microsoft.com/en-us/documentation/articles/operational-insights-search/>).

Automatización por medio de PowerShell

Azure dispone de una interface REST (Azure Resource Manager API, ARM) que se puede utilizar para consultar todos los datos monitoreados por Operational Insights. Para utilizar PowerShell como consola Shell con Azure ARM es recomendable utilizar alguno de los varios programas de ayuda que instalan los necesarios cmdlets en PowerShell, como por ejemplo “ARMClient” (<https://github.com/projectkudu/ARMClient>) o “ARMPowerShell” (<https://github.com/dushyantgill/AzureResourceManagerPowerShell/archive/master.zip>). Una vez instalado el último programa mencionado, para logearse en Azure ejecute el comando “Connect-ARM” desde una consola de PowerShell. El comando abre una ventana en donde se puede escribir el login y clave de Azure y realiza la conexión con el sistema.

El siguiente script muestra el ultimo error monitoreado por Operational Insights:

```
$subscriptionID = “[Subscription ID]”
$workSpace = “[Nombre Workspace]”
$baseBusqueda = “/subscriptions/$subscriptionID/
resourcegroups/OI-Default-West-Europe/providers/microsoft.
operationalinsights/workspaces/$workSpace/search”
$myConsulta = “Error”
$myResultado = Execute-ARMQuery -SubscriptionId
$subscriptionID -HTTPVerb Post -Base $baseBusqueda -Data
@{Query=$myConsulta} -APIVersion “2015-03-20”
$myResultado
```

Los valores de “\$subscriptionID” y “\$workSpace” se pueden encontrar en la página de Dashboard de Azure (seleccione la pestaña de “Operational Insights” y luego haga clic sobre el nombre del Workspace y utilice la pestaña de “Dashboard”).

La consola de manejo de Operational Insights ofrece acceso directo a algunos paneles de control, como por ejemplo el monitoreo de malware y de updates

El resultado se puede ver directamente en PowerShell:

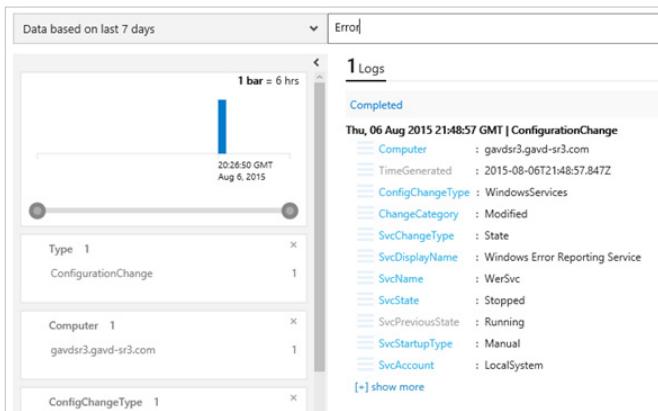
```
Administrator: Windows PowerShell
PS C:\> $myResultado

ComputerName : gavids3.gavid-pc.com
ManagementGroupName : AOT-elb279d4-cefb-4bfa-a115-a5c821f7c5c
SourceSystem : OpsManager
TimeGenerated : 2015-08-06T21:48:57.847Z
ConfigChangeType : WindowsServices
ChangeCategory : Modified
SrvName : Windows Error Reporting Service
SrvDisplayName : WerSvc
SrvName : Stopped
SrvStatus : Running
SrvStartType : Manual
SrvType : LocalSystem
SrvPath : \Windows\System32\svchost.exe -k WerSvcGroup
SrvDescription : Allows errors to be reported when programs stop working or responding and allows existing
  services to be delivered. Also allows logs to be generated for diagnostic and repair services.
  If this service fails to start, programs might not work correctly and results of diagnostic
  services and repairs might not be displayed.
Id : 74d36423-ccb1-5ca4-fa6e-cdb898652b2c
Type : ConfigurationChange
__metadata : {{"@type": "ConfigurationChange", "TimeGenerated": "2015-08-06T21:48:57.847Z"}}

PS C:\>
```

Imagen 14 - El ultimo Error monitoreado con PowerShell

Que coincide perfectamente con el valor que se puede obtener por medio de la página de búsqueda de la consola de Operational Insights:



The screenshot shows a log entry from 'Operational Insights' with the following details:

- Type:** ConfigurationChange
- Computer:** gavdsr3.gavd-sr3.com
- ConfigChangeType:** 1
- Completed:** Thu, 06 Aug 2015 21:48:57 GMT | ConfigurationChange
- Properties:**
 - Computer : gavdsr3.gavd-sr3.com
 - TimeGenerated : 2015-08-06T21:48:57.847Z
 - ConfigChangeType : WindowsServices
 - ChangeCategory : Modified
 - SvcChangeType : State
 - SvcDisplayName : Windows Error Reporting Service
 - SvcName : WerSvc
 - SvcState : Stopped
 - SvcPreviousState : Running
 - SvcStartupType : Manual
 - SvcAccount : LocalSystem

Imagen 15.- El ultimo Error monitoreado con la consola de Operational Insights.

¿Es Operational Insights seguro?

Operational Insights recopila numerosa información de los servidores que monitoriza, en el caso que se discute aquí, servidores de una granja de SharePoint OnPremises. Algunos de esos datos pueden ser confidenciales, lo que plantea la pregunta si el servicio tiene los niveles de seguridad suficientes para mantener la información secreta y garantizar que no va a ser utilizada de ninguna forma fuera del control de la empresa. El esquema de seguridad utilizado por Microsoft es expuesto en la siguiente imagen (tomado de MSDN, <https://azure.microsoft.com/en-us/documentation/articles/operational-insights-security/>)

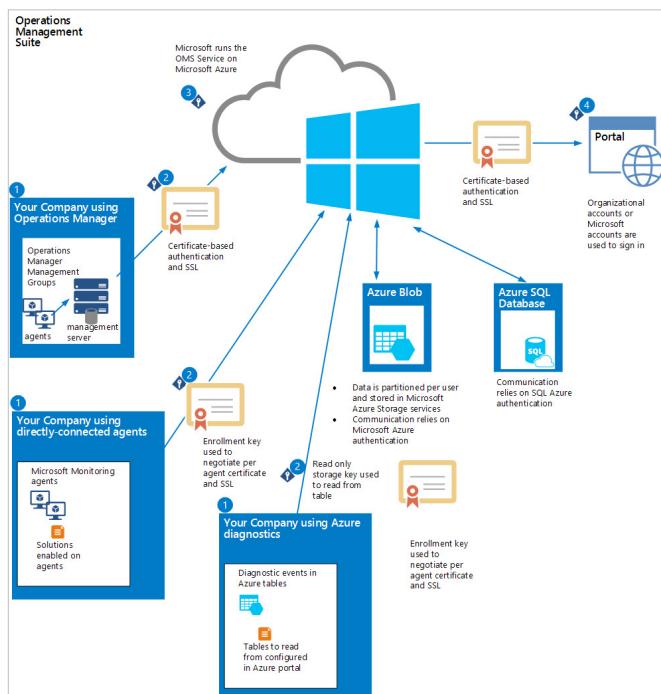


Imagen 16.- Esquema de seguridad de Operational Insights.

Todas las líneas de transferencia de información utilizan Security Layers de HTTP (canales HTTPS), tanto para la comunicación entre los servidores con Azure, como dentro de Azure mismo. Los datos se mantienen separados lógicamente dentro de las bases de datos utilizando un BLOB encriptado con claves separadas para cada cliente, y los períodos de retención están definidos en los contratos de Azure. Los centros de datos de Azure son mantenidos por Microsoft mismo, garantizando todo tipo de medidas de seguridad para el acceso físico y/o remoto. Además, Microsoft garantiza que los datos no se utilizarán para ningún otro objetivo fuera del monitoreo, y nunca serán vendidos comercialmente a otras empresas.

Conclusión

Un requisito indispensable cuando se crea una granja de SharePoint OnPremises es crear un sistema de monitorización que permita mantener información sobre el funcionamiento de los servidores. Operational Insights es un servicio de Azure que permite recopilar toda la información necesaria para garantizar esa monitorización, ofreciendo además herramientas para crear paneles de control, un motor de búsqueda para compendiar la información y acceso remoto por medio de un API REST.

Operational Insights se puede configurar fácilmente y comenzar a utilizar en cuestión de minutos. Azure ofrece varias posibilidades de retención de datos, variando desde la utilización del sistema totalmente gratis con retención por una semana, hasta períodos de retención de hasta un año (servicio de pago). Microsoft garantiza también la seguridad del sistema, su mantenimiento on-line y que los datos no serán utilizados para ningún otro fin que no sea monitoreo por parte del propietario.

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

Mentoring



Comparti **MOSS**

Un servicio experto alrededor de su SharePoint



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

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



Generando máquinas como churros: Azure DevOps y Powershell Desired State Configuration (DSC)

Al hilo de los dos últimos artículos de mi blog, sobre Azure Automation Service, descubrí una funcionalidad que se me había pasado por alto hasta ahora, llamada Desired State Configuration o DSC. Esta funcionalidad, que forma parte del estándar de Powershell 4.0, y está disponible desde Windows Server 2012 R2, permite a los servidores realizar un auto-aprovisionamiento durante el despliegue y/o un auto-arreglo cuando su configuración se desvía del estándar marcado. Esta funcionalidad está disponible como descarga gratuita para Windows Server 2008 R2 y 2012 dentro del Windows Management Framework 4.0.

Esta funcionalidad, que a priori pudiera resultar poco atractiva, es muy útil cuando pensamos en el mundo de la nube, donde la velocidad y fiabilidad de aprovisionamiento de máquinas que cumplan determinados estándares es altamente requerida. DSC proporciona un framework estándar para definir y gestionar cambios de configuración, y proporciona una solución que puede ser replicada para gestionar grandes sistemas. ¡Todo son ventajas! Con esta funcionalidad podemos desde configurar roles y características, tanto en la plataforma Hyper-V, como en Azure (¡menos mal! Si no, no estaría escribiendo este artículo).

Con DSC se puede gestionar máquinas Windows Server tanto físicas, virtuales en hosts, como en plataformas de nube... digamos que solo en la de Azure y además, puede de ser utilizado más allá de la plataforma Microsoft. Está construido sobre el estándar Desktop Management Task Force (DMTF), como el Managed Object Format (MOF) y el protocolo de servicios WS-Management. De esta manera, se integra de forma nativa con herramientas de terceros como Puppet, Chef y otras herramientas de gestión de la configuración del mundo Linux... y ahora también usadas como agentes en las máquinas virtuales de Azure.

Prerrequisitos para DSC

Para poder hacer los ejemplos del presente artículo es necesario descargar e instalar los siguientes complementos:

- Powershell DSC Resource Kit: Este kit proporciona 40 nuevos módulos para DSC y proporciona recursos para configurar Directorio Activo, SQL Server, IIS, Hyper-V y otros. Puedes descargarlo desde <https://gallery.technet.microsoft.com/DSC-Resource-Kit-All-c449312d>, aunque lo están subiendo a GitHub
- Módulo Microsoft Azure Powershell: En este artículo

lo utilizaremos para subir fichero DSC de configuración a una cuenta de almacenamiento desde la que enlazarlo a las máquinas virtuales recién creadas

Programación Declarativa

¿Qué es esto de Declarativa? En pocas palabras, es que en lugar de picarnos unos inacabables scripts de Powershell, vamos a declarar un estado final deseado para un servidor o conjunto de servidores, de forma que sea el propio sistema en encargado de “alcanzar” este estado. Cuando se le asigna un estado deseado a un servidor, este automáticamente compara su estado actual con el estado deseado y realiza todos los cambios de configuración necesarios para alcanzar el estado deseado, sin necesidad de escribir código. Cómodo ¿no?

