

COMPARTIMOSS

Revista especializada en Tecnologías SharePoint



LA EVOLUCIÓN DE OFFICE DELVE, UN FUTURO PROMETEDOR

MIGRACIÓN VS ACTUALIZACIÓN. ¿QUÉ HAGO CON MI SHAREPOINT?

¿QUÉ SABEMOS DE SHAREPOINT 2016?

INTEGRACIÓN DE POWERSHELL LOCAL CON POWERSHELL DE AZURE



02

COMPARTIMOSS

Revista especializada en Tecnologías SharePoint

Staff

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

DIRECCIÓN GENERAL

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

Contacte con nosotros

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

BLOGS

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

REDES SOCIALES

Facebook:

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

LinkedIn:

<http://www.linkedin.com/groups/CompartiMOSS-3776291>

Twitter:

@CompartiMOSScom

Contenidos

03

Editorial

Tal como lo prometimos en el número anterior, CompartiMOSS ha comenzado a explorar las fronteras de “La Nube” de Microsoft.

04. La evolución de Office Delve, un futuro prometedor

09. Active Directory Authentication Library (ADAL) y su aportación al desarrollo en Office 365

14. Búsqueda de datos empresariales en SharePoint 2013 – II

19. Entrevista - Juan Manuel Herrera Ocheita

21

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

23. Migración Vs Actualización. ¿Qué hago con mi SharePoint?

25. Provider-Hosted Aps + Remote Provisioning (Parte I)

28. Nuevo modelo de desarrollo para SharePoint: El tiempo de las apps y APIs de cliente.

33

¿Qué sabemos de SharePoint 2016?

35. Movilidad sobre SharePoint - Parte II

40. Protegiendo VMWare con Azure Site Recovery

43. ¿Hasta dónde podemos llegar con Nintex y SharePoint? – Análisis y casos prácticos

47. La importancia de diseñar y aplicar un buen uso de Gobernanza en nuestro entorno SharePoint 2013 – Parte II

52. La unión de dos grandes: Azure Machine Learning y PowerBI

55. Integración de PowerShell local con PowerShell de Azure



03

Editorial

Tal como lo prometimos en el número anterior, CompartiMOSS ha comenzado a explorar las fronteras de “La Nube” de Microsoft. En el número que estamos presentando hoy hemos incluido varios artículos relacionados tanto con Office 365 (y su implementación de SharePoint 365), como de Azure. Pero por supuesto no nos olvidamos, ni nunca nos olvidaremos de SharePoint OnPremises, aunque Microsoft nos empuja cada vez a que lo dejemos de lado.

Las últimas semanas han estado marcadas por la información que Microsoft liberó en la conferencia Ignite. Si el futuro cercano de SharePoint se pudiera resumir en un párrafo, podríamos decir que los próximos años estarán enmarcados en “Sistemas Híbridos”. Aunque Microsoft hasta hace poco tiempo había dado señales de que la próxima versión de SharePoint (2016) OnPremises sería la última, y de ahí en adelante solamente existiría SharePoint 365, en Ignite mostraron un cambio radical de opinión y aseguraron que seguirían entregando nuevas versiones OnPremises cada cuantos años. Y aunque esto ha sido un alivio para muchos de nuestros clientes que no quieren o pueden moverse a La Nube, se han abierto algunas nuevas inquietudes, como por ejemplo cual será el futuro de Yammer (el que parece bastante incierto en este momento, aunque Microsoft no haya dicho nada al respecto).

Hablar sobre el futuro de SharePoint en este momento es pura y simple especulación: probablemente Microsoft mismo no sabe cuál será. Así que la única seguridad de que disponemos ahora mismo es que ustedes, nuestros lectores, disfrutarán leyendo este número, tanto como los autores de los artículos, y nosotros, los editores, hemos disfrutado creándolo.

EL EQUIPO EDITORIAL DE COMPARTIMOSS

La evolución de Office Delve, un futuro prometedor

A estas alturas la mayoría de nosotros conocemos ya a Office Delve, un actor relativamente nuevo en la nube de aplicaciones de Office 365 que viene a romper el paradigma tradicional de los portales de información, mostrando la información que “el sistema” entiende que es relevante para cada uno de nosotros en forma de “cards” visuales que hacen de Delve una plataforma dinámica y atractiva de navegar.

Para ser francos, mi primera experiencia con Office Delve no fue del todo satisfactoria por la escasez de contenidos y rigidez de los mismos, así como unos “acabados” un tanto cuestionables en algunos puntos de la aplicación. Sin embargo debo decir que mi interés por Delve crece a medida que voy viendo cómo la plataforma va evolucionando a ritmos acelerados para ir ofreciendo cada vez más y más contenidos provenientes de un número de fuentes también en crecimiento.

Para recordar muy rápidamente las “tripas” de Delve, en un artículo de Javier Menéndez Pallo del número 22 de COMPARTIMOSS ya se explicó en detalle cómo Office Graph, haciendo uso de ciertos algoritmos base definidos con Machine Learning, consigue relacionar los contenidos relativos a un usuario y extrapolar la relevancia y actualidad de los mismos para mostrarlos directamente en Office Delve.



Imagen 1.- Esquema gráfico del algoritmo que usa Graph para mostrar al usuario las cards de Delve.

Actualmente Delve extrae información de otras fuentes de Office 365, tales como SharePoint Online, Yammer, Exchange, Office Video u OneDrive. En el futuro ya han anunciado que también se integrará con otras fuentes como Skype for Business o entornos híbridos de SP 2016, pero lo más interesante es que ya han sido liberadas las “Graph Query API’s” que permitirán construir apps a medida mediante conexiones a LOB y servicios de terceros. Así pues no hay límites sobre de qué fuente puede informar Office

Delve, y eso le otorga un gran abanico de posibilidades y potencialidad.

De todas esas fuentes, Graph extrapola la información relevante (diferente para cada persona) estableciendo relaciones sobre a quién estoy siguiendo, a qué grupos pertenezco, quién me ha contestado un mail, post o conversación que yo he iniciado, quien ha modificado un documento que yo he creado, a quién reporto, qué tareas tengo asignadas, etc. Y con toda esa información nos muestra pantallas de “cards” tan dinámicas y atractivas como las que muestra la siguiente imagen.

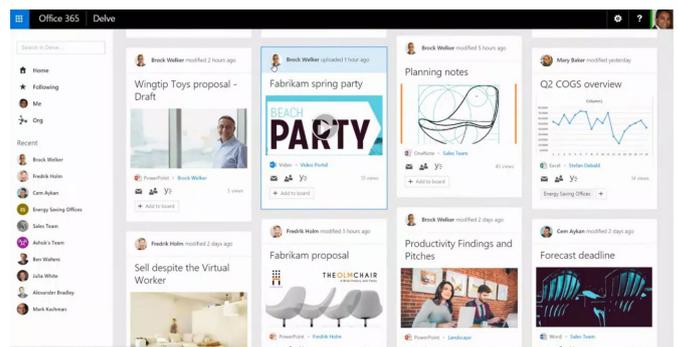


Imagen 2.- Contenidos Delve provenientes de diversas fuentes (SharePoint Online, Office Video...)

Además, dejemos claro que Office Graph siempre tiene en cuenta los permisos de la fuente de información, mostrando al usuario únicamente aquella información a la que tiene acceso. Si alguien nos envía un documento adjunto en un mail privado, únicamente nosotros podremos ver el documento, y si se publica información en un grupo privado de Yammer, únicamente los integrantes del grupo la visualizarán en Office Delve.

Perfil del usuario

En Office Delve se puede consultar el perfil tanto de nosotros mismos como de cualquier persona en nuestra granja de Office 365. Esta es una característica que ya está actualmente disponible, pero que ha evolucionado respecto a su versión original y sigue teniendo previsto una mayor evolución en próximas versiones, incluyendo nuevos componentes como el “Blog”, “Working On” o los premios recibidos “Praises”, y redistribuyendo el espacio en diversas columnas y secciones, así como una personalización de la cabecera de nuestro perfil (banner de fondo). Personalmente siempre he pensado que el profile del “My Site” estaba mal resuelto, y aquí tenemos una versión más ami-

gable y moderna de presentar la información relevante de cada usuario

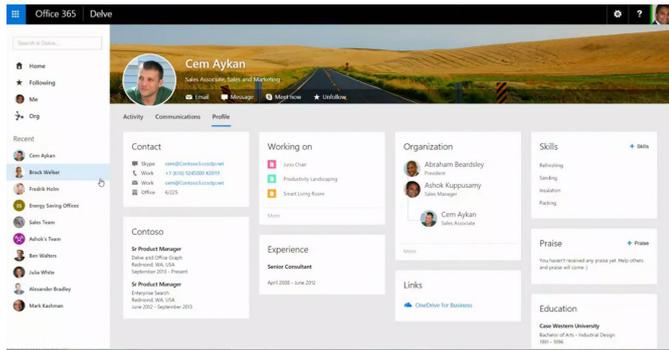


Imagen 3.- Ejemplo de página de perfil de un usuario.

Blog

Microsoft ha mostrado en Ignite otra característica que estará disponible en breve para Office Delve y que es ciertamente interesante, se trata de la posibilidad de mantener nuestro propio blog directamente desde Delve. La diferencia con los actuales blogs de SharePoint es que estos tirarán de un nuevo motor de creación de páginas con contenido enriquecido (imágenes, enlaces, contenido multimedia...) con una interface nueva y muy sencilla de manejar. A esta nuevo motor de creación de páginas web le llaman "microsites" y lo usaremos en Delve para crear posts de nuestros blogs desde la página de perfil de nuestro usuario, simplemente haciendo click en el enlace de "empezar a escribir" que se habilitará a tal efecto. Microsites también se ha anunciado para otros componentes que estarán presentes en el futuro de Office 365, tales como "InfoPedia" (antes conocida como bibliotecas de páginas wiki en SharePoint) y otras muchas que seguro iremos oyendo hablar durante los próximos meses.

Una de las características destacadas de los microsites es que serán responsive y por lo tanto se podrán visualizar correctamente desde cualquier dispositivo móvil. Esta funcionalidad en sí misma justifica dejar de usar los Blogs de SharePoint (al menos hasta que tengamos acceso a la versión 2016) y dar el salto a Office Delve.

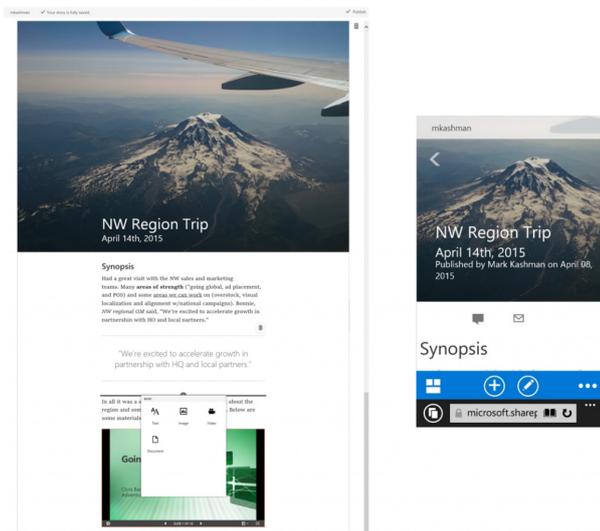


Imagen 4.- Creación de un post (sencillo y rápido) desde Delve con microsites. Mobile friendly.

Seguir a personas

En el momento de escribir este artículo todavía no estaba disponible esta opción, pero el equipo de Delve ha anunciado que cuando en Delve busquemos a una persona en concreto y vayamos a su página de perfil, podremos hacer un seguir "follow" o dejar de seguir "unfollow", siendo este uno de los parámetros básicos para que el algoritmo de Graph nos muestre resultados de dicha persona como relevantes en nuestra homepage de Delve.

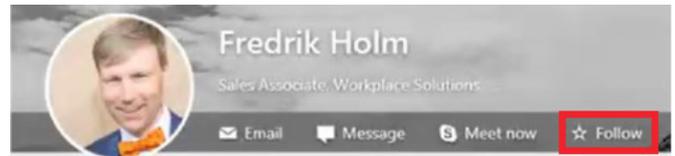


Imagen 5.- Desde la página de perfil de un usuario se podrá seguir/dejar de seguir al mismo.

Las personas que seguimos y visitamos más frecuentemente serán las que tengamos acceso rápido desde el menú de navegación vertical izquierdo.

mi primera experiencia con Office Delve no fue del todo satisfactoria por la escasez de contenidos y rigidez de los mismos, así como unos "acabados" un tanto cuestionables en algunos puntos de la aplicación

Actividades de personas que sigo

Office Delve no solo nos muestra actividades en nuestra página principal, también permite ir a las páginas de actividades de cualquier otra persona y ver qué actividades son relevantes para él/ella (siempre que tengamos permisos adecuados para visualizar dicho contenido, claro está). El tipo de "cards" que veremos serán el mismo que en nuestra página principal, obtenidas de las mismas fuentes de información.

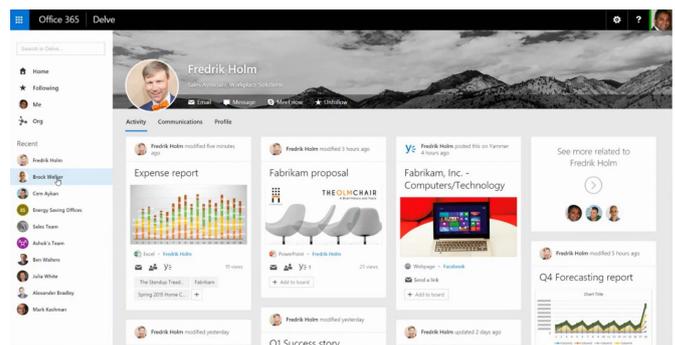


Imagen 6.- Ejemplo de página principal de actividades de un usuario al que estoy siguiendo en Delve.

Ver más contenido relacionado de...

Otra funcionalidad que actualmente podemos encontrar en Delve es el acceso a “ver más contenido relacionado de...” que nos mostrará las personas vinculadas con ese usuario en concreto, además de más “cards” de actividades que no se mostraban en la página principal.

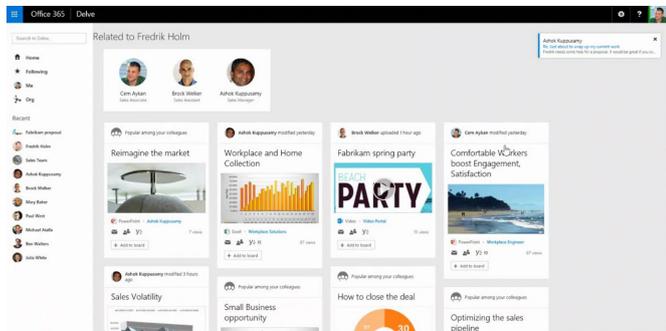


Imagen 7.- Ejemplo de página de “más a cerca de...” en Office Delve.

Premios

Esta es otra característica muy interesante que está por llegar a Office Delve, y se basa en que se podrá otorgar premios (felicitaciones) específicos a los usuarios de Office 365, y ser visualizados desde la página de perfil de los mismos, así como hacer un “me gusta” o “no me gusta” del mismo. El concepto de gamificación llega también a Delve, dispuesto a dinamizar y fomentar la calidad de contenidos y la participación del resto de usuarios.

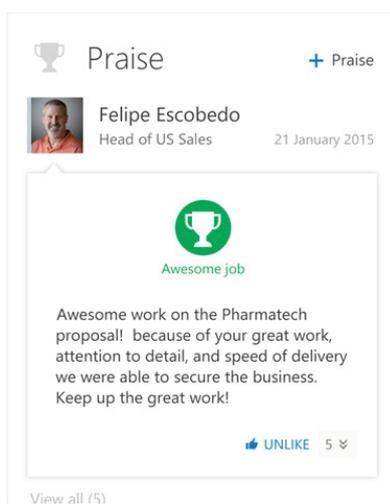


Imagen 8.- Visualización de un premio concedido a un usuario desde su página de perfil.

Trabajando en

De nuevo otro componente que todavía está por llegar en Office Delve, es el “trabajando en” que se acoplará en la página de perfil de cada usuario. De nuevo los algoritmos basados en machine learning de Graph serán capaces de dilucidar en qué proyectos estamos trabajando, basándose en los últimos documentos que hemos subido, actualizado, seguido, comentado... en los grupos de Office 365, SharePoint y Yammer. El componente mostrará de forma directa en qué proyectos hemos registrado mayor activi-

dad durante los últimos días.

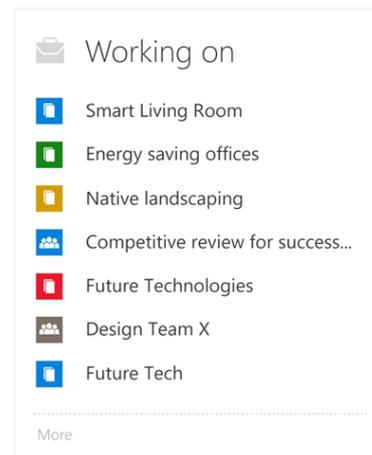


Imagen 9.- Visualización del futuro “Trabajando en” de un usuario desde su página de perfil.

Comunicaciones

La pestaña de comunicaciones todavía no estaba disponible en el momento de la redacción del presente artículo, aunque Microsoft la mostró ampliamente en Ignite y se aprecia una nueva funcionalidad muy sugerente, como la posibilidad de leer nuestros últimos mails y comentarios de Yammer, así como responderlos directamente desde Delve mediante una interface muy intuitiva y sencilla. De nuevo vemos aquí actuar a Delve como front-end unificador (centralizador) de las diversas piezas que componen la nube de Office 365, y es en estos detalles donde podemos dilucidar lo que pretende Microsoft realmente con esta plataforma: Que el usuario la utilice en primera instancia para comprobar el estado de su trabajo diario, y a partir de aquí, si se quiere profundizar en algún aspecto concreto derivar a la aplicación que corresponda. Entendemos además que mediante la propia evolución del producto y las API's que vayamos teniendo acceso podremos ir añadiendo más fuentes de contenido a la pestaña de comunicaciones (Skype for Business, por ejemplo).

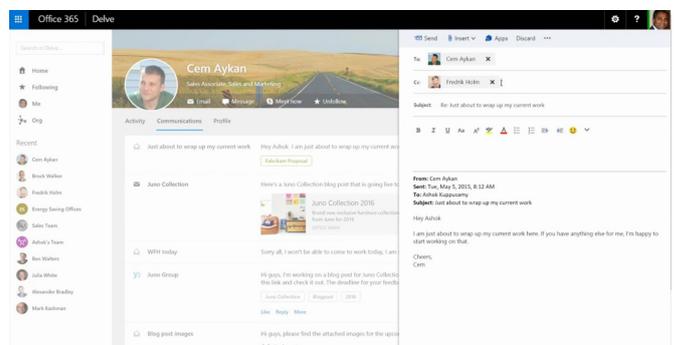


Imagen 10.- Contestando un mail directamente desde la página de comunicaciones de Delve.

Actualmente Delve extrae información de otras fuentes de Office 365, tales como SharePoint Online, Yammer, Exchange, Office Video u OneDrive

Boards

Durante la navegación por Delve, el usuario puede seleccionar cualquier actividad (card) y enviarla a un Board (nuevo o existente). Esto permite algo inaudito hasta ahora, que es poder agrupar en una única página diversos ficheros o contenidos provenientes de fuentes tan dispares como SharePoint, Yammer, Outlook, Office Video, Skype... y guardar ese contenedor de información relacionada para compartirlo con quien queramos. Imaginad por ejemplo que os llega un correo electrónico con un enlace a Delve que contiene diversos documentos en Word, PDF, un video formativo, una conversación de Yammer... al respecto de un proyecto en el que acabamos de aterrizar.

Además los "Boards" se mostrarán automáticamente en el panel de navegación izquierdo de Office Delve si los algoritmos de Graph detectan que tienen algún tipo de vinculación con el usuario (lo ha creado una persona a la que seguimos o hemos sido autores de alguno de los documentos que incluye el Board, por ejemplo).

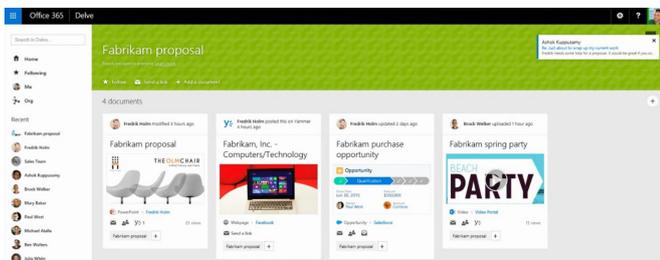


Imagen 11.- Ejemplo de diversos contenidos agrupados en un "Board".

Organizational Dashboards

Otra de las futuras novedades de Delve presentadas en Ignite es el nuevo Dashboard de organización ("Organizational Dashboard"), que muestra gráficas y números estadísticos muy visuales y atractivos sobre la actividad que hemos realizado desde Delve y Skype. Por ejemplo permite establecer un mapa de relaciones de nuestro equipo con otros equipos (tanto corporativos como externos). En dicho mapa, el grueso o fino de las líneas dibujadas indican lo fuerte o débil que son estas relaciones (relaciones fuertes se marcan con líneas gruesas, y viceversa).

También permitirán saber de forma muy visual en qué hemos invertido nuestro tiempo de trabajo (emails, meetings, videoconferencias de Skype, conversaciones de Yammer), o incluso con qué países del mundo nos comunicamos más según el número de interacciones registradas en cada una de ellas.



Imagen 12.- Ejemplo de un "Organizational Dashboard" de Office Delve.

Mobile

Otra de las grandes novedades de Office Delve es que ya está mundialmente disponible la App para móviles Android, se puede descargar directamente de Play Store, y disfrutar de la experiencia responsive en nuestro dispositivo móvil. Es indudable también que esta App irá evolucionando y ofreciendo nuevas funcionalidades a medida que evolucione también papá Delve en Office 365.



Imagen 13.- Diversas pantallas de la versión App para móviles ya existente de Office Delve.

En Office Delve se puede consultar el perfil tanto de nosotros mismos como de cualquier persona en nuestra granja de Office 365

Grupos de Office 365

Los grupos de Office 365 fueron lanzados hace ya unos meses, y aunque en principio resulten algo confusos en su concepto (¿Una bandeja de entrada más en nuestro buzón de correo para hacer algo similar a Yammer?) en Ignite se vio que estos también van a seguir evolucionando para ofrecer una experiencia integradora entre diversas fuentes de información, añadiendo nuevos orígenes de datos (como Dynamics CRM), o integraciones desde el propio Outlook cliente que pueden resultar interesantes. También Office Delve va a ser capaz de mostrarnos páginas de actividades relativas a grupos, con sus "cards" específicas para los mismos, tratándolos como si de un usuario más se tratara.

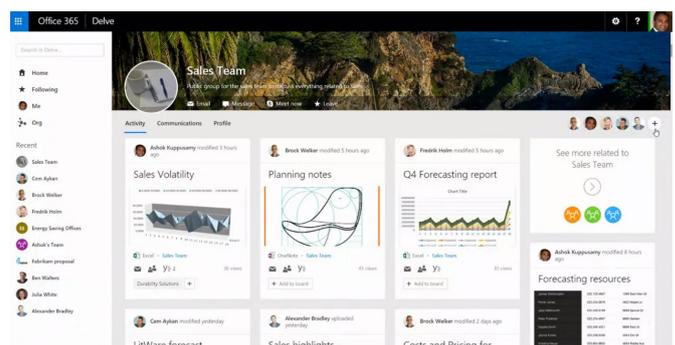


Imagen 14.- Ejemplo de página principal de un grupo de Office 365 visualizado en Office Delve.

Perfil de grupos de Office 365

También hay que tener en cuenta que los grupos de Office 365 también van a tener su propia página de perfil, donde podremos ver por ejemplo los miembros del grupo que están conectados en el momento de la consulta, los siguientes eventos en la agenda del grupo, las conversaciones de email mantenidas, o los ficheros del grupo que hayamos podido guardar en OneDrive, así como las notas de OneNote relativas al mismo. Sin duda le da mucha más vistosidad a los Grupos de Office 365 que hasta el momento eran bastante precarios en cuanto a contenidos y presentación. Atención, ¿Estamos viendo nacer a un futuro rival interno de Yammer?

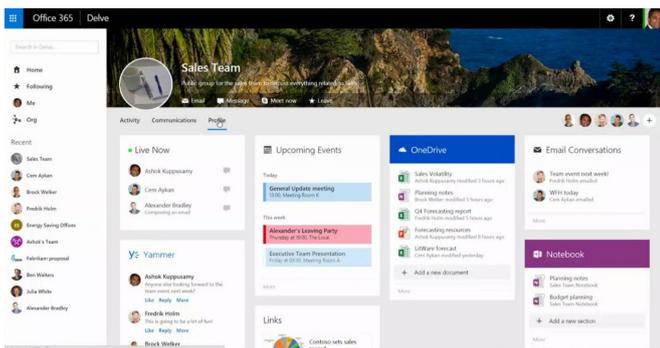


Imagen 15.- Ejemplo de página de perfil de un grupo de Office 365 visto desde Office Delve.

Conclusiones

Hasta aquí la ráfaga de novedades conocida a día de hoy sobre Office Delve. Como veréis son unas cuantas novedades importantes y la sensación que destila es que es un

producto que está siendo mimado por Microsoft (al igual que toda la plataforma de Office 365). No podemos decir lo mismo de SharePoint 2016, del que apenas se vieron unas cuantas diapositivas explicando algunas novedades en cuanto a arquitectura y servicios que reflejan lo verde que se encuentra todavía la nueva versión de SharePoint.

En conjunto está claro que con Office Delve nos encontramos con un producto que Microsoft va a intentar usar como centralizador o catalizador de la información del resto de aplicaciones en la nube y también en entornos híbridos, incluyendo SharePoint, por supuesto. El hecho de que las Graph Query API's permitan a aplicaciones de terceros enviar "cards" de acciones a Delve no hace sino incrementar las posibilidades de la plataforma.

¿Por dónde empezar a mirar cuando uno tiene SharePoint, Office Video, Skype for Business, Yammer, Exchange, Dynamics CRM, Power BI, One Drive...? Sin duda alguna Delve aparece como "el pegamento" para unirlos a todos en una única experiencia, y tampoco cabe duda que Office Delve seguirá evolucionando a buen ritmo durante los próximos meses, así que os recomiendo no le perdáis el ojo, porque puede acabar convirtiéndose en una plataforma realmente útil e interesante como puerta de entrada a la nube de Office 365.

IGNASI TEBÉ

SharePoint Professional

ignasitt@gmail.com

<http://www.sharepointer.info>

Active Directory Authentication Library (ADAL) y su aportación al desarrollo en Office 365

Mucho se está hablando del giro que se está produciendo en Microsoft desde la llegada de Satya Nadella al puesto de CEO de Microsoft. Uno de sus principales objetivos es hacer posible que todos los productos de su marca aumenten la productividad de las empresas desde cualquier dispositivo y en cualquier lugar, lo que bautizó como "FIRST CLOUD, FIRST MOBILE". Esta afirmación radicaba naturalmente en la mejora de su plataforma Cloud (Azure) y en su plataforma de productividad (Office 365).

Desde los últimos meses la comunidad de desarrolladores de Office 365 ha ido creciendo cada vez más y más, liderada de la mano de grandes gurús, MVPs o personal de Microsoft como Andrew Cornell, Jeremy Thake, Vesa,... De esta Comunidad ha surgido un proyecto Open Source llamado Patterns and Practice (PNP) para servir de adopción/evangelización a los desarrolladores de Office 365. Quizás sea el proyecto estrella y el que más ha evolucionado en los últimos meses. Pero para que cumplir el objetivo de poder consumir servicios de Office 365 en cualquier dispositivo/plataforma ha sido muy importante la creación de una librería como ADAL. El funcionamiento de esta librería es abstraer al desarrollador de la forma en la que se tiene que autenticar contra Office 365 y simplifica mucho la tarea al desarrollador.

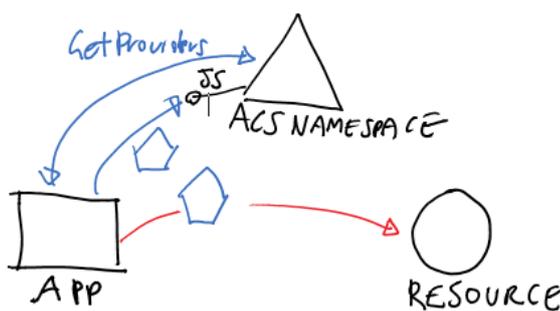


Imagen 1.- Funcionamiento de ADAL.

¿Qué es Active Directory Authentication Library (ADAL)?

Active Directory Authentication Library es una librería tanto de .NET como JavaScript cuyo funcionamiento es abstraer al programador de la forma en la que nuestra aplicación se autentifica contra Office 365. No es una librería nueva, ya tiene más de dos años desde que salió su primera versión. Esta librería se puede utilizar tanto en desarrollos para IOS, Android como la nueva versión de ASP.NET

llamada vNEXT.



Imagen 2. - First Cloud First Mobile.

Principal patrón de ADAL

La autenticación es un proceso bastante difícil, engorroso y sobre todo muy importante para la aplicación. Hay que conocer todo el proceso y verificar que se está haciendo correctamente, y no podemos perder de vista muchos aspectos para poder autenticarnos. Los formatos de Token, qué protocolo a utilizar para una topología dada, qué parámetros de trabajo para un proveedor de identidad, pero no para el otro, cómo prevenir al usuario de que se le pregunte cada vez, cómo evitar guardar contraseñas y secretos, qué hacer cuando se necesita múltiples factores de autenticación, y muchos detalles más oscuros. Utilizando ADAL nos hace centrarnos en una solución a alto nivel tan simple como:

- Tengo una aplicación cliente
- Quiero llamar a un servicio, pero me obliga a presentar una ficha

De esta Comunidad ha surgido un proyecto Open Source llamado Patterns and Practice (PNP) para servir de adopción/evangelización a los desarrolladores de Office 365

Si queremos saber en detalle cual es el proceso de Autenticación que hace nuestra aplicación para consumir información de Office 365 podemos revisar artículos anteriores donde se añaden más detalle. <http://www.compartimoss.com/revistas/numero-22/Como-utilizar-API-Office-365-aplicaciones-multiplataformas-Xamarin>

Escenarios donde podemos utilizar ADAL

1. Aplicación Cliente Nativa

Este es el caso en el que se está desarrollando una aplicación nativa (todo lo que no es un navegador y que puede mostrar la interfaz de usuario, que incluye aplicación de consola) y desea acceder a un recurso con un token obtenido como el usuario interactivo actual. Para implementarla se puede utilizar un código similar al siguiente:

```
AuthenticationContext _authenticationContext =
    new AuthenticationContext("https://login.windows.net/mytenant.onmicrosoft.com");
AuthenticationResult _authenticationResult =
    _authenticationContext.AcquireToken("http://myservices/service1",
        "a8cb2a71-da38-4cf4-9023-7799d00e09f6",
        new Uri("http://TodoListClient"));
```

La primera línea crea una AuthenticationContext atada al inquilino AAD "mytenant". La segunda línea pide que mytenant emitir un token para el servicio con identificador "http://misServicios/service1", se comunica que la solicitud proviene de cliente con ID a8cb2a71-da38-4cf4-9023-7799d00e09f6 y pasa en el retorno URI (utilizado como un terminador en el flujo contra el punto final autorización: un detalle protocolo OAuth2).

La autenticación es un proceso bastante difícil, engorroso y sobre todo muy importante para la aplicación

Si esta es la primera vez que ejecutamos ese código, el efecto de la llamada a AcquireToken será para que salga un cuadro de diálogo navegador recopilación de las credenciales de usuario:

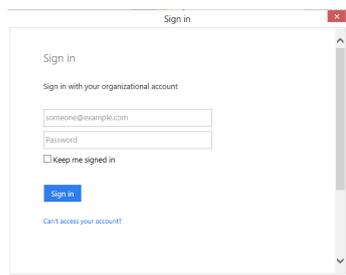


Imagen 3.- Pantalla de Autenticación

ADAL maneja la creación del diálogo. Este diálogo es en realidad sólo una superficie del navegador y la autoridad decida qué enviar: aquí especificamos el nombre de usuario y contraseña, pero los usuarios específicos podría obtener múltiples factores de autenticación, mensajes de consentimiento y similar. Tras la autenticación exitosa, el AccessToken estará en _authenticationResult. El contenido se almacena en caché _authenticationResult: las llamadas posteriores a AcquireToken utilizando los mismos parámetros se obtendrán los resultados de la memoria caché

hasta que se produzca una expiración.

2. Aplicación Server to Server

Este es el escenario en el que la aplicación cliente en sí tiene sus propias credenciales (independiente de las credenciales de Office 365) y necesita obtener información de algún servicio de Office 365. Para ello lo hace utilizando el estándar OAuth2 (el cual permite que la aplicación pueda acceder a Office 365). Es un protocolo de autorización que permite a terceros (clientes) acceder a contenidos propiedad de un usuario (alojados en aplicaciones de confianza, servidor de recursos) sin que éstos tengan que manejar ni conocer las credenciales del usuario. Es decir, aplicaciones de terceros pueden acceder a contenidos propiedad del usuario, pero estas aplicaciones no conocen las credenciales de autenticación. De todas formas ADAL se encarga de gestionar este protocolo con total transparencia. Para este escenario se utiliza un código similar al siguiente.

```
_authenticationContext =
    new AuthenticationContext("https://login.windows.net/21211b42-1e25-4046-8872-d8832a38099b");

ClientCredential clientCred2 =
    new ClientCredential("2188a797-7d21-41dc-84f3-3c1720262614",
        "HBheXXXXXXXXXXXXXXXXXXXXXXXXXXXXM=");
AuthenticationResult _authenticationResult =
    _authenticationContext.AcquireToken("https://localhost:9001",
        clientCred2);
```

No sólo autentica sino que tiene varias características destacadas

1. Refresca el token de uso: El token obtenido tiene una duración de tiempo determinada y es necesario volver a obtenerlo, con ADAL se encarga automáticamente de este refresco.
2. Caché propia: ADAL viene con una memoria caché de forma predeterminada en memoria que se extiende por el proceso, y que se utiliza de forma automática. Eso caché es totalmente consultable a través de LINQ, y contiene mucho más que información como datos usuario (como identificadores, nombre y apellido, etc.).
3. Protección y validación: ADAL le protege de los recursos que le hagan llegar a las autoridades maliciosos mediante la validación de la URL autoridad contra las plantillas conocidas.

Caso Práctico utilizar ADAL en una aplicación JavaScript (Cordova)

En artículos previos hemos visto cómo utilizar ADAL, en aplicaciones Nativas sobre Windows 8, Android o IOS, así como en ASP.NET 4. Todos estos casos están muy bien, pero tiene algo en común y es que se realiza desde código C# lo cual no nos puede extrañar que tengamos una comunicación correcta con Office 365. Para demostrar que el mensaje que dice Microsoft es real vamos a ver como en

una aplicación móvil implementada con Apache Cordova podemos consultar las API's de Office 365.

Requisitos Previos:

- Visual Studio 2013 Update 3 o Superior / Visual Studio 2015.
- Herramientas para Apache Cordova (el setup incluye herramientas como NodeJS, Sdk de Android,...).
- Office 365 API Tools (<http://visualstudiogallery.msdn.microsoft.com/7e947621-ef93-4de7-93d3-d796c43b-a34f>).

Nota: Dentro de los emuladores que vienen con Cordova hay un plugin de Chrome llamado Riple que emula el comportamiento de un teléfono dentro de un navegador Web. Este emulador no funciona con ADAL debido a que dentro de una aplicación móvil no se espera que se abra una aplicación fuera de su contexto. Para que funcione deberemos de utilizar un emulador bien de Android, IOS o Windows Phone.

Manos a la obra

Abrimos Visual Studio, creamos un proyecto JavaScript-> Blank App (Apache Cordova). A continuación, a nuestro proyecto le vamos a añadir la conexión a los servicios REST de API 365. Para añadirlos, nos dirigimos al Explorador de soluciones, Agregar Servicio conectado tal y como se muestra en la siguiente imagen:

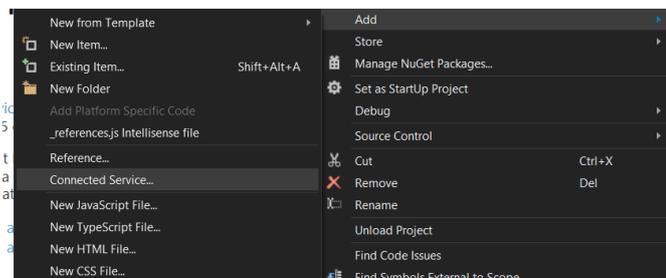


Imagen 4.- Agregamos Servicios a nuestro proyecto

A continuación seleccionamos la API que va a utilizar nuestra aplicación:

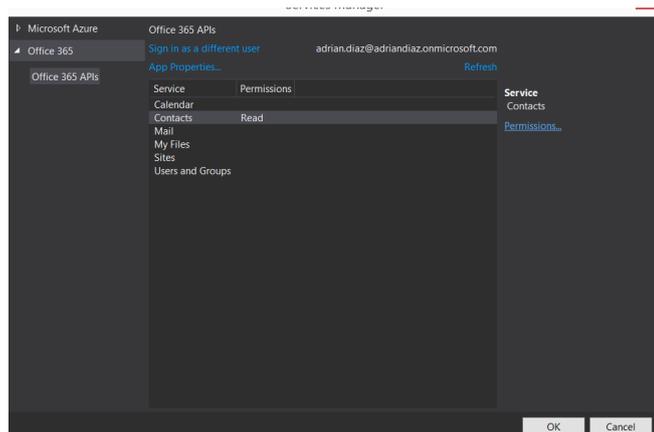


Imagen 5.- Damos permisos sobre la API de Office 365.

Si observamos dentro de nuestro proyecto se ha creado una carpeta dentro de Services-> Office 365 en la que están las funciones necesarias para poder utilizar las API's de Office 365.

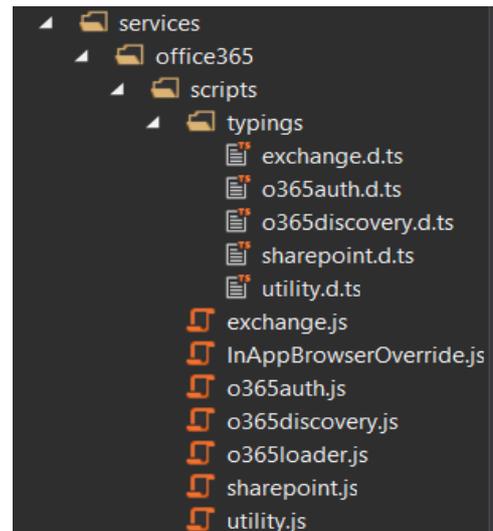


Imagen 6.- Ficheros TypeScript.

Con el fin de minimizar el uso de JavaScript vamos a utilizar un framework como AngularJS. Para ello nos dirigimos al repositorio de Nuget y nos descargamos el paquete Angular Core.

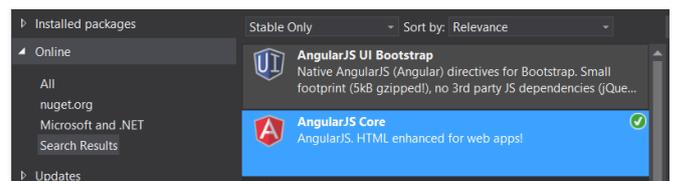


Imagen 7.- Utilización de AngularJS en nuestro desarrollo.

Una vez ya tenemos todas las herramientas preparadas, vamos con la parte más divertida: el código.

El primer paso en toda aplicación Angular es crearnos el punto de entrada de la aplicación, para ellos nos crearemos un fichero App.JS, en este fichero se define el nombre de nuestra aplicación, así como que dependencias tiene (la inyección de dependencias es uno de los aspectos que hacen muy potente a Angular y evitan que el código JavaScript se convierta en código espagueti).

```
var app365 = angular.module('app365', [
    'ngRoute']);
```

```
app365.config(['$routeProvider',
function ($routeProvider) {
    $routeProvider.
        when('/', {
            templateUrl: 'partials/sign-in.html',
            controller: 'signInCtrl'
        });
        when('/home', {
            templateUrl: 'partials/contact-list.html',
            controller: 'contactCtrl'
        });
    });
}
]);
```

A continuación implementamos un Módulo en AngularJS en el que vamos a tener las funciones para autenticarnos contra Office 365. Este módulo contiene las siguientes funciones de login y de logout.

```
function login(callback) {
    if (!authContext) {
        authContext = new O365Auth.Context();
    }

    authContext.getIdToken("https://outlook.office365.com/")
        .then((function (token) {
            // Get auth token.
            authToken = token;
            // Get user name from token object.
            userName = token.givenName + " " + token.familyName;
            // Create Outlook client object.
            outlookClient = new Microsoft.OutlookServices.Client('https://outlook.office365.com/api/v1.0', authToken.accessTokenFn('https://outlook.office365.com'));
            // Callback without parameter to indicate successful sign-in.
            callback();
        })).bind(this, function (reason) {
            // Log sign-in error message.
            console.log('Failed to login. Error = ' + reason.message);
            callback(reason.message);
        });
};

function logout() {
    if (!authContext) {
        authContext = new O365Auth.Context();
    }

    authContext.logout();
};
```

Este ejemplo tendremos dos Controladores uno para la página de Login y otro para mostrar los contactos. El código a utilizar es el siguiente:

```
(function () {
    'use strict';

    function signInCtrl($scope, $location, app365api) {

        $scope.signIn = function () {
            app365api.login(onlogin);
        };

        var onlogin = function (reason) {
            $location.path("/home");
        };
    }

    angular.module('app365').controller('signInCtrl', ['$scope', '$location', 'app365api', signInCtrl]);
})();
```

Una de las grandes ventajas que tiene AngularJS es están bastantes claras las responsabilidades, cumpliendo un patrón MV*, ya hemos implementado los Controladores y ahora añadiremos las vistas necesarias. Para finalizar con el ejemplo tan solo nos queda modificar el HTML inicial, este HTML siguiendo con los patrones de Angular es un HTML sintáctico en el que por un lado tendremos que incrustar determinadas marcas para que Angular pueda aplicarlos, estas marcas comienzan todas por "ng". En nuestro caso quedaría como el siguiente:

```
(function () {
    "use strict";
    function contactCtrl($scope, app365api) {
        var vm = this;
        var outlookClient;

        function getContacts() {
            // Fetch all the contacts.
            outlookClient.me.contacts.getContacts().fetch()
                .then(function (contacts) {
                    // Get the current page. Use getNextPage() to fetch next set of contacts.
                    vm.contacts = contacts.currentPage;
                    $scope.$apply();
                });
        };

        vm.loadList = function () {
            // Get the Outlook client object.
            outlookClient = app365api.outlookClientObj();
            // Get contacts.
            getContacts();
        };

        vm.loadList();
    }

    angular.module("app365").controller("contactCtrl", ['$scope', 'app365api', contactCtrl]);
})();
```

Una de las grandes ventajas que tiene AngularJS es están bastantes claras las responsabilidades, cumpliendo un patrón MV*, ya hemos implementado los Controladores y ahora añadiremos las vistas necesarias

Por entender un poco el funcionamiento de AngularJS es el siguiente:

Encuentra una marca con el atributo “ng-app”, este atributo indica el lugar donde va a arrancar la aplicación. Dependiendo del código introducido en nuestro modulo App.js realizaremos una acción u otra. En este ejemplo hemos indicado que cuando estemos en la pantalla inicial nos muestre el template de login así como el código del controlador establecido, por poner un símil tenemos el fichero ASPX y su código behind. De esta forma es mucho más sencillo y más fácil tener el código JavaScript organizado.

Conclusión

El desarrollo en Office 365 está evolucionando a pasos de gigante, cada vez hay más desarrolladores sobre esta

plataforma. Una de los principales motivos son todos los esfuerzos que está poniendo Microsoft para intentar que desarrollar sobre cualquier plataforma propia se pueda realizar con las herramientas que utiliza la mayoría de la gente.

Adrián Diaz Cervera

SharePoint - Architect at Encamina

MVP SharePoint Server

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

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

adiaz@encamina.com @AdrianDiaz81

KWizCom Forms App

Solución para formularios 100% nativos en SharePoint

Mejore a SharePoint, no lo reemplace...



www.kwizcom.com

Búsqueda de datos empresariales en SharePoint 2013 – Parte II

En muchas ocasiones necesitamos presentar en nuestro sitio de búsqueda datos empresariales de otros orígenes (fuera de SharePoint) como pueden ser base de datos (SQL Server, Oracle, MySQL), ERP (Dynamics, SAP, ECI, NetSuite), ECi's M1ECi's M1 CRM(Dynamics, SAP, Sugar), etc.

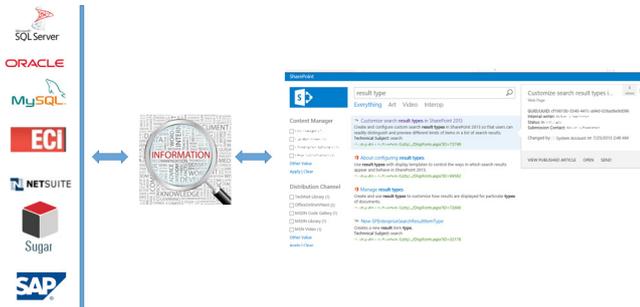


Imagen 1.- Muestra un gráfico ilustrativo para presentar datos de orígenes externos en SharePoint.

Esto es posible mediante la búsqueda de datos externos utilizando los servicios de conectividad empresarial, a continuación vamos a ver paso a paso como realizar dicha integración utilizado como conector un servicio web.

Paso 5 - Creación del origen de datos para su rastreo.

Para indexar el contenido externo es necesario crear un nuevo origen de contenido, para ello seguimos los siguientes puntos:

- Abrimos el formulario principal del servicio de aplicaciones de búsqueda. En el caso de no estar en la página inicial de la administración central pulsamos sobre la imagen de SharePoint, seleccionamos la opción de gestión de servicios de aplicaciones (Manage services applications), posteriormente seleccionar nuestra aplicación de servicio de búsqueda (Search Service Application).

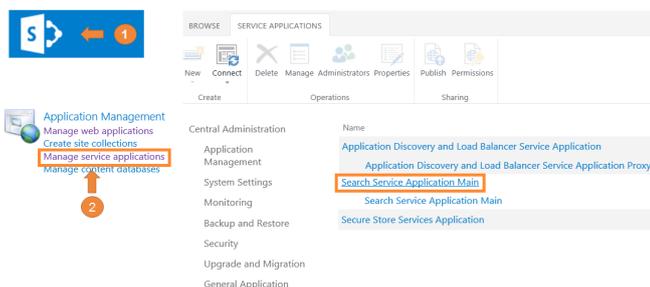


Imagen 2.- Muestra la página de las aplicaciones del servicio en la administración central.

vamos a hablar acerca de cómo integrar PowerShell local con PowerShell de Azure para poder realizar de manera más ágil nuestras operaciones en Azure.

- Abrimos el formulario de origen de contenidos para crear uno nuevo. En el apartado (Crawling) del menú vertical derecho seleccionamos origen de contenido (Content Source), posteriormente pulsamos sobre la opción de crear un nuevo origen de contenido (New Content Source).

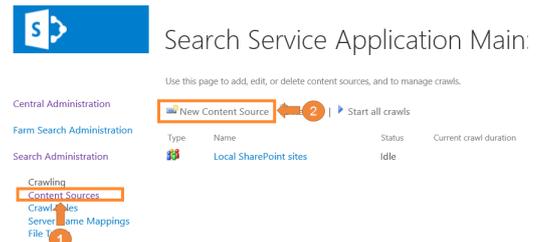


Imagen 3.- Muestra la página de orígenes de contenidos.

- Introducimos los datos solicitados. En la caja de texto de name (Name) introducimos "Proveedores", en el tipo de origen de contenido seleccionamos (Line of Business Data"), en el origen de datos externo (External Data Source) seleccionamos "ServiciosProveedores".

Search Service Application Main: Add Content Source

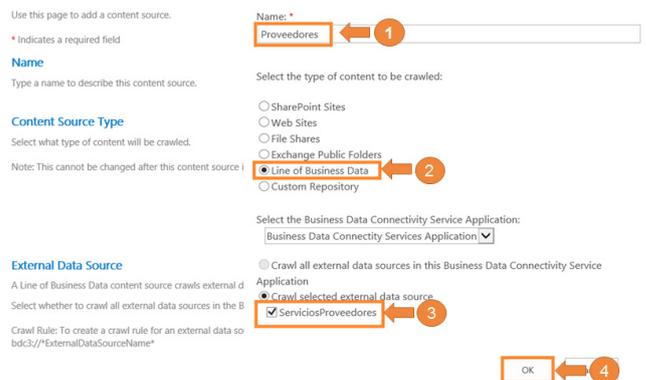


Imagen 4.- Muestra el formulario de nuevo origen de contenido.

- Realizamos un rastreo completo. Para indexar y recoger las propiedades rastreadas es necesario realizar un rastreo completo, para ello pulsamos sobre el botón dere-

cho del ratón sobre el nuevo origen de datos “Proveedores” y seleccionamos inicializar un rastreo completo (Start Full Crawl).

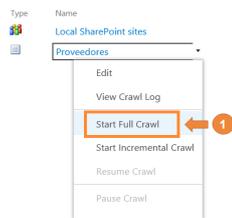


Imagen 5.- Muestra el menú de opciones del origen de contenido.

- Comprobamos el resultado de rastreo. Comprobamos el resultado rastreo para ello pulsando sobre el botón derecho del ratón sobre el nuevo origen de datos “Proveedores” y seleccionamos ve el log de rastreo “View Crawl Log”.

View a summary of items crawled per content source.

Content Source	Average Crawl Duration					Summary			
	Last crawl	Last 24 hours	Last 7 days	Last 30 days	Successes	Warnings	Errors	Top Level Errors	Deletes
Local SharePoint sites	No data	No data	No data	No data	0	0	0	0	0
Proveedores	00:02:06	No data	00:02:06	00:02:06	102	0	0	0	0

Imagen 6.- Imagen P5-5: Muestra el log del rastreo del contenido.

Paso 6 - Creación de las propiedades administradas.

Es necesario incluir las propiedades administrada para posteriormente su utilización en las búsquedas, por ejemplo en el panel de refinamiento, plantillas de visualización de elementos, panel activable, etc. para ello seguimos los siguientes puntos:

- Visualizamos las propiedades rastreada de nuestro tipo de contenido externo. Seleccionamos en el menú vertical derecho la opción de esquema de búsqueda (Search Schema) en el apartado de consultas y resultados (Queries and Results), posteriormente pulsamos sobre la opción propiedades rastreada (Crawled Properties), seleccionamos en categoría (Category) “Business Data” y pulsamos sobre el botón buscar para presentar dichas propiedades.

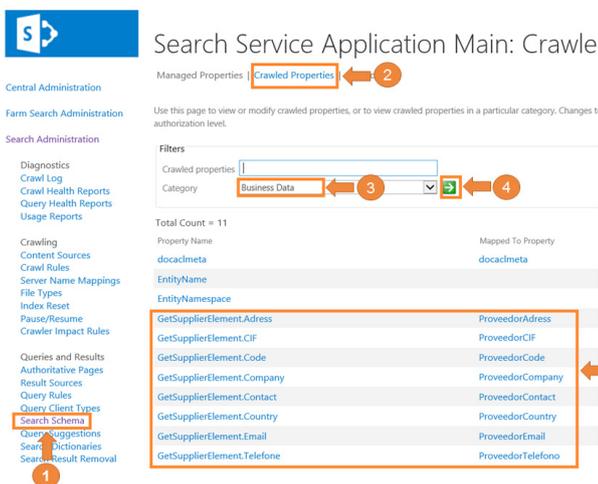


Imagen 7.- Muestra la página de propiedades rastreada de nuestra aplicación servicio de búsqueda.

Añadimos nuestras propiedades administradas. Seleccionamos la opción de “Managed Properties” y pulsamos sobre “New Managed Property”

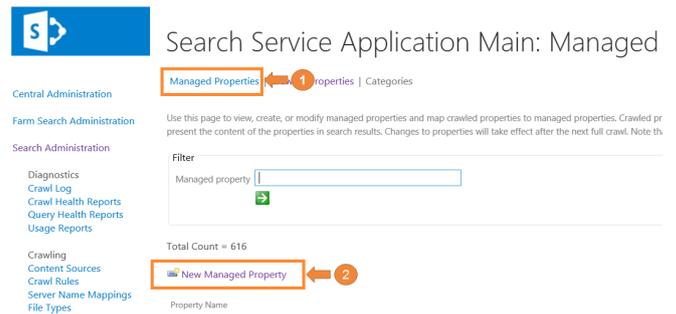


Imagen 8.- Muestra la página de propiedades administrada de nuestras aplicaciones de servicio de búsqueda.

- Introducimos los datos solicitados. En nuestro caso en el nombre de la propiedad (PropertyName) introducimos “ProveedoresContry”, en el tipo (Type) seleccionamos “Text”, dependiendo la propiedad podemos indicar si deseamos que deseamos permitir realizar consultas contra el contenido de la propiedad administrada mediante índice de texto completo (Searchable), también si deseamos que se pueda incluir en la consulta activamos la opción (Query), también si deseamos que sea devuelto en el resultado de búsqueda activamos la opción (Retrievable), también si deseamos que se pueda refinar por dicha propiedad seleccionamos la opción correspondiente en (Refinable), introducimos el mapeo entre la propiedad rastreada y nuestra propiedad administrada, para ver los paso a seguir ver el siguiente punto antes de aceptar.

Search Service Application Main: New Managed Property

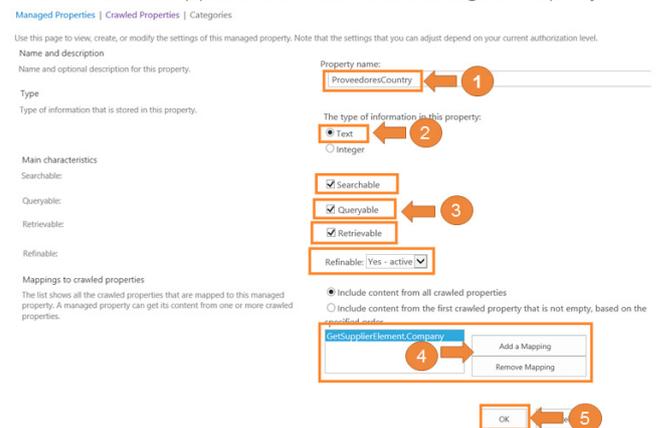


Imagen 9.- Muestra el formulario para crear una propiedad administrada.

Para indexar y recoger las propiedades rastreadas es necesario realizar un rastreo completo, para ello pulsamos sobre el botón derecho del ratón sobre el nuevo origen de datos “Proveedores”

- Añadimos el mapeo con la propiedad rastreada. Pulsamos sobre el botón de un añadir mapeo (Add a Map-

ping), en el formulario realizamos un filtro por la categoría de datos de negocio (Business Data) para ello pulsamos sobre el botón de buscar (Find) después de indicar el valor correspondiente, en la lista de resultados de las propiedades rastreada (Select a crawled property) seleccionamos la propiedad deseada y pulsamos el botón de aceptar (OK).

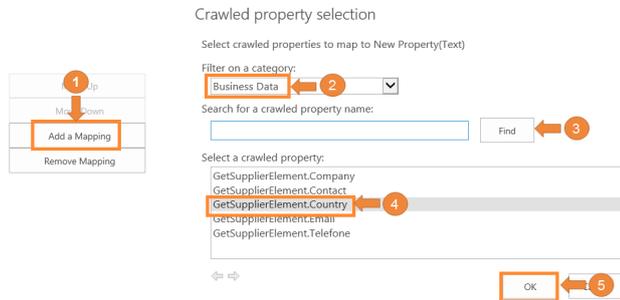


Imagen 10.- Muestra el formulario para realizar el mapeo con la propiedad administrada.

- Añadimos todas las propiedades. Repetimos los puntos anteriores hasta incluir todas las propiedades deseadas en nuestro caso.

Property Name	Type	Multi	Query	Search	Retrieve	Refine	Mapped Crawled Properties
ProveedorAddress	Text	-	Query	-	-	-	GetSupplierElement.Address
ProveedorCIF	Text	-	Query	-	-	-	GetSupplierElement.CIF
ProveedorCode	Integer	-	Query	-	-	-	GetSupplierElement.Code
ProveedorCompany	Text	-	Query	-	-	-	GetSupplierElement.Company
ProveedorContact	Text	-	Query	-	-	-	GetSupplierElement.Contact
ProveedorCountry	Text	-	Query	-	-	Refine	GetSupplierElement.Country
ProveedorEmail	Text	-	Query	-	-	-	GetSupplierElement.Email
ProveedorTelefono	Text	-	Query	-	-	-	GetSupplierElement.Telephone

Imagen 11.- Muestra todas las propiedades administradas de nuestro origen de datos externo.

Paso 7 - Creación de un origen de resultado.

Para realizar un aislamiento del contenido externo de proveedores para posteriormente su utilización de un modo cómodo es necesario crear un origen de resultado, para ello seguimos los siguientes puntos.

- Abrimos el formulario de nuevo origen de resultado. Seleccionamos en el menú vertical derecho la opción de orígenes de resultado (Result Sources) en el apartado de consultas y resultados (Queries and Results) y posteriormente pulsamos sobre la opción de nuevo origen de resultado (New Result Source).

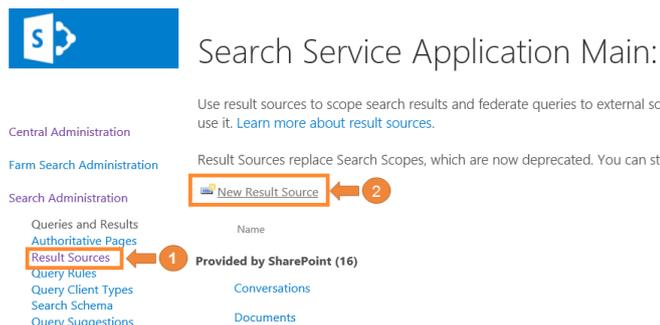


Imagen 12.- Muestra la página de gestión de orígenes de resultado de nuestro servicio de aplicaciones de búsqueda.

- Introducimos los datos solicitados. En nuestro caso introducimos “Proveedores” en cuadro de texto de nombre (Name), “{searchTerms} ContentSource:Proveedores” en la consulta a transformar (Query Transform) y pulsamos en guardar para crear el origen.