Esto se consigue por medio de una nueva palabra reservada en Powershell 4.0, Configuration, que es usada para definir estados. La estructura sintáctica es muy similar a la de una función PowerShell, por lo que es fácil entender y seguir la sintaxis. Por ejemplo, a continuación, vamos a escribir la configuración para un servidor de SharePoint.

```
Configuration SharePointHost # Define the configuration name
{
    param ($NodeName) # Accept server names as parameters
    Node $NodeName # Begin block of configuration settings for
    each assigned server
    {
        #Install the Net Framework 4.5
        WindowsFeature Net-Framework-Features
        {
            Ensure = "Present" # Can be "Present" or "Absent"
            Name = "Net-Framework-Features"
        }
        #Install the Web Server
        WindowsFeature Web-Server
        {
            Ensure = "Present" # Can be "Present" or "Absent"
            Name = "Web-Server"
        }
        WindowsFeature Web-Common-Http
        {
            Ensure = "Present" # Can be "Present" or "Absent"
            Name = "Web-Common-Http"
        }
        WindowsFeature Web-Static-Content
        {
            Ensure = "Present" # Can be "Present" or "Absent"
            Name = "Web-Static-Content"
        }
        WindowsFeature Web-Default-Doc
        {
            Ensure = "Present" # Can be "Present" or "Absent"
            Name = "Web-Default-Doc"
        }
        WindowsFeature Web-Dir-Browsing
        {
```

```

        Ensure = "Present" # Can be "Present" or "Absent"
        Name = "Web-Dir-Browsing"
    }
WindowsFeature Web-Dir-Browsing
{
    Ensure = "Present" # Can be "Present" or "Absent"
    Name = "Web-Dir-Browsing"
} # ... completar las que faltan. El espacio en la revista no es
infinito!!!
}

```

Este ejemplo (si lo completáis) asegura que todo servidor provisionado cumple con los requisitos necesarios para desplegar un servidor de SharePoint, aunque no solo se pueden añadir características. Se pueden incluir recursos para manejar los siguientes conceptos:

- Habilitar o deshabilitar roles y características.
- Gestionar configuraciones de Registro.
- Gestionar Archivos y Características.
- Arrancar, parar y gestionar proceso y servicios.
- Gestionar usuarios locales, y grupos.
- Desplegar nuevos paquetes de software.
- Gestionar variables de entorno.
- Ejecutar Scripts de PowerShell.

Una vez hayamos definido la configuración DSC, tenemos que ejecutar dos pasos para asignarla a los servidores en la que queramos ejecutarla:

1. Compilar la configuración definida en un conjunto de fichero MOF para cada servidor. Esto es posible realizarlo ejecutando la configuración, como si fuera una llamada a una función

```
# Compile MOF files for each assigned server
SharePointHost -NodeName "SP2013SV1","SP2013SV12"
```

2. Usar el cmdlet Start-DSCConfiguration para copiar el archivo MOF a cada uno de los servidores e invocar la configuración deseada

```
# Copy MOF file to node(s) and invoke the configuration
Start-DSCConfiguration -Path .\SharePointHost -Wait -Verbose
-Force
```

Cuando se lanza este comando, los ficheros MOF son copiados localmente a cada servidor y el motor Local Configuration Manager (LCM) los procesa. Como parte del proceso, el LCM va a hacer los cambios necesarios para que cada servidor designado cumpla con la configuración expresada en el archivo.

Con DSC se puede gestionar máquinas Windows Server tanto físicas, virtuales en hosts, como en plataformas de nube

Que usar, cuando y donde

Hay tantas piezas que podemos utilizar para automatizar nuestros despliegues de IaaS (hablando del mundo Cloud, por ser el más dinámico y el más relacionado conmigo) que en ocasiones hay que decidir entre varias alternativas cual es la más adecuada para un determinado escenario. Sin querer ser dogmático aquí hay una clasificación que puede ser válida:

- Orquestar todo el proceso de aprovisionamiento con Azure Automation.
- Aprovisionar recursos de cloud fabric con PowerShell Workflow Activities usando Azure Runbooks.
- Configurar el sistema operativo y requisitos de aplicación dentro de las máquinas virtuales de Azure con PowerShell DSC.
- Publicar las configuraciones de Powershell DSC a una cuenta de Azure Storage.
- Usar los Agentes de Azure VM para aplicar las configuraciones de Powershell DSC.

Otra herramienta muy interesante es el Azure Resource Management (ARM) para manejar subconjuntos de recursos relacionados con dependencias entre sí.

Ejemplo práctico: Aprovisionamiento extremo a extremo en Azure

Vamos a hacer algo con todo lo que hemos visto. Partimos de una suscripción de Azure en la que tenemos al menos:

- Un grupo de afinidad: xxxdscag001 (substituir xxx por las iniciales de cada uno).
- Una cuenta de almacenamiento: xxxdscstor001.
- Red virtual: xxxdscnet001 (en el ejemplo, espacio 172.16.x.x).
- Servidor de AD en una máquina virtual: xxxdscad001 (en el ejemplo, 172.16.0.4).
- Una cuenta de Azure Automation (xxxautomation001).

Todos ellos configurados preferentemente en el mismo cpd. En el ejemplo Oeste de Europa.

fevdscstore001	Cuenta de almacenamiento	✓ En línea	Microsoft Partner Network	Oeste...
fevdscnet001	Red virtual	✓ Creado	Microsoft Partner Network	Norte...
fevdscad001	Máquina virtual	✓ Creado	Microsoft Partner Network	Oeste...
fevdscad001	Servicio en la nube	✓ Ejecutándose	Microsoft Partner Network	Oeste...
fevautomation001	Cuenta de automatización	✓ Listo	Microsoft Partner Network	Oeste...

Imagen 1.- Elementos necesarios en la suscripción de Azure.

En primer lugar, vamos a crear un nuevo Workflow en Azure Automation, al que denominaremos New-AzureDSCProvision, y al cual dotaremos de una serie de parámetros que nos servirán posteriormente para que podamos reutilizar el proceso. Estos parámetros, en los casos en los que el flujo se lance de forma manual, se pedirán por pantalla. En caso de ser lanzado de forma automatizada se los deberemos proporcionar. El código inicial es el siguiente:

```
workflow New-AzureDSCProvision
{
    param(
        [parameter(Mandatory=$true)]
        [String]
        $subscriptionName,
        [parameter(Mandatory=$true)]
        [String]
        $deploymentPrefix,
        [parameter(Mandatory=$true)]
        [Int]
        $numberOfVMs,
        [parameter(Mandatory=$true)]
        [String]
        $vmInstanceSize,
        [parameter(Mandatory=$true)]
        [String]
        $vmDomain,
        [parameter(Mandatory=$true)]
        [String]
        $vmDomainNetBIOS
    )
}
```

Así mismo, en el flujo introduciremos una serie de variables que utilizaremos para crear todas las máquinas.

```
# Obtener credenciales Azure
$cred = Get-AutomationPSCredential -Name
<AzureAutomationAccount>
# Obtener credenciales del administrador de las
máquinas
$vmAdmin = Get-AutomationPSCredential -Name
'AzureAdmin'

# Variables comunes
$storageName = $deploymentPrefix + 'store001'
$vmServiceName = $deploymentPrefix + 'app'
$affinityGroupName = $deploymentPrefix + 'ag001'
$availabilitySetName = $deploymentPrefix + 'as'
$vNetName = $deploymentPrefix + 'net001'
$subnetName = 'Subnet-1'
$dscArchive = 'ADSCSharePointHost.ps1.zip'
$dscConfigName = <SharePointHost>
$vmInstanceSize = <Large>
```

muy interesante es el Azure Resource Management (ARM) para manejar subconjuntos de recursos relacionados con dependencias entre sí

A continuación, vamos a introducir un bucle para crear todos los recursos necesarios (en este caso, las máquinas virtuales). Como la creación de recursos puede ser una tarea que se alargue en el tiempo, y para poder retomar el flujo en caso de error o suspensión del mismo, es conveniente utilizar los Workflow-CheckPoint, que nos permitirán retomar por donde lo dejamos. Así mismo, para la creación de las máquinas utilizaremos un bloque de inline-script, que se ejecutará de forma atómica

```
for ($i=1; $i -le $numberOfVMs; $i++)
{
    # Salvar estado del workflow
    Checkpoint-Workflow

    # Conectar a la suscripción de Azure
    Add-AzureAccount -Credential $cred
    Select-AzureSubscription `
        -SubscriptionName $subscriptionName

    # Fijar la cuenta de almacenamiento
    Set-AzureSubscription `
        -SubscriptionName $subscriptionName `
        -CurrentStorageAccountName $storageName

    # Aprovisionar VM
    InlineScript {
        $currentVM = "{0:D2}" -f $Using:i
        $vmName = $Using:deploymentPrefix + 'app' +
        $currentVM
        $vmImage = @((Get-AzureVMImage |
            Where-Object Label -like
            "Windows Server 2012 R2 Datacenter*").ImageName)[-1]
        $vmAdmin = $Using:vmAdmin

        # Especificar nombre, imagen y tamaño
        $vm = New-AzureVMConfig `
            -Name $vmName `
            -ImageName $vmImage `
            -InstanceSize $Using:vmInstanceSize

        # especificar administrador local y credenciales
        para unirla al dominio
        $vm = Add-AzureProvisioningConfig `

            -VM $vm `
            -WindowsDomain `

            -AdminUserName $vmAdmin.Username `
            -Password $vmAdmin.GetNetworkCredential().Password `

            -JoinDomain $Using:vmDomain `
            -Domain $Using:vmDomainNetBIOS `

            -DomainUserName $vmAdmin.Username `
            -DomainPassword `

            $vmAdmin.GetNetworkCredential().Password

        # Especificar subnet para las VM
        $vm = Set-AzureSubnet `

            -VM $vm `
            -SubnetNames $Using:subnetName

        # Especificar Availability Set
        $vm = Set-AzureAvailabilitySet `

            -VM $vm `
            -AvailabilitySetName $Using:availabilitySetName

        # Provisionar la máquina con la configuración
        especificada
        New-AzureVM `

            -VMs $vm `
            -ServiceName $Using:vmServiceName `
            -VnetName $Using:vNetName `
            -AffinityGroup $Using:affinityGroupName `
            -WaitForBoot
    }
}
```

Y hasta aquí podemos guardarlo, y probarlo. Ahora mismo estamos en la situación en la que se van a crear las máquinas configuradas como nosotros creemos, pero aún no les hemos aplicado la configuración que nosotros queremos. Para cerrar el ejemplo, vamos a crear un fichero DSC con

la especificación de máquinas para granja de SharePoint, de forma que sea tan solo instalar y configurar (que joh casualidad! También podríamos hacerlo por PowerShell ☺)

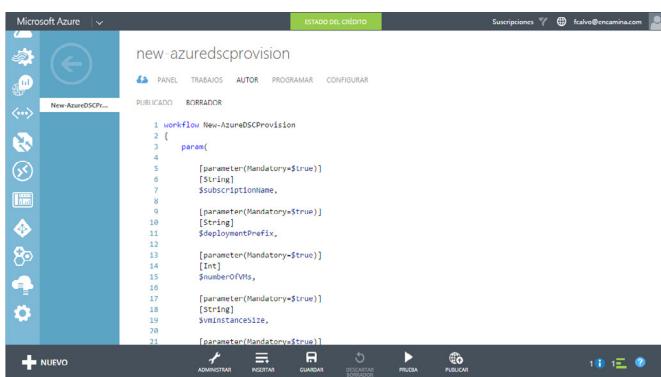


Imagen 2.- Workflow creado.

Para ello, los pasos a seguir van a ser un poco diferentes de los pasos que vimos en la parte inicial del artículo, ya que no disponemos de las máquinas creadas para generar los archivos MOF, ni podemos aplicarlos con Start-DSCConfiguration.

En primer lugar, cogeremos el fichero que creamos en el primer ejemplo del presente artículo, y lo salvaremos como fichero de PowerShell Script de nombre ADSCSharePointHost.ps1

A continuación, subiremos el fichero a una cuenta de almacenamiento desde la que podamos enlazar por medio de Azure Automation este archivo a las máquinas que vamos a crear con el Workflow. Para ello, vamos a ejecutar el siguiente script en una ventana de PowerShell ISE:

```

# Autenticar contra Azure
Add-AzureAccount
# Seleccionar Subscripción y Cuenta de almacenamiento
$subscriptionName = (Get-AzureSubscription).SubscriptionName | Out-GridView ` -Title "Selecciona la subscripción" ` -PassThru

Select-AzureSubscription ` -SubscriptionName $subscriptionName

$storageName = (Get-AzureStorageAccount).StorageAccountName | Out-GridView ` -Title "Selecciona la cuenta de almacenamiento" ` -PassThru

Set-AzureSubscription ` -SubscriptionName $subscriptionName ` -CurrentStorageAccountName $storageName
# Publicar el fichero de configuración a la cuenta de almacenamiento
Publish-AzureVMDscConfiguration ` -ConfigurationPath .\ADSCSharePointHost.ps1 ` -Force

```

Durante la publicación, el fichero será convertido en un archivo .zip, junto con los módulos importados mediante Import-Module dentro del bloque de configuración. Por defecto, este fichero se dejará dentro de un contenedor denominado Windows-PowerShell-dsc, y ahí es donde deberemos ir a buscarlo.

Por último, para aplicarlo a las máquinas que creamos con el Workflow de Azure Automation, deberemos introducir en el script anterior, justo antes de la provisión de las máquinas, la asignación del fichero subido a la cuenta de almacenamiento como fichero de DSC mediante el siguiente trozo de script

```

# Especificar la configuración DSC de la VM
$vm = Set-AzureVMDSCExtension ` -VM $vm ` -ConfigurationArchive $Using:dscArchive ` -ConfigurationName $Using:dscConfigName

# Provisionar la máquina con la configuración especificada

```

Y ahora solo falta lanzar el Workflow y comprobar que, efectivamente, todas las máquinas provisionadas tienen todos los roles y características necesarias para montar una granja de SharePoint sobre ellas.