Search Service Application Main: Result Source

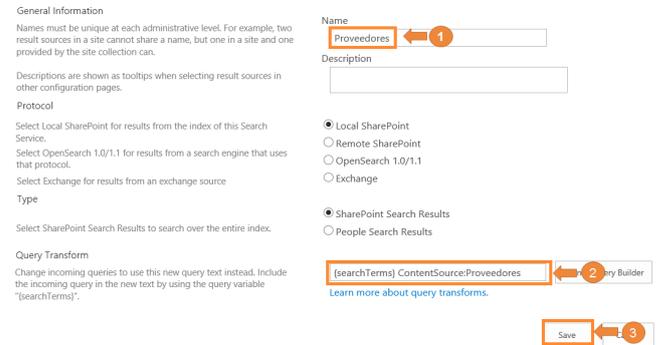


Imagen 13.- Formulario para crear un nuevo origen de resultado.

Paso 8 - Creación de la página de resultado.

Como deseamos realizar búsquedas aislada sobre nuestro contenido externo de proveedores, será necesario la creación de una nueva página de resultado limitando a dicho ámbito, para ello seguimos los siguientes puntos:

- Abrimos la página de contenido de sitio. En caso de no tener abierto nuestro sitio de búsqueda introducimos en el navegador la URL correspondiente, en nuestro caso <http://spmartin15:39910>, en la parte superior derecha pulsamos sobre el icono del menú de configuración y seleccionamos la opción de contenidos del sitio (Site Contents).

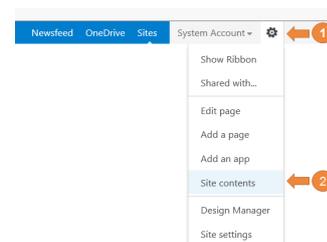


Imagen 14.- Muestra menú de opciones de configuración de sitio en nuestro centro de búsqueda.

- Abrimos la librería de páginas. Pulsamos sobre el icono de Páginas (Pages).

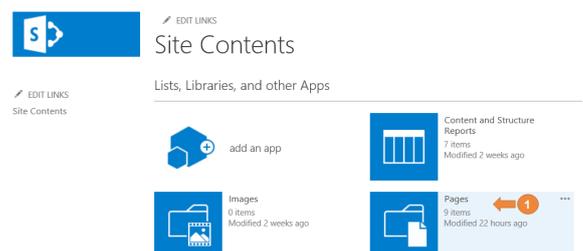


Imagen 15.- Muestra el contenido del sitio de nuestro centro de búsqueda.

- Introducimos los datos solicitados. Seleccionamos ficheros (Files) dentro de la barra superior de la cinta de opciones, pulsamos sobre el icono de nuevo documento (New Document) y seleccionamos página (Page) después introducimos los datos solicitado en nuestro caso “Resultado de proveedores” como título (Title), URL “ProveedoresResult” y pulsamos el botón de crear (Create).

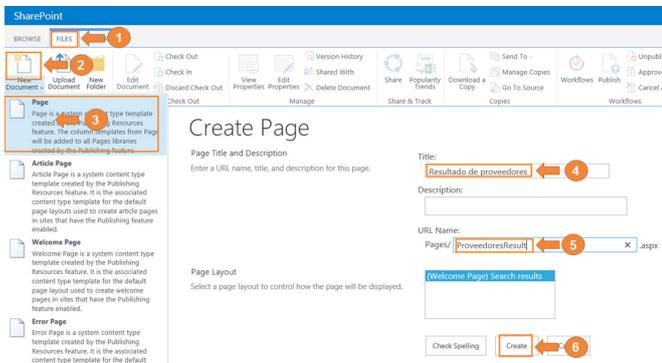


Imagen 16.- Muestra el formulario para crear una nueva página en nuestro sitio de búsqueda.

- Modificamos elemento web de resultado de búsqueda. Seleccionamos la página que acabamos de crear “ProveedoresResult”

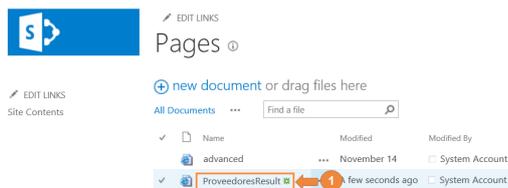


Imagen 17.- Muestra el formulario para crear una nueva página en nuestro sitio de búsqueda.

- Editamos el elemento web de resultado de búsqueda (Search Result) para ello tenemos que editar la página.

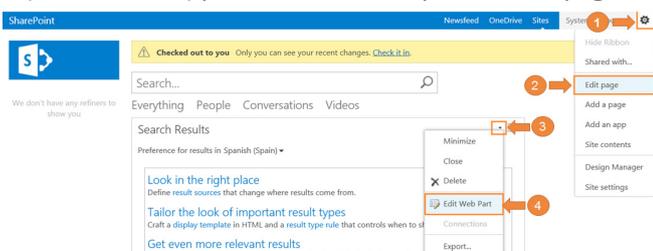


Imagen 18.- Muestra la página en modo edición.

- Cambiamos la consulta para ello seleccionamos el botón de cambiar la consulta (Change query), en el formulario introducimos “Proveedores (Services)” en la selección de la búsqueda (Select a query) y guardamos pulsando sobre el botón de aceptar (OK).

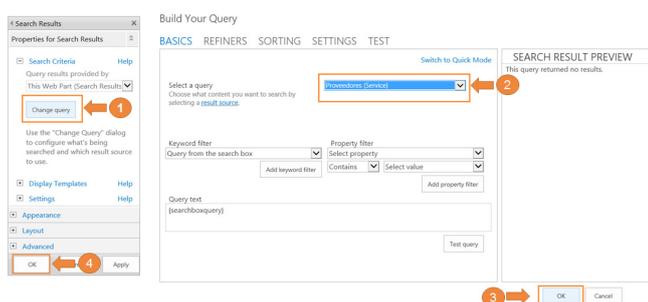


Imagen 19.- Muestra la herramienta de propiedades del elemento web de resultado de búsqueda.

Después es necesario publicar la página para que sea visible por todos los usuarios.

Paso 9 - Creación de vínculo de navegación.

Como deseamos que el usuario realice búsquedas de forma cómoda en nuestro contenido externo de proveedores, será necesario la creación de un nuevo vínculo de navegación apuntado a nuestra página de resultado de proveedores, para ello seguimos los siguientes puntos:

- Abrimos la página de configuración de nuestro sitio de búsqueda. En la parte superior derecha pulsamos sobre el icono del menú de configuración y seleccionamos la opción de configuración de sitio (Site Settings).

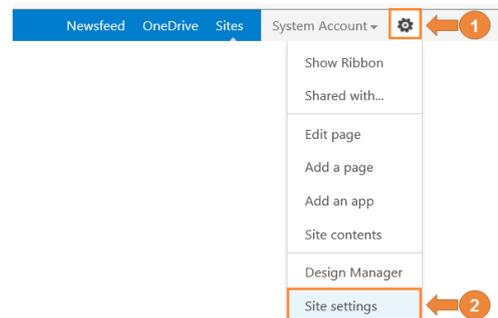


Imagen 20.- Muestra menú de opciones de configuración de nuestro sitio de búsqueda.

- Abrimos la página de configuración de búsqueda. Seleccionamos la opción de configuración de búsqueda (Search Settings) dentro de la categoría de búsqueda (Search).

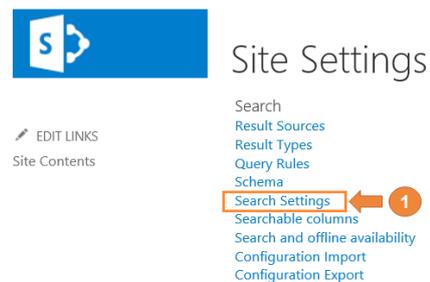


Imagen 21.- Muestra menú de opciones en la categoría de búsqueda en nuestro sitio de búsqueda.

podemos realizar otras configuraciones como puede ser el panel de refinamiento, plantillas de visualización de elementos, panel activable

- Introducimos los datos solicitados. Dentro de la caja de la configuración de navegación (Configuration Search Navigation) pulsamos sobre la opción de añadir enlace (Add Link), introducimos los datos solicitados en nuestro caso “Proveedor” como título (Title), Url “/Pages/ProveedoresResult.aspx” y guardamos los datos.

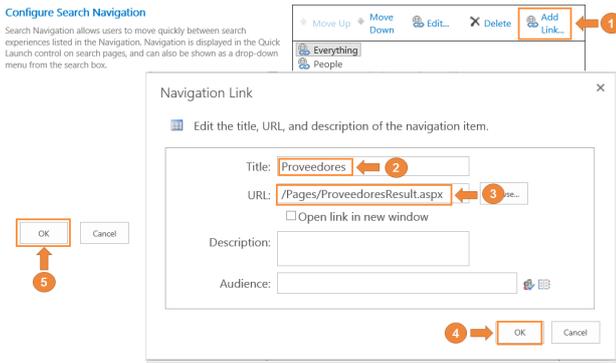


Imagen 22.- Muestra la página de configuración de búsqueda en nuestro sitio de búsqueda.

- Visualizamos sólo en el ámbito de proveedores. Si pulsamos sobre la pestaña de proveedores podemos comprobar que sólo presentará los resultados que se encuentra en el ámbito de proveedores.

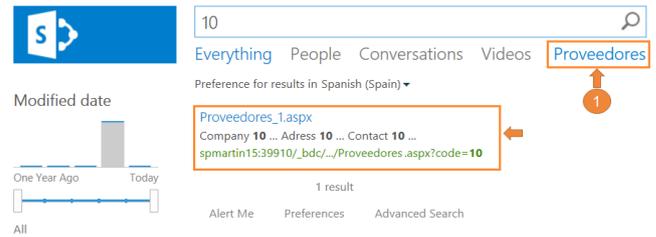


Imagen 55.- Muestra la página de resultado en el ámbito de proveedores en nuestro sitio de búsqueda.

Paso 10 - Validación del funcionamiento.

Aunque podemos realizar otras configuraciones como puede ser el panel de refinamiento, plantillas de visualización de elementos, panel activable, etc. Y para no ser más extenso de lo debido en este artículo, vamos a concluir con una comprobación del funcionamiento, para ello seguimos los siguientes puntos:

- Realizamos una búsqueda. En caso de no estar en la página principal de nuestro sitio de búsqueda pulsamos sobre la imagen de SharePoint, introducimos en la caja de texto el valor que deseamos encontrar por ejemplo "Proveedor 10" y pulsamos sobre el botón de búsqueda.



Imagen 23.- Muestra la página principal de nuestro sitio de búsqueda.

- Visualizamos los resultados. Comprobamos que aparecen los resultados deseado de nuestro contenido externo de proveedores.

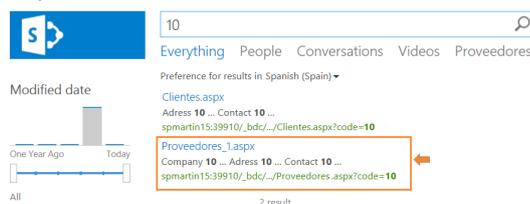


Imagen 24.- Muestra la página de resultado genérico de nuestro sitio de búsqueda.

- Visualizamos los datos del proveedor. Como podemos comprobar presenta toda la información de nuestro proveedor.



Imagen 26.- Muestra la página de los datos del proveedor seleccionado en nuestro sitio de búsqueda.

Conclusiones

Si combinamos el motor de búsqueda con Business Connectivity Services (BCS) nos brinda grandes posibilidades de integración con datos empresarial sin ningún tipo de desarrollo personalizado en la mayoría de los cosas además de una gran facilidad y flexibilidad en su configuración.

MARTIN LUIS LOPEZ REQUENA

SharePoint Solution Architect & Trainer at Everis

martinluislopez@hotmail.com



19

Entrevista Juan Manuel Herrera Ocheita

Mi nombre es Juan Manuel Herrera Ocheita vivo en Ciudad de Guatemala.

Tengo más de 20 años desarrollando software y funde una empresa con un amigo hace más de 5 años de consultoría sobre los productos Microsoft SharePoint, Project Server, Team Foundation Server y recientemente Azure.

Desde la incorporación de los computadores personales o PC en los hogares de las personas y de conectarlas a través de la Internet, ha sido una revolución sin precedentes en la era de la información y la era digital una es la meta y la otra el medio. El término el cielo es el límite con la incorporación de la nube hace que tenga un sentido tangible y emocionante. Estamos a la puerta de ampliar más aún las capacidades humanas para resolver infinidad de problemas, enfermedades y desafíos que enfrentamos día a día y la investigación ya no es un asunto de pocos sino es una realidad de esta abrazando las generaciones jóvenes. Aunque hay retos y desafíos en ello es emocionante lo que hemos logrado y lo podremos alcanzar si tenemos sustento



suficiente en la sociedad que cambia y se adapta a todos los avances tecnológicos que usamos día con día.

SharePoint ha sido uno de los productos que ha impactado en las organizaciones la forma que se colabora, comparte y disemina la información. Hay muchas soluciones y formas de implementar la solución. La plataforma aún requiere de muchos desafíos y no ha logrado aún satisfacer del todo las demandas de los usuarios corporativos, pero su uso e interés no disminuye y cada vez más empresas adoptan su uso. Y Azure permitirá expandir aún más las funcionalidades de SharePoint y con ambientes heterogéneos.

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

Desde niño mi padre compró un Atari con teclado que permitía escribir programas con Basic para dibujar figuras en 2 dimensiones. Luego adolescente Papá compró la primera computadora personal Clone XT sin discos solo disquetes al principio y con 256 Kb de RAM y un procesador 8086 más o menos a mediados de los 80.

Siempre me llamo la atención crear programas y disfrutar de las imágenes gráficas en generaban las computadoras que en ese tiempo era color verde pixelado, pero que luego fue evolucionando a color ámbar, blanco y negro, 16 colores, VGA, Super VGA y hasta la era led que conocemos hoy.

¿Cuáles son tus principales actividades tecnológicas hoy en día?

Asesorar a los clientes para ofrecerles a los clientes soluciones tecnología sobre tecnologías Microsoft, como: Azure, SharePoint, Project, Team Foundation Server, desarrollo de Apps. Para ello es necesario investigar, capacitarse y probar los productos que soy especialista o experto.

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

Cuidar a mi hija de 16 años y de mis seres amados y familia.

¿Cuáles son tus hobbies?

Soy corredor y he corrido muchas carreras pequeñas y medias maratones. Me estoy preparando actualmente para correr una maratón de 42 kms, es un reto al cual me estoy preparando a mis 43 años, y espero alcanzarla en noviembre de este año.

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

Muy emocionante, creo que vamos a lograr cosas que en sueños nos imaginábamos, tendremos al alcance una vasta información y optimizaremos nuestro tiempo y estaremos conectados más que nunca. Además recibiremos una inteligencia que no teníamos por la información que dispondremos sobre todo. Estamos a la puerta de un boom

digital que impactará la forma en que tomamos decisiones y vivimos.

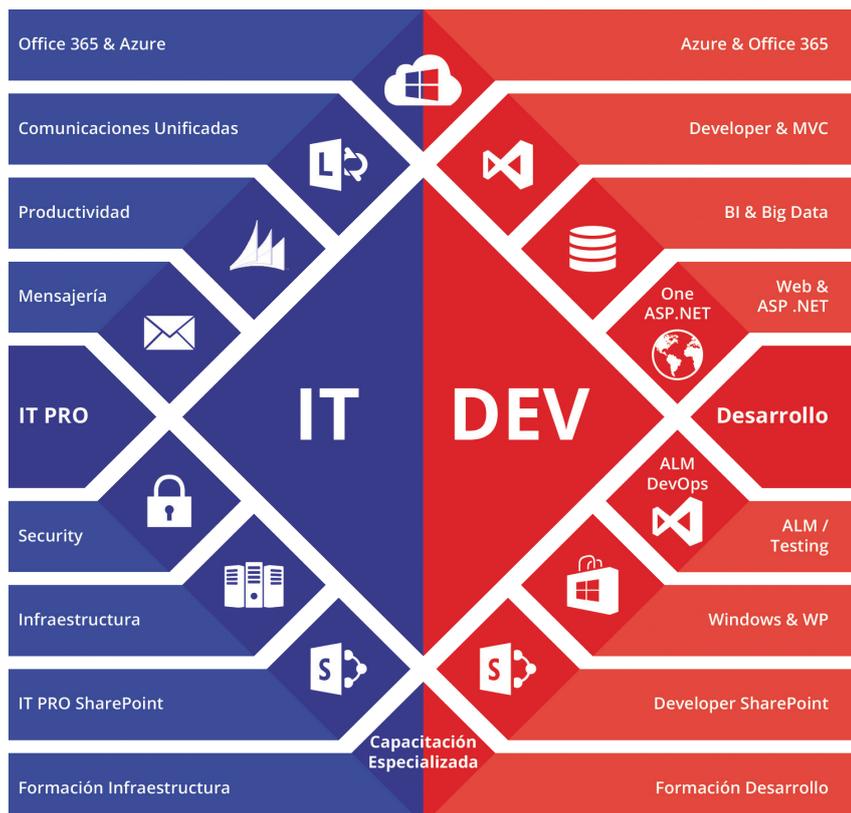
JUAN MANUEL HERRERA OCHEITA

Microsoft SharePoint MVP

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

Correo: mherrera@infowareplus.com **Twitter:** [jmhogua](https://twitter.com/jmhogua)

Expertos en Plataformas y Tecnologías Cloud & OnPremise



CLUSTER
a FiveShare IT Company



www.mvpcluster.es

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

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. Adicionalmente, los Grupos de Office 365 cuentan con opciones adicionales de configuración como se verá en el presente artículo.

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 dotándoles de funcionalidad de valor añadido como muro conversacional

Mejoras en el Sitio Documental de Grupos

En el momento en el que se crea un Grupo de Office 365 a través de OWA, se provisiona de forma silenciosa una Colección de Sitios asociada al mismo y que, por el momento, está oculta para Administradores de SharePoint Online ya que no se muestra en la Administración de SharePoint Online en Office 365. Esta Colección de Sitios cuenta con una única Biblioteca de Documentos que dispone de un subconjunto limitado de funcionalidad si la comparamos con Bibliotecas de Documentos en Sitios convencionales de SharePoint Online. Con respecto a la versión inicial de Grupos liberada por Microsoft, esta Biblioteca ha ido incorporando una serie de capacidades adicionales (que no serán las últimas) como las siguientes:

- Si accedemos al Sitio asociado al Grupo, veremos cómo la Biblioteca por defecto incorpora una opción "Manage" que nos da acceso a ciertas funciones no disponibles hasta ahora.

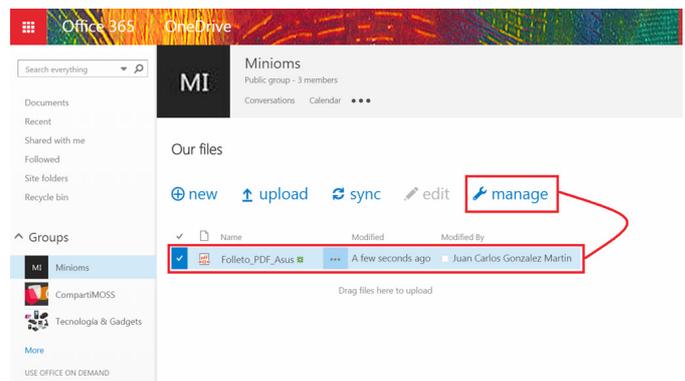


Imagen 1.- Acceso a las opciones de administración de un documento almacenado en el sitio de Grupo.

- La opción "Manage" nos da acceso a las operaciones típicas de "Check Out", "Versionado", "Descargar una copia" o "Borrar". Por ejemplo, la opción de versionado nos da acceso a las versiones del documento y a comprobar como por defecto el versionado está habilitado (versiones mayores) en la Biblioteca.

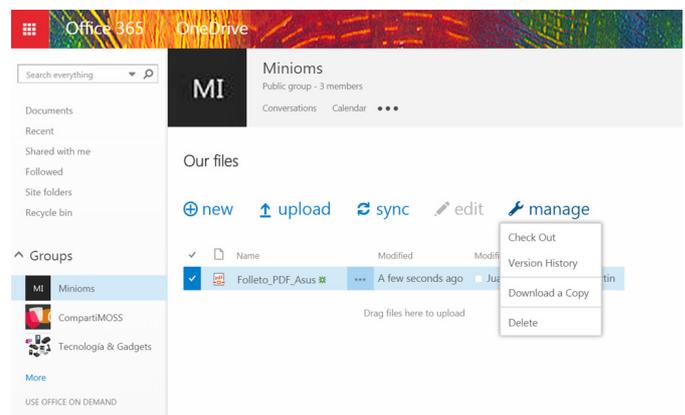


Imagen 2.- Opciones de administración disponibles para un documento almacenado en el Sitio del Grupo.

Junto con estas capacidades adicionales de gestión en la Biblioteca de Documentos del Sitio asociado a un Grupo, otra de las novedades incorporadas es la de disponer de un Bloc de Notas en el mismo. Se trata de un archivo de OneNote asociado al Grupo y pensado para que los integrantes del Sitio puedan colaborar, añadir sus ideas, etc. Para acceder a esta nueva funcionalidad:

- Entre las opciones disponibles en el menú del Grupo, veremos que aparece una nueva opción denominada "Notebook" ("Bloc de notas").

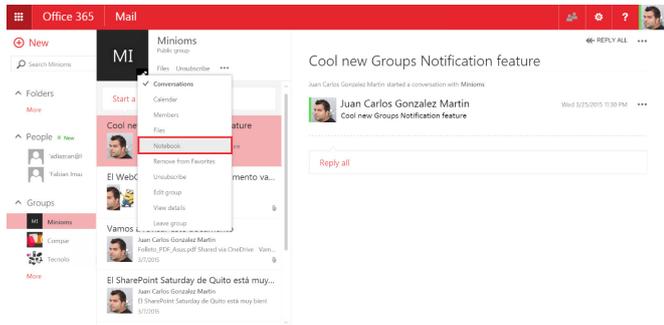


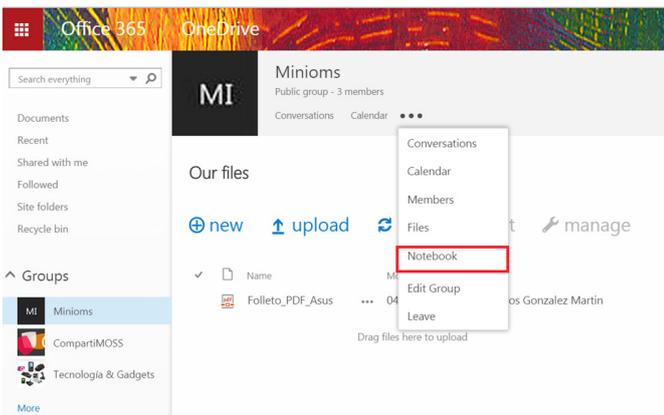
Imagen 3.- Acceso a la funcionalidad de Notebook en un Grupo de Office 365.

- Si hacemos clic en la misma, se abre un archivo de OneNote en el que los integrantes del Grupo pueden colaborar.



Imagen 4.- Bloc de Notas por defecto en un Grupo.

- Adicionalmente, el Bloc de notas del Grupo será accesible desde el Sitio del Grupo y haciendo uso de las opciones de administración disponibles:



En el momento en el que se crea un Grupo de Office 365 a través de OWA, se provisiona de forma silenciosa una Colección de Sitios asociada al mismo

Nuevas características “Sociales” en Grupos

La falta de características sociales como “Me Gusta”, uso de “hashtags” o poder seguir conversaciones es al algo que desde el principio se ha echado en falta en los Grupos de Office 365 para poder modelar escenarios avanzados de colaboración más allá del proporcionado por la generación de discusiones y debates en el seno de un Grupo o desde el correo electrónico. En este sentido, parece que Microsoft irá incorporando poco a poco capacidades más

sociales como demuestra la funcionalidad de “Me Gusta” que está disponible de nuevo para tenants de Office 365 en los que se haya habilitado la configuración “First Release”:

- El detalle de esta funcionalidad lo podéis localizar en el roadmap de Office 365: <http://roadmap.office.com/en-us>. En concreto, la funcionalidad de “Me Gusta” facilita “categorizar” conversaciones que tienen lugar en un Grupo de Office 365 y además se muestran las correspondientes notificaciones en el panel de notificaciones de Office 365.



Imagen 6.- Detalle de la funcionalidad de “Me Gusta” en el roadmap de Office 365.

- Como se comentaba al inicio, de momento esta funcionalidad está disponible únicamente para tenants “First Release” y no en todos los tenants en los que se haya habilitado “First Release”.

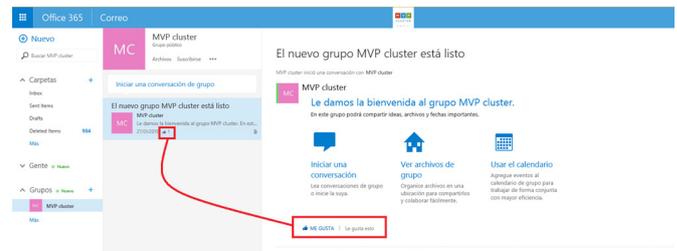


Imagen 7.- Funcionalidad de “Me Gusta” en un tenant de Office 365.

Conclusiones

En este artículo se han detallado las últimas novedades incorporadas por Microsoft a los Grupos de Office 365 que ponen de manifiesto los nuevos conceptos de comunicación y colaboración desde el punto de vista del gigante de Redmond. Frente a la apuesta tradicional en centralizar ambas características en torno a las posibilidades de la plataforma SharePoint, Microsoft ha apostado por definir una capa de colaboración sobre Exchange Online y SharePoint Online en la que se facilita que los usuarios colaboren desde su correo electrónico y dotándoles además de funcionalidades compartidas como el Sitio de Grupo, el Bloc de notas del Grupo o el Calendario de Grupo.

JUAN CARLOS GONZÁLEZ MARTÍN

MVP Office 365

Cloud & Productivity Advisor en MVP CLUSTER

E-Mail de Contacto: juancarlos.gonzalez@fiveshareit.es

Twitter: @jcg1978

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

Web: www.mvpcluster.es

Migración Vs Actualización. ¿Qué hago con mi SharePoint?

Para dar respuesta a la pregunta “¿Migrar o actualizar?”, comencemos con las definiciones:

Hacer un upgrade o actualizar consiste precisamente en eso, en actualizar una versión de SharePoint (por ejemplo SharePoint 2007 ó 2010) a una versión superior como puede ser SharePoint 2013. Una actualización esencialmente es la adición y/o sustitución de los binarios o ficheros del software en cuestión por los nuevos ficheros que el fabricante ha desarrollado como parte de la mejora del propio sistema y como corrección de errores o vulnerabilidades detectadas en versiones anteriores.

Cuando hablamos de Plataformas SharePoint, los cambios de una actualización están íntimamente relacionados a cambios en el Schema de base de datos.

Y, por otro lado, una migración consiste en mover el contenido de nuestra Plataforma SharePoint de una versión antigua a una posterior.

Atendiendo a ambas definiciones, parece claro que siempre una actualización va a ser más rápida que una migración. Sin embargo, existen varias razones por las que una migración será siempre preferible. Aquí van cuatro:

1. Empezar de cero

Cuando algo no funciona, a veces lo mejor es hacer borrón y cuenta nueva. Para muchas organizaciones el despliegue inicial de SharePoint no ha sido el correcto. Bien sea por el desconocimiento general de la tecnología en versiones anteriores, bien sea porque la propia plataforma se está utilizando de un modo totalmente distinto a como fue originalmente pensada, lo cierto es que hay muchos casos en los que SharePoint ha crecido de un modo totalmente incontrolado (lo sé, todos estamos pensando en gobernanza ahora, pero hace no tanto ni conocíamos la palabra). En otras ocasiones, nuestro SharePoint, el SharePoint que queremos actualizar, es una plataforma inconsistente en la que, generalmente por desconocimiento, el administrador IT ha venido configurando y administrando de manera no recomendada. Estos casos son fáciles de identificar, basta con sugerir al administrador SharePoint que haga el upgrade. Si le cambia el color significa que esa granja es inestable.

Por ello, en estas circunstancias la migración es una gran oportunidad para dar estabilidad a esa plataforma y para poner en práctica un adecuado plan de Gobernanza.

Aprender de los errores del pasado nos ayudará en la nueva plataforma a estructurarla para mejorar la adopción y productividad del usuario final. Una actualización, por el contrario, solo cambiará la interfaz de nuestro SharePoint. En el mejor de los casos nuestro SharePoint seguirá teniendo los mismos problemas de estructura, adopción y gobernanza y en aquellos despliegues particularmente inestables o problemáticos seguramente agravaremos el problema.

2. Conservar el contenido útil, deshacernos del resto

Una migración a veces es como cambiarse de casa. Podemos meter todo rápidamente en cajas y llevárnoslo o podemos aprovechar la ocasión para poner orden. Con SharePoint ocurre lo mismo. Después de varios años y de un uso intensivo nos podemos encontrar con una cantidad ingente de contenido en nuestro SharePoint. Este puede ser el momento perfecto para reflexionar si realmente todo el contenido es válido, si hay contenido irrelevante, si existe contenido duplicado o, directamente, si hay contenido que necesitamos archivar. Deshacernos de contenido irrelevante nos ahorrará espacio y mejorará el rendimiento. Y para el resto de información útil un buen proyecto de migración nos ayudará a clasificarla y en base a ello podremos establecer reglas para filtrar contenido irrelevante y reglas para el archivado. Esta operación de clasificación, además, nos permitirá detectar irregularidades en nuestro contenido como incumplimiento de algún tipo de legislación (LOPD por ejemplo) y también nos ayudará a entender mejor el manejo de información en nuestra organización. Esto será clave para la construcción del entorno SharePoint final.