Conclusiones

A lo largo de este artículo hemos visto cómo podemos utilizar las herramientas de automatización a nuestro alcance para hacer nuestra vida, en el caso de que seamos administradores de sistemas, mucho más sencilla. Escenarios como el tener que provisionar unos entornos de desarrollo idénticos para un equipo grande, resultan mucho más sencillos, y con el ejemplo desarrollado, simplemente consistiría en cambiar nuestro fichero DSC, y trabajando una configuración en la que incluyéramos el despliegue desatendido de todas las herramientas que nos soliciten.

En todo caso, y para aquellos de vosotros que estéis interesado en este tipo de soluciones, os recomiendo que empiecéis por el curso de Microsoft Virtual Academy llamado Automating the Cloud with Azure Automation. Os gustará, y os abrirá la mente a nuevas oportunidades... ¿para qué trabajar pudiendo darle a un botón?

FABIÁN CALVO

Gerente Soluciones Cloud & Pseller

fcalvo@encamina.com

@fcvspain

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

<http://www.encamina.com>

i

35

Aplicaciones empresariales en Office 365

Un gran número de empresas que usan SharePoint como intranet corporativa, terminan implementando algún tipo de escritorio virtual o concentrador de aplicaciones. No todas las aplicaciones tienen cabida en SharePoint, pero sí que podemos usar este como herramienta de distribución y/o autenticación de aplicaciones externas.

Office 365 ha tenido en cuenta esta necesidad funcional y hace algunos meses liberaron una versión con un concentrador de aplicaciones llamado “Iniciador de aplicaciones”, y lo podemos encontrar en el menú superior izquierdo de cualquier aplicación de Office 365.

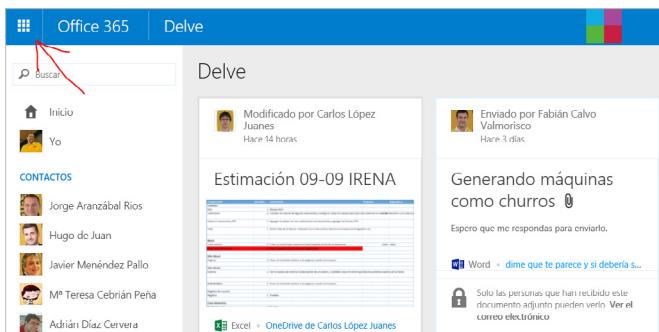


Imagen 1.- Menú del iniciador de aplicaciones.

Cuando abrimos el menú, accederemos a las aplicaciones que tenemos ancladas al más puro estilo Windows Phone.



Imagen 2. Aplicaciones principales del iniciador de aplicaciones.

Office 365 se ha convertido en el centro de trabajo para los usuarios de las empresas que lo usan

Donde en el enlace de “Mis aplicaciones” tenemos todas las aplicaciones que se han publicado para mi usuario en Office 365.

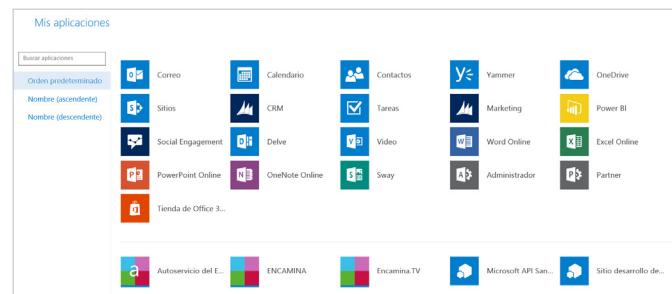


Imagen 3.- Todas las Aplicaciones del iniciador de aplicaciones.

¿Qué tipo de aplicaciones puedo añadir?

De momento, se pueden añadir enlaces personalizados, aplicaciones (AddIns) de SharePoint y/o aplicaciones de Azure.

Los enlaces permiten crear mosaicos para todos los usuarios de la organización a modo de ícono con enlace a una URL, por ejemplo, a la web pública de la compañía.



Imagen 4.- Administración de enlaces de Office 365

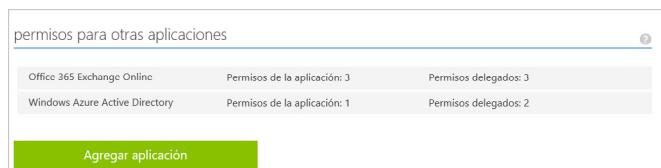
Si ser más que un enlace, puede resultar útil poner aquellos vínculos relevantes para los usuarios de la organización, aunque lo que nos puede hacer madurar como empresa es implementar nuestras aplicaciones como AddIns de SharePoint o integradas en el Directorio Activo de Azure de nuestro Office 365.

¿Qué nos aporta un AddIn de SharePoint o una aplicación del Directorio Activo de Azure?

Básicamente un salto hacia el nivel empresarial que necesitan nuestras aplicaciones. Con este tipo de aplicaciones tenemos autenticación integrada y el acceso a toda la API de Office 365.

Esto nos permite desarrollar nuestras aplicaciones usando la misma autenticación que usa SharePoint o Exchange Online con lo que podemos ofrecer a nuestros usuarios una experiencia de inicio de sesión que ya usan y conocen, además de poder administrar esas cuentas desde un único punto centralizado.

Con la API podemos integrar nuestras aplicaciones con el calendario de Outlook, con el correo electrónico o con los documentos de OneDrive, entre otras funcionalidades, usando el mismo mecanismo de autenticación y habilitando el acceso a estas capacidades al registrar o instalar la aplicación, bien el Directorio Activo de Azure o en SharePoint.



Office 365 Exchange Online	Permisos de la aplicación: 3	Permisos delegados: 3
Windows Azure Active Directory	Permisos de la aplicación: 1	Permisos delegados: 2

Agregar aplicación

Imagen 5.- Administración de permisos de la aplicación.

Si bien ya conocemos el modelo de Aplicaciones (AddIns) de SharePoint, del que puedes leer multitud de artículos en la revista, creo que el modelo que se nos presenta en Office 365 tiene mucho más sentido. Ofrecer aplicaciones seguras desde nuestro tenant con permisos para acceder a SharePoint o Exchange, nos abre un amplio abanico de posibilidades que una Aplicación de SharePoint no nos

ofrece y que está algo limitada al ámbito del sitio donde se instala.

Para todas las aplicaciones o procesos que desarrollamos habitualmente en SharePoint, con un modelo de desarrollo que se ha quedado en OnPremises, nos permite desarrollar aplicaciones nativas para móviles, aplicaciones de escritorio o aplicaciones web, usando las mejores prácticas de desarrollo y los Frameworks actuales.

¿Por qué no pensar en una intranet implementada en ASP.NET MVC que use la gestión documental y los sitios de colaboración de SharePoint, junto a los grupos de Office 365 y los nuevos blogs de los usuarios? Office 365 nos ofrece esa capa de servicio para que nuestras aplicaciones hagan uso de los mejores sistemas de colaboración y productividad empresarial como SharePoint o Exchange.

En próximos artículos entraremos más en detalle de cómo implementar una aplicación con el Directorio Activo de Azure que pueda conectarse a Office 365.

ALBERTO DIAZ MARTIN

MVP SharePoint

adiazcan@hotmail.com

@adiazcan

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

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



15

Octubre

Santiago, Chile

Hotel Neruda

Credentials and Examples

- Industry Experience
- Focus

Te invitamos a participar de este evento junto a los más destacados expertos, colaboradores y clientes de tecnologías Microsoft.

www.simposiosharepoint.com



Por un SharePoint Sostenible – Controla tus listas y bibliotecas, Conjunto de documentos.

Me podría permitir la licencia de decir que la gran mayoría de las veces nuestras aplicaciones de SharePoint se van ralentizando, o en casos más extremos directamente sufriremos caídas del sistema de forma repentina.

Evidentemente pueden existir muchos factores para provocar estos síntomas, como un mal gobierno de nuestra granja, una actualización inesperada...; pero yo quiero centrarme en un motivo muy concreto y es “el desbordamiento de nuestra información”. Me voy a explicar, me refiero a que nuestras listas o bibliotecas empiezan a superar los límites de elementos recomendados por Microsoft, y esto provoca que nuestras consultas o no funcionen o vayan muy lentas, lo que provoca una sensación de ralentización de nuestras aplicaciones.

En este artículo vamos a recordar cuales son nuestros límites, y algunas pautas a tener en cuenta cuando nos planteemos montar una aplicación sobre listas de SharePoint 2013 OnPremises, que ya podemos anticipar que si lo hacemos bien vamos a conseguir una aplicación con un rendimiento óptimo.

Gestiona y controla Tú la información

A partir de ahora nos referiremos al límite máximo de nuestras listas como “umbral”, y vamos a tener en cuenta los siguientes umbrales de lista/biblioteca de SharePoint antes de analizar las opciones para gestionar nuestra información:

- Límite de elementos por vistas: 5.000 (20.000 para usuarios administradores).
- Límite máximo de columnas de búsqueda por lista: 12.
- Límite recomendado de permisos exclusivos por ítem: 5.000.

Cuando vayamos a analizar nuestro sistema y plantear una solución técnica nos tenemos que fijar mínimo en estos tres umbrales (existen muchos más expuestos por Microsoft), ya que si los respetamos tendremos mucho terreno ganado.

Ahora bien ¿Existe alguna solución mágica que nos permite cumplir todos estos objetivos?

La respuesta es sencilla, “NO”, en estos casos es más fácil

por la experiencia vivida saber muchas formas de cómo no hacerlo, que la realmente perfecta, dado que cada proyecto es un mundo y cubrir todos los casos es casi imposible.

Lo que si se recomienda es no hacer lo siguiente:

- Subir los límites desde administración central
- Los umbrales que hemos recordado antes son “modificables” desde la administración central, pero es muy poco recomendable. Por un lado, si llegamos a este punto podemos confirmar que nuestra fase de Análisis y Diseño es un fracaso porque ya notamos el mal rendimiento, y por otro lado es nadar para morir en la orilla, ya que la degradación es exponencial y llegado a este punto cada vez nos va a funcionar menos subir los límites.
- Pensar en nuestras listas como un modelo “RELACIONAL”
- Dejemos claro que las listas de SharePoint nunca pueden representar un modelo relacional como podríamos conseguir directamente en tablas con SQL. Tender a este modelo nos implica llenar nuestras listas de campos del tipo búsqueda para poder reproducir los joins entre tablas, y estamos claramente tendiendo a superar el segundo umbral (nº máximo de columnas de búsqueda sobre listas).
- Cultivo de carpetas desmedido para gestionar nuestras vistas
- Añadir carpetas a las listas puede ser una buena solución, ya que organizamos nuestra información de forma estructurada y además empezamos a limitar el número de elementos por vista. Tenemos que entender que una carpeta es un ítem más y podemos tener igualmente 5k carpetas por vista como máximo. Pero esta solución alberga un gran problema, y es el usuario. Yo mismo como usuario en mi maquina personal tiendo a tener millones de carpetas para organizar mi información, llevar esto a SharePoint puede ser la perdición para el desarrollador, ya que realizar búsquedas recursivas en ocasiones puede ser muy complejo.
- En conclusión, no es malo añadir carpetas, pero añadir sin control es mala práctica.
- Vistas infinitas
- ¿Si el problema es 5k elementos por vista, porque no crear vistas a demanda y controlamos así los umbrales?. Ya podemos deducir que esta solución tampoco

es la correcta, ya que no podemos asignar vistas a usuarios, es decir todos los usuarios verían las mismas vistas, además de que es casi imposible controlar los umbrales filtrando por columna o agrupando por varias, siempre llega un momento que nos despistamos y perdemos el control de nuestra información, obligándonos incluso a subir los límites para poder crear más vistas aún.

Entonces visto todo lo anterior, ¿No podemos conseguir nuestros objetivos?, evidentemente la respuesta es sí, si no SharePoint no tendría tanto recorrido en el mercado. Si conjugamos una buena fase de análisis y diseño de nuestra información, ajustamos unos límites de listas realistas y controlados que podrían ser incluso un poco más altos del estándar, no damos el control total al usuario para crear carpetas si no que llevamos un proceso lógico de crecimiento, y lo mezclamos con vistas acotadas / permisos por carpetas, podemos llegar a conseguir sin mucho esfuerzo a futuro un sistema accesible y sobre todo con un rendimiento óptimo.

Pero para realizar todo esto tenemos que tener unos conocimientos altos de SharePoint y de arquitectura en general, por lo que vamos a plantear el uso de los Conjuntos de Documentos para gestionar nuestras bibliotecas, ya que son sencillos de utilizar y nos pueden ayudar a cumplir todos los requisitos anteriores.

Conjunto de Documentos en SharePoint 2013

Un conjunto de documentos de SharePoint es un tipo especial de carpeta con metadata, que nos permite administrar y gestionar documentos y archivos. Además, tiene su propio tipo de contenido, y permite limitar que tipos de contenido pueden tener los archivos que contiene. Los límites de un conjunto son los mismos que vimos anteriormente, no aplica ninguno nuevo. Un conjunto puede tener permisos exclusivos o se le pueden asignar Workflows, como a cualquier carpeta o elemento de una biblioteca.

Caso práctico: Gestión de proyectos

Vamos a ver un claro caso de uso de conjuntos sobre SharePoint, que puede ser un área de proyectos para el departamento de Finanzas de una empresa. Pongamos que una empresa necesita archivar y gestionar los gastos, pedidos y seguimientos de un proyecto, y que cada proyecto tendrá un grupo acotado de usuarios que podrán aportar la documentación. Cada proyecto vamos a suponer que tiene documentos del tipo Gastos, Pedidos y Resultados; y además un proyecto tendrá usuarios Gestores que puedan administrar la documentación y usuarios Lectores que puedan consultarla.

Activar características

Para poder usar conjuntos de documentos, lo primero que debemos hacer es activar la característica “Conjunto de documentos o document sets” desde las características de la colección de sitios.



Imagen 1.- Característica con las funcionalidades de conjuntos de documentos.

Crear tipos de contenido

Necesitamos crear los tipos de contenido para el Proyecto (conjunto padre) y para los documentos del tipo Gastos, Pedidos y Resultados. Para ello accedemos a configuración del sitio raíz, y en tipos de contenido creamos los que necesitamos como muestra la imagen.

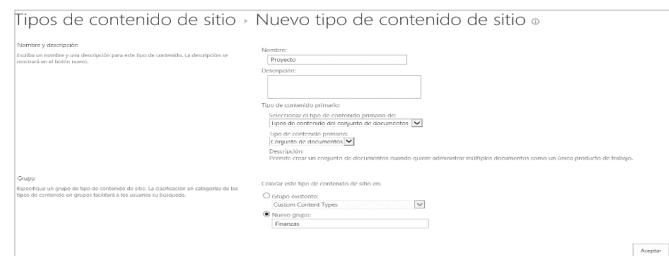


Imagen 2.- Creamos un tipo de contenido para nuestro conjunto de documentos.

Para el proyecto crearemos un tipo de contenido primario Conjunto de documentos, para el resto del tipo Documento. Agruparemos todos los tipos de contenido en un nuevo grupo que llamaremos Finanzas.

Tipo de contenido de sitio	Primario
Finanzas	
Gastos	Documento
Pedidos	Documento
Proyecto	Conjunto de documentos
Resultados	Documento

Imagen 3.- Documentos y Conjuntos de Documentos del grupo Finanzas.

Podemos añadir alguna columna al tipo de contenido, como por ejemplo he hecho con los gastos.

Columnas	Tipo	Estado	Origen
Nombre	Archivo	Requerida	Documento
Nombre	Una línea de texto	Opcional	Documento
Título	Varias líneas de texto	Opcional	Documento
Descripción	Persona o grupo	Opcional	Documento
Aprobador	Una línea de texto	Opcional	Documento
Fecha Registro		Opcional	Documento

Imagen 4.- Columnas del tipo de contenido Gastos.

Los gastos contienen un nombre, descripción del mismo, un usuario aprobador que consume un grupo de seguridad que he creado como Aprobadores_Gastos_Proyectos, y por ultimo una fecha para auditar el registro del gasto en el proyecto. Como Gastos hereda del tipo Documento, tendremos que subir por ejemplo un PDF o una Excel con el resumen de Gastos para que el Aprobador revise los números. Para el resto de documentos deberíamos hacer algo similar, así como con el proyecto que en mi caso queda con Título, Descripción, Gestor Proyecto (Persona), Fecha Inicio y Fecha de Finalización. El campo Gestor Proyecto se alimenta del nuevo grupo de seguridad Gestores_Proyectos.

Columnas	Tipo	Estado	Origen
Nombre	Una linea de texto	Oculto	Elemento
Título	Archivo	Requerida	Carpeta
Nombre	Varias líneas de texto	Opcional	Conjunto de documentos
Descripción	Persona o grupo	Opcional	
Gestor Proyecto	Fecha y hora	Opcional	
Fecha Inicio	Fecha y hora	Opcional	
Fecha Finalización	Fecha y hora	Opcional	

Imagen 5.- Columnas del tipo de contenido Proyecto.

Crear biblioteca de documentos

Creamos una biblioteca de documentos con el tipo Proyecto como tipo de contenido primario, con lo que tendremos una biblioteca con Conjunto de documentos implementada. Para conseguir esto, en contenido del sitio Agregamos una aplicación del tipo Biblioteca de Documentos, con el nombre Proyectos.



Tipos de contenido
Especifique si se permite la administración de tipos de contenido en esta biblioteca de documentos. Cada tipo de contenido aparecerá en el botón nuevo y puede tener un solo conjunto de columnas, flujos de trabajo y otros comportamientos.

¿Desea permitir la administración de tipos de contenido?
 Sí No

Imagen 6.- Habilitando el uso de tipos de contenidos.

Una vez creada la biblioteca en la configuración avanzada de la biblioteca => configuración avanzada, permitimos administrar los tipos del contenido como muestra la imagen. Ahora desde la página de configuración de la lista en el apartado “Tipos de documentos” podemos borrar el tipo de contenido Documento y agregar el tipo de contenido Proyecto creado anteriormente.



Tipos de contenido
Esta biblioteca de documentos está configurada para permitir diversos tipos de contenido. Los tipos de contenido siguientes se encuentran disponibles actualmente en esta biblioteca:
 Tipos de contenido: Gastos, Pedidos, Resultados
 Visible en botones nuevos: ✓
 Tipos de contenido predeterminado: ✓
 Agregar a partir de tipos de contenido de sitio: ✓

Imagen 7.- Agregamos el tipo de contenido Proyecto a la biblioteca.

Configurar Conjunto de documentos

En este punto ya podríamos crear un nuevo proyecto, pero antes vamos a configurar algunos aspectos interesantes de nuestro conjunto de documentos. Desde la configuración de la biblioteca, en Tipos de Contenido vamos a acceder al tipo de contenido Proyecto. Una vez allí seleccionamos configuración del conjunto de documentos, y en esta pestaña podemos gestionar lo siguiente:

- **Tipos de contenido permitidos:** Se pueden elegir qué tipos de contenido van a tener los documentos del conjunto, en este caso vamos a seleccionar los creados para Finanzas (Gastos, Pedidos, Resultados).



Tipos de contenido permitidos
Permite seleccionar de la lista de tipos de contenido de sitio disponibles para agregarlos al conjunto de documentos.

Seleccionar tipos de contenido de sitio:
 Finanzas

Tipos de contenido de sitio disponibles:
 Agregar >
 < Quitar

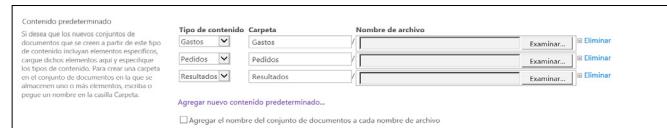
Tipos de contenido permitidos en el conjunto de documentos:
 Gastos
 Pedidos
 Resultados

Descripción:

Imagen 8.- Administramos los tipos de contenido del conjunto de documentos.

- **Contenido predeterminado:** Una opción interesante es que el conjunto coloque en distintas carpetas los documentos en función del tipo de contenido, o que

incluso cuando creamos un documento del tipo Gasto se genere un documento predeterminado como podría ser un contrato a firmar o un documento de privacidad.

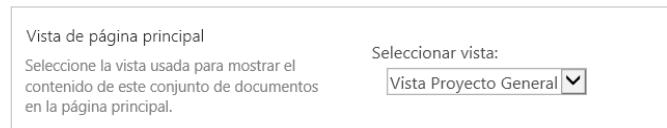


Tipo de contenido	Carpeta	Nombre de archivo
Gastos	Gastos	<input type="button"/> Examinar... <input type="button"/> Eliminar
Pedidos	Pedidos	<input type="button"/> Examinar... <input type="button"/> Eliminar
Resultados	Resultados	<input type="button"/> Examinar... <input type="button"/> Eliminar

Agregar nuevo contenido predeterminado...
 Agregar el nombre del conjunto de documentos a cada nombre de archivo

Imagen 9.- Documentos predeterminados.

- **Vista página principal:** Podemos crearnos y elegir que vista queremos ver en la página principal del conjunto de documentos en la relación de documentos del proyecto. Nos vamos a crear una vista que va a contener los campos Titulo, Gestor Proyecto, Fecha de Inicio, Fecha de Finalización de proyecto, y la llamaremos “Vista Proyecto General”.



Vista de página principal
Seleccione la vista usada para mostrar el contenido de este conjunto de documentos en la página principal.

Seleccionar vista:
 Vista Proyecto General

Imagen 10.- Vista principal.

- **Personalizar página principal:** Cuando creamos un conjunto de documentos, este tiene una página principal que se puede asemejar a la vista de detalle de una biblioteca pero que es mucho más completa. Esta página es configurable al punto que podemos añadir WebParts ya existentes o desarrolladas por nosotros de forma sencilla, como en una página cualquiera de bibliotecas de SharePoint. Para personalizarla seleccionamos “Personalizar la página principal” y una vez en la página, podemos seleccionar editar Página en la ribbon superior y editar la página a nuestro gusto. Por defecto la página añade un logo predeterminado, y las dos WebPart que tenemos de serie para conjuntos de documentos.



Zona 1: Agregar elemento web
 Imagen: 

Zona 2: Agregar elemento web
 Propiedades del conjunto de documentos:
 Conjunto de documentos
 El elemento web Propiedades del conjunto de documentos mostrará las propiedades del conjunto de documentos en la página principal.

Propiedad	Valor de la propiedad
Propiedad	Valor de la propiedad
Propiedad	Valor de la propiedad

Zona 3: Agregar elemento web
 Contenido del conjunto de documentos
 El elemento web Contenido del conjunto de documentos mostrará el contenido del conjunto de documentos en la página principal.

Imagen 11.- Página del conjunto de documentos en modo edición.

- **Propiedades del conjunto:** Esta WebPart visualiza las columnas indicadas en el apartado “Columnas de página principal” en la configuración de nuestro conjunto de documentos.
- **Contenido del conjunto de los documentos:** Lista todos los documentos pertenecientes al conjunto, y en este caso a nuestro proyecto.

Crear un nuevo proyecto

Hemos configurado nuestra biblioteca de Proyectos de

forma correcta, por lo que solo nos queda probar el crear un nuevo proyecto y familiarizarnos con el proceso de dar de alta nuevos documentos.

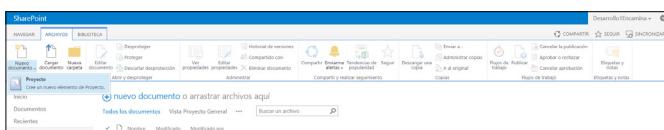


Imagen 12.- Creando un nuevo proyecto.

Necesitaremos dar de alta nuestro proyecto con los datos que configuramos en los pasos anteriores.

Imagen 13.- Formulario de creación del conjunto de documentos.

Una vez creado el documento, vemos la página principal del conjunto definida en la configuración de la biblioteca, en este caso hemos dejado la que nos viene por defecto.

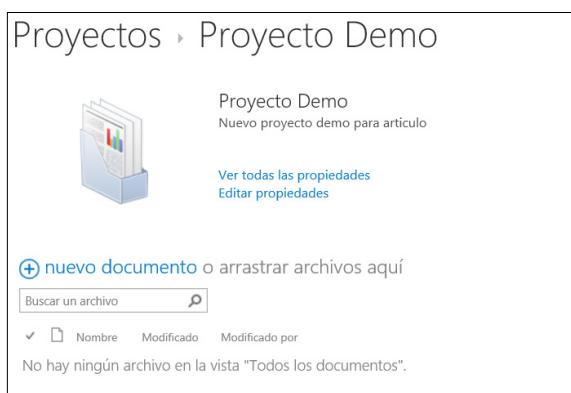


Imagen 14.- Conjunto de documentos de Proyecto.

El proceso de creación es similar a crear un ítem en una lista de SharePoint, y luego subir los documentos es igual que subir documentos a una biblioteca. Por ejemplo, vamos a subir un documento del tipo Gasto, seleccionando "Nuevo Documento". Se nos solicita que adjuntemos el archivo y los datos que configuramos en el tipo de documento Gastos.

Imagen 15.- Formulario de creación de documento de gastos.

El resultado es el siguiente, un nuevo conjunto del tipo proyecto llamado Proyecto Demo, que contiene un nuevo

documento del tipo Gasto.