Hacer un upgrade o actualizar consiste precisamente en eso, en actualizar una versión de SharePoint (por ejemplo SharePoint 2007 ó 2010) a una versión superior como puede ser SharePoint 2013

3. Evitar incompatibilidades entre versiones

Otra de las razones fundamentales para decantarse por la migración es salvar el escollo de las incompatibilidades

des ocasionadas por la discontinuidad de una funcionalidad en la nueva versión o por su modificación. El ejemplo claro son las plantillas. ¿Alguno recuerda las “Faboulous 40 templates” que teníamos disponibles en MOSS 2007? Si alguno ha intentado actualizarlas a SharePoint 2010 ó 2013 sabrá de lo que hablo, una auténtica pesadilla hacer el upgrade. Y es que, en el caso de plantillas, la referencia existente a ellas en la base de datos de contenido se rompe, quedando nuestros sitios personalizados imposibles de visualizar correctamente. Una migración en este caso nos permitirá solventar este problema sustituyendo estas plantillas por otras nuevas o recreadas.

en estas circunstancias la migración es una gran oportunidad para dar estabilidad a esa plataforma y para poner en práctica un adecuado plan de Gobernanza

4. Actualizar usuarios y seguridad

Por último, otra mejora que podemos poner en práctica solo con una migración sobre nuestro SharePoint es revisar nuestro mapa de seguridad de usuarios. Nuestra organización está en continuo cambio, y parte de ese cambio son las personas. Personas que estaban en la organización y que colaboraban en SharePoint habrán creado o editado documentos y elementos en la plataforma. Los campos Creado Por, Modificado Por por ejemplo, seguirán referenciando esos antiguos usuarios, aunque dichas cuentas hayan sido deshabilitadas o eliminadas de nuestro Directorio Activo.

Y, por otro lado, migrar nuestro SharePoint puede ser el momento ideal para recertificar permisos. Si hoy concedemos permisos a un usuario lo hacemos en base a su rol en la organización, sus necesidades y la estructura existente. ¿Qué ocurre años después? ¿Estamos seguros de que las condiciones no han cambiado y el usuario sigue teniendo que acceder a los mismos sitios? La migración es el momento ideal para comprobarlo.

Otra de las razones fundamentales para decantarse por la migración es salvar el escollo de las incompatibilidades ocasionadas por la discontinuidad de una funcionalidad en la nueva versión o por su modificación

Conclusión

Por todo ello, una migración puede ser tan compleja o más que una actualización, pero las ventajas, como veis, innumerables. Para terminar, si todavía no estamos convencidos sobre qué nos conviene más, desde aquí os animo a utilizar la [SharePoint Discovery Tool](#) de AvePoint. Dicha herramienta, gratuita, escaneará a fondo nuestro SharePoint, creando un completo informe con todo el inventario de elementos, customizaciones y arquitectura de información. Con todo esto, saca tu informe y reflexiona: ¿seguro que es mejor actualizar?

GONZALO MARCOS ANSOAIN

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



Provider-Hosted Apss + Remote Provisioning (Parte I)

Como parte de la estrategia de Microsoft de mover todos sus productos/clientes hacia la nube, ha hecho una serie de recomendaciones a los desarrolladores entre las que se encuentra el hecho de tratar de abandonar el clásico modelo de desarrollo basado en Features, para comenzar a pensar en el modelo de aprovisionamiento remoto (Remote Provisioning).

Si bien existen varias formas de aplicar este patrón de desarrollo (entre las que se encuentran Scripts en PowerShell o Aplicaciones de consola) en esta serie de artículos vamos a utilizar el modelo de Aplicaciones de SharePoint 2013, específicamente aplicaciones de tipo Provider-Hosted. Vale aclarar que este patrón también se puede aplicar utilizando aplicaciones de tipo SharePoint-Hosted, a través del uso de REST / JSOM en el lado del cliente desde código JavaScript.

Comenzaremos haciendo una breve introducción a los dos tipos de aplicaciones disponibles: SharePoint-hosted y provider-hosted.

Hasta hace poco existía una tercera solución, auto-hosted, pero nunca salió de la versión "Preview" y fue abandonada en junio de 2014.

Entre algunas de las diferencias que tienen estos dos tipos de aplicaciones encontramos que las SharePoint-Hosted solamente pueden contener código JavaScript, ya que se ejecutan en el navegador, mientras que las Provider-Hosted tienen al menos un componente que se hospeda de forma externa a la granja de SharePoint o la suscripción de SharePoint Online, permitiéndonos ejecutar código de servidor.

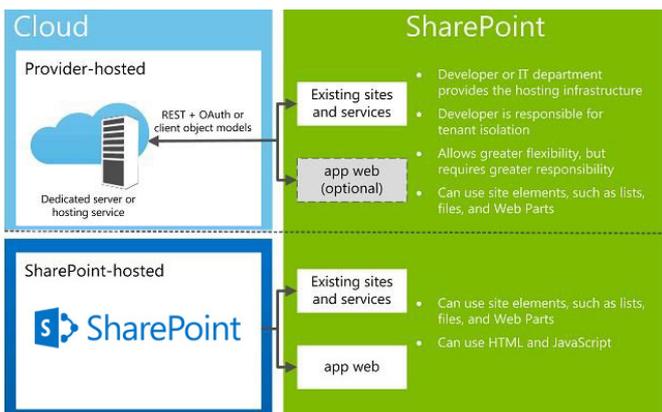


Imagen 1.- Arquitectura de aplicaciones.

Las Aplicaciones **SharePoint-Hosted** se instalan en un web site llamado Host Web. A su vez todos los recursos se hospedan en un sub sitio aislado del Host Web, denominado App Web. Es importante tener clara la diferencia entre el host web y el App Web. Este tipo de hospedaje nos permite poder autenticar a los usuarios con el mismo nivel de autorización que maneja la granja.

Comenzaremos haciendo una breve introducción a los dos tipos de aplicaciones disponibles: SharePoint-hosted y provider-hosted.

Las Aplicaciones **Provider-Hosted** también son instaladas en el Host Web, pero sus componentes remotos son alojados en otro servidor externo a la granja. El proceso de autenticación que se lleva a cabo es OAuth.

Vamos a comenzar desarrollando nuestra aplicación provider-hosted. Lo primero que vamos a hacer es configurar Azure creando un nuevo web site.

- Accedemos al panel de control y hacemos clic en la opción "Nuevo" / "Aplicación Web" / "Creación Rápida".

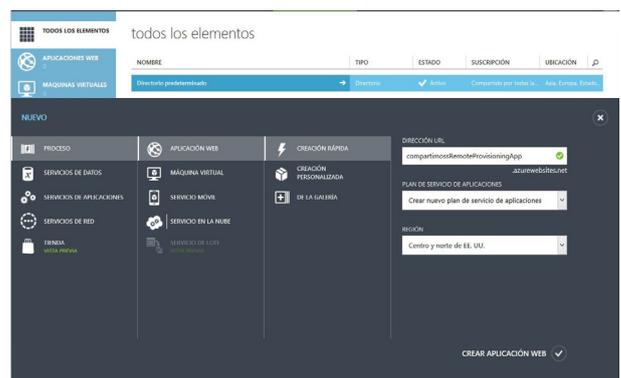


Imagen 2.- Panel de control, Azure.

- Especificamos una URL y hacemos clic en "Crear Aplicación Web". Esperamos unos segundos y el sitio se habrá creado correctamente.
- Una vez que tenemos creado el sitio en Azure el cuál utilizaremos para hospedar nuestra aplicación, debemos descargar el "Perfil de Publicación" ya que lo vamos a necesitar más adelante para poder publicar nuestra aplicación desde Visual Studio.

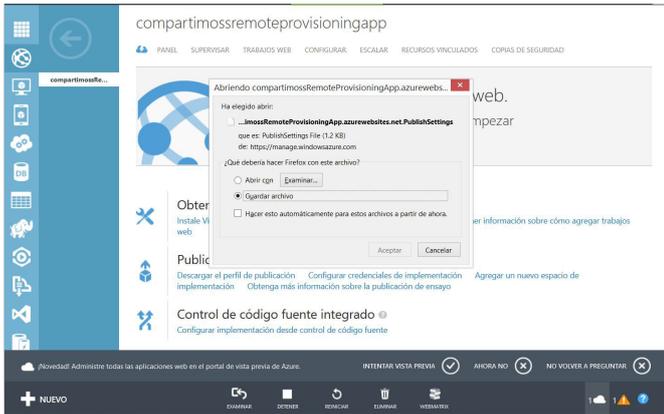


Imagen 3.- Perfil de Publicación, Aplicación Web.

- A su vez, vamos a necesitar crear un sitio de desarrollo (Developer Site) en nuestro SharePoint Online (Office 365), el cuál utilizaremos para poder instalar nuestra aplicación.

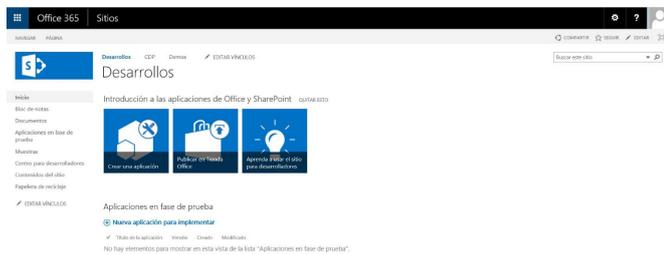


Imagen 4.- Developer Site en Office 365.

- Cuando hayamos creado nuestro sitio en SharePoint, vamos a proceder a registrar nuestra aplicación manualmente y generar el Client Id y el Client Secret (requerido para la autenticación a través de ACS).
- Ambos valores son usados por OAuth para enlazar nuestra aplicación hospedada en Azure con SharePoint Online.
- Navegamos a la siguiente URL: https://<your site>/_layouts/15/appregnew.aspx

Id. de cliente:

Clave secreta de cliente:

Título:

Dominio de aplicación:

Ejemplo: "www.contoso.com"

Dirección URI de redireccionamiento:

Ejemplo: "https://www.contoso.com/default.aspx"

Imagen 5.- Formulario de registro, SharePoint.

- Generamos el "Id del cliente" y la "Clave secreta del cliente" de manera aleatoria haciendo clic en el botón "Generar". El Id del cliente es un Guid que identifica la App de manera única. Mientras que la "Clave Secreta del cliente" es una clave que utiliza SharePoint para validar los permisos de nuestra aplicación.
- También seteamos el título de nuestra aplicación, el domino (nuestro sitio en Azure) y una dirección de redireccionamiento (nuestro sitio en Azure con https). El

resultado será el siguiente:

El identificador de la aplicación se ha creado correctamente.

Id. de cliente: 8344ab4d-94d7-40dc-94f0-b21f2fffd7cc
 Clave secreta de cliente: lz18LcqGKS00bYachLROP21XMiTL0smmFQllgr8NRGc=
 Título: Compartimos Remote Provisioning App
 Dominio de aplicación: compartimosremoteprovisioningapp.azurewebsites.net
 Dirección URI de redireccionamiento: https://compartimosremoteprovisioningapp.azurewebsites.net/

Imagen 6.- Registro de la aplicación.

Ahora ya estamos listos para crear nuestra aplicación desde Visual Studio. En mi caso para este ejemplo voy a utilizar Visual Studio 2012 con "Microsoft Office Developer Tools" instalado.

- Creamos un nuevo proyecto, seleccionamos el template "App for SharePoint 2013" y configuramos las características de nuestra App.
- Debemos de especificar un nombre a nuestra aplicación, la dirección de nuestro sitio de SharePoint y seleccionamos la opción de "Provider-hosted".

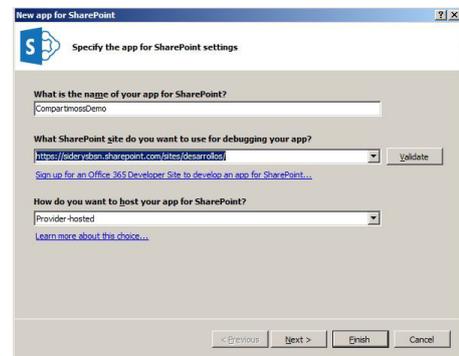


Imagen 7.- Asistente para crear la Aplicación en Visual Studio 2012.

- Luego debemos de seleccionar la forma de autenticación. En nuestro caso vamos a seleccionar "Client Secret" ya que utilizaremos el servicio ACS (Access Control Service) disponible en Azure.

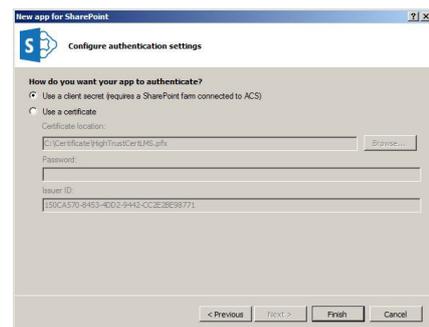


Imagen 8.- Configuración de la autenticación.

Las Aplicaciones Provider-Hosted también son instaladas en el Host Web, pero sus componentes remotos son alojados en otro servidor externo a la granja

Finalmente se crean dos proyectos:

- Un proyecto de SharePoint: desde donde vamos a poder configurar nuestra App.



- Un proyecto Web: donde se aloja el código y las páginas de nuestra App.

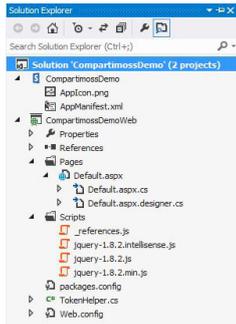


Imagen 9.- Estructura del Proyecto en Visual Studio.

Antes de publicar nuestra app debemos de configurar algunas cosas:

- Hacemos clic en el botón derecho del mouse, sobre el AppManifest.xml y seleccionamos la opción "View Code".
- Reemplazamos el "*" en la siguiente línea con nuestro "Id del cliente" (generado cuando registramos al App en SharePoint).

```
<RemoteWebApplication ClientId="*" />
```

- En "StartPage" cambiamos la URL de la aplicación Web, utilizando nuestro dominio del sitio de Azure + /pages/default.aspx?{StandardTokens}

```
<StartPage>https://compartimosremoteprovisioningapp.azurewebsites.net/Pages/Default.aspx?{StandardTokens}</StartPage>
```

- Abrimos el Web.config del proyecto Web y seteamos el "ClientId" y el "ClientSecret" con los generados anteriormente.

```
<configuration>
  <system.web>
    <compilation debug="true" targetFramework="4.5" />
    <httpRuntime targetFramework="4.5" />
  </system.web>
  <appSettings>
    <add key="ClientId" value="8344ab4d-94d7-4bdc-94fd-82112ff1f1cc" />
    <add key="ClientSecret" value="ln7BlCqj2K300bYachE8CP21X9M6.ClamenFCIgrBNW0c" />
  </appSettings>
</configuration>
```

- Modificamos la página "Default.aspx" tradicional de ASP.NET y agregamos el código que queremos.
- Finalmente, hacemos clic derecho en el proyecto web y elegimos la opción de publicar "Publish". En este punto es donde vamos a utilizar el "Perfil de Publicación" que guardamos cuando creamos nuestra aplicación web desde Azure.

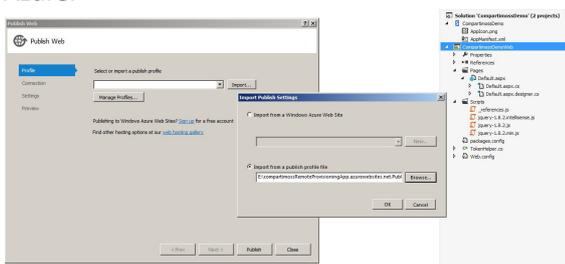


Imagen 11.- Publicación de la App.

- Importamos el perfil de publicación y hacemos clic en "Publish".

- Ahora debemos de volver a la consola de administración de Azure y crear las entradas "ClientId" y "ClientSecret" en la configuración de aplicaciones.



Imagen 12.- Entradas ClientId y ClientSecret.

- Finalmente volvemos a Visual Studio y presionamos F5 para hacer deploy de nuestra aplicación en SharePoint Online.
- Se abrirá un popup del navegador preguntándonos si confiamos en nuestra aplicación: Obviamente le decimos que si confiamos, y seremos redirigidos a nuestra App y deberíamos de ver el título de nuestra página (y en mi caso, otro texto que le agregué).



Imagen 13.- Aplicación publicada.

A su vez, en el contenido del sitio de SharePoint deberíamos de tener acceso a nuestra aplicación.

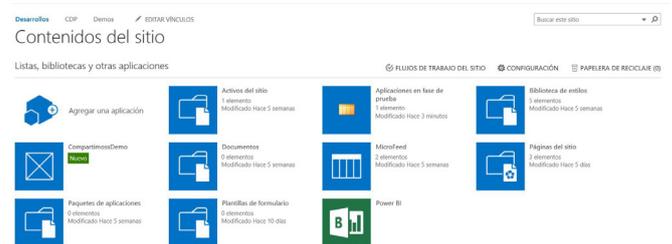


Imagen 14.- Acceso a la App desde la página de Contenidos del Sitio.

Conclusión

Sin perder el foco en la recomendación de Microsoft de tratar de abandonar el clásico modelo de desarrollo basado en Features, para comenzar a pensar en el modelo de aprovisionamiento remoto, ahora que ya tenemos nuestra aplicación Provider-Hosted hospedada en Azure, en el próximo artículo veremos la codificación necesaria para crear un ejemplo de aprovisionamiento remoto.

JOSE LUIS EROSA
SharePoint Solution Architect en Siderys & BSN
jlerosa@hotmail.com

Nuevo modelo de desarrollo para SharePoint: El tiempo de las apps y APIs de cliente.

¿Por qué este nuevo modelo de desarrollo?

En la versión Online, se hace evidente que el modelo tradicional de desarrollo, en el que predomina el código del lado del servidor, no es compatible para extender una plataforma multitenant en la que miles de usuarios comparten un mismo servidor, por lo que cualquier problema provocado por un código mal escrito podría afectar no solo a los usuarios de nuestra organización sino a otros muchos. Desde la propia Microsoft se ha confirmado que SharePoint Online ni soporta ni soportará nunca el despliegue de soluciones de tipo granja. Esto se une al hecho de que la suite de productividad de Office 365, de la que como se ha mencionado forma parte SharePoint Online, es una de las principales apuestas para el futuro de Microsoft, lo que se reflejará en un aumento cada vez mayor de la adopción de esta plataforma por parte de las empresas. Por ello, desarrollar soluciones que puedan ser compatibles con la misma se convierte casi en una obligación.

Siendo éste un motivo de suficiente peso como para adoptar esta nueva propuesta, no es el único. Hay otros motivos que llevarán a considerar este nuevo modelo de desarrollo no sólo como la única solución posible para extender SharePoint Online, sino también como una alternativa deseable para la versión OnPremise.

Por un lado, tenemos la oportunidad de generar soluciones que permitan la actualización y migración de versiones de SharePoint de una forma más sencilla, tanto entre versiones OnPremise, como de versiones OnPremise a Online. Por otro lado, con este nuevo modelo, se pretende minimizar los riesgos asociados a las soluciones de granja, que pueden causar caídas de rendimiento de los servidores provocadas por leaks de memoria o un mal uso de la API de servidor. Finalmente, un modelo de desarrollo basado en el despliegue de aplicaciones desde un catálogo corporativo, resolverá problemas como el downtime, que se produce cada vez que se despliega una solución de tipo granja en SharePoint.

¿En qué consiste este nuevo modelo de desarrollo?

Este nuevo modelo de desarrollo propone olvidar el modelo de soluciones de granja basadas en la API de servidor

y adoptar un modelo que utilice código alejado del contexto de SharePoint. Para ello, las nuevas versiones de SharePoint han evolucionado de una forma muy importante sus APIs de cliente proporcionando una API REST, una API cliente en C# (CSOM) y otra API de cliente para JavaScript (JSOM).

En la versión Online, se hace evidente que el modelo tradicional de desarrollo, en el que predomina el código del lado del servidor, no es compatible para extender una plataforma multitenant

Asimismo refuerza también el concepto de aprovisionamiento remoto, que nos propone usar las aplicaciones de SharePoint para, junto con las APIs mencionadas anteriormente, desplegar todos los elementos tales como listas, librerías de documentos, tipos de contenido, etc. o usar las mismas para la creación de sitios o personalizar su diseño, entre otras opciones. No obstante, no sólo se limita al uso de las aplicaciones de SharePoint, sino que también podremos usar el modelo de objetos de cliente tanto en scripts PowerShell, aplicaciones de consola, o características propias de Azure para extender remotamente SharePoint, tanto OnPremise como Online.

Además este modelo de desarrollo propone un cambio de filosofía a la hora de afrontar los proyectos. A partir de ahora, el desarrollo no se debe centrar tanto en la tecnología como en la funcionalidad y los requerimientos a los que hay que dar respuesta. Hasta ahora, con la API de servidor, se tenía acceso a todas las capacidades para extender SharePoint con los riesgos que esto podía suponer. Obviamente desde la API de cliente no se tienen disponibles todas las opciones, pero ¿quiere decir esto que solo podremos extender SharePoint con las opciones que permitan las APIs de cliente?. Vamos a ver un ejemplo para responder a esta pregunta.

En el siguiente supuesto, se quiere implementar un Timer Job en SharePoint Online que procese una lista de elementos. Obviamente, desde la API de Cliente y para esta versión de SharePoint, no tenemos la oportunidad de crear Timer Jobs como sí podíamos hacer con la API de servidor y las soluciones de granja. Aquí es donde entra la idea de

dejar de pensar en la tecnología, para pensar en la funcionalidad a la que queremos dar respuesta. ¿Qué es un Timer Job?. En resumen, es solo una tarea programada que hace un procesamiento en background sobre SharePoint. ¿Cómo podríamos cubrir esta funcionalidad?, bastaría con crear un Worker Role de Azure que usara la API de Cliente para hacer el procesamiento de la lista que deseamos con una programación establecida.

Por tanto, con la propuesta de este nuevo modelo, no solo es suficiente con conocer la API de Cliente. Es necesario conocer el abanico de posibilidades que nos ofrecen las API de SharePoint, las de Office 365, conocer Azure y sus oportunidades, etc. Esto permitirá saber en cada momento, cuales son las mejores opciones a emplear para resolver el requerimiento que se desea.

¿Debemos dejar de utilizar el modelo tradicional?

Cuando se piensa en desarrollar para SharePoint Online la respuesta a esta pregunta es obvia, pero no tanto cuando se va a abordar un proyecto de desarrollo sobre una plataforma OnPremise. La realidad es que no es obligatorio adoptar el nuevo modelo de desarrollo para la infraestructura OnPremise. Desde Microsoft se garantiza el soporte, al menos de momento, para las soluciones de granja. Sin embargo, por todas las ventajas de su uso, es recomendable adaptar progresivamente los desarrollos al nuevo paradigma.

No obstante, en el caso de que se opte, a la hora de desarrollar para SharePoint, por continuar trabajando con soluciones de granja, puede resultar interesante adaptar las mismas para aproximarlas, en la medida de lo posible, a las recomendaciones y buenas prácticas que nos sugieren con el nuevo modelo de desarrollo. Esto permitirá reducir algunos de los inconvenientes que provocan este tipo de soluciones.

A continuación vamos a ver qué cambios se podrían aplicar sobre una solución mediante un ejemplo en el que la intención es crear varias columnas de sitio y un tipo de contenido que usa las mismas.

Este nuevo modelo de desarrollo propone olvidar el modelo de soluciones de granja basadas en la API de servidor y adoptar un modelo que utilice código alejado del contexto de SharePoint

Desplegando Columnas de sitio y Tipos de Contenido vía XML

Una primera aproximación para resolver este requeri-

miento, y que se ha hecho muy habitual en las soluciones de granja, gracias a las facilidades que proporciona Visual Studio, podría ser emplear ficheros XML donde se definieran estos elementos, como se ve a continuación, y después desplegar los mismos a través de características. En el siguiente ejemplo de código se puede ver la creación de columnas de sitio y tipos de contenido.

```
<?xml version="1.0" encoding="utf-8"?>
<Elements xmlns="http://schemas.microsoft.com/sharepoint/">

  <Field ID='{014AB8C6-02AA-41EC-B5A1-CD46668E6087}'
    Group="Ejemplo" Type="Number" Name="EjemploNumero1"
    DisplayName="EjemploNumero1" StaticName="EjemploNumero1"
    SourceID="http://schemas.microsoft.com/sharepoint/v3"/>

  <Field ID='{9F764C8E-D966-416D-8148-3CC0E4E88BDA}'
    Group="Ejemplo" Type="Text" Name="Ejemplo1"
    DisplayName="Ejemplo1" SourceID="http://schemas.microsoft.com/sharepoint/v3" StaticName="Ejemplo1" />

</Elements>
```

```
<?xml version="1.0" encoding="utf-8"?>
<Elements xmlns="http://schemas.microsoft.com/sharepoint/">
<!-- Tipo de contenido primario: Elemento (0x01) -->
<ContentType
ID="0x01002bledca8d3f94d9db9ca8da78b2460"
Name="EjemploTipoContenido"
Group="Ejemplo"
Description="Descripción de ejemplo de tipo de contenido"
Inherits="TRUE"
Version="0">
  <FieldRefs>
    <FieldRef ID='{014AB8C6-02AA-41EC-B5A1-CD46668E6087}'
      Name="EjemploNumero1" Required="TRUE"/>
    <FieldRef ID='{9F764C8E-D966-416D-8148-3CC0E4E88BDA}'
      Name="Ejemplo1" Required="TRUE" />
  </FieldRefs>
</ContentType>
</Elements>
```

Como se puede ver en la Figura 1, esta aproximación tiene sus inconvenientes. Cuando usamos esta definición vía ficheros XML se crean dependencias con la base de datos de contenido que pueden dificultar la migración de las soluciones a nuevas versiones o nuevas granjas de servidores.

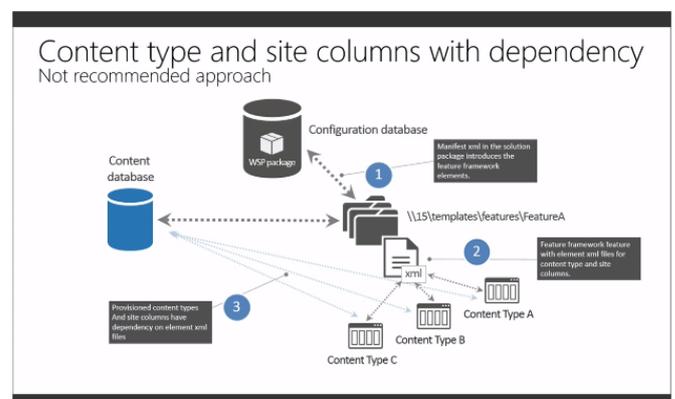


Imagen 1.- Dependencia que se crea al desplegar elementos usando XML. Fuente: Microsoft Virtual Academy.

Adaptando el modelo tradicional al nuevo modelo de desarrollo

Los problemas de dependencias que se crean usando la aproximación anterior se pueden solucionar cambiando

la forma en la que se despliegan los elementos. Cuando por cualquier motivo tenemos que seguir desarrollando soluciones de granja, en lugar de usar ficheros XML, aprovisionaremos los elementos por medio de código usando los EventReceiver de las características. En este caso usaremos el evento de activación de la característica de la siguiente forma.

```
public class Feature1EventReceiver : SPFeatureReceiver
{
    public override void
    FeatureActivated(SPFeatureReceiverProperties properties)
    {
        SPSite site = properties.Feature.Parent as SPSite;
        using (SPWeb web = site.OpenWeb())
        {
            Helper.CreateTextColumn(web, "Ejemplo1");
            Helper.CreateNumberColumn(web, "EjemploNumero1");

            Helper.CreateContentType(web,
            "EjemploTipoContenido", "Descripcion de ejemplo de tipo de
            contenido", "Ejemplo");
            Helper.AddColumnToContentType(web,
            "EjemploTipoContenido", "Ejemplo1", true);
            Helper.AddColumnToContentType(web,
            "EjemploTipoContenido", "EjemploNumero1", false);
        }
    }
}
```

Con esto, como se aprecia en la siguiente imagen, además de crear las columnas de sitio y tipos de contenido sin dependencias, se avanza en el cambio de filosofía que nos lleva al ya comentado aprovisionamiento de los distintos elementos por medio de código. Esta técnica es recomendable que sea usada para desplegar todos los elementos propios de SharePoint como podrían ser Listas, Bibliotecas de documentos, etc.

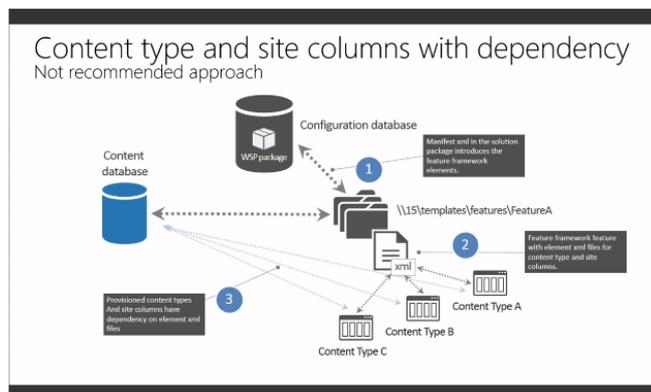


Imagen 2.- Aprovisionamiento de elementos sin dependencias usando la API de servidor. Fuente: Microsoft Virtual Academy.