Proyectos

Proyectos > Proyecto Demo

Imagen 16.- Conjunto de documento creado.

Permisos del proyecto y biblioteca

Esta parte la podemos complicar todo lo que queramos, lo más sencillo sería crearnos un grupo de seguridad por cada proyecto con el nombre "Miembros_NombreProyecto", y le daríamos lectura sobre ese nuevo conjunto de documentos. A su vez reutilizaremos el grupo de Gestores_Proyectos para darles permisos de colaboración. Sobre la biblioteca podemos dar permisos de Colaboración a Gestores_Proyecto y lectura al resto de grupos de lectores que vayamos creando.

Esto es una opción que yo acabo de elegir, para demostrar que un conjunto no deja de ser un ítem más al estilo de una carpeta en una biblioteca y podemos dar permisos exclusivos. Debemos recordar que podríamos crear un máximo de 5k proyectos sin sobrepasar el umbral de permisos exclusivos, lo cual en este caso está bastante bien.

Conclusiones

Como hemos podido ver sin tirar una línea de código hemos conseguido administrar información por valor de 5k proyectos, y respetando los límites de SharePoint. Podemos dar seguridad a los conjuntos, personalizar la página de resultados e incluso añadir componentes propios si fuera necesario. Podemos concluir que no hay solución mágica para todo, pero esta es una buena forma de ordenar tus bibliotecas y conseguir que un usuario tenga la información controlada respetando los umbrales que nos marca SharePoint, lo cual nos dará un mayor rendimiento en nuestras aplicaciones.

Conjunto de Documentos: Acceso a Datos desde código

Como hemos visto en el caso práctico con conjunto de documentos y los componentes Out of the box de SharePoint

podemos conseguir resultados interesantes, pero para los que les gusta desarrollar, o para casos más complejos vamos a ver un poco como crear y consumir un conjunto de documentos desde código.

Código Servidor / Modelo de Objetos de SharePoint

La primera forma y la más común es desarrollar código servidor en una WebPart, servicio web o una aplicaction page con modelo de objetos de SharePoint en servidor, que gestione nuestros conjuntos de documentos. Para poder realizar esto necesitamos añadir la dll “Microsoft.Office.DocumentManagement”, a nuestro proyecto (Viene con la instalación de SharePoint).

Aunque no lo vamos a tratar en este artículo también podemos consultar conjuntos vía CSOM en JavaScript o por PowerShell.

Crear un ítem del tipo conjunto de documentos

```
private SPFolder CreateDocumentSet(string name, string description, SPList library, SPUUser currentUser, SPContentType contentType)
{
    int docsetId;
    SPFolder folder = library.RootFolder;
    Hashtable properties = new Hashtable();
    properties.Add("DocumentSetDescription", description);
    DocumentSet docSet = documentSet.Create(folder, name, contentType.ID, properties, true);
    if (docSet != null)
    {
        docSetId = docSet.Item.ID;
        SPListItem docSetItem = library.GetItemById(docSetId);
        if (docSetItem != null)
        {
            docSetItem[SPBuiltInFieldId.Author] = currentUser;
            docSetItem[SPBuiltInFieldId.Editor] = currentUser;
            docSetItem.Update();
        }
    }
    return docSet.Folder;
}
```

Imagen 17.- Código para crear un conjunto de documentos.

Como vemos con la instrucción DocumentSet.Create que nos proporciona la dll que hemos añadido antes, podemos crear un conjunto de documentos de forma sencilla. Necesitamos el SPContentType con el tipo de contenido primario del conjunto a crear y que debe existir en SharePoint, el rootFolder de la biblioteca donde vamos a añadir el nuevo conjunto y el nombre del nuevo conjunto de documentos.

Insertar documentos a un conjunto

```
private Boolean UploadFile(SPFolder folder, Stream stream, String documentName, String title, SPList library, String contentTypeID)
{
    try
    {
        SPFile document = folder.Files.Add(documentName, stream, true);
        if (document != null)
        {
            document.CheckOut();
            SPListitem documentListitem = document.Item;
            documentListitem[Constants.TITLEFIELD] = title;
            documentListitem[Constants.CONTENT_TYPE_ID_FIELD] = contentTypeID;
            documentListitem.Update();
            document.CheckIn("Empty");
            document.Update();
        }
    }
    return true;
}
```

Imagen 18.- Subir un documento a un Conjunto de documentos-

La función de insertar un nuevo conjunto nos devolvía un SPFolder, que usaremos para insertar sobre este un documento obtenido por pantalla. Como vemos en el código los parámetros recibidos en el método son la carpeta del conjunto, el fichero, el nombre del documento, el título del documento a mostrar por pantalla y el id del tipo de contenido del documento (este debe ser alguno de los permitidos por el conjunto o nos arrojará una excepción).

No es muy distinto a añadir un documento a una librería de SharePoint, ya que como hemos dicho antes un conjunto de documentos no deja de ser un SPFolder para SharePoint.

Leer los documentos de un conjunto

```
using (SPSite site = new SPSite(currentSite.Site.RootWeb.Url))
using (SPWeb web = site.OpenWeb())
{
    SPList targetList = web.Lists.TryGetList("LIST_NAME");
    SPFolder targetFolder = targetList.RootFolder.SubFolders["Nombre_DocumentSet"];
    var documentSet = DocumentSet.GetDocumentSet(targetFolder);
    if (documentSet != null)
    {
        foreach (SPFile file in documentSet.Folder.Files)
        {
            //codigo que se necesite hacer con los ficheros obtenidos
        }
    }
}
```

Imagen 19.- Obtener un documento del conjunto de documentos

Como podemos ver en el código, podemos obtener el conjunto con el nombre de la lista y el nombre del conjunto de documentos, y con una simple iteración sacar todos los SPFiles del conjunto. Aclarar que también podríamos hacer una consulta con CAML o LINQ to SharePoint, si no queremos iterar sobre el listado de ficheros.

En resumen, podemos comprobar que vía código se nos simplifica bastante realizar consultas o insertar elementos con conjunto de documentos, si lo comparamos con estructurar nuestras bibliotecas con carpetas, ya que nos implicaría tener que ir haciendo recorridos recursivos sobre una biblioteca de SharePoint.

SERGIO HERNÁNDEZ MANCEBO

Team Leader en **ENCAMINA**
shernandez@encamina.com

Machine Learning: Usos y Aplicaciones

Por mi vinculación y pasión con esta tecnología son muchas las personas que me preguntan periódicamente acerca de los usos y avances del Machine Learning, por lo que inspirado en algunas de esas conversaciones es que surge este artículo.

Para los más profanos en el tema y sin entrar en definiciones, recordemos que Machine Learning no es un método de adivinación, y aunque pertenece a una forma de inteligencia artificial, está lejísimos de ser una base para crear una forma de pensar “fabricada”. Lo que Machine Learning nos brinda es la búsqueda del patrón de comportamiento de una cosa basándose en lo que esa cosa ha hecho antes bajo determinadas circunstancias, y una vez obtenido el patrón aplicarlo con el objetivo de “acertar” o “predecir” el comportamiento futuro. Como para muestra un botón, comparto con vosotros un debate (y un deseo) que hace muy poco tenía con un también autor de esta revista: Sergio Hernández.

Conducir con rutas inteligentes, ¿deseo o realidad próxima?

El debate con este colega se centraba en la gran utilidad que tendría incorporar en los actuales coches una mejora al sistema GPS o de ayuda a la conducción que, mediante la incorporación de la información sobre hábitos, costumbres, condiciones de tráfico y meteorológicas, y nuestra forma de conducir, personalice las rutas que el sistema GPS nos recomienda.

Como ejemplo de aplicación y para aclarar la idea recuerdo una situación de la que he sido partícipe el mes pasado: dado que era un trayecto desconocido para mí, configuré el sistema GPS para llegar desde mi posición hasta el destino deseado, y este me recomendó como opción principal la ruta más rápida, con menos tráfico y haciendo uso de autopistas. En ese momento observo que esa ruta incluye peajes, y como el tiempo de conducción no era una prioridad en ese momento, configuro el GPS para que evite los peajes y este vuelve a calcular la ruta presentándome dos posibles opciones diferenciadas apenas por unos pocos minutos de tiempo de conducción. Una vez hecha la elección y siguiendo la ruta seleccionada comienzo el ascenso y descenso a un par de carreteras de montaña en las cuales la cantidad de carriles se reducía a uno, los arcenes eran inexistentes y los precipicios a borde de carretera eran im-

portantes. Llegado al destino pensaba lo que le hubiese pasado ante esta misma situación a una persona que sufriese de vértigo, o con menos experiencia o poco acostumbrada a inclinaciones superiores a 9 grados, o con un vehículo tirando de una auto caravana, ya que la lógica del GPS los habría metido por el mismo camino y cuando llegasen a los puntos críticos se encontrarían con el problema de lleno. La verdad es que ninguna de esas 3 suposiciones la habría pasado bien y probablemente hubieran dado la vuelta y desecharon camino.

Contar con asistentes a la conducción auto-adaptables a nuestros hábitos es ya casi una realidad

Ahora imaginemos la potencia de un sistema GPS que, aunque necesite tecnología adicional, recoja información sobre los hábitos del conductor. A decir verdad, detectar cosas como “aminora o acelera excesivamente en las pendientes de bajada”, “conduce muy lentamente en carreteras con acantilados a los lados”, “aumento de correcciones al rumbo”, “aumento de los descuidos/indiferencia a las señales”, etc., no son tan difíciles de detectar y al día de hoy tenemos tecnología de sobra para ello. El rol de Machine Learning en este punto es que sería la herramienta perfecta de ayuda a la conducción basándose en la búsqueda y aprendizaje de los patrones de cada conductor. La cantidad de aplicaciones sería enorme y no solo en materia de GPS, sino que podría aportar desde cambios en los parámetros de consumo de los consumibles y la periodicidad de las revisiones técnicas, hasta el precio de nuestro seguro. Me consta que existen iniciativas que ya están tratando algunas de estas mejoras, aplicando Big Data y Machine Learning, por lo que estoy seguro que las veremos disponibles e incorporadas como un accesorio más antes de lo que creemos.

Como consumidores: ¿Queremos la personalización?

Como consumidores hay infinidad de ejemplos donde nuestros proveedores podrían mejorar el producto o servicio que nos brindan haciendo uso de Machine Learning y no cabe duda que lo notarían inmediatamente en sus

ventas.

Por mencionar algún ejemplo de información que ya se recoge al día de hoy pero no se explota de esta forma, pensemos en el cómo utilizas los servicios que tienes contratados en el teléfono móvil que llevas encima.



Imagen 1.- ML aplicado a la conducción.

Simplificando mucho podríamos decir hay grupos de personas que utilizan e incluso sobrepasan el caudal de datos mensual, pero que por el contrario no llegan a utilizar la cantidad de minutos previstos/contratados para llamadas de voz. Al mismo tiempo tendremos otro grupo que actúa a la inversa, es decir muchas llamadas y pocos datos, y obviamente quedan dos grupos más que utilizan ambos servicios por debajo o por encima de los límites.

Como consumidores hay infinidad de ejemplos donde nuestros proveedores podrían mejorar el producto o servicio que nos brindan

Podríamos decir que esta es la división más simple de los hábitos de consumo de los usuarios de una operadora telefónica, pero que al mismo tiempo esboza el modelo que siguen las operadoras para ofrecer las distintas modalidades de contratación. Es verdad que siempre existen paquetes adicionales que permiten personalizar la tarifa contratada, e incluso me consta que se realiza una inversión en Big Data para detectar esas necesidades y ofrecer esos paquetes al usuario vía la clásica llamada del departamento de marketing, pero esto no deja de ser Big Data pura y dura donde el estudio de patrones de comportamiento por usuario es nulo.

Muchas empresas ya cuentan con el Big Data suficiente para ser explotado mediante Machine Learning.

Dudo que un usuario se niegue a adaptar la tarifa contratada si el ofrecimiento estuviese especialmente adaptado a sus hábitos en ese momento temporal. Es decir, las técnicas de Machine Learning podrían detectar cambios en el comportamiento del usuario como ser el momento del día

en que se utilizaba antes y ahora un determinado servicio, o los cambios en la cantidad de unidades que consume (minutos, bytes, etc.), o las características de ese cambio (llama a determinados números, conecta a determinados servidores, cambia o no su ubicación frecuentemente...). Una de las tareas fundamentales para dar este salto ya se realiza y no es otra que la recopilación de la información de uso (hábito), por lo que deberíamos estar cerca de empezar a recibir estas ventajas.

Utilizando este mismo ejemplo de la telefonía pensemos que es aplicable a casi todos los ámbitos. Pensar que el supermercado nos ofrezca una oferta exclusiva para nosotros si compramos verduras y pañales al mismo tiempo podría no ser algo tan descabellado para algunos usuarios que hasta hace poco solo compraban pizzas y cerveza, y en otros casos la oferta de ropa podría ser un descuento especial por compra de conjunto de camisa y pantalón, ambos a juego y en azul, los primeros dos martes de cada mes...



Imagen 2.- ML aplicado a compras cotidianas.

¿Desarrollar negocio con Machine Learning?

Ya os he mencionado algunos ejemplos de cómo muchas empresas podrían mejorar lo que nos ofrecen, ¿pero qué pasa del lado de esas empresas? Ya existe un gran número de empresas que están incorporando poco a poco técnicas de Machine Learning a su modelo productivo o de ventas. Por mencionar algunos podéis ver los casos que publica Microsoft en <http://azure.microsoft.com/en-us/services/machine-learning/>.

Lo más importante a tener en cuenta desde el punto de vista empresarial es que el uso de Machine Learning genera impacto positivo en todo el modelo corporativo, desde la producción hasta las ventas, y para los más escépticos dejo algunas cuestiones planteadas con el objetivo de arrojar un poco más de luz sobre el tema.

Imaginemos lo útil que sería obtener los parámetros de aceptación de un nuevo producto o servicio basándonos en los comportamientos de los usuarios tanto en lo que respecta a las campañas de marketing como a las primeras ventas que se realizan en web o tienda física de ese

producto. Podríamos saber qué características son las más solicitadas y cuáles son rechazadas, lo que permitiría ajustar mucho mejor el modelo de negocio y la previsión de ROI, al mismo tiempo que esa misma información es básica para optimizar las cadenas de montaje, elaboración y/o producción, e incluso permitiría ajustar mejor la petición y almacenamiento de la materia prima, la contratación de personal, etc.

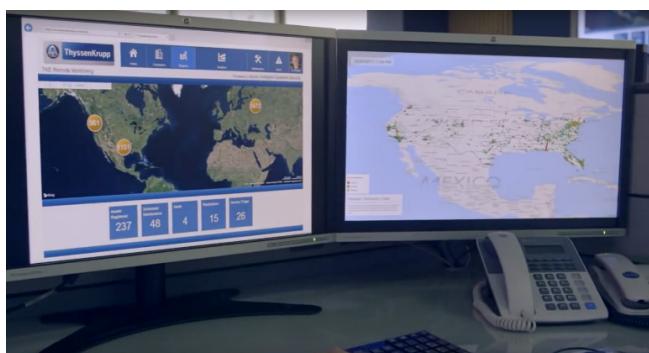


Imagen 3.- Ejemplo práctico de uso de ML (Fuente: <http://www.microsoft.com/en-us/server-cloud/customer-stories/thyssen-krupp-elevator.aspx>)

Machine Learning para prevenir enfermedades

En base a una campaña realizada por Microsoft hace poco tiempo que anunciaba un slogan llamativo a modo de “Esta es la nube que está ayudando a curar el cáncer”, unos colegas me preguntaban si era real, si realmente habíamos llegado a ese punto.



Imagen 4.- Imagen de Microsoft en relación a la lucha contra el cáncer.

Nota: Más información sobre este caso en particular en <http://bit.ly/PalloCmpMOSS1> y en <http://bit.ly/PalloCmpMOSS2>

Afortunadamente la respuesta es que sí. Específicamente no estamos hablando de un caso exclusivo de Machine Learning, pero sí de la gran capacidad de computación que ofrece Azure y como Virginia Tech hace uso de esa capacidad brutal de potencia de cálculo en la búsqueda de una cura para el cáncer. El resultado es demoledor: donde antes necesitaban 2 semanas para secuenciar y analizar un genoma, ahora logran procesar hasta 100 por día, bajando el coste de genoma secuenciado de U\$S 95 MM a U\$S 6.000.

Aunque no cabe la menor duda de lo que Machine Learning puede aportar al mundo de la medicina, estamos todavía muy lejos de sacar todo el provecho que nos puede llegar a dar. Por ahora es un hecho la aportación en la búsqueda de patrones en el comportamiento y mutación de virus y bacterias, la aportación en la pre-diagnóstico clínica o el soporte al tratamiento de enfermedades coronarias, por mencionar algunas.

Y la lista sigue

Describiros todos los casos en los que se está aplicando Machine Learning haría este artículo el más pesado de digerir, por lo que para terminar os dejo algunas de las posibles aplicaciones que ya se están utilizando:

- Reconocimiento Facial, Reconocimiento de Voz, Interpretación de Voz, Traducción automática: no puedes dejar de visitar <https://www.projectoxford.ai/>, o jugar con <http://how-old.net> (prueba subir una foto tuya...), o si ya tienes Windows 10 prueba a utilizar Cortana para gestionar tu Windows o prueba la traducción en tiempo real de Skype.
- Prevención de delitos y Fraude: Policía de Cambridge, PayPal, American Express y muchos otros.
- Adecuación de resultados en los motores de búsqueda.
- Prevención de fenómenos atmosféricos.
- Modelado de nuevas moléculas.
- Realidad aumentada.
- Reconocimiento de gestos, formas y posiciones.
- Educación adaptativa al alumno
- Mejora y Adaptación de proceso de producción y mantenimiento.

JAVIER MENÉNDEZ PALLO

Director Oficina Centro – España y Portugal
ENCAMINA
jmenendez@encamina.com

5 Razones por las que es importante clasificar la información

Una de las características más importantes y que mejor lleva a cabo SharePoint en una plataforma de colaboración es la clasificación de la información. Metadatos administrados, tipos de contenido, columnas, taxonomías, perfil de usuario... Son muchas las funcionalidades en SharePoint que nos proporcionan mecanismos de clasificación de nuestra información. En muchas ocasiones, pese a contar con todo este arsenal de posibilidades, hay organizaciones que apenas realizan una mínima clasificación. En otras ocasiones, sí que existe una clasificación y una utilización de estas funcionalidades pero no es la correcta. A menudo se decide utilizar tipos de contenido, columnas o metadatos administrados solo por el hecho de justificar su uso, pero sin ningún fin en particular. Obviamente, no existe una manera objetiva de determinar si una clasificación es buena o mala, una taxonomía puede ser muy útil para una organización pero totalmente ineficiente para otra.

Es muy importante reflexionar acerca de qué tipo de información vamos a manejar, cómo está estructurada nuestra organización

Por eso, es muy importante reflexionar acerca de qué tipo de información vamos a manejar, cómo está estructurada nuestra organización y la relación que existe entre esta estructura organizativa y la tipología de documentos y/o procesos con los que trabajamos día a día. Aquella clasificación que sea reflejo de estos tres conceptos probablemente será la más adecuada, la más natural. Pero quizás no la más efectiva. Debemos, además, enfocar esta clasificación de manera que potencie la productividad en una o varias áreas de trabajo de nuestra plataforma SharePoint: colaboración, publicación, procesos de negocio, seguridad... Debemos pensar cómo definir esta clasificación para mejorar estas áreas en el caso concreto de nuestra organización. El problema es que, en general, en la fase de definición no se es consciente de cómo esta clasificación nos va ayudar en un futuro. Por eso, aquí os dejo varios ejemplos de escenarios donde la clasificación de la información resulta clave.

Experiencia de usuario

El éxito de una plataforma SharePoint está obviamente

determinado por la adopción del usuario final. Por ello, resulta más que necesario aprovechar las funcionalidades que hacen de SharePoint una verdadera plataforma de colaboración. Los servicios de búsqueda, perfil de usuario o metadatos administrados pueden (y deben) marcar la diferencia entre nuestra Plataforma SharePoint y otros sistemas de gestión documental. La verdadera potencia de estos servicios se alcanza cuando se apoyan sobre una clasificación sólida de nuestra información. Aquí tenemos un par de ejemplos:

- La búsqueda devolverá resultados más relevantes si somos capaces de filtrar resultados en función del departamento, del tipo de proyecto... y también ofrecerá opciones de refinamiento al usuario si existe algún criterio de clasificación.
- Nuestro perfil de usuario se construye a partir de propiedades de usuario. El usuario, por tanto, es parte de la información de nuestra plataforma. Definir propiedades de usuario significa clasificar dicha información y, en definitiva, nos proporciona un criterio para seleccionar qué información mostrar en nuestra plataforma en función del usuario que accede a ella.

Ciclo de vida del contenido

Otra característica que ayuda a la adopción del usuario final consiste en mantener continuamente la información más relevante a primera vista. Esto conlleva un proceso de gestión de vida del contenido en el que se requiere definir reglas que establezcan cuándo un tipo de información debe mostrarse en páginas principales, cuándo debe ser movida de ubicación, cuándo debe ser considerada obsoleta y, por último, cuándo debe ser archivada o eliminada. Sólo podremos definir y automatizar estas reglas cuando exista un criterio, cuando exista una propiedad, valor, metadato o como lo queramos denominar, que nos indique en qué parte de dicho ciclo se encuentra un documento o elemento. En definitiva, cuando exista una clasificación.

El éxito de una plataforma SharePoint está obviamente determinado por la adopción del usuario final

Seguridad de la información

Aunque no lo parezca, el hecho de que nuestra información esté bien clasificada aportará un mayor nivel de seguridad por dos motivos:

- En primer lugar, porque nos ayudará a evitar fugas de información. El riesgo que se corre en una plataforma de colaboración es que información confidencial aparezca en un área pública de nuestra plataforma. Si somos capaces de tener clasificados (y sobre todo de forzar esta clasificación) estos documentos confidenciales, nos será mucho más fácil identificar estos casos y sobre todo prevenirlos.
- Y, en segundo lugar, considerando como decíamos los usuarios y sus propiedades como parte de la información de la organización, una buena definición del perfil de usuario nos puede servir para conceder permisos en nuestra estructura de sitios en función de la pertenencia a un departamento, a una sede o a un grupo de trabajo del usuario, por ejemplo.

Restaurar un documento a partir del backup de ayer puede ser aceptable para un grupo de usuarios, pero inaceptable para otro, especialmente si es información que se actualiza continuamente

Protección de datos

Uno de los errores más comunes a la hora de definir un plan de backup para nuestra plataforma es otorgar a todo nuestro contenido la misma importancia. Son muchas, demasiadas, las organizaciones que definen un único plan de backup para todo. No todos los documentos tienen la mis-

ma criticidad. Restaurar un documento a partir del backup de ayer puede ser aceptable para un grupo de usuarios, pero inaceptable para otro, especialmente si es información que se actualiza continuamente. Una vez más, la manera de crear diferentes planes de backup en función del tipo de información requiere tenerla clasificada.

Migraciones

Por último, la mayor ventaja que podemos tener de cara a una migración a SharePoint, ya sea desde una versión anterior o desde otro sistema de gestión documental, es que la información a migrar esté clasificada, o que al menos, se pueda clasificar. Al contrario de lo que la gran mayoría piensa, un proyecto de migración no conlleva solo mover contenido. Conlleva, además, la reestructuración de dicha información, el archivado de la información obsoleta y la eliminación de contenido redundante, entre otras cosas. Y cada una de ellas sólo es viable si existe un criterio que nos permita definir el contenedor de destino para cada tipo de documento en nuestra plataforma final. Y también aquí, esto solo es posible mediante una clasificación de la información.

Conclusión

Estos son solo unos ejemplos de cómo la clasificación de la información puede ser determinante en una Plataforma de Colaboración donde, en principio, no consideramos relevante tener nuestros documentos clasificados de un modo u otro. Por eso, cuando pensemos en clasificar nuestra información, tengamos en cuenta más aspectos que la mera tipología o extensión de un documento.

GONZALO MARCOS ANSOAIN

Technical Solutions Professional en AvePoint
gonzalo.ansoain@avepoint.com



Llega... SharePoint Saturday Barcelona

Institut Químic de Sarrià, Via Augusta 390
<http://tinyurl.com/spsbcn>

Charlas técnicas de primerísima calidad
Ponentes de lujo
Comunidad de SharePoint
Evento gratuito de todo el día
Sorteos con valiosos regalos
Te esperamos el 26 de septiembre

La importancia de diseñar y aplicar un buen uso de Gobernanza en nuestro entorno SharePoint 2013 (3^a Parte)

Personalizando el servicio

Aquellos que personalizan el servicio de SharePoint también comprueban un alto grado de satisfacción en su experiencia con el entorno mediante dicha personalización. En este último artículo sobre la gobernanza, nos centraremos en los factores específicos del diseño y desarrollo personalizado que se relacionan con el gobierno del servicio de SharePoint, destacando estrategias y claves para los diferentes tipo de personalizaciones, incluyendo enfoques sobre cómo podemos diseñar, desarrollar y realizar pruebas de procesos.