Evolucionando hacia el nuevo modelo de desarrollo de SharePoint

La solución al requerimiento anterior, siguiendo el nuevo modelo, consistiría en el desarrollo de una Aplicación Provider-Hosted de SharePoint, que podría ser una aplicación ASP.NET, ya sea WebForm o MVC. Esta aplicación tendrá la opción de aprovisionar remotamente los elementos deseados por medio de un botón que va a usar la API Cliente. Para el ejemplo que se verá a continuación, y por simplici-

dad, se creará una aplicación Provider-Hosted de tipo WebForms en la que toda la funcionalidad se encontrará en el evento asociado al botón que se ha añadido a la misma. El código del evento podría ser el que se muestra a continuación.

```
var spContext = SharePointContextProvider.Current.
GetSharePointContext(Context);

using (var clientContext = spContext.
CreateUserClientContextForSPHost())
{
    Web web = clientContext.Site.RootWeb;

    web.Fields.AddFieldAsXml("<<Field
ID='{014AB8C6-02AA-41EC-B5A1-CD46668E6087}'
Group='Ejemplo' Type='Number' Name='EjemploNumero1'
DisplayName='EjemploNumero1' StaticName='EjemploNumero1'
SourceID='http://schemas.microsoft.com/sharepoint/v3/'> ",
false, AddFieldOptions.AddFieldInternalNameHint);
    web.Fields.AddFieldAsXml("<<Field ID='{9F764C8E-
D966-416D-8148-3CC0E4E88BDA}' Group='Ejemplo' Type='Text'
Name='Ejemplo1' DisplayName='Ejemplo1' SourceID='http://
schemas.microsoft.com/sharepoint/v3/' StaticName='Ejemplo1'
/>", false, AddFieldOptions.AddFieldInternalNameHint);
    clientContext.ExecuteQuery();
    var parentContentTypes = clientContext.
LoadQuery(web.ContentTypes.Where(ct => ct.Name ==
"Elemento"));
    clientContext.ExecuteQuery();

    ContentType parentContentType = parentContentTypes.
FirstOrDefault();

    if (parentContentType != null)
    {
        web.ContentTypes.Add(new
        ContentTypeCreationInformation
        {
            Name = "EjemploTipoContenido",
            Group = "Ejemplo",
            Description = "Descripción de ejemplo de tipo de
            contenido",
            ParentContentType = parentContentType
        });
        clientContext.ExecuteQuery();
    }
    else
    {
        throw new InvalidOperationException("Item Content
        Type not found");
    }

    Field ejemplo1 = web.Fields.
GetByInternalNameOrTitle("Ejemplo1");
    Field ejemploNumero1 = web.Fields.
GetByInternalNameOrTitle("EjemploNumero1");

    var createdContentTypes = clientContext.
LoadQuery(web.ContentTypes.Where(ct => ct.Name ==
"EjemploTipoContenido"));
    clientContext.ExecuteQuery();

    ContentType createdContentType =
createdContentTypes.FirstOrDefault();
    createdContentType.FieldLinks.Add(new
    FieldLinkCreationInformation
    {
        Field = ejemplo1
    });
    createdContentType.FieldLinks.Add(new
    FieldLinkCreationInformation
    {
        Field = ejemploNumero1
    });
    createdContentType.Update(true);
    clientContext.ExecuteQuery();
}
```

Del mismo modo que en los casos anteriores, en la Imagen 3 se muestra, el esquema de despliegue de elementos

del modelo basado en aprovisionamiento remoto y APIs de cliente.

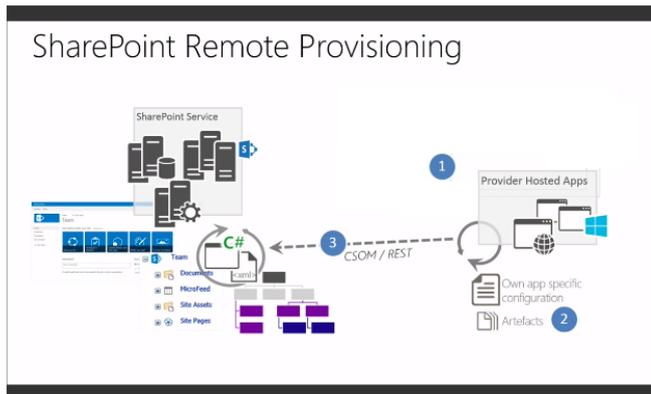


Imagen 3.- Aprovisionamiento remoto usando el nuevo modelo de desarrollo. Fuente: Microsoft Virtual Academy.

Se trata de un ejemplo sencillo para el aprovisionamiento remoto de elementos tales como columnas de sitio y tipos de contenido. Para facilitar a los desarrolladores la adopción de este nuevo modelo se dispone del mejor repositorio de código mantenido por la propia Microsoft, y cuya visita puede resultar muy interesante para ver qué se puede hacer con las APIs de cliente y cómo resolver casi cualquier situación. Se trata del repositorio de GitHub de Pattern & Practices de Office 365 cuyo enlace se recoge en la bibliografía del artículo.

Los problemas de dependencias que se crean usando la aproximación anterior se pueden solucionar cambiando la forma en la que se despliegan los elementos

Conclusiones

La propuesta de Microsoft de avanzar hacia este nuevo pa-

radigma para el desarrollo de soluciones nos provoca, a muchos desarrolladores de SharePoint, un necesario cambio de filosofía y concepto a la hora de afrontar nuestros proyectos. Este cambio supone un reto, que nos obligará a conocer mejor las tecnologías de Front-End, MVC .NET, Azure, etc. Así como a volver a trabajar sobre aspectos como el ALM de nuestras soluciones. Un reto que por otro lado, nos ayudará a abrir nuestro abanico de posibles soluciones a los requerimientos de nuestros clientes. Obviamente, si bien el modelo de apps, está aquí para quedarse, aún tiene que evolucionar y mejorar en muchos aspectos. No obstante, creo que resultará muy interesante y todas las oportunidades que nos ofrece nos ayudarán a estar al día, en toda la vorágine de cambios en la que Microsoft y sus productos estrella, se encuentran inmersos. En próximos artículos seguiremos avanzando en aspectos como el aprovisionamiento remoto de sitios o aspectos tan cuestionados como el branding.

Bibliografía

- <https://jcgonzalezmartin.wordpress.com/2015/02/12/sharepoint-2013-sharepoint-online-cambios-en-el-empaquetado-y-despliegue-de-soluciones-i/>
- <http://www.microsoftvirtualacademy.com/training-courses/transform-sharepoint-customizations-to-sharepoint-app-model>
- <https://github.com/OfficeDev/PnP/wiki>
- <http://blog.mastykarz.nl/programmatically-creating-site-columns-content-types-app-model/>

JOSÉ CARLOS RODRÍGUEZ AVILÉS

Analista Programador en Soluciones de SharePoint en UCI

josecarlos.rodriquez@uci.com

@jcroav

<http://elblogdelprogramador.wordpress.com>

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.



¿Qué sabemos de SharePoint 2016?

A principios de mayo se celebró en Chicago la conferencia Microsoft Ignite, donde, entre otras novedades, descubrimos por primera vez de manera oficial a la próxima versión de nuestro servidor favorito. De entre todas las sesiones que se realizaron, veamos qué novedades contaron sobre SharePoint 2016.

Las mejoras en gestión y administración que se han implementado en SharePoint Online, se han trasladado a SharePoint Server 2016, incluyendo un gran esfuerzo mejorando los entornos híbridos para garantizar que sea mucho más sencillo que las organizaciones den el salto a la nube, cuando estén preparadas.

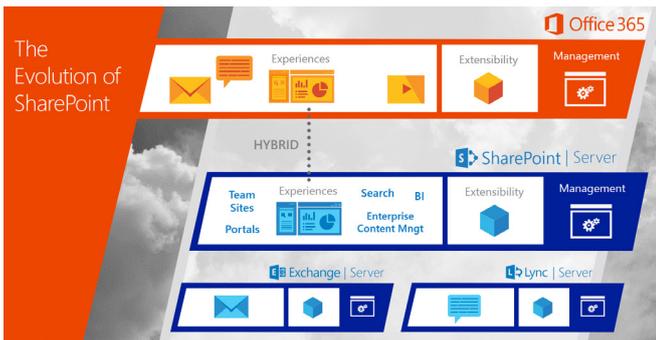


Imagen 1.- SharePoint 2016 podrá implementar servicios de Office 365.

La nueva generación de portales nos traerá una nueva experiencia de usuario a la hora de generar contenido web o escribir un artículo en nuestro blog, más adaptada a las interfaces actuales y simplificando el proceso de publicación, además de un moderno y mejorado look & feel. Por supuesto, con una experiencia responsive para cualquier dispositivo.

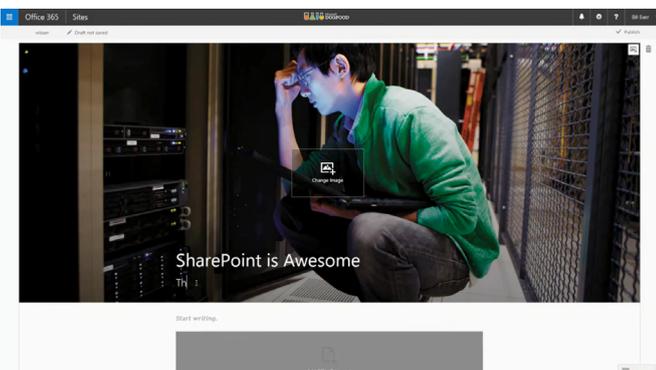


Imagen 2.- Imagen de un blog personal.

Esta evolución de los Portales, llamada Next-Gen Portales, traerá un nuevo Framework para generar portales más

Responsive y adaptados a los usuarios actuales y a sus dispositivos, como podemos ver en Office 365 con Video o el perfil de los usuarios en Delve.



Imagen 3.- Qué mejoras se han incluido a nivel de Portales.

Como plataforma de colaboración documental, SharePoint es perfecto para la gestión del contenido empresarial, sin embargo, los usuarios reclaman una forma más sencilla de trabajar con sus ficheros y, por eso, se han centrado en mejorar y simplificar OneDrive para conseguir que sea el centro natural de los ficheros para los usuarios.

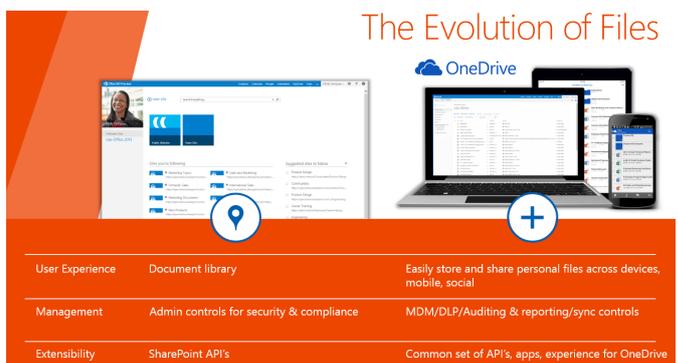


Imagen 4.- Una evolución hacia los ficheros.

SharePoint salió al mercado en 2001 siendo una plataforma de colaboración, que fue evolucionando a plataforma de gestión de contenido en 2007 hasta que llegó a la nube, junto al Social Enterprise, en 2013. Sin embargo, SharePoint 2016 será la primera versión que ha sido definida desde la nube, la evolución no camina hacia la nube, sino que se ha evolucionado la versión OnPremises usando las experiencias en administración y desarrollo de SharePoint Online en Office 365. Una versión inspirada en la nube, que mejora las experiencias de los usuarios y que se centra en la seguridad de las personas y la información que gestionan.

Investment Areas



Imagen 5. Dónde se han hecho los esfuerzos en esta nueva versión.

Para mitigar los silos de información que se producen entre las infraestructura OnPremises y en la nube, SharePoint 2013 nos permite configurar servicios en modo híbrido, como, por ejemplo, el servicio de búsqueda que nos permitía federar el contenido de SharePoint Server 2013 en SharePoint Online y viceversa. SharePoint 2016 da un paso más allá en cuanto a las integraciones híbridas, y, entre otras mejoras, el servicio de búsqueda será capaz de indexar contenido en la nube y OnPremises, mejorando la experiencia de usuario cuando se realiza una búsqueda, o permitiendo utilizar las capacidades de descubrimiento de Delve con información en la nube y OnPremises.

Desde el punto de vista de infraestructura, se define un nuevo concepto en SharePoint 2016 llamado "MinRole". MinRole simplifica el despliegue y escalado de SharePoint asegurando que las peticiones de los usuarios se dirigen a los servidores que tienen los servicios o roles adecuados.

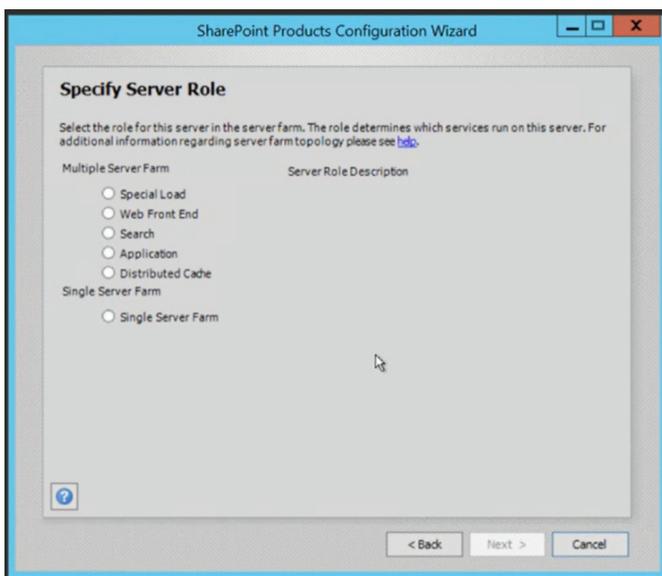


Imagen 6.- El nuevo Configuration Wizard con la instalación por roles.

Como podemos ver en la imagen anterior, a la hora de unir un servidor a la granja podemos especificar qué rol va a

tener este servidor, algo que nos ayudará a mejorar el plan de capacidad, escalado y rendimiento de nuestros servidores. Podéis leer más sobre la Instalación y Despliegue de SharePoint 2016 en <http://blogs.technet.com/b/wbaer/archive/2015/05/12/what-s-new-in-sharepoint-server-2016-installation-and-deployment.aspx>

Se ha hecho un gran esfuerzo, provocado por la necesidad de mantener numerosas granjas de SharePoint Online en Office 365, en los procesos de actualización, que tan delicados son en la plataforma actual. SharePoint 2016 implementa un nuevo mecanismo de orquestado de actualizaciones que permitirá realizar los procesos sin parada, el tan deseado "0 downtime at Business Hours".

Otro gran reto en SharePoint es el tema de los límites y parece que se ha hecho otro gran avance aumentando los mimos de manera exponencial. Por ejemplo, se podrán subir ficheros de hasta 10Gb, se ha incrementado el umbral de vista por encima de 5.000 o se podrán indexar unos 500 millones de elementos.

Por último, parece que tendremos beta pública de SharePoint 2016 en el Q4 de este año 2015, la primera Release Candidate en el Q1 de 2016 y, al final, la RTM en el Q2 de 2016.



Imagen 7.- Planificación de las entregas.

Si os resumo el evento en una frase, sería la siguiente: "Casi todas las mejoras que se han implementado en SharePoint Online durante los últimos 3 años, se han trasladado a SharePoint 2016, incluyendo las experiencias de Microsoft administrando Office 365"

ALBERTO DIAZ MARTIN

MVP SharePoint

adiazcan@hotmail.com

@adiazcan

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

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

Movilidad sobre SharePoint - Parte II

En el artículo anterior de Movilidad sobre SharePoint – Parte I, hicimos una breve introducción a las distintas opciones que tenemos a la hora de desarrollar soluciones móviles que se integren con nuestro SharePoint.

Continuando con la serie, esta vez realizaremos un ejemplo práctico en el que desarrollaremos una aplicación híbrida multiplataforma.

Usaremos:

- Apache Cordova para empaquetar la solución.
- Windows Azure Active Directory para gestionar la identidad y controlar el acceso a nuestro Office 365.
 - KnockoutJS para facilitarnos separar la capa de presentación y la lógica de nuestra aplicación, siguiendo el patrón model-view-viewmodel (MVVM).

Introducción a Apache Cordova

El núcleo de una aplicación Apache Cordova está compuesto por HTML5 y CSS3 para gestionar el renderizado y JavaScript (o TypeScript) para la lógica de la aplicación. El uso de HTML5 nos posibilita el acceso tanto a funcionalidades nativas como al hardware del dispositivo.

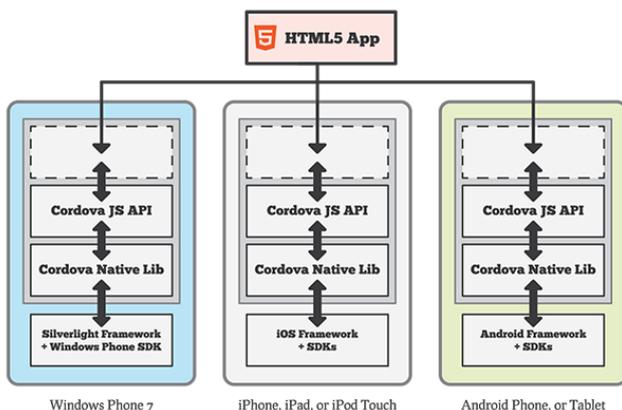


Imagen 1.- Capas de interacción con funcionalidades nativas.

El framework nos proporciona la manera de encapsular y paquetizar nuestro código HTML y esto lo hace a través de un Webview, que es el que se encarga de renderizar dicho contenido. Por otro lado nos proporciona una API javascript para poder acceder a los componentes y funciones del dispositivo

A continuación una tabla de funcionalidades nativas soportadas por Apache Cordova. Como podemos comprobar para los dispositivos móviles más comunes el framework soporta prácticamente la totalidad de características nativas.

	iPhone / iPhone 3G	iPhone 3GS and newer	Android	BlackBerry OS 5.x	BlackBerry OS 6.0+	WebOS	Windows Phone 7+	Symbian	S60
Accelerometer	✓	✓	✓	✓	✓	✓	✓	✓	✓
Camera	✓	✓	✓	✓	✓	✓	✓	✓	✓
Compass	X	✓	✓	X	X	✓	✓	X	✓
Contacts	✓	✓	✓	✓	✓	X	✓	✓	✓
File	✓	✓	✓	✓	✓	X	✓	X	X
Geolocation	✓	✓	✓	✓	✓	✓	✓	✓	✓
Media	✓	✓	✓	X	X	X	✓	X	X
Network	✓	✓	✓	✓	✓	✓	✓	✓	✓
Notification (Alert)	✓	✓	✓	✓	✓	✓	✓	✓	✓
Notification (Sound)	✓	✓	✓	✓	✓	✓	✓	✓	✓
Notification (Vibration)	✓	✓	✓	✓	✓	✓	✓	✓	✓
Storage	✓	✓	✓	✓	✓	✓	✓	X	X

Imagen 2.- Tabla de funcionalidades nativas soportadas por dispositivo

Requisitos

En este apartado, enumeraremos todo lo que necesitamos para poder desarrollar una aplicación como la que vamos a presentar más adelante.

Infraestructura

- Office 365 E3 o superior: Se puede crear una Trial de Office 365 desde <http://bit.ly/1FD6Lvb>.
- Azure Active Directory: Vinculado con Office 365.

Herramientas

- Visual Studio 2013 Update 4
- Visual Studio Tools for Apache Cordova CTP3.0

APIs para el Desarrollo

- Referencia de la API de Cordova:
 - http://cordova.apache.org/docs/en/4.0.0/guide_support_index.md.html#Platform%20Support
- Referencia de la API REST de SharePoint:
 - <https://msdn.microsoft.com/en-us/library/office/jj860569.aspx>
- Referencia de Knockout:
 - <http://learn.knockoutjs.com/#/?tutorial=intro>

Publicación de la Aplicación en el Marketplace de aplicaciones

Es necesario tener cuenta y licencia para publicar en el marketplace de aplicaciones de las distintas plataformas:

- Android: Pagando 25€ y ya tienes la licencia de desarrollador para siempre.
- Windows Phone: Pagando 19 € y ya tienes la licencia de desarrollador para siempre.
- Apple:
 - Se pagan 100€ al año para ser desarrollador.
 - Necesitas obligatoriamente un Mac para subir tu aplicación a revisión.

Conexión a Office 365 en Aplicaciones Apache Cordova

A continuación describiremos los pasos para poder conectar nuestra aplicación con Office 365. Tenemos que tener en cuenta los requisitos mencionados anteriormente para poder realizarlo con éxito.

- Abrimos Visual Studio y creamos un proyecto Apache Cordova.

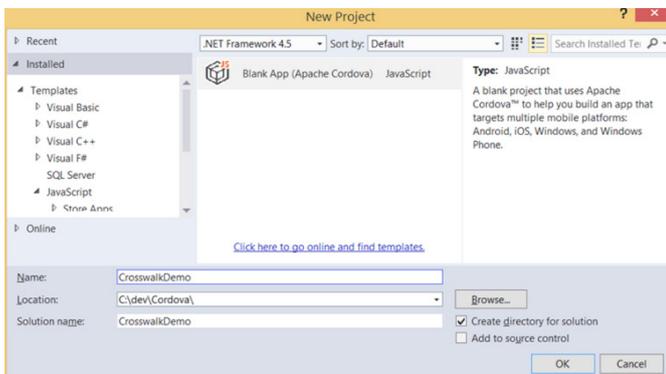


Imagen 3.- Añadiendo un proyecto Apache Cordova

- Desde el explorador de soluciones incluiremos en el proyecto un servicio conectado:

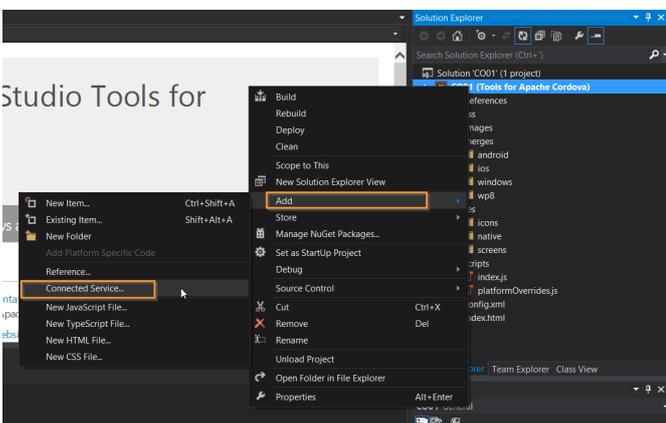


Imagen 4.- Añadiendo un servicio conectado.

- Registramos nuestra App en Azure Directory Services, definiendo los permisos para cada servicio al cual que-

ramos conectar. En nuestro caso, vamos a proceder a conectar a SharePoint y para ello seleccionaremos Sites:

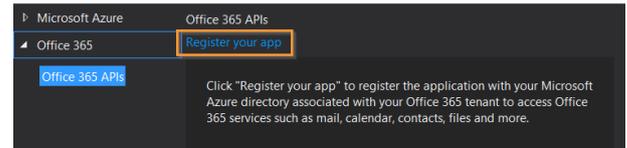


Imagen 5.- Registrando nuestra aplicación en Azure Active Directory.

El framework nos proporciona la manera de encapsular y paquetizar nuestro código HTML y esto lo hace a través de un Webview, que es el que se encarga de renderizar dicho contenido

- Una vez registrada nuestra aplicación nos aparecerá en el panel de Azure nuestra aplicación, del tipo cliente nativo.



Imagen 6.- Detalle de nuestra aplicación en Azure Active Directory.

- Si vemos el detalle, podremos ver el nombre de la solución, el Id de cliente y la URL de la redirección.

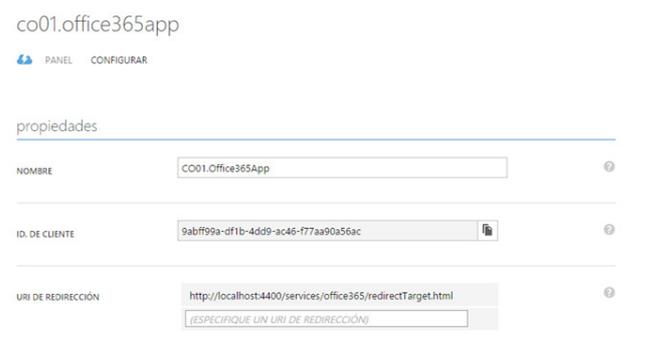


Imagen 7.- Propiedades de nuestra aplicación en Azure Active Directory.

- Al haber creado el servicio conectado desde Visual Studio, las propiedades de la conexión quedan automáticamente configuradas en el fichero settings.js



Imagen 8.- Archivo JavaScript con las propiedades del servicio.

- En la solución de Visual Studio nos incluirá la carpeta Services con todos los scripts necesarios:

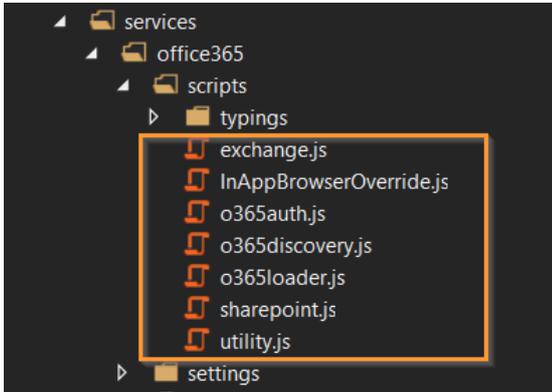


Imagen 9.- Archivos JavaScript agregados a través del servicio conectado.

Introducción a KnockoutJS

Se trata de una librería JavaScript que facilita la creación de interfaces basándose en el patrón MVVM desde el lado del cliente. Las características de este Framework JavaScript son las siguientes:

- Permite separar los datos del dominio, los componentes y los datos a mostrar.
- Contiene una capa definida por el código para gestionar las relaciones entre los componentes y las vistas.
- Utiliza funciones de gestión de eventos nativos de JavaScript.
- Permite simplificar las relaciones y hacer que la visualización sea dinámica y rica en experiencia de usuario.

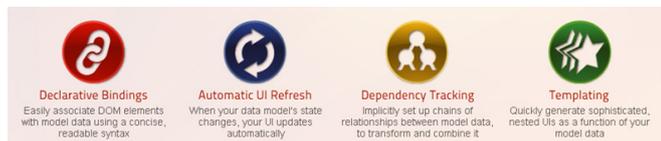


Imagen 10.- Características del Framework KnockoutJS.

Patrón MVVM

Es un patrón que se originó a partir de Microsoft (John Gossman) como una especialización del patrón de diseño modelo de presentación introducida por Martin Fowler.

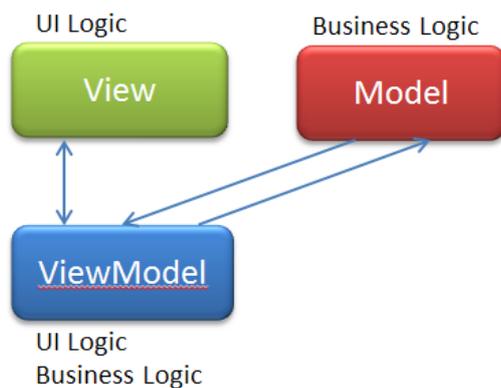


Imagen 11.- Separación de las capas en el patrón MVVM.

El siguiente ejemplo extraído de <http://learn.knockoutjs>.

[com/#/?tutorial=intro](#) muestra las características del Framework de acuerdo a la imagen 11.

Vista:

```
<p>First name: <input data-bind="value: firstName" /></p>
<p>Last name: <input data-bind="value: lastName" /></p>
```

VistaModelo:

```
// This is a simple *viewModel* - JavaScript that defines the data
and behavior of your UI
function AppViewModel() {
  this.firstName = ko.observable("Bert");
  this.lastName = ko.observable("Bertington");
}
// Activates knockout.js
ko.applyBindings(new AppViewModel());
```

¿Porque KnockoutJS y no Angular?

Elegimos KnockoutJS y no Angular porque como muchos de vosotros, preferimos seleccionar cada herramienta en función de lo que sabe hacer mejor frente a la alternativa de todo en uno que representa Angular, aunque también sea una opción más que válida en escenarios más complejos.

Creación de la Aplicación

Nuestro objetivo es consumir en un dispositivo móvil información hospedada en sitios de SharePoint Online en Office 365. En nuestro caso de ejemplo, se tratará de una lista de noticias que tiene la siguiente estructura:

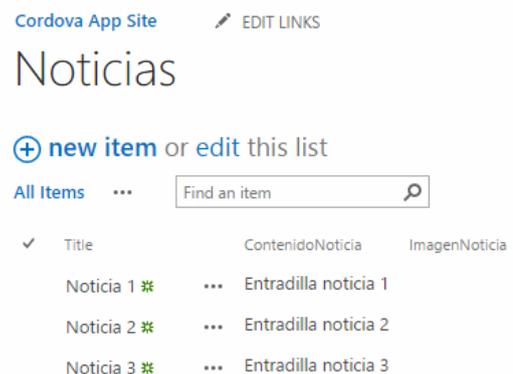


Imagen 12.- Lista de ejemplo donde obtendremos nuestra información

Como parte de nuestros objetivos, conectaremos a nuestro Office 365 dejando que Azure, y en concreto el servicio de Azure Directory, se encargue de gestionar tanto los permisos como el acceso. A nivel de arquitectura de la Aplicación, tendremos:

Capas

- Vista: Usamos esta capa para mostrar la información.
- Vista Modelo: Usamos esta capa para relacionar nuestros datos con la capa de presentación.