"No todo a lo que nos enfrentamos se puede cambiar, pero nada se puede cambiar si no nos enfrentamos"

Un enfoque para organizar las personalizaciones globales con los propietarios de las personalizaciones es desplegarlos en el Catálogo de Aplicaciones de SharePoint. Los desarrolladores pueden desarrollar aplicaciones que contienen funcionalidad para una aplicación, la funcionalidad que se ejecuta en otro servidor fuera de nuestra granja de SharePoint. Estas aplicaciones también proporcionan funcionalidad a cualquier propietario del sitio que quiere añadir la App a su sitio, y pueden descubrirla mediante el catálogo de aplicaciones.

Podemos crear un sitio de SharePoint para alojar y administrar el catálogo interno de Apps. Además, podemos compartir el catálogo entre granjas, lo que le permite centralizar el alojamiento y la gestión de su granja principal, mientras otras granjas subordinadas pueden consumir el catálogo y las aplicaciones que se enumeran.

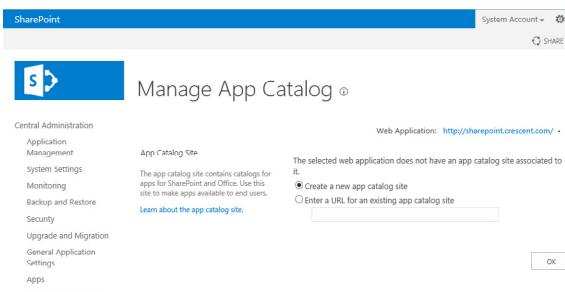


Imagen 1.- Creación de un catálogo de aplicaciones para SharePoint.

Una vez que se configura el catálogo y se habilitan las Apps, los administradores del sitio y los diseñadores de páginas pueden añadir las aplicaciones adicionales a su sitio, tam-

bien los usuarios finales pueden seleccionar la Aplicaciones disponibles que deseen agregar a su sitio. Desde aquí, pueden agregar las aplicaciones publicadas internamente, así como aquellas que vienen con SharePoint. Además, los administradores del sitio pueden ir a comprar Aplicaciones adicionales disponibles de los vendedores que publican sus aplicaciones en la tienda de Microsoft.

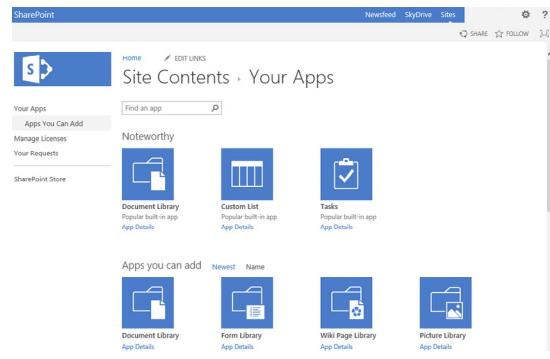


Imagen 2.- Página de Contenidos del Sitio en un sitio de SharePoint.

Habilitar la tienda de SharePoint y permitir que los propietarios de sitios puedan comprar sus propias aplicaciones directamente proporcionará una forma efectiva para que los usuarios puedan adquirir la funcionalidad que necesiten sin tener que solicitar un cambio global en el entorno de SharePoint.

Aquellos que personalizan el servicio de SharePoint también comprueban un alto grado de satisfacción

Delegar responsabilidades en los Administradores de Sitio

Una de las razones por las que SharePoint tiene éxito es su capacidad para distribuir y delegar la administración. Se elimina la carga (y cuello de botella) para el departamento de TI a la hora de responder a todas las peticiones de los usuarios. Suele ocurrir muy a menudo que los administradores de sitio no estén preparados para adquirir esa responsabilidad y sobre todo que no estén interesados en adquirirla, pero se puede cambiar dicho enfoque si, a través de procesos sencillos, como pueda ser cambiar el aspecto de un sitio a través de las plantillas que nos ofrece el entorno, vayan tomando sus propias decisiones.

Facilitar y Aislar las personalizaciones del Usuario Final

"Ningún copo de nieve en una avalancha se puede sentir responsable de la misma"

Tengo el total convencimiento de que el éxito del servicio de SharePoint radica en potenciar y enriquecer el manejo de los usuarios avanzados a los que les interese la herramienta, y dotarlos en la mayor medida posible de flexibilidad y libertad en sus acciones. Esto a muy corto plazo nos pueden aportar beneficios tales como:

- Mayor adopción global ya que los usuarios pueden adaptar sus experiencias para satisfacer sus necesidades.
- Reducir las cargas y cuellos de botellas al departamento de IT para cambios y personalizaciones
- Menos demanda de aplicaciones globales ya que los usuarios pueden desarrollar soluciones aisladas.
- Mejorar la sensación de enriquecimiento del usuario y su apoyo a nuestro objetivo.

La clave para lograr esta designación comienza en la decisión de en qué medida vamos a dotar de poder a nuestros usuarios. SharePoint no ofrece una herramienta muy útil como es SharePoint Designer para poder dotar a nuestros usuarios del enriquecimiento necesario para convertirse en usuarios autónomos que puedan gestionar sus propias necesidades y sus tiempos. Podemos comenzar enseñando aspectos como:

- Administrar niveles de suscripción de sitio y permisos.
- Crear recipientes de contenido como nuevas listas o bibliotecas.
- Adquirir aplicaciones a través del Catálogo de Aplicaciones.
- Cerrar o Eliminar un sitio.
- Construir flujos de trabajo personalizado.
- Construir paquetes de diseño personalizados.
- Implementar paquetes de soluciones de usuario.
- Activar y Desactivar características del sitio
- Cambiar el tema, logo o aspecto del sitio

No se tiene que capacitar a estos usuarios con todas las capacidades disponibles para un administrador de colección de sitios. Sin embargo, tenemos que decidir y luego tomar las acciones a implementar y gobernar y cuáles desea permitir y cuales se desea bloquear y restringir. Si decidimos capacitar a sus usuarios, dándoles la opción de personalizar sus sitios, entonces querremos asegurarnos de proporcionar un lugar seguro y aislado para sus personalizaciones. Por ejemplo, no queremos interrumpir el servicio para toda la comunidad por temas de consumo de recursos, y desde luego no queremos que se realicen cambios que entren en conflicto con otros usuarios. La so-

lución es el aislamiento de la colección de sitios.

Yo prefiero un diseño del sitio que distribuye a los usuarios y contenidos a través de muchas colecciones de sitios en lugar de estructurar una profunda jerarquía de sitios y sitios secundarios dentro de sólo unas colecciones de sitios. Me gusta el diseño con muchas colecciones de sitios por varios motivos, y el uso de una colección de sitios como contenedor para aislar los usuarios y sus personalizaciones es uno de los motivos. SharePoint utiliza colecciones de sitios como el límite principal para el contenido y la seguridad, por lo que sirven como recipientes naturales a utilizar en la planificación de sus diseños de sitios para el aislamiento y la delegación de la administración.

Una colección de sitios proporciona un área para que los usuarios puedan gestionar su experiencia en el sitio como mejor les parezca. Esto les permite hacer esto sin tener que preocuparse por la superposición con otros usuarios o experiencias. Esta segregación de la personalización de sitio puede tomar la forma de diseños visuales de sitios o aplicaciones personalizadas tales como los flujos de trabajo. También puede simplemente proporcionar un espacio para la forma en la que organizar las aplicaciones para el contenido y la forma en que quieren mostrar elementos web en la página de bienvenida. La mayoría de los usuarios estarán encantados con lo que el valor predeterminado que se les proporciona, lo que para ellos la colección de sitios es simplemente un contenedor para ayudarles a organizar su contenido. También le ayudará a mantenerse a salvo de los usuarios de otras colecciones de sitios que deseen personalizar sus sitios.

Diseñando nuestro desarrollo y proceso de pruebas

"Calidad significa hacer lo correcto cuando nadie está mirando."

Se debe elaborar y proporcionar un documento de normas para el desarrollo y un proceso de pruebas para los mismos, con el objetivo de minimizar los riesgos que puedan afectar a la disponibilidad del servicio de SharePoint. Para ello, defino los puntos claves a tener en cuenta en la elaboración de dichos estándares:

- Explicar el rol de un arquitecto.
- Considerar la disponibilidad Global.
- Decidir entre las soluciones de SharePoint y Aplicaciones.
- Establecer los límites para los desarrolladores.
- Definir puntos de referencias de pruebas.
- Incorporar desarrollo basado en pruebas.
- Automatizar los controles de calidad de código.

El arquitecto tiene que proporcionar el equilibrio entre el desarrollo y la distribución de la funcionalidad, el análisis de los impactos a largo plazo y las consideraciones de un diseño sostenible. La arquitectura juega un papel crítico en un proyecto de desarrollo, ya que abarca tanto el diseño

de una solución que pueda satisfacer la necesidad actual, como las necesidades que se puedan presentar a futuro. Necesita considerar todas las piezas en movimiento y visualizar como se unirán los diferentes aspectos. Se debe pensar en el ciclo de vida de una solución, no sólo tener presente la funcionalidad inmediata a desarrollar.

Una vez que se configura el catálogo y se habilitan las Apps, los administradores del sitio y los diseñadores

Se debe hacer especial énfasis a la envergadura de las personalizaciones en las colecciones de sitio, ya que pueden afectar a una aplicación web completa o incluso a la granja de servidores. Suele haber personalizaciones globales que un administrador de la granja despliega y pueden tener en su alcance un impacto potencial si algo sale mal, por lo que puede afectar a todos los usuarios del servicio de SharePoint. Para ello los desarrolladores deben tener siempre presente los peligros inherentes que pueden llevar un desarrollo erróneo. Los principales puntos que se han de tener en cuenta en un despliegue global son:

- Acceso a los datos compartidos o cualquier otro recurso compartido.
- Ejecución con privilegios elevados.
- Gestión de cualquier estado de la sesión o el almacenamiento en caché
- Añadir datos de varios objetos desecharables.
- La integración con sistemas externos directamente.

Una limitación de las apps, es que no se puede ejecutar código personalizado en los servidores de SharePoint en la granja, el código en su lugar se tiene que ejecutar en el cliente, como puede ser el uso de JavaScript, o ejecutar en un host, o a través de una solución de la nube como puede ser Azure. Si nuestra solución necesita ejecutar código en los servidores de SharePoint, necesitaremos utilizar un paquete de soluciones de SharePoint. También tendrá que usar paquetes de soluciones para desarrollar ciertos otros componentes para su granja de SharePoint. En particular, es necesario implementar lo siguiente a través de una solución de SharePoint en lugar de una App:

- Definiciones de sitios personalizados.
- Controles Delegados.
- Temas personalizados.
- Acciones personalizadas.
- Controles de usuario.

Los límites no tienen que venir como un montón de restricciones de áreas. Deben tomar la forma de directrices generales o áreas específicas para evitar ciertas cosas que desea restringir. Para todo lo demás, podemos decidir si dejarlos o no trabajar con un arquitecto y usar su propia creatividad para diseñar una solución, o si se quiere definir los límites para desarrolladores a nivel granular. A conti-

nuación, se enumeran los principales límites que se deben establecer para evitar estas cosas específicas:

- Todas las personalizaciones deben empaquetar los archivos de implementación y los archivos de manifiesto tanto en un paquete de solución o un paquete de aplicación para su despliegue en la granja de SharePoint.
- El paquete de implementación debe incluir todas las instrucciones de implementación y configuración de SharePoint para aplicar automáticamente, incluyendo cualquier configuración Web.config o cambios.
- Ninguna personalización puede modificar o sobrescribir los archivos del producto y del “core” de SharePoint directamente, incluyendo todos los archivos ASPX, XML, JavaScript, CSS y de imagen.
- Todas las personalizaciones que almacenan datos se deben persistir en una base de datos de SharePoint a través del API de SharePoint o proporcionando su propio almacén de datos dedicado.
- Ninguna Personalización puede realizar consultas directas sobre la BBDD de SharePoint.
- Todas las personalizaciones sólo deben acceder al sistema de archivos en modo de sólo lectura, a excepción de una copia de archivos de la propia personalización y ensamblados durante su despliegue inicial.
- Ninguna Personalización puede implementar su propia solución de almacenamiento en caché y en su lugar utilizar el SharePoint AppFabric u otra estrategia de persistencia disponible a través de la API de SharePoint.
- Todas las personalizaciones deben controlar todas las excepciones y proporcionar un mensaje de error amigable mientras graba cualquier detalle del error en el log de registro cuando existe una condición de error o excepción.

Escribir información para el registro de seguimiento (trazas) es muy fácil de incluir en el código. En su forma más simple, se puede escribir información básica información mediante una llamada al método SPDiagnosticsService.Local.WriteTrace y establecer la categoría correspondiente y la gravedad. Por ejemplo, se podría incluir el siguiente código en el bloque de la captura de una excepción al escribir el mensaje de la excepción en el registro de seguimiento. Podemos encontrar estos diagnósticos en el espacio de nombre Microsoft.SharePoint.Administration:

```
SPDiagnosticsCategory diagnosticsCategory =
diagnosticsService.Areas["SharePoint Foundation"].Categories["General"];
SPDiagnosticsService.Local.WriteTrace(
0,
diagnosticsCategory,
TraceSeverity.Unexpected,
ex.Message
);
```

Esto puede proporcionar información útil para detectar y rastrear el origen de un error mientras se ejecuta el código del entorno de producción. También se puede escribir esta información en el registro de eventos de Windows llamando al método WriteEvent en su lugar, como muestro en el

siguiente ejemplo de código:

```
SPDiagnosticsCategory diagnosticsCategory =
diagnosticsService.Areas["SharePoint Foundation"].
Categories["General"];
SPDiagnosticsService.Local.WriteEvent(
0,
diagnosticsCategory,
EventSeverity.Error,
ex.Message);
```

Para escribir la información de error en el registro de eventos de Windows para que sea más fácil para los Administradores identificar el error:

```
SPDiagnosticsCategory diagnosticsCategory = new
SPDiagnosticsCategory(
"Ejemplo Personalizado", // "Poner la categoría deseada aquí"
TraceSeverity.Unexpected,
EventSeverity.Error);
```

Para supervisar las estadísticas del uso de un código específico, se coloca el código en un bloque de código usando esa instancia un nuevo Objeto SPMonitoredScope. Se ejecuta dentro del bloque utilizando, el alcance monitoreado mide las estadísticas del código y los escribe en el registro de SharePoint ULS log y el panel del desarrollador. El espacio de nombres Microsoft.SharePoint.Utilities contiene la clase MonitoredScope.

```
using (new SPMonitoredScope(<Nombre del ámbito>))
{
// Código a supervisar
}
```

Varias métricas de rendimiento que se pueden utilizar para medir y monitorizar los diferentes niveles de rendimiento en el servicio de SharePoint. Estas medidas pueden ser útiles para monitorizar el desempeño de una aplicación personalizada, y sobre todo si para supervisar las tendencias de rendimiento antes y después de implementar una personalización. Estas métricas y tendencias pueden darle una idea general del impacto en el rendimiento en el servidor y en toda la granja, mientras que las herramientas descritas en la sección anterior proporcionan estadísticas detalladas de uso de un componente en una página individual.

Se puede usar el siguiente código para crear sus propios contadores de rendimiento en cada servidor SharePoint en la granja:

```
if (!PerformanceCounterCategory.Exists("Aplicación Personalizada"))
{
PerformanceCounterCategory.Create (
" Aplicación Personalizada ", // categoría del contador
"Descripción de la categoría personalizada",
PerformanceCounterCategoryType.SingleInstance,
"Contador de Aplicación", // Este es el nombre del contador
"Descripción del Contador");
}
```

Una vez que se crea un contador de rendimiento personalizado en el servidor, puede comprobar que existe y empezar a ver sus indicadores con la herramienta de Windows: Monitor de Rendimiento. Por supuesto, tener una categoría de monitor de rendimiento y el contador no será hasta

que empecemos a grabar los datos de rendimiento en su aplicación personalizada. Se puede utilizar el siguiente código dentro de la aplicación para crear una instancia del contador de rendimiento y luego incrementar su contador:

```
PerformanceCounter counter = new PerformanceCounter();
counter.CategoryName = "Aplicación Personalizada";
counter.CounterName = "Contador de Aplicación";
counter.ReadOnly = false;
// Incrementar el rendimiento del contador.
counter.Increment();
counter.Close();
```

Tener un proceso de prueba eficaz puede servir como su última línea de defensa para asegurar que todas sus normas y directrices son aplicadas en un desarrollo personalizado. También puede ayudar a evitar los problemas relacionados con las personalizaciones, en particular para los problemas comunes que se pueden detectar. Un proceso de prueba da un proceso formal para asegurar que las personalizaciones son de calidad, y esto le puede dar la confianza en la sostenibilidad de su servidor SharePoint, incluso después de añadir varios componentes personalizados a la misma. Tras un proceso de pruebas formales también puede establecer las expectativas de sus desarrolladores, para que sepan el baremo de calidad que tienen que cumplir a medida que desarrollan. Se puede diseñar su proceso de pruebas para que tome la forma de un proceso sistemático para revisar las solicitudes, validación de las normas, y luego trabajar a través de un procedimiento de gestión de cambios. Seguir un proceso disciplinado, mejorará la estabilidad general en la granja de SharePoint. Este resultado positivo se consigue tomando decisiones conscientes y coordinadas entre diferentes roles - dos actividades tienen un impacto positivo significativo, incluso si las pruebas reales no son demasiado sofisticadas o completas.

Expongo un ejemplo de un proceso de prueba para desarrollos personalizados:

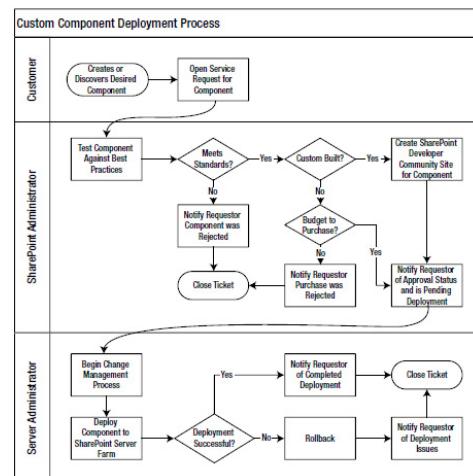


Imagen 3.- Proceso de pruebas para desarrollos personalizados.

Lo primero y más fácil de testear en su proceso de pruebas son las pruebas de las normas y directrices que hemos definido, y esto es especialmente más fácil si hemos automatizado estas pruebas a través de una secuencia de comandos. Por ejemplo, se puede utilizar las clases del espacio

de nombres Reflection de .NET para obtener información sobre los ensamblados cargados y los tipos definidos dentro de ellos para probar si utiliza conexiones de base de datos o cualquier otra cosa. Esto puede ayudarnos a crear un informe rápido de las áreas que desea resaltar y realizar un seguimiento con el equipo de desarrollo. El ejemplo de código siguiente se utiliza la reflexión para poner a prueba un ensamblado y recorrer todos los ensamblados a los que hace referencia:

```
Assembly assembly = Assembly.LoadFile("EjemploComponente.dll");
foreach (AssemblyName a in assembly.GetReferencedAssemblies())
{
    // Código
}
```

El desarrollo basado en pruebas (TDD) es una práctica de desarrollo ágil, donde los desarrolladores codifican un test unitario por función, antes de codificar la funcionalidad global. La idea es que si siempre construir la prueba primero, entonces tendremos una prueba por cada método de funcionalidad en nuestra aplicación personalizada cuando terminemos el desarrollo.

No se tiene que capacitar a estos usuarios con todas las capacidades disponibles para un administrador de colección de sitios

El desarrollo basado en pruebas generalmente sigue un ciclo de adición de pruebas en el que, inicialmente, se producirá un error (debido a que la funcionalidad en la aplicación aún no existe), y luego se codifica la funcionalidad y la lógica para obtener la prueba para pasar. Por último, limpiar el código por refactorización tanto el de la aplicación como la prueba unitaria. Este proceso se resume a menudo como “rojo, verde, Refactor.” resumo este proceso en los siguientes 7 pasos:

- Escribir una prueba unitaria para probar una parte específica de la funcionalidad o condición.
- Ejecutar la prueba para verificar no pasa la prueba (rojo).
- Tan pronto como sea posible, escribir la cantidad mínima de código necesario para hacer que pase la prueba.
- Ejecutar la prueba para verificar la validez de la prueba (verde).
- Refactorice su código para eliminar la duplicación, ineficiencias, las convenciones pobres de nomenclatura, o cualquier otra cosa que refactorizar y mejorar (Refactor).
- Ejecutar todas las pruebas unitarias para asegurar que todas las pruebas aún pasan.
- Repetir este proceso para la siguiente parte de la funcionalidad que desee añadir.

A continuación, expongo un ejemplo de prueba unitaria:

```
[TestMethod]
public void SiteTitleTest()
{
    string title = "título del sitio";
    MockWeb web = new MockWeb();
    SiteConfigurer c = new SiteConfigurer();
    c.UpdateTitle(title, web);
    Assert.AreEqual<string>(title, web.Title);
    Assert.IsTrue(web.HasCalledUpdate);
}
```

En este punto, la prueba sería un fracaso, porque aún no he implementado la funcionalidad. El siguiente código proporciona un ejemplo de implementación de la funcionalidad de esta hipotética clase. Nótese que he utilizado la interfaz IWeb para el parámetro del sitio.

```
public class SiteConfigurer
{
    public void UpdateTitle(string title, IWeb web)
    {
        web.Title = title;
        web.Update();
    }
}
```

Por motivos de integridad, el siguiente código muestra un ejemplo de envolver el objeto real de SharePoint en un objeto que implementa la interfaz. En el código de la aplicación, este objeto envoltorio sería el objeto que pase al método; mientras tanto en mi código de pruebas unitarias, paso el objeto “falso”. Esto funciona porque ambos comparten el mismo interfaz.

```
public class Web : IWeb
{
    SPWeb web;
    public Web(SPWeb web)
    {
        this.web = web;
    }
    public Title
    {
        get { return web.Title; }
        set { web.Title = value; }
    }
    public void Update()
    {
        web.Update();
    }
}
```

Conclusiones

En estos tres artículos sobre gobernanza, he intentado abarcar algunas de las medidas que se pueden tomar para gobernar el entorno de SharePoint. La documentación puede ayudar a establecer un entendimiento compartido de cómo su servicio de SharePoint se ejecuta y las acciones que se necesitan para gobernarlo. Os animo a considerar muchas de las plantillas y normativas de orientación sobre la documentación de artefactos de gobierno, debido a que muchos de ellos podrían tener partes interesantes y valiosas que se pueden adoptar y adaptar.

FRANCISCO RICARDO GIL GONZÁLEZ

MVP CLUSTER | Especialista en SharePoint & Office 365

francisco.gil@fiveshareit.es

<https://es.linkedin.com/pub/ricardo-gil-gonzalez/63/99b/689>

<http://www.mvpccluster.es>

Nosotros



Alberto Diaz

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

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

Email: adiazcan@hotmail.com

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

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



Fabián Imaz

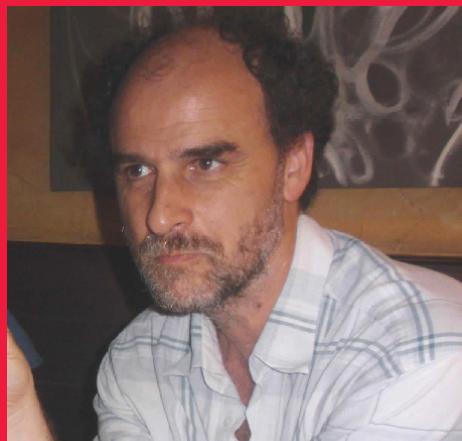
Fabián Imaz, MVP de SharePoint Server trabaja en el mundo del desarrollo de software desde hace más de 10 años, teniendo la suerte de trabajar en distintas arquitecturas y tecnologías Microsoft. Pertenece a la firma Siderys, <http://www.siderys.com> empresa de desarrollo de Software especializada en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

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

Email: fabiani@siderys.com.uy

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

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



Gustavo Velez

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

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

Email: gustavo@gavd.net

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



Juan Carlos González Martín

Juan Carlos González Martín. Ingeniero de Telecomunicaciones por la Universidad de Valladolid y Diplomado en Ciencias Empresariales por la Universidad Oberta de Catalunya (UOC). Cuenta con más de 11 años de experiencia en tecnologías y plataformas de Microsoft diversas (SQL Server, Visual Studio, .NET Framework, etc.), aunque su trabajo diario gira en torno a las plataformas SharePoint & Office 365. Juan Carlos es MVP de Office 365 desde 2015 (anteriormente fue reconocido por Microsoft como MVP de SharePoint Server desde 2008 hasta 2015), coordinador del grupo de usuarios .NET de Cantabria (Nuberos.Net) y co-fundador del Grupo de Usuarios de SharePoint de España (SUGES, www.suges.es), del Grupo de Usuarios de Cloud Computing de España (CLOUDES) y de la Comunidad de Office 365. Desde el año 2011 participa junto con Gustavo Vélez y Fabián Imaz en la dirección de la revista CompartiMOSS (www.compartimoss.com). Hasta la fecha, ha publicado 6 libros sobre SharePoint & Office 365, así como varios artículos en castellano y en inglés sobre ambas plataformas. Actualmente es Cloud & Productivity Advisor en MVP CLUSTER.

Email: jcgonzalezmartin1978@hotmail.com

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

¿Desea colaborar con CompartiMOSS?



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

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

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

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

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

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

revista@compartimoss.com

adiazcan@hotmail.com

fabiani@siderys.com.uy

jcgonzalezmartin1978@hotmail.com

gustavo@gavd.net