- Modelo: Usamos esta capa para obtener la información de SharePoint.

Acciones

- Creamos un fichero HTML , donde pondremos la Vista

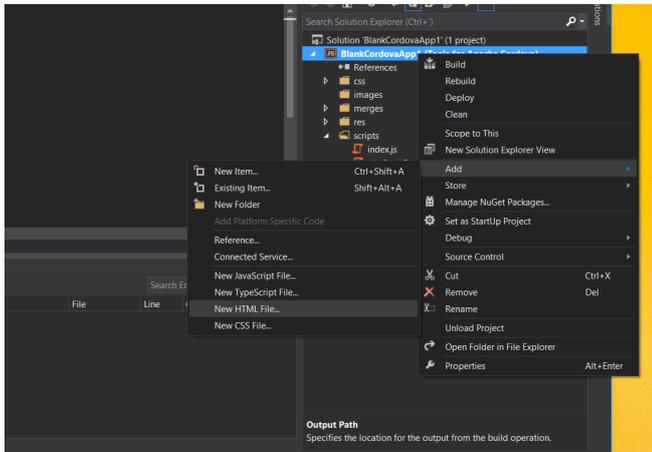


Imagen 13.- Creación de un archivo HTML en la Aplicación

A continuación vamos a ir viendo como quedarían configurado cada uno de estos elementos:

Vista:

```

<script src="cordova.js"></script>
<script src="scripts/platformOverrides.js"></script>
<script src="scripts/jquery-2.1.1.js"></script>
<script src="services/office365/settings/settings.js"></script>
<script src="services/office365/scripts/InAppBrowserOverride.js"></script>
<script src="services/office365/scripts/utility.js"></script>
<script src="services/office365/scripts/o365auth.js"></script>
<link href="Content/bootstrap.min.css" rel="stylesheet">
<script src="scripts/bootstrap.min.js"></script>
<script src="services/office365/scripts/aadgraph.js"></script>
<script src="scripts/app/index.js"></script>
<script src="scripts/app/listClient.js"></script>
<script src="app/index.js"></script>
<script src="scripts/knockout-3.2.0.js"></script>

<h3>Noticias <span class="label label-default">New!</span></h3>
<div class="list-group">
  <!-- ko foreach: news -->
  <!-- ko if:($index0 === 0)-->
  <a href="#" class="list-group-item active" data-bind="text:Title"></a>
  <!--/ko-->
  <!-- ko if:($index0 !== 0) -->
  <a href="#" class="list-group-item" data-bind="text:Title"></a>
  <!--/ko-->
  <!--/ko-->
</div>

```

podemos concluir que es posible crear una aplicación con tecnologías HTML y JavaScript de forma sencilla y encapsulada en una aplicación para poder consumir la información desde nuestro dispositivo portátil, sin importar cuál sea

VistaModelo:

Creamos un fichero JS y agregamos Vista Modelo:

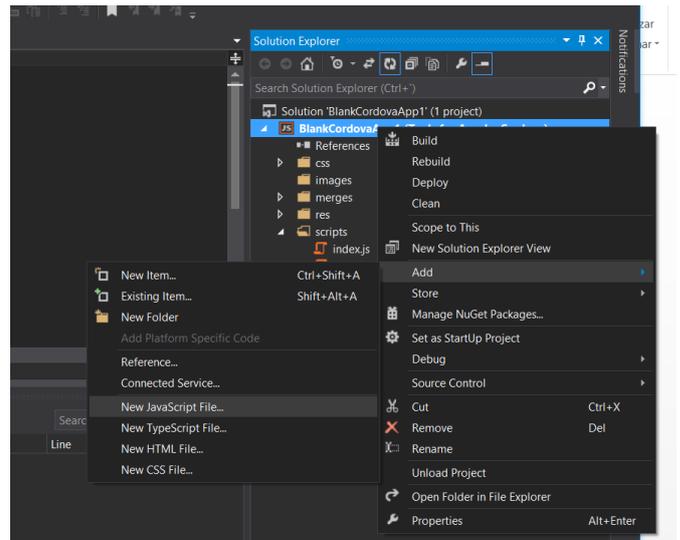


Imagen 14.- Creación de un archivo JavaScript en la Aplicación

```

$(document).ready(function () {
  listClient.getAccessToken(getAccessTokenSuccess, getAccessTokenFail);
});

function GetNews() {
  var url = O365Auth.Settings.sitecollectionUrl + "/_api/lists/GetByTitle('Noticias')/Items";
  listClient.getListItems(url, (OnShowNewsSucceed).bind(this), (OnShowNewsFail).bind(this));
}

function OnShowNewsSucceed(results) {
  var viewModel = {
    news: ko.observableArray(results)
  };
  ko.applyBindings(viewModel);
}

function OnShowNewsFail(xhr) {
  alert("Load Fail " + xhr);
}

```

Modelo:

Creamos un fichero JS y agregamos el Modelo:

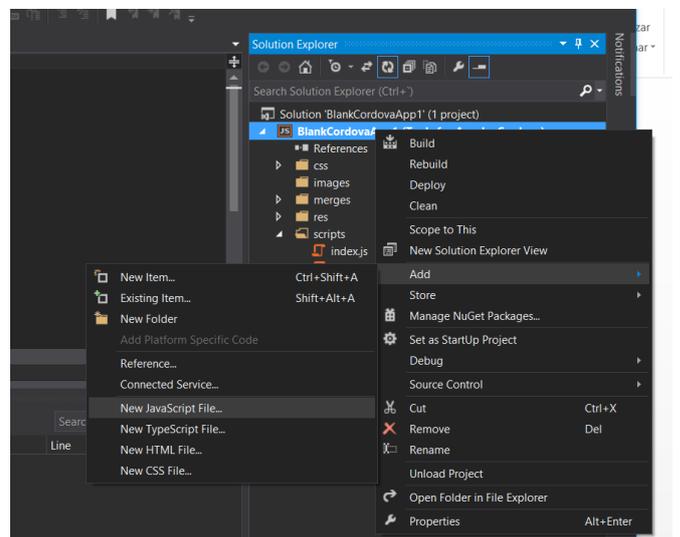


Imagen 15.- Creación de un nuevo archivo JavaScript en la Aplicación

```

var listClient =
{
  accessToken: "",
  emailaccessToken: "",
  authContext:null,

  getListItems: function(url , successcallback, failcallback)
  {
    $.ajax({
      type: "GET",
      async:true,
      url: url,
      headers: { "Authorization": "Bearer "+this.getToken(),
        "accept": "application/json;odata=verbose"},
    });
    done(
      (function (data) {
        //console.log(data);
        var listitems = this.parseJsonDataToArray(data);
        successcallback(listitems);
      }).bind(this)
    )
    .fail(function (jqXHR, textStatus) {
      failcallback(textStatus)
    });
  }
  getAccessToken: function(successCallBack, rejectCallBack)
  {
    //authenticate to Office 365
    if (this.authContext == null)
    {
      this.authContext = new O365Auth.Context();
    }

    //authContext.logout();
    this.authContext.getAccessToken(O365Auth.Settings.
resourceId)
    .then(
      (function (token) {
        this.setToken(token);
        successCallBack();
      }).bind(this)
    )
    ,
    function (reason) {
      rejectCallBack(reason);
    }
  )
  }
}

```

Flujo de Operación

Como se puede apreciar, el flujo de operación implica:

- En el document ready obtenemos el token de acceso con la función listClient.getAccessToken().
- Una vez obtenemos el token llamamos a la función listClient.getListItem() y hacemos una llamada API REST

donde le pasamos el token de acceso.

- Una vez obtenemos los elementos llamamos a la función OnShowNewsSucceed() y pintamos los datos.

Resultado del caso Práctico

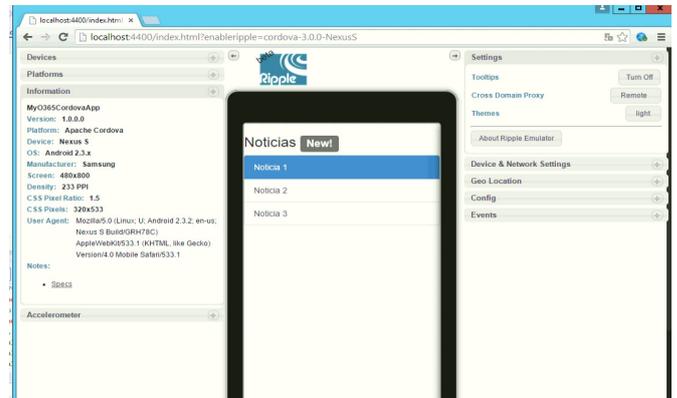


Imagen 13.- Resultado de nuestra aplicación.

Conclusiones:

A través del ejemplo mostrado, podemos concluir que es posible crear una aplicación con tecnologías HTML y JavaScript de forma sencilla y encapsulada en una aplicación para poder consumir la información desde nuestro dispositivo portátil, sin importar cuál sea. En el próximo artículo hablaremos sobre cómo crear una Aplicación nativa y cómo consumir información de nuestro SharePoint.

RUBÉN TORIBIO

SharePoint Architect en Tokiota

@rtoribiog

JAIME ARROYO

SharePoint Architect en Neccsia

@jaumeouyea

Protegiendo VMWare con Azure Site Recovery

Azure Site Recovery es el servicio de recuperación que contribuye a la continuidad de negocio a la estrategia de recuperación ante desastres (BCDR) orquestando la replicación, la conmutación por error y recuperación de máquinas virtuales y servidores físicos. Este servicio, cuya capacidad y versatilidad está más que comprobada en entorno multihost (Enterprise to Enterprise E2E) o híbridos (Enterprise to Azure E2A) se ha enriquecido en los últimos tiempos con la adquisición por parte de Microsoft de InMage. Este software permite la salvaguarda de prácticamente cualquier plataforma, pero en particular viene a resolver un caso que a menudo nos encontramos en muchos clientes: La plataforma de virtualización VMWare.

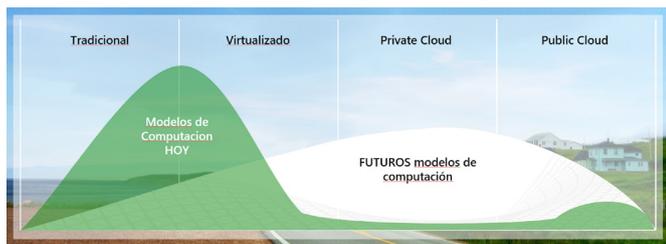


Imagen 1.- Etapas de madurez en modelos de computación

Muchos clientes, en su etapa de virtualización, virtualizaron sus máquinas sobre plataformas host VMWare. Esta situación, hasta ahora, ha sido un impedimento para proteger sus máquinas con Azure Site Recovery, y en muchos casos ralentizaba su adopción por la necesidad de convertir de forma manual sus máquinas antes de ser subidas a la nube. Mediante el uso de InMage, y de nuevas herramientas que hay en camino nacidas de la integración entre ambos servicios, este camino ha sido allanado, y en este artículo vamos a hablar sobre cómo implementar Site Recovery en Azure para proteger VMs alojadas en un VMWare ESXi.

Azure Site Recovery es el servicio de recuperación que contribuye a la continuidad de negocio a la estrategia de recuperación ante desastres

Algunas de las ventajas que ofrece son:

- Replicación simple, conmutación por error y recuperación con el portal de Azure sitio de recuperación.

- La replicación de datos a través de Internet, una conexión de sitio a sitio VPN, o sobre Azure ExpressRoute.
- La conmutación por recuperación (restauración) de Azure a una infraestructura de VMWare en las instalaciones.
- Consistencia multi-VM para que las máquinas virtuales y servidores físicos que ejecutan cargas de trabajo específicas se pueden recuperar junto a un punto de datos consistente.
- Los planes de recuperación para la conmutación por error simplificada y la recuperación de las cargas de trabajo escalonados en varios equipos.

Arquitectura de la solución

Para empezar, un pequeño mapa de cómo va a funcionar:

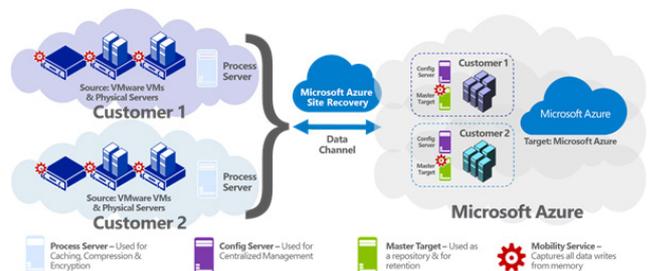


Imagen 2.- Diagrama de Azure Site Recovery.

Máquinas On-Premise

Las máquinas On-Premise a proteger, que pueden ser principalmente de dos tipos: Máquinas físicas, o máquinas virtuales que se ejecutan en un hipervisor VMWare ESXi.

Servidor de proceso On-Premise

Las máquinas protegidas envían datos de replicación al servidor de proceso, el cual realiza una optimización sobre estos datos y los envía al servidor maestro de destino en Azure. Este servidor cuenta con una memoria caché basada en disco para almacenar en los datos de replicación recibidos y también se encarga de coordinar los diferentes agentes del servicio de movilidad que se deberán instalar en cada máquina virtual o servidor físico que se desee proteger. Además, también se encarga de realizar la detección automática de servidores VMWare vCenter.

El servidor de proceso debe ser un servidor virtual o físico Windows Server 2012 R2 y se recomienda que esté situado en el mismo segmento LAN y red que las máquinas a proteger, aunque es posible ejecutarlo en una red diferente siempre y cuando las máquinas protegidas tengan visibilidad con el servidor. Durante despliegue podrás configurar el servidor de procesos e inscribirte para el servidor de configuración.

Servicio de Almacenamiento de Azure Site Recovery

También denominado Vault o bóveda, el servicio de almacenamiento sirve tanto como repositorio de las imágenes (apoyándose en un blob storage), validador de seguridad, y gestor de failover y recuperación, todo de una forma integrada y sencilla, como un servicio de Azure.

Servidor de Configuración

El servidor de configuración coordina la comunicación entre las máquinas protegidas, el servidor de procesos y los servidores de destino en Azure. Para ello, establece la replicación y coordina la recuperación en Azure cuando se produce una conmutación por error. El servidor de configuración se ejecuta en una máquina virtual Azure Standard A3 dentro de la misma suscripción Azure en la que se ha establecido el Servicio de Almacenamiento, y durante el despliegue es necesario configurar el servidor e inscribirlo en el Servicio de Almacenamiento de Azure Site Recovery por medio de una key de seguridad.

Servidor Maestro Destino

El servidor maestro de destino posee los datos replicados de sus máquinas protegidas utilizando VHD adjuntos creados en el almacenamiento de blob en su cuenta de almacenamiento Azure. Un servidor maestro se puede implementar como una máquina virtual Azure ejecutando Windows 2012 R2 y basado en una de las imágenes de la galería cuando se van a proteger máquinas Windows, y como un servidor OpenLogic CentOS 6.6 cuando se van a proteger máquinas Linux. Para ambos casos, las opciones de dimensionamiento son A3 de la serie estándar, o D14 de la serie D. Este servidor debe estar conectado a la misma red de Azure que el servidor de configuración, y también ha de ser registrado en el servicio de Almacenamiento, y en el servidor de configuración.

Agente de Movilidad

El agente de movilidad (InMage) se instala en cada servidor físico o máquina virtual VMWare que se desea proteger, y es el encargado de enviar los datos de replicación al servidor de procesos, que a su vez los envía al servidor maestro de destino en Azure. El servidor de proceso puede instalar automáticamente el agente de movilidad en las máquinas protegidas, o se puede desplegar el servicio manualmente descargándolo desde el servicio de almacenamiento.

Para la comunicación de datos y el canal de replicación existen un par de opciones. En ninguno de los casos se requiere abrir puertos de red en las máquinas protegidas, por lo que la seguridad se mantiene en todos los casos:

Opción 1: A través de una conexión site-to-site: Este canal seguro permite la comunicación y replicación desde la red local hasta Azure a través de una conexión segura y cifrada mediante un protocolo de IpSec

Opción 2: A través de una conexión ExpressRoute: Para los casos en los que se estima que existirá una alta demanda de traspaso de información hacia/desde la nube, y se requieren unos estándares muy altos de seguridad, es posible establecer la conexión VPN a través de ExpressRoute, la cual ejerce como una fibra negra entre nuestros entornos On-Premise y la nube y requiere ser contratada a través de un ISP Provider

El servidor de proceso debe ser un servidor virtual o físico Windows Server 2012 R2 y se recomienda que esté situado en el mismo segmento LAN y red que las máquinas a proteger

Manos a la obra

Ya hemos repasado la teoría que sustenta la conexión, pero ahora, vamos a configurar una conexión segura.



Imagen 3.- Creación Almacén Azure Site Recovery.



Imagen 4.- Almacén creado.

Nos descargamos la versión más reciente de Azure Site Recovery para el servidor de Configuración.

Nota: Cada vez que se descarga una clave de registro, se genera una nueva y solo la última será la válida.



- 1 Preparar servidores VMM
 - Descargue la última versión del proveedor de Microsoft Azure Site Recovery y la clave de registro. Después de instalar el proveedor en el servidor VMM, especifique la clave para registrar el servidor en el almacén. Cada vez que haga clic en "Generar archivo de clave de registro", se creará una nueva clave, y solo será válida la última clave generada.
 - [Descargar una clave de registro](#) [Descargar el proveedor de Microsoft Azure Site Recovery para instalarlo en servidores VMM](#)

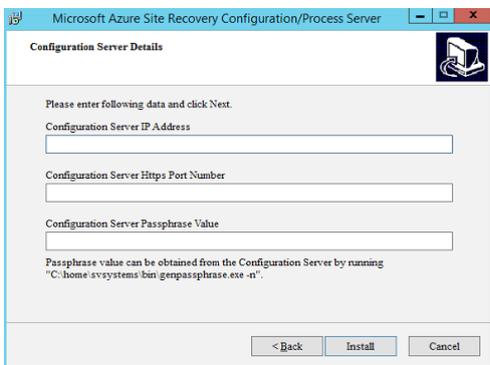
Imagen 5.- Preparación VM.

Añadimos una cuenta de almacenamiento. Debe ser con geo-redundancia activa en la misma región que la vault.

En este paso también vamos a instalar el agente del servidor de Proceso On-Premise

- 2 Preparar recursos
 - [Agregar cuenta de almacenamiento de Azure](#) [Descargar el agente de los Servicios de recuperación de Microsoft Azure para instalarlo en los servidores host de Hyper-V.](#)

Debemos indicar la IP interna así como la URL del servidor de configuración por HTTPS



Una vez registrado veremos una pantalla como esta:

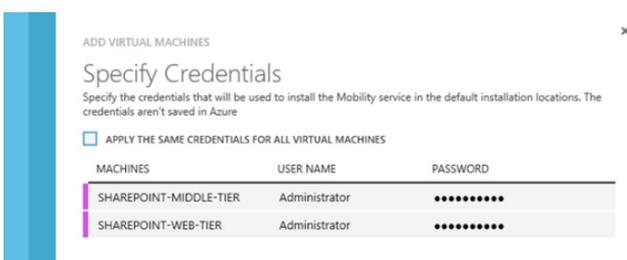
NAME	IP ADDRESS	HEALTH	PROTECTED ITEMS	VERSION
INVTST28	10.10.0.28	Healthy	0	RELEASE_8.2.0.0_PREVIEW_3728_Mar_18_2015
CONTOSO-PS	10.150.25.84	Healthy	0	RELEASE_8.2.0.0_PREVIEW_3728_Mar_18_2015

Configuramos la nube que va albergar las máquinas virtuales.

Nos descargamos el agente y lo instalamos en las VMs que vamos a proteger

- 3 Configurar protección de la nube
 - [Configurar protección para nubes VMM](#)

Una vez instalado los agentes los veremos en la consola



Asignamos las redes de destino para las VMs tras la con-

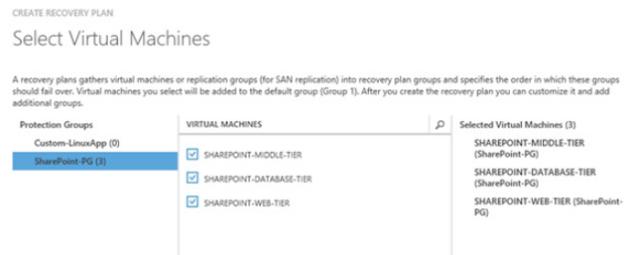
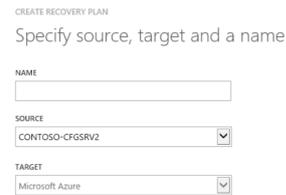
mutación por error

- 4 Asignar recursos
 - [Asignar redes](#)

Activamos la protección de las VMs ubicadas en la nube VMM

- 5 Proteger máquinas virtuales
 - [Habilitar protección de máquinas virtuales](#)

y por último, definimos el plan de recuperación, y lanzamos un test para comprobar que todo funciona adecuadamente



El agente de movilidad (InMage) se instala en cada servidor físico o máquina virtual VMWare que se desea proteger, y es el encargado de enviar los datos de replicación al servidor de procesos

Una vez lanzado podemos ver usar la pestaña de Trabajos para seguir el proceso de replicación.

y eso es todo! esperamos que os haya parecido sencillo y si tenéis cualquier duda sobre el detalle del proceso, podéis aprovechar y escribirnos a cualquier de los dos autores.

FABIÁN CALVO VALMORISCO

Gerente Soluciones Cloud Encamina

fcalvo@encamina.com

@rainfc

<https://blogs.encamina.com/elsextosharepoint>

ISRAEL URREA CARBALLO

Consultor Infraestructuras y Cloud Encamina

iurrea@encamina.com

@israurrea

¿Hasta dónde podemos llegar con Nintex y SharePoint? – Análisis y casos prácticos

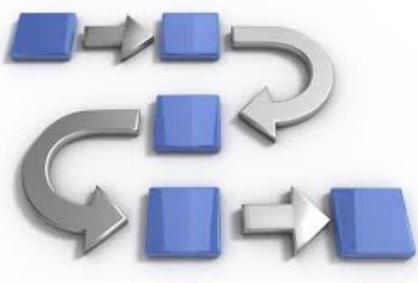
Como siempre la comunidad .Net avanza a pasos agigantados, salen herramientas nuevas que ni conocemos, nos llegan las versiones de SharePoint casi sin darnos cuenta y para colmo entra en liza el mundo Cloud. ¿Pero estamos realmente utilizando todo el poder de los componentes que tenemos a nuestro alrededor? Un ejemplo claro es Nintex, un producto integrado 100% con SharePoint desde la versión 2007, y que en mi modesta opinión al menos en los proyectos que yo he tenido el lujo de colaborar lo hemos usado sin realmente expresar todo su potencial, y que hemos introducido más como un añadido que como un “requisito técnico” indispensable.

Nintex se puede definir como una herramienta de diseño de flujos de trabajo y procesos de negocio, que nos permite desde una completa interfaz gráfica definir nuestros procesos de trabajo sin necesidad de desarrollar código. Además, Nintex añade un potente editor de formularios “Nintex Forms” para listas de SharePoint, que nos permite editar los formularios de Creación, Edición y Detalle de forma sencilla desde la propia interfaz mediante un editor gráfico y un conjunto de reglas. Dicho esto ¿Por qué acabamos usando Nintex para pequeños proceso de negocio? ¿Podemos basar todas nuestras soluciones en Nintex sin tener preocupación alguna? ¿Podemos hacer un componente 100% basado en Nintex?

Nintex como todo en esta vida, tiene sus ventajas y sus desventajas, y en este artículo voy a intentar aclarar cuando es más recomendable o no basar nuestras aplicaciones en Nintex para soluciones SharePoint On Premises.

Nintex como motor de reglas

Una de las preocupaciones más presentes en un cliente es el mantenimiento de la aplicación, y por ello tienden a pedir que los desarrollos sean configurables y parametrizables para evitar costes futuros en despliegues y evolutivos.



Con esta idea, podíamos pensar inicialmente que Nintex es una buena solución para un usuario, dado que podemos implementar en flujos gran parte de la lógica de negocio, y si necesitamos incorporar algún cambio a futuro no nos va a implicar realizar ningún despliegue.

Esto es real, y salvo una inversión inicial para desarrollar los flujos, el mantenimiento es mucho más bajo, además de que se pueden realizar cambios en los propios entornos casi en tiempo real y publicándolos sin afectar al usuario.

¿Entonces cuál es el problema? – Como siempre el rendimiento es el problema, la realidad es que Nintex 2013 es un motor de workflow y el rendimiento es cuestionable cuando hablamos de una aplicación de negocio donde el usuario espera respuestas instantáneas.

Debemos entender que los flujos de Nintex son procesos asíncronos que se ejecutan en cola y que nos puede causar retardos si el volumen de peticiones concurrentes llega a ser muy elevada.

Nintex como editor de formularios: Nintex Forms

Nintex Forms se nos presenta como una clara alternativa a la hora de desarrollar formularios de forma intuitiva y sin requerir tener amplios conocimientos de programación. Desde una solución grafica bastante conseguida, al más estilo Infopath podemos desarrollar formularios contra listas de Sharepoint de forma sencilla y rápida.

InfoPath vs. Nintex Forms

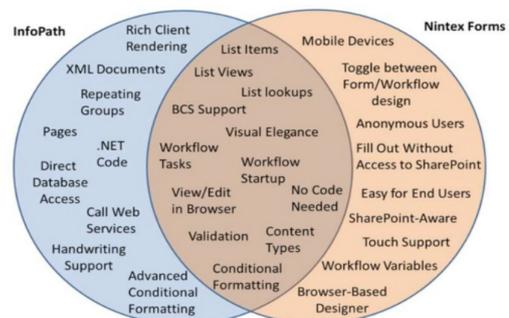


Imagen 1.- Comparativa InfoPath vs Nintex.

Pero claro está para poder decir que Nintex es “el rey” de los formularios, debemos comparar antes con las otras opciones disponibles. Omitiendo las opciones de SharePoint

Designer y XSLT o JSLINK, que son soluciones más orientadas a un programador (sobre todo JSLINK), vamos a intentar comparar al discontinuado InfoPath (que no retirado, ya que continúa en SharePoint 2016) con Nintex Forms, para poder poner en situación que es Nintex Forms:

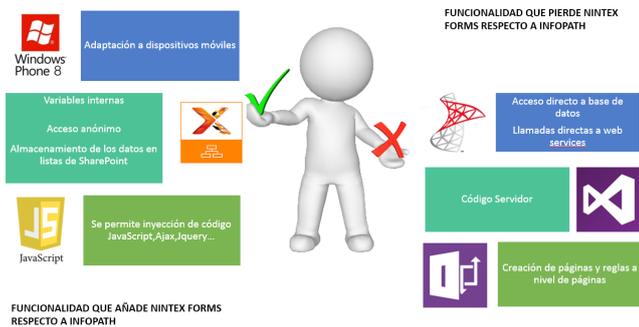


Imagen 2.- Qué me ofrece Nintex frente a InfoPath.

Analizando la comparación podemos deducir que, si conseguimos que nuestros metadatos estén 100% sobre listas de SharePoint, podemos decir que Nintex Forms es una solución idónea, dado que es mucho más mantenible que lo era InfoPath, nos permite inyectar código cliente y al no tener código servidor asociado no necesitamos desplegar para poder realizar cualquier cambio.

Por otro lado, esta solución no nos sirve para formularios personalizados que requieran llamadas a servicios web, integración directa contra base de datos o que tenga una gestión de vistas compleja. Aclarar que Nintex Forms se basa en el tipo de contenido de la lista, y que por cada cambio que hagamos sobre las columnas de la lista, deberemos revisar tanto el formulario como las reglas configuradas en él.

En conclusión, podemos decir que para un escenario en el que nuestros datos estén en listas de SharePoint, y en el que sabemos que no vamos a sufrir demasiados cambios en la estructura de la lista, esta solución es idónea y podemos obtener unos resultados excelentes.

En ocasiones, usamos Nintex por el hecho de darle un añadido a nuestras soluciones, limitando su importancia dentro de los proyectos

Caso práctico: Gestión de aprobaciones

En este apartado vamos a ver cómo realizar un componente con Nintex y Nintex Forms del cual obtenemos un buen resultado tanto a nivel de rendimiento como de costes en mantenimiento. Vamos a desarrollar un componente que permita a los miembros de un departamento de una empresa firmar un documento de Word como pudiera ser un documento de Compra de forma automática. Para ello vamos a seguir los siguientes pasos:

Creación de una biblioteca

En primer lugar vamos a crear una biblioteca en SharePoint, por ejemplo una biblioteca de documentos Estándar, y añadimos una columna del tipo persona que sea Firmante. Para esta versión vamos a indicar que solo admita un usuario ya que solo vamos a permitir un aprobador por documento

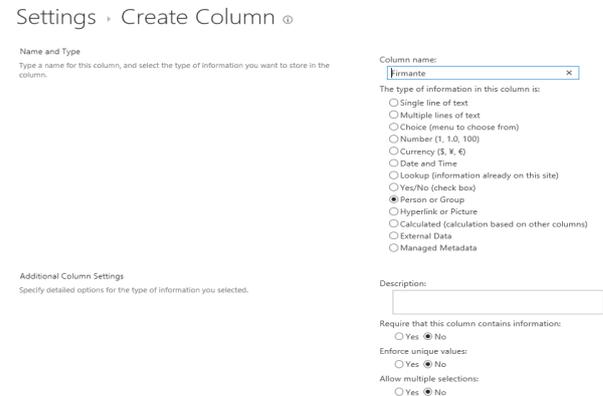


Imagen 3.- Creación de la columna.

Creación de Nintex Workflow y Nintex Forms

Creamos desde la ribbon un nuevo formulario de Nintex muy básico que únicamente va a solicitarnos el usuario firmante de este documento.

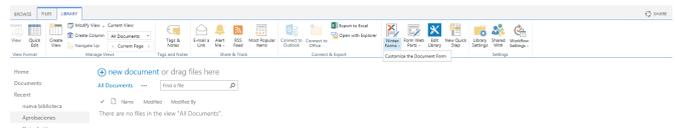


Imagen 4.- Creación del formulario de Nintex.

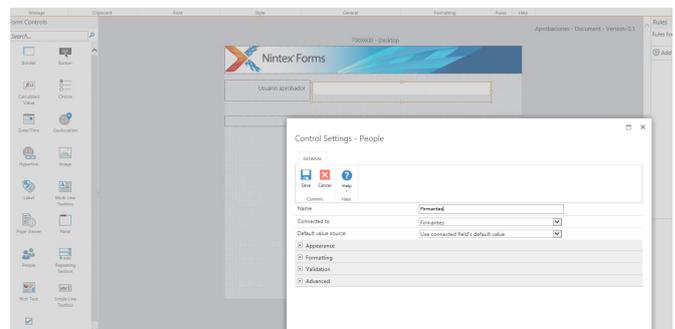


Imagen 5.- Pantalla de edición del formulario.

Creamos un nuevo workflow utilizando una máquina de estados de Nintex tal cual muestra la siguiente imagen:

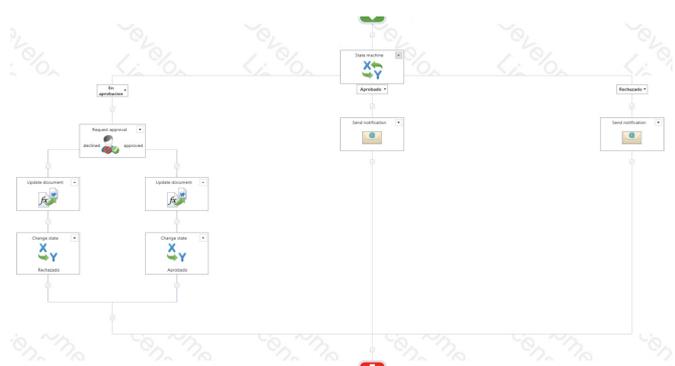


Imagen 6.- Diseño del Workflow de Nintex

Realmente para este caso concreto en el que solo hay un

aprobador, podríamos no usar la máquina de estados, pero como veremos después para una aprobación múltiple de usuarios nos da muchas ventajas. Si observamos el flujo lo primero que hacemos es obtener el usuario firmante, posteriormente creamos una tarea de aprobación (Request approval) que le hará llegar un mail al usuario firmante. Podemos configurar el texto del mensaje tal cual vemos en la imagen, siempre dejando un enlace a la página de aprobación.

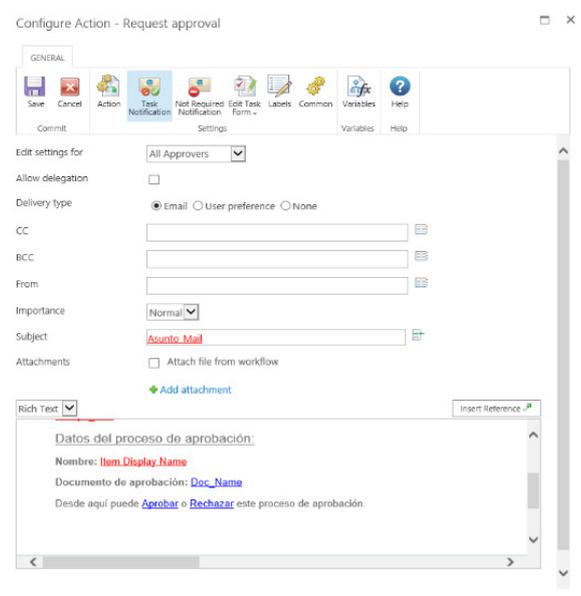


Imagen 7.- Edición de la actividad de aprobación,

Esta caja de Nintex “Request approval”, lo que hace es generarnos una tarea para el usuario destinatario de la aprobación. Una vez el usuario apruebe o no la tarea firmaremos el documento de Word desde Nintex, dejando el resultado de la tarea de aprobación y la firma en el documento.

Configuraremos el flujo para que se inicie siempre que editemos el documento desde la biblioteca, así cuando guardemos el usuario firmante desde el formulario de Nintex Form se iniciará el flujo (lo encontramos en propiedades del flujo).

Nintex se puede definir como una herramienta de diseño de flujos de trabajo y procesos de negocio, que nos permite desde una completa interfaz gráfica definir nuestros procesos de trabajo sin necesidad de desarrollar código

Plantilla Word

Para que Nintex pueda grabar datos en el documento adjunto vamos a decantarnos por introducir marcadores especiales en el documento. Esta es la forma más fácil, pero a la vez la más limitada comparada con hacerlo usando Open XML. Necesitamos acceder al documento de Word

que usaremos como plantilla, y habilitar en la barra de herramientas la opción “Desarrollador”. Lo conseguimos accediendo a Opciones, Personalizar Cinta de Opciones y marcamos en la parte de la derecha el check “Desarrollador” con todos sus componentes.

Ahora sobre la ribbon en Desarrollador seleccionamos **Aa**, y arrastramos el control de texto enriquecido a nuestro documento, hasta conseguir un diseño de documento parecido a esto:

FIRMA DE DOCUMENTO DE PRUEBA

En el presente documento se presentan las condiciones del contrato pactado en el acta de la reunión celebrada el 12-05-2015:

- Presupuesto: 1.000.000 €
- Fecha Inicio Proyecto: 07-07-2015
- Fecha Fin Proyecto: 12-12-2015
- Condiciones de cobro: 90 días

Estado aprobación condiciones proyecto: Rechazado Aprobado

Fdo.: Firma

Imagen 8.- Diseño final del documento.

Es muy importante darle nombres a nuestros controles de texto enriquecido (Rechazado, Aprobado), ya que este nombre es interno y único, y deberemos configurarlo en la caja del flujo para escribir en Word de la siguiente forma:

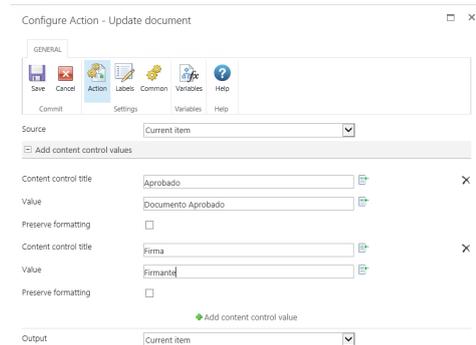


Imagen 9.- Actividad de actualización de documento.

Configuramos que añada el texto “Documento Aprobado” al marcador de Aprobado e inserte el valor del usuario firmante desde la columna creada en el primer paso (proceso similar para escribir el rechazo del documento). Al terminar la firma del documento (aprobado o rechazado), la maquina salta a estado Aprobado o Rechazado en función de la aceptación del usuario firmante, y le llega un mail de fin de proceso al usuario iniciador del flujo (usuario que configuró la aprobación).

Iniciar Aprobación

Tal cual hemos montando el componente, basta con subir un documento de Word con la plantilla definida (podemos configurar la biblioteca para que use esta plantilla), y desde el formulario de creación (nuestro Nintex Forms), insertamos el usuario al que queremos que llegue la notificación. Debe de ser un usuario con permisos en el sitio de la biblioteca y con correo en su perfil de SharePoint.

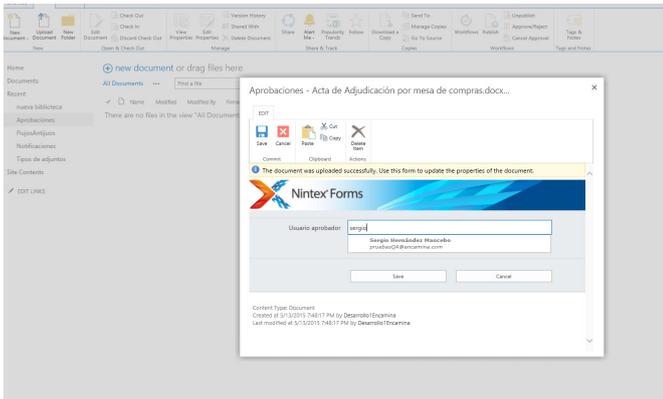


Imagen 10.- Formulario del documento.

Versión con múltiples aprobadores

Como hemos visto esta máquina de estados es muy sencilla porque solamente tenemos un aprobador por documento, pero si cambiáramos la columna firmante y aceptáramos multi-selección, se complicaría bastante el proceso. En este caso podríamos evolucionar nuestro flujo a algo parecido a esto:

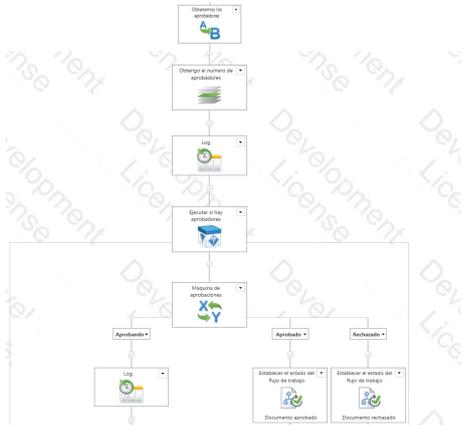


Imagen 11.- Ejemplo de diagrama de flujo de aprobación.

En este caso ya tratamos más de un aprobador, y la máquina de estados coge más peso en la gestión del flujo. Necesitamos obtener el número de aprobadores, y los aprobadores. Mediante una expresión regular obtenemos los aprobadores y los almacenamos en una colección de datos (similar a una lista en C#).

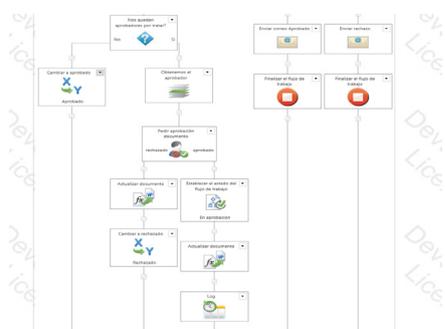


Imagen 12.- Continuación del ejemplo de diagrama de flujo de aprobación.

Como vemos el flujo se encarga de evaluar si nos quedan aprobadores por tratar, y en caso afirmativo sigue mandando notificaciones vía mail para que aprueben o rechacen el documento. Si no nos quedan aprobadores y nadie ha rechazado se notifica la aceptación, en caso contrario se notifica el rechazo.

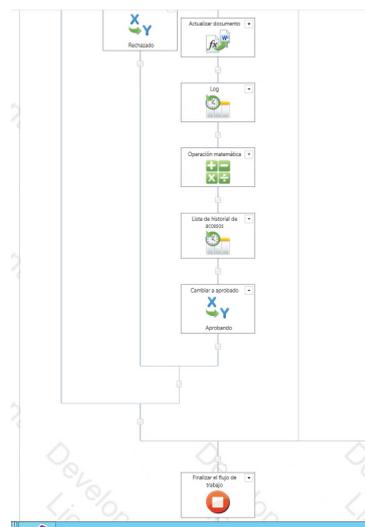


Imagen 13.- Continuación del ejemplo de diagrama de flujo de aprobación

Esta parte del flujo se encarga de modificar el documento, y recalcular cuantos aprobadores nos faltan por tratar.

Solo nos quedaría adaptar nuestra plantilla de Word para que acepte más aprobadores o bien concatenar los aprobadores en una variable en el workflow y escribirlos en el documento. Lo importante de este ejemplo es ver que con la máquina de estados podemos crear componentes complejos con el mínimo coste de diseño. Si intentásemos diseñar este flujo mediante recorridos o condiciones (cosa que sería totalmente viable), veríamos que el tamaño del flujo crecería proporcionalmente al número de estados que tengamos en nuestra máquina.

Conclusiones

En mi opinión personal con Nintex se pueden hacer cosas bastantes completas como las que hemos podido ver, y no limitarnos a aprobar tareas, mandar notificaciones o cambiar valores de una columna en SharePoint. Le queda camino por andar todo sea dicho, a día de hoy sigue siendo una cola de procesos asíncrona que para aplicaciones de alta concurrencia y que requieran una cierta velocidad se nos queda un poco pequeño.

En cuanto al editor de formularios, es una solución que globalmente supera con creces a InfoPath, aunque aún se limita bastante a Formularios de Listas en SharePoint, y carece de integración con fuentes de datos externas.

La unión de Nintex Forms y Nintex Workflow, sumado a los potentes servicios web que tiene publicado Nintex en Sharepoint, nos proporcionan herramientas suficientes como para hacer cosas interesantes, eso sí sin huir del Visual Studio que en ocasiones nos dará menos dolores de cabeza.

SERGIO HERNÁNDEZ MANCEBO

Team Leader en ENCAMINA

shernandez@encamina.com

La importancia de diseñar y aplicar un buen uso de Gobernanza en nuestro entorno SharePoint 2013 – Parte II

Enfocando el modelo operativo de SharePoint donde el usuario es un cliente conseguimos dotar de mentalidad positiva a las operaciones, al definir los límites y limitaciones del servicio, implícitamente estamos dotando a los Administradores del Sistema capacidades para establecer expectativas con los usuarios del servicio, beneficiándose de una estructura de planificación que orienta las decisiones presupuestarias y de dotación de recursos.

El punto importante es definir lo que el servicio que ofrecemos, lo que implica que el servicio, y qué recursos es responsable de qué acciones.

La idea es que sepamos como:

- Hacer que el ámbito de aplicación del servicio de SharePoint sea explícito e intencional.
- Establecer diferentes niveles de servicio para dirigirse a diferentes necesidades.
- Organizar y priorizar las entradas de solicitud de servicio.

La determinación del alcance de su servicio de SharePoint

Hay varias razones por las que podrían querer un amplio margen para su primer lanzamiento de SharePoint. A continuación se enumeran algunas razones que a menudo encontramos:

- Consultoría de empresas quieren un amplio margen para que puedan asegurar su propia tubería de suministro.
- Los promotores de proyectos a menudo quieren hacer una mezcla grande con un wow-factor en un comunicado de que puedan adjuntar su nombre;
- Los usuarios están entusiasmados con todas las diferentes características de SharePoint y no pueden priorizar entre cuáles quieren primero.

Me gusta pensar en ello como una muerte por mil cortes de papel. Cada aspecto marginal en el proyecto en sí mismo no parece demasiado grande. Cuanto mayor sea el número de requisitos a definir en el servicio, más se aumenta el riesgo y se ralentiza el proceso. Cada corte de papel solamente podría venir con una pequeña picadura, pero todos se suman y trabajan en nuestra contra. Con un ciclo de entrega regular, puede evitar la carga de cualquier ciclo

de entrega individual.

Usted debe entregar rápidamente, ofrecer con frecuencia, y entregar de forma incremental.

Una de las cosas que más me gusta de SharePoint es la forma en que puede crecer y adaptarse con el tiempo. No es como otros sistemas que necesitan de decisiones por adelantado. SharePoint no te obliga a desplegar la mayoría en el primer despliegue. Puede desplegar la infraestructura básica, con una línea de base y luego rápida y regularmente.

el modelo operativo de SharePoint donde el usuario es un cliente conseguimos dotar de mentalidad positiva a las operaciones, al definir los límites y limitaciones del servicio, implícitamente estamos dotando a los Administradores

A la hora de definir el ámbito de un servicio, podemos precisamente definir una línea básica y posteriormente ir incrementando funcionalidad, un poco más adelante veremos cómo determinar las características de SharePoint y su funcionalidad. Yo uso la metáfora de la fruta madura: lo que está en esas ramas SharePoint metafóricas más al alcance, que yo pudiera llegar rápida y fácilmente. No hay una respuesta única para todas las preguntas, al igual que con cualquier tipo de actividad de diseño, ya que su elección depende de su situación particular y prioridades individuales.

Digamos, por ejemplo, que las bases de operaciones resultan ser que fruta madura para usted. Sus usuarios utilizan actualmente recursos compartidos de archivos, pero su experiencia de intercambio de archivos no es tan rica o de colaboración, que frecuentemente suele ser. Queremos ofrecerles la posibilidad de revisar los documentos para la edición, el seguimiento de las versiones, para comentar sobre los cambios, e incluso establecer alertas de notificación cuando hay cambios. SharePoint hace esto muy bien, y este escenario lo convierte en una excelente fruta madura para encargarse de ello.

Ahora, ¿cómo se dibuja un cuadro alrededor de la entrega de bases de operaciones para proporcionar una experiencia de colaboración de documentos más rica para los usuarios? En primer lugar, necesitamos los servidores desplegados y SharePoint instalado, a continuación generamos un sitio web con una dirección URL a la que los usuarios podrán acceder, y entonces necesitamos decidir si los usuarios pueden aprovisionar sus propios sitios o si van a solicitar y tener el servicio para crear sitios para ellos. Ahí lo tienen: fruta madura que se puede entregar rápidamente y que va a proporcionar un valor inmediato a sus usuarios.

No hay marca, no hay plantillas de sitio, no hay desarrollo, no hay flujos de trabajo y sin campanas y silbatos. Usted sólo tiene el núcleo de los sitios de grupo de SharePoint, y que le ha hecho estos sitios disponibles para los usuarios en casi ningún momento. Sitios de grupo por defecto ya están desbordados con las campanas y silbatos integrados en los que hará las delicias de los usuarios que están acostumbrados a usar recursos compartidos de archivos. Los usuarios pueden asignar tareas, crear discusiones, e incluso pueden blog y enviar encuestas! Hay un montón de características que ya están disponibles para mantenerlos satisfechos, por lo que no necesitan inflar su alcance del proyecto de servicios de SharePoint inicial y retardo haciendo este valor a disposición de sus usuarios.

Entregar el servicio tan rápido como pueda, y luego extender y mejorar en una fase de seguimiento. La fase dos para este ejemplo podría incluir lo siguiente:

- Un diseñador puede diseñar los colores de la interfaz de usuario y logotipos, y el branding necesario para que los usuarios se sientan bien.
- Un desarrollador puede adjuntar una página maestra y los temas a los nuevos sitios usando la función de grado.

La identificación de diferentes niveles de servicio para diferentes necesidades

Cada cliente es diferente, y eso es cierto, independientemente si el cliente es un cliente externo o si son sus usuarios internos. Todos ellos tienen diferentes necesidades, capacidades diferentes, y diferentes prioridades. Cada uno tiene una combinación única de las tres que traen con ellos, ya que consumen el servicio de SharePoint, y tienen sus propias expectativas.

Tratando de individualizar el servicio para complacer a cada una de estas necesidades se puede desbordar el servicio rápidamente. Si tratamos de ser todo para todos, se termina arriesgando no siendo mucho a nadie. No podemos simplemente escalar el servicio si usted consume a sí mismo con el enfoque en cada usuario individual, y por lo que necesita una manera de agrupar usuarios en

grupos con necesidades comunes. Al mismo tiempo, no queremos alejarnos de estos clientes o desarrollar una actitud militante en el que tienen que ajustarse a lo que usted ofrece. No, queremos satisfacer las necesidades de nuestros clientes y ofrecer un servicio valioso que los apoyará en sus funciones diarias y funciones de trabajo. En lugar de individualizar el servicio que ofrecemos y lo que es específico para cada persona, puede simplemente dar la impresión de un servicio individualizado

¿Cómo se puede dar la impresión de hacer frente a las diferentes necesidades? Podemos ofrecer unos sabores del servicio donde la funcionalidad y uso donde agrupar características comunes en un nivel de servicio. Esto nos permite centrarnos en el uso más pesado para obtener los mayores beneficios para el momento de invertir, mientras que también proporciona un conjunto repetible y de bajo mantenimiento de las ofertas que satisfagan la mayoría de las necesidades con la menor cantidad de nuestra participación. A continuación, puede dividirse y dar prioridad a sus clientes, agrupándolos en niveles de servicio adecuados.

Otro enfoque para definir los niveles de servicio, si las devoluciones de cargos no son una opción consiste en supervisar el uso y la identificación de los usuarios pesados. Me refiero a estos como aquellos usuarios del servicio de SharePoint que son mis principales clientes. Si no tengo un modelo de cargo donde los clientes pueden suscribirse y auto-identificarse a su nivel de servicio más apropiado, entonces yo les identifico con base en sus tasas de uso y adopción reales. Utilizo una serie de medidas para identificar a estos clientes. En primer lugar, me veo en el número de usuarios activos que utilizan su sitio y la cantidad de contenido que almacenan en el sitio. Estas dos métricas me dan un indicador razonable de cuáles son mis clientes más grandes y más activos. Identificarlos mediante la ejecución de una secuencia de comandos en el servidor que enumera los sitios por tamaño o actividad.

La siguiente secuencia de comandos de PowerShell SharePoint 2013 lista los sitios en una granja y los ordena de mayor a menor en base a la cantidad de contenido que almacenan. Puedo usar esto como un método rápido y fácil para obtener una lista de todas las colecciones de sitios en un servidor y su tamaño. Usando la salida, puedo dirigir mi atención a los sitios en la parte superior de la lista, que sitios almacenan la mayor cantidad de contenido. Puedo usar esto en diferentes escenarios, por ejemplo, si quiero auditar almacenamiento para identificar oportunidades para reducir cualquier espacio desperdiciado de almacenamiento de contenido.

```
Get-SPSite | Select Url, @{Label="Size";Expression=($_.usage.Storage/1MB)} | Sort-Object -Descending -Property "Size" | ConvertTo-Html -Title "Site Collection List" | Set-Content SiteCollectionSizeReport.html
```

El Diseño de nuestro nivel de servicio

Podemos nombrar a sus niveles de servicio basado en números, como nivel uno, nivel dos, y así sucesivamente. Este método para mi gusto es bastante genérico y aburrido; pero lo que es peor, es cuando el número de niveles de servicio que restringirá a sí mismo por el propio sistema de numeración. Me siento obligado en este sistema de numeración de dos maneras: los números implican una jerarquía u orden, y esto se convierte en un problema cuando desea incluir niveles paralelos que ofrecen el mismo nivel de servicio con diferentes conjuntos de características; y la segunda manera es que la numeración hace que sea difícil insertar niveles de servicio adicionales más adelante, tales como 1B oferta nivel.

Cuanto mayor sea el número de requisitos a definir en el servicio, más se aumenta el riesgo y se ralentiza el proceso

Me gusta mucho más nombres para los niveles de servicio y los nombres más comunes parecen ser platino, oro, plata y bronce. Podemos utilizar otros nombres que muestren más descriptivo de sus servicios, tales como: básico, extranet, portal y repositorio. Alternativamente, es posible utilizar otros nombres que también ofrecen descripción de sus necesidades de recursos del servidor, tales como: la residencia, semiprivada, aislar las aplicaciones de servicio, aplicación web privada y agrícola dedicada. Puede ser que incluso mezclar y combinar entre estas estrategias de denominación. Todas estas estrategias de nombres son descriptivos y pueden ser significativos para los clientes, así que escoja uno que resuena con su organización y el tipo de niveles de servicio que desea ofrecer.

Generalmente empiezo con un nivel de servicio básico, y esto normalmente incluye un sitio de grupo de SharePoint predeterminado con mi cuota de sitio más restrictiva. Yo uso este nivel de servicio para incluir a todos los sitios de grupo que los usuarios crean con la creación de sitios sin servicio integrado para SharePoint. Esto permite a los usuarios a la libre prestación de un sitio genérico bajo demanda. Estos sitios bajo una ruta de acceso administrada y que comparten las aplicaciones de servicio asociadas a su aplicación web. Los usuarios pueden “inflar” estas páginas a varios gigabytes de tamaño, personalizar el diseño visual, y añadir funcionalidad adicional. Podrían agregar funcionalidad personalizada mediante un paquete de soluciones de usuario o un SharePoint 2013 App Del SharePoint Store o desde un catálogo de aplicaciones internas. Dependiendo de cómo me desglose de los niveles de servicio, que pueden ofrecer lo suficiente en este nivel básico para dar cabida a este tipo de necesidades para que pueda satisfacer las necesidades de la mayoría de mis usuarios, sobre todo si tengo una base de usuarios que es razonablemente

autosuficiente y que quiere ser habilitada para gestionar sus propios sitios.

El nivel de servicio básico puede ofrecer un montón de campanas y silbatos, pero a fin de que a escala y en gran parte de autoservicio impulsado, sitios de usuario en el nivel de servicios básicos tienen que compartir la URL de la raíz y de provisión de sus sitios bajo una ruta de acceso administrada. La mayoría de los usuarios no les importará, y probablemente ni siquiera pensar en esto, pero algunos prefieren una URL fácil y corta. A menudo me llamo a estas peticiones URL Vanity. Por sí solo, probablemente no es razón suficiente para romper los sitios en el nivel de servicio básico para comenzar a ofrecer los nombres de host individuales, pero todavía se puede satisfacer la mayoría de nuestros clientes que preguntar por una dirección URL Vanity. En mi respuesta a estas peticiones, recomiendo el aprovisionamiento del sitio con una URL normal bajo una ruta de acceso administrada, y luego ofrecer un servicio de redireccionamiento que utiliza DNS, IIS, o una aplicación ASP.NET sencilla que recibe las solicitudes realizadas a la URL de Vanity más corto y luego les redirige a la URL real de la colección de sitios.

- Una opción para redireccionar las URLs de Vanity a la URL de una colección de sitios bajo una ruta de acceso administrada implica añadir una regla de redireccionamiento con la función de reescritura de URL de IIS. Puedo crear las reglas para hacer una redirección a la URL y aprobar la solicitud en el servidor SharePoint frontend web. Puede descargar y aprender más acerca de la herramienta de reescritura de URL de IIS desde la siguiente URL: www.iis.net/download/urlrewrite.
- También he visto a los clientes a implementar un servicio centralizado de dirección de la URL donde crean una entrada DNS para un simple URL, como `http://ir`. A continuación, construir una aplicación de redirección de URL en el lugar donde reciban a esta Go URL. El Diseño es muy similar a cómo iba a funcionar un servicio de acortamiento de URL pública y lo utilizan para el sitio o página de referencias, perfil de las personas de referencias, referencias de mapa, y similares. Pueden anunciar un anuncio en un ascensor o de un pasillo con una URL corta y fácil como `http://go/102`, y el uso que para redirigir las solicitudes a las direcciones URL largas.

Puede acoger esta aplicación de redirección Ir URL centralizado en IIS utilizando una simple aplicación ASP.NET. Me gustaría crear una aplicación ASP.NET para proporcionar una forma donde los usuarios pueden crear su propio Go URLs, y almacenar la lista de direcciones URL de redirección en una base de datos de Microsoft SQL Server para mantener una lista dinámica de URLs.

Un ejemplo común es un nivel de servicio GOLD, que podría ofrecer cosas tales como una aplicación web dedicado para alojar un portal personalizado, aplicaciones de servicios aislados, una frecuencia de copia de seguridad mayor contenido, a un público más amplio de usuarios de los

contenidos, y similares. Estos niveles de servicio más altos suelen ser para los clientes internos que quieren construir un sitio de portal de departamento o algún otro tipo de aplicación personalizada alojado en SharePoint. Podrían ser un portal que aloja los formularios de informe de gastos y procesa los flujos de trabajo, un portal de aprobación de viajes y reservas, o un sistema de gestión de aprendizaje empresarial. Uno podría albergar un portal del Departamento de Recursos Humanos, que contiene varias aplicaciones personalizadas tales como las relacionadas con las revisiones de desempeño y planificación de la carrera. Estas son todas las aplicaciones que añaden valor empresarial enriquecido en SharePoint y podrían requerir más de su participación.

A medida que nuestros clientes internos adoptan el servicio de SharePoint, algunos tendrán requisitos muy sencillos, mientras que otros tendrán necesidades de servicios más complejos o más involucrados. Podemos proporcionar a nuestros clientes con un servicio básico para empezar, y luego ofrecerles diferentes opciones para que puedan aumentar el alcance de las capacidades disponibles y el grado con el que se puede construir una aplicación personalizada.

- Un mayor número de usuarios que utilizan el sitio.
- Una mayor cantidad de contenido en el sitio.
- Una gama más amplia de funciones disponibles para el sitio.

Organizar y priorizar las entradas de solicitud de servicio.

Una de las cuestiones a las que las personas se enfrentan con las solicitudes de servicio implica el manejo de entradas de solicitud de servicio de un cliente, o la falta de manejo para ser más precisos. Por ejemplo, cuando un usuario experimenta algo o un indicio de algo que él o ella quiere experimentar viene a la mente, que se abra un ticket de solicitud de servicio. Por supuesto, el usuario piensa que su solicitud de servicio es importante, de lo contrario, el usuario no se habría molestado en abrir la compra de entradas y para que asigne una alta prioridad a la compra de entradas. Entonces el proceso le asigna el billete, pero ese recurso no cambia el nivel de prioridad, probablemente debido a que el recurso de apoyo no quería insultar al solicitante.

Las solicitudes de servicio van a venir cuando los usuarios tienen problemas con el sistema, y cuando los usuarios imaginan todo tipo de cosas que les gustaría que el sistema haga para el desarrollo de su trabajo. Si un problema afecta a un usuario, él o ella es, naturalmente, va a percibirlo como de alto impacto. El usuario simplemente no tiene una visión global del servicio de SharePoint y por lo general no es consciente de los costos que hay detrás de su petición. En el caso de solicitar una nueva funcionalidad, el usuario no puede incluso saber si la funcionalidad

va a resolver su problema o incluso ser factible. Por otro lado, un usuario puede solicitar una nueva funcionalidad que no es la solución óptima para el problema que están tratando de resolver. Si un experto no analiza un problema real del usuario y el equipo simplemente salta a cualquier funcionalidad que el usuario solicita, entonces el equipo confía en un usuario para jugar el papel de un arquitecto de la solución; sin embargo, un usuario normal no tiene la misma experiencia de SharePoint como arquitecto para soluciones reales. Por otra parte, no creo que esas cuestiones deben ser incluso problemas de los usuarios, ya que tienen sus propios puestos de trabajo. Deben confiar en nosotros para responder a sus solicitudes mediante la adopción de una visión empresarial del sistema y sus costos. Estamos allí para dar sentido a sus peticiones ya que como profesionales de TI, analizamos y comprendemos la función de negocios que están tratando de lograr, y luego diseñar una solución o proporcionar orientación de una manera que ayude a continuación, a que el usuario realice su tarea.

En la imagen de abajo podemos ver el ejemplo de un proceso de flujo de trabajo de entradas de solicitud de servicio:

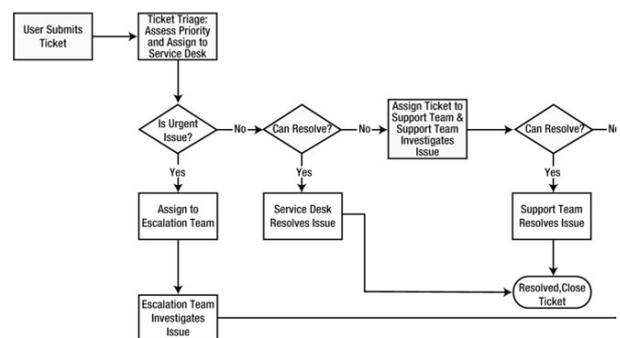


Imagen 1.- Ejemplo de un proceso de flujo de trabajo de entradas de solicitud de servicio.

Si no podemos confiar en los usuarios para priorizar sus propios billetes, ¿cómo se puede confiar en los socorristas y personal urgencias para dar prioridad a una urgencia correctamente? ¿Cómo se asegura la coherencia entre el equipo? Necesitamos un protocolo que defina lo que cada uno de los niveles de prioridad son y los criterios de umbral del ticket debe caber dentro con el fin de asignar a un determinado nivel de prioridad. Necesitamos umbrales precisos para indicadores que puedan medir, y por supuesto, que los indicadores tienen que ser medibles.

Un indicador que me parece revelador de la prioridad de un ticket es la medición de la cantidad de personas afectadas por el problema. Podría ser sólo una persona que presenta la solicitud de servicio, un pequeño grupo de trabajo que esté colaborando con el que consta de ocho personas, su departamento que consta de un centenar de personas, o de toda la organización. Si tenemos en cuenta estos diferentes umbrales, deberían mostrarnos los diferentes impactos que una solicitud de servicio tiene desde el punto de vista de la empresa, y por lo tanto proporcionar una forma consistente para asignar una prioridad válida para

la compra de entradas. Determinando el número de personas afectadas por un problema proporciona una medida fiable para determinar un nivel de prioridad para asignar a una solicitud de servicio durante su diseño inicial. Esto también tiene una perspectiva global, ya que sus umbrales serán en proporción al tamaño de la organización.

Por ejemplo, en una organización más pequeña con menos de 1.000 usuarios, los umbrales pueden variar desde un único usuario a todos los 1000. En contraste, una organización más grande con 100.000 o más usuarios se extenderán esos mismos umbrales en un rango mucho más grande. He encontrado que estas proporciones también representan propia percepción de la organización de la prioridad para un número determinado de usuarios. Esa organización más pequeña se sentiría los efectos de 500 personas afectadas por un corte de luz, la mitad de sus usuarios totales, y se daría prioridad a una resolución mucho más urgente. El mismo número de usuarios de la organización mucho más grande seguiría siendo importante, pero la criticidad no estaría en la misma medida porque el número de usuarios afectados solo representa apenas la mitad del uno por ciento de su población total de usuarios, en lugar de un medio como en el que se hizo para la compañía más pequeña.

podemos precisamente definir una línea básica y posteriormente ir incrementando funcionalidad, un poco más adelante veremos cómo determinar las características de SharePoint y su funcionalidad

Otro indicador que uso para determinar la prioridad de un ticket es la pérdida potencial de ingresos cuando la solicitud de servicio es en relación a un corte de luz que afecta a los clientes externos, como en un ejemplo de un sitio de comercio electrónico. En un proceso similar a la determinación de umbrales medibles para los usuarios afectados, me gustaría determinar los umbrales de la posible pérdida de los ingresos que van desde ninguno hasta tener todos los ingresos detenidos. Cualquier pérdida de ingresos es importante. Tener este tipo de medidas y umbrales razonables le ayudará a mantener el incidente en perspectiva para que pueda reaccionar adecuadamente.

Otra medida para la prioridad: comparar el coste de volver a crear frente al coste para recuperar el contenido. En un sentido más general más allá del costo de producción de contenidos, puede medir el valor de la disponibilidad del contenido. Me gusta tener en cuenta tanto el costo y el valor, en función de las circunstancias. Yo uso el costo como una medida para el contenido generado en las operaciones diarias donde los usuarios pueden fácilmente volver

a crear o de contenido que no es verdaderamente misión crítica y sensible al tiempo. Esto proporciona una decisión métrica más fácil de usar en la priorización de la solicitud de servicio. Por otro lado, el contenido puede ser de misión crítica por razones sensibles al tiempo u otros. Este tipo de contenido se ajusta mejor a la medición del valor que aporta más que el costo de producir, ya que está proporcionando más valor en curso. Después de haber determinado el valor que el contenido más crítico ofrece, puede utilizar esta en la determinación de la prioridad de la solicitud de servicio. Ahora lo ideal es que haya identificado que el contenido antes de un corte de luz o antes de que un usuario envía una solicitud de servicio urgente.

Algunos ejemplos incluirían recuperación de desastres o grandes planes y procedimientos de respuesta a incidentes. Estos son valiosas piezas de contenido cuando los usuarios los necesitan, y en el caso de un incidente grave, que será fundamental para poner a disposición en lugar de considerar la re-creación. Puede identificar qué tipo de contenido y, posiblemente, la almacena en una base de datos designada para el contenido crítico, y esto podría ser la primera base de datos que se restaura en caso de un incidente grave. Otro caso que se puede usar para clasificar contenido es si está considerando la sensibilidad del contenido en sí mismo, y se podía utilizar eso para ayudar a determinar la prioridad de un ticket de solicitud de servicio. Por ejemplo, saber un contenido sensible que contiene información de identificación personal nos proporciona un indicador que puede utilizar para determinar la prioridad en la respuesta a la solicitud de servicio relacionado. Coste contenido, valor crítico, la sensibilidad, la seguridad, y la urgencia proporcionan medidas que podría utilizar para determinar la prioridad de la solicitud de vale de servicio, y lo más importante, se puede medir a todos sin depender de una gran cantidad de subjetividad.

Ahora que podemos dar prioridad a las entradas con una prioridad válida, puede priorizar su respuesta. Cuando sabemos los umbrales de las prioridades de las entradas y las rúbricas que definan en qué consiste cada prioridad, entonces podemos tener confianza en nuestro sistema de venta de entradas y tendremos la seguridad cierta para su funcionamiento. Nuestros usuarios pueden tener confianza en el proceso, y una vez que tengamos los niveles de prioridad bien definidos con todos los parámetros que intervienen en ellos, podemos compartir esta información con los usuarios para ayudar a establecer sus expectativas también.

FRANCISCO RICARDO GIL GONZÁLEZ

MVP CLUSTER | Especialista en SharePoint & Office 365

francisco.gil@fiveshareit.es

LinkedIn

<http://www.mvpcluster.es>

La unión de dos grandes: Azure Machine Learning y PowerBI

¿Como es Azure Machine Learning?

El Machine Learning que nos ofrece Azure es accesible como un servicio más de los disponibles en nuestro panel de control de Azure y todo su potencial es accesible desde ahí o desde azureml.net. Una de las principales ventajas que nos ofrecerá Azure ML es la potencia de procesamiento que implica Azure en sí mismo, ya que montar una infraestructura de estas características en entornos on-premise tendría unos costes muy elevados como punto de partida.

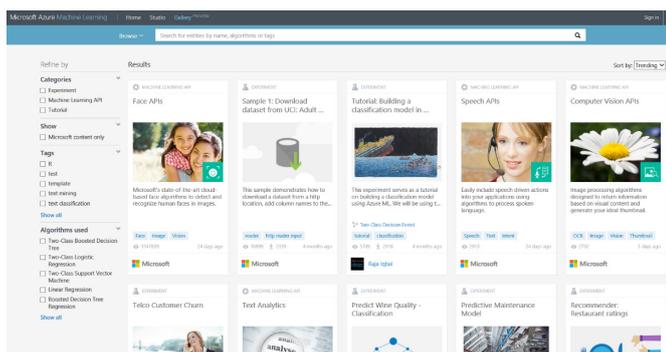


Imagen 1.- Galería de Machine Learning en Microsoft Azure.

Otra de las ventajas ofrecidas es que todas las herramientas necesarias están accesibles a través del portal, el cual os invito a visitar desde ya mismo. Allí encontraremos todo lo que necesitamos para empezar a trabajar con Azure ML, desde un buen repositorio de ejemplos hasta la herramienta de implementación que necesitamos para implementar nuestros proyectos, o como los llamamos en ML: experimentos.

El Machine Learning que nos ofrece Azure es accesible como un servicio más de los disponibles en nuestro panel de control de Azure y todo su potencial es accesible desde ahí o desde azureml.net

Se nota el esfuerzo realizado por parte de Microsoft para que podamos utilizar ML desde el primer momento ya que

podemos acceder a un sinnúmero de tutoriales y experimentos ya creados en la galería. También se agradece la forma de trabajar en los experimentos, muy al estilo de los diseñadores de ETL's que nos ofrecen las Data Tools de SQL Server o mejor dicho, Visual Studio configurado con las plantillas para Integration Services.

Nuestro primer experimento ML

Una vez que hemos iniciado el Studio del panel encontraremos que la forma de implementar un experimento se compone de ir agregando los pasos mediante los cuales vamos a leer, procesar, trabajar y volcar la información.

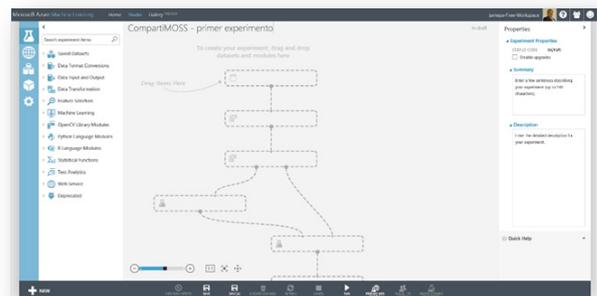


Imagen 2.- ML Studio.

Aunque la mejor forma de empezar a trabajar es con uno de los ejemplos que nos podemos encontrar en la galería, debemos tener en cuenta que el proceso es más simple de lo que parece. En realidad se trata de 3 pasos bien diferenciados:

- **Extraer la información:** la capacidad de leer información de múltiples orígenes de datos es enorme, incluso de aquellos no estructurados.
- **Procesarla y tratarla:** aquí está el trabajo principal ya que la cantidad que necesitamos hacer con la información es infinita pero siempre debemos tener en cuenta el paso a paso y una regla importante a tener en cuenta en este paso es siempre: preparar la información (quitar información que no queremos), normalizarla y evaluarla según los algoritmos que consideremos y que trataré más adelante.
- **Salida:** una vez que tenemos los resultados procesados tendremos gran cantidad de posibilidades donde volcar la información aunque si nuestro objetivo es explotarla más adelante deberíamos almacenarla en un sitio igual

de accesible en el futuro (Azure Databases, Blobs, etc.)

Un punto radical en cualquier experimento de ML es elegir bien el algoritmo de previsión que vamos a aplicar. Y con esto recalco lo de “previsión” ya que recordemos que el objetivo final de ML no es analizar grandes cantidades de información, para eso ya tenemos Big Data, sino buscar los patrones de comportamiento en un conjunto finito de información. En otras palabras, para ML “analizar la información es un medio y no un destino. Su objetivo es buscar el patrón de lo que va a suceder y no centrarse tanto en lo que está sucediendo, aunque obviamente hay que mirar el presente obligatoriamente para intentar averiguar el futuro.

Como comentaba antes la elección de un correcto modelo de predicción, nos llevará al éxito o al fracaso en nuestro experimento, pero desde ya os adelanto que no hay una fórmula mágica para decidir cuál es el mejor modelo ya que aplican distintos dependiendo del caso. Con lo que si contamos es con la posibilidad de evaluar los modelos en todo momento y eso es lo que nos dará pistas sobre los mejores algoritmos a aplicar según la información de la que partimos, y estos métodos son la evaluación y la validación cruzada.

la elección de un correcto modelo de predicción, nos llevará al éxito o al fracaso en nuestro experimento, pero desde ya os adelanto que no hay una fórmula mágica para decidir cuál es el mejor

Hacer la información accesible

Una vez que tengamos nuestros juegos de datos finales procesados nos interesará volcar la información para que esta sea accedida desde otros sitios, como por ejemplo PowerBI. Debemos tener en cuenta que es posible acceder a información en tiempo real desde PowerBI utilizando EventHubs y ASAs (Azure Stream Analytics) los cuales serán seguramente motivo de otro artículo.

Cuando optamos por dar salida a la información desde nuestro experimento, tenemos varias opciones como podemos ver aquí:

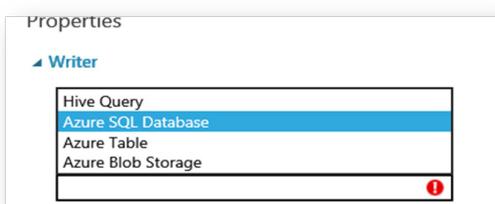


Imagen 3.- Opciones de salida de información para el experimento.

Dependiendo de nuestras necesidades deberemos optar por una u otra opción, y lo mejor es que en todo momento podríamos hacer que esta información esté accesible a través de un web service.

Para que veáis la potencia y la simplicidad de conexión para este artículo he cogido uno de los ejemplos que ofrece el portal de Machine Learning y he agregado un paso de escritura a un base de datos de Azure, a la cual voy a acceder inmediatamente desde Power BI. Para que lo podáis reproducir el paso a paso resumido es bastante simple:

1. Lo primero que necesitamos es Azure SQL Database que podemos crear con una suscripción de prueba y en cuanto esté creada debemos darle permisos para que sea accesible a través de Azure.
2. Ahora, en el panel de control de Machine Learning cogemos uno de los ejemplos que os ofrecen, en mi caso el “Sample 4: Cross Validation for Regression: Auto Imports Dataset” y al final de todo el proceso adjuntamos un paso de escritura a la base de datos que hemos creado en el paso anterior.
3. Le damos a ejecutar y nuestra base de datos estará cargada.

Este es un pantallazo de cómo queda dibujado el final del proceso:

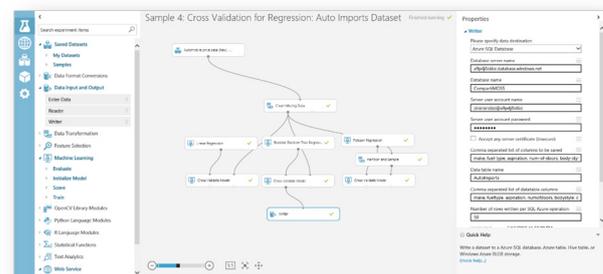


Imagen 4.- Aspecto final del experimento.

Explotando los datos con PowerBI

Ahora que ya contamos con la información disponible a través de nuestro Azure SQL Database el proceso de explotación de esa información es relativamente sencillo. Dado que Power BI puede acceder directamente a múltiples orígenes de datos y uno de ellos es Azure SQL Database, no tenemos que hacer nada más que darle los parámetros de conexión para que pueda acceder a la información.

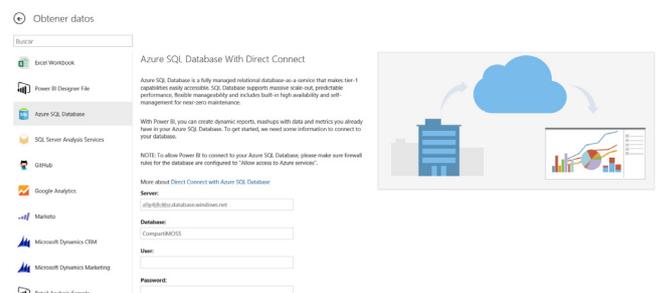


Imagen 5.- Configuración de conexión en Power BI.

Vuelvo a insistir en la necesidad de que en el panel de control de la base de datos en Azure, y especialmente en las reglas de firewall hayamos marcado la opción de permitir acceder desde Azure a esta base de datos, ya que sin esta opción no hay conectividad posible.

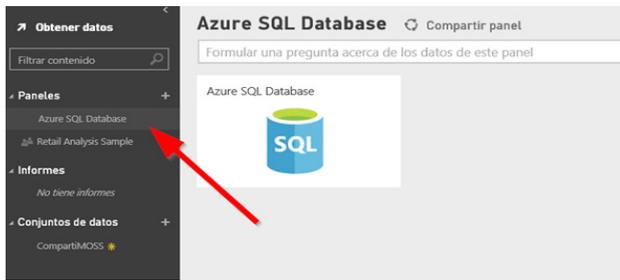


Imagen 6.- Configuración del acceso a la base de datos desde Azure.

Una vez que tengamos nuestra base de datos conectada, la veremos en el panel de navegación de PowerBI y en este punto solo nos queda construir el informe. Al hacer clic en la base de datos se abrirá el diseñador de informes de PowerBI, donde una vez más podemos ver que la conexión es correcta viendo los campos que en su momento exportamos desde nuestro experimento de Machine Learning.



Imagen 7.- Diseñador de informes de Power BI.

Desde mi humilde punto de vista Machine Learning tiene un potencial increíble que todavía no está siendo explotado del todo por el mercado, y está en nosotros, los expertos técnicos y de negocio

A partir de este punto podemos crear el informe arrastrando campos hacia el diseñador de informes y aplicando los filtros que consideremos necesarios.

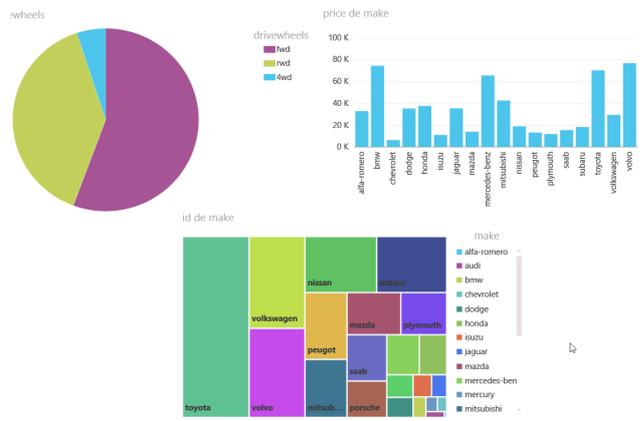


Imagen 8.- Informe final generado.

Conclusión

Desde mi humilde punto de vista, Machine Learning tiene un potencial increíble que todavía no está siendo explotado del todo por el mercado, y está en nosotros, los expertos técnicos y de negocio, en hacer que llegue de una forma rápida y sencilla al público en general. Ya en el día de hoy explotar los datos con PowerBI es posible, tal como hemos visto en el ejemplo, y como os comentaba, contamos con piezas que nos permiten hacer esa interconexión en tiempo real de forma relativamente sencilla. Como frase final decir que no cabe duda que oiremos mucho de Machine Learning en el futuro cercano y más aún cuando se amplíen sus capacidades de volcado e interconexión de forma directa.

JAVIER MENÉNDEZ PALLO

Director Oficina Centro – España y Portugal
ENCAMINA

jmenendez@encamina.com

Integración de PowerShell local con PowerShell de Azure

PowerShell puede ser utilizado tanto de forma local en nuestro equipo haciendo uso de las funciones disponibles, como de forma remota para configurar y administrar servicios OnPremises o en la nube. En este artículo hablaremos de cómo integrar PowerShell local con PowerShell de Azure para poder realizar de manera más ágil nuestras operaciones en Azure.

vamos a hablar acerca de cómo integrar PowerShell local con PowerShell de Azure para poder realizar de manera más ágil nuestras operaciones en Azure

Pasos a seguir para la integración

Para integrar nuestro PowerShell local con PowerShell en Azure hay que seguir los pasos que se indican a continuación:

- Llegamos a la dirección: <http://Azure.microsoft.com/en-us/downloads/> . En el cual encontramos diferentes herramientas que podemos utilizar para interactuar con Azure. Elegimos la Opción : Azure command-line interface, Windows Install (Download .exe)

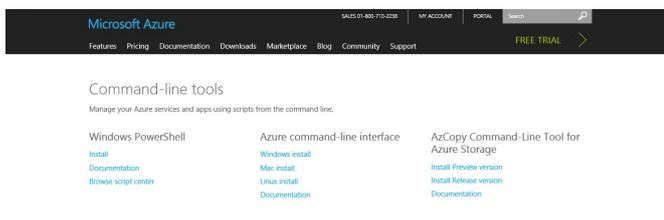


Imagen 1.- Microsoft Azure Download.

- Procedemos a la instalación del mismo presionando el botón “Instalar”

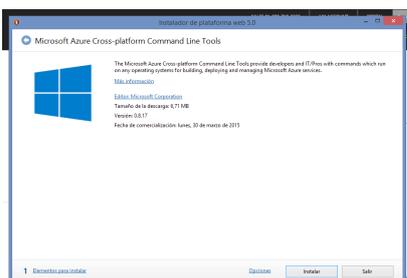


Imagen 2.- Instalación de Azure command-line interface

- En requisitos, tomamos la opción Windows Azure PowerShell y le damos aceptar. Se iniciará el proceso de descarga e instalación de los componentes para PowerShell de Azure.

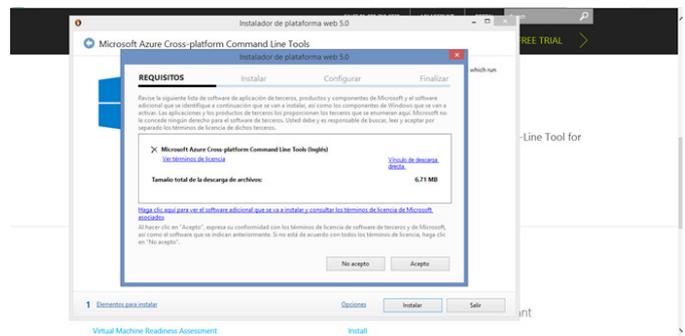


Imagen 3.- Instalación de Azure command-line interface

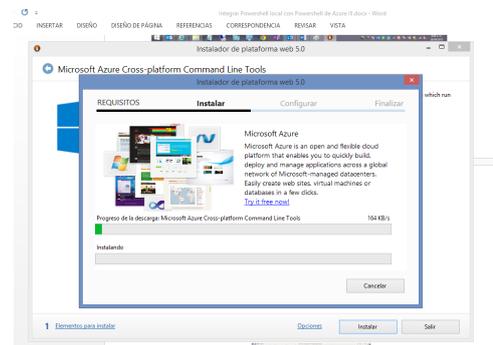


Imagen 4.- Instalación de Azure command-line interface

- Y aquí continuamos con la finalización de la instalación inicial.

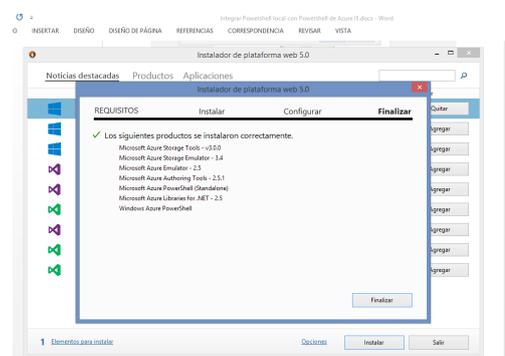


Imagen 5.- Herramienta instalada.

- Luego de tener instalada la herramienta procedemos a ir al panel de Windows y elegimos Azure PowerShell y tomamos la opción “ejecutar como administrador”





Imagen 6.- Acceso a Azure PowerShell en nuestro equipo.

- Estando en la opción de Azure PowerShell ejecutamos la sentencia Set-ExecutionPolicy RemoteSigned que nos sirve para dar permisos de ejecución de scripts PowerShell en nuestro ambiente.



Imagen 7.- Ejecución sentencia en Azure PowerShell.

PowerShell puede ser utilizado tanto de forma local en nuestro equipo haciendo uso de las funciones disponibles, como de forma remota para configurar y administrar servicios OnPremises o en la nube

- Luego de realizar lo anterior procedemos a integrar las herramientas de Azure al ambiente normal de PowerShell ejecutamos lo siguiente:

```
Import-Module "C:\Program Files (x86)\Microsoft SDKs\Windows Azure\PowerShell\Azure\Azure.psd1
```

- Allí elegimos la opción "Certificate".
- Posteriormente nos aparece la opción de loguearnos en Azure, para lo cual entramos con el usuario y contraseña que tenemos asignada para realizar esta operación.

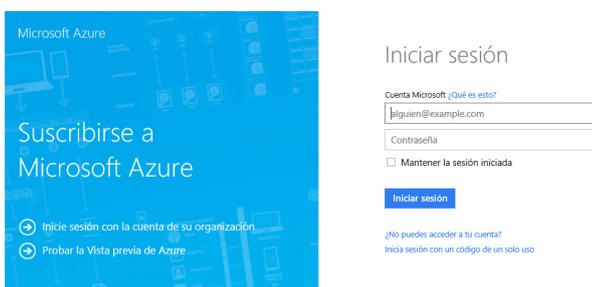


Imagen 8.- Login en Azure.

- Posteriormente procedemos a salvar el archivo que utilizaremos para la suscripción

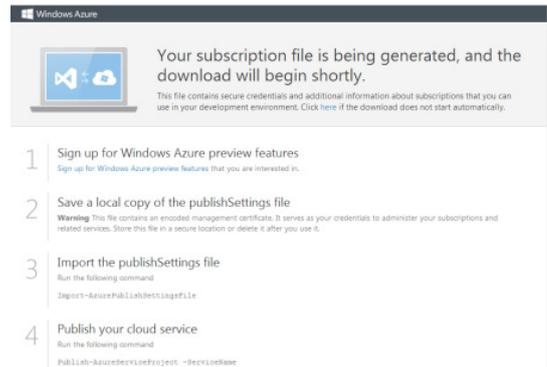


Imagen 11.- Guardar el archivo correspondiente a la suscripción de Azure.

- Si queremos establecer una conexión entre nuestra cuenta en la nube de Azure y nuestro equipo debemos instalar un certificado en nuestro PC proveniente de Azure.

```
Import-AzurePublishSettingsFile -PublishSettingsFile D:\V2\credentials.publishsettings
```

Donde D:\V2\ es la ubicación del archivo

- Realizamos la verificación de nuestra suscripción del comando PowerShell a nuestro pc con la sentencia : Get-AzureSubscription .
- Con la siguiente sentencia verificamos nombre de suscripción que tenemos por PowerShell:

```
Get-AzureSubscription
```

vimos como configurar nuestro pc local para poder conectarnos a Windows Azure desde el shell de sistema, y así poder administrar desde línea de comandos cualquier aspecto de nuestra infraestructura en la nube

Conclusiones

En este artículo vimos como configurar nuestro pc local para poder conectarnos a Windows Azure desde el shell de sistema, y así poder administrar desde línea de comandos cualquier aspecto de nuestra infraestructura en la nube.

ENRIQUE RHENALS BÁRCENAS

MCT, MCP SharePoint 2010 y 2013

erhenalsb@hotmail.com

@enriquerhenals

http://enriquerhenalsb.blogspot.com



Alberto Díaz

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

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

Email: adiazcan@hotmail.com

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

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



Fabián Imaz

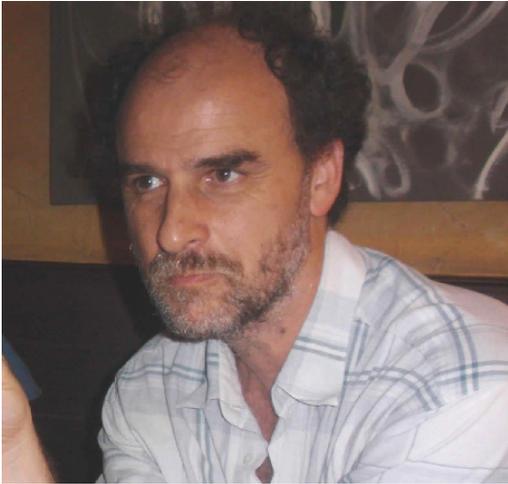
Fabián Imaz, MVP de SharePoint Server trabaja en el mundo del desarrollo de software desde hace más de 10 años, teniendo la suerte de trabajar en distintas arquitecturas y tecnologías Microsoft. Pertenece a la firma Siderys, <http://www.siderys.com> empresa de desarrollo de Software especializada en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

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

Email: fabiani@siderys.com.uy

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

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



Gustavo Velez

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

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

Email: gustavo@gavd.net

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



Juan Carlos González Martín

Juan Carlos González 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



